

Pensamiento Computacional

Programa de Estudio para la Educación
de Personas Jóvenes y Adultas

FORMACIÓN INSTRUMENTAL

Educación Media



Pensamiento Computacional

Programa de Estudio para la Educación
de Personas Jóvenes y Adultas

FORMACIÓN INSTRUMENTAL

Educación Media

DECRETO EN TRÁMITE

Estimados profesores, profesoras y equipos directivos:

Me complace presentar esta nueva versión de los Programas de Estudio para la Educación de Personas Jóvenes y Adultas (EPJA), que busca actualizar y enriquecer la experiencia de aprendizaje de quienes, bajo esta modalidad, buscan retomar sus trayectorias educativas y finalizar la escolaridad.

Los programas que a continuación se presentan ofrecen un marco de acción claro para la gestión pedagógica de docentes y equipos directivos, permitiéndoles orientar su trabajo según las circunstancias específicas en que se lleva a cabo el proceso de enseñanza-aprendizaje, así como las características, necesidades e intereses de los y las estudiantes.

Bajo dicha perspectiva, continuar fortaleciendo la cobertura y abogar por la mejora continua de la EPJA contribuye significativamente a cumplir con la importante misión que tenemos como país de avanzar hacia una educación más pertinente, inclusiva y flexible, que confiera herramientas apropiadas para los desafíos actuales, y promueva el desarrollo de capacidades conducentes a un mayor bienestar y una mejor calidad de vida en nuestra sociedad.

Los Programas están organizados según los Objetivos de Aprendizaje estipulados en las Bases Curriculares de la EPJA, en línea con los aprendizajes significativos que demandan las Habilidades del Siglo XXI. En este esfuerzo, se potencia la formación integral, permitiendo el desenvolvimiento efectivo de cada estudiante en contextos cada vez más exigentes, observando la perspectiva de ciclo vital y de trayectoria educativa.

En esa clave, estos Programas de Estudio no solo ayudan a concretar el anhelo de las personas jóvenes y adultas de finalizar su etapa escolar, sino que también les forma en oficios para el trabajo, y los alienta para que prosigan estudios superiores y enriquezcan su quehacer. Todo esto está mediado por un proceso pedagógico virtuoso, que reconoce las experiencias y proyecto de vida de cada joven y persona adulta como referencias significativas para el trabajo con los contenidos.

Como todo nuestro sistema, la educación para personas jóvenes y adultas afronta desafíos importantes que requieren rigurosidad y empatía. Les invito a seguir orientando su acción pedagógica en esos términos, que quizás son todavía más críticos tratándose de esta modalidad de enseñanza. La EPJA desarrolla una contribución invaluable en el propósito de garantizar e incluso actualizar el derecho a la educación en nuestro país y, de este modo, brindar nuevas oportunidades a quienes desean aportar con sus conocimientos y habilidades al desarrollo de un país donde nadie quede atrás.



Nicolás Eduardo Cataldo Astorga
Ministro de Educación de Chile

Pensamiento Computacional

**Programa de Estudio para la Educación
de Personas Jóvenes y Adultas**

FORMACIÓN INSTRUMENTAL

Educación Media

Documento aprobado por el Consejo Nacional de Educación
mediante el Acuerdo N°019/2022

DECRETO EN TRÁMITE

Unidad de Currículum y Evaluación
Ministerio de Educación
Gobierno de Chile

Primera edición: 2026



Contenidos

PRESENTACIÓN

- 8** **Nociones básicas**
- 10** **Habilidades y Actitudes para el siglo XXI**

CONSIDERACIONES GENERALES

- 16** Perfil de Egreso
- 18** Contextualización Curricular
- 19** Inclusión y Diversidad

ORIENTACIONES PEDAGÓGICAS PROGRAMAS DE ESTUDIO EPJA

- 22** **Organización Modular del Programa de Estudio**
- 22** Módulos Obligatorios
- 25** Módulos Electivos
- 26** Aprendizaje Basado en Proyectos
- 29** Aprendizaje Basado en la Resolución de Problemas
- 30** **Orientaciones para Evaluar los Aprendizajes**
- 32** **Referencias**

PROPÓSITOS FORMATIVOS DE LA ASIGNATURA PENSAMIENTO COMPUTACIONAL

- 36** Enfoque de la Asignatura
- 37** Estructura Curricular
- 39** Objetivos de Aprendizaje
- 39** Conocimientos Esenciales

ORIENTACIONES DIDÁCTICAS Y PEDAGÓGICAS

- 42** **Orientaciones Didácticas y Pedagógicas**
- 47** **Visión panorámica de los Módulos de Educación Media**
- 48** **Visión panorámica de los Objetivos de Aprendizaje y Conocimientos Esenciales**

MÓDULOS OBLIGATORIOS DE LA ASIGNATURA

54 **Módulo Obligatorio 1**

- 54** Visión panorámica
- 56** Actividad de desempeño 1
- 70** Actividad de desempeño 2
- 81** Actividad de desempeño 3
- 88** Actividad de desempeño 4

98 **Módulo Obligatorio 2**

- 98** Visión panorámica
- 100** Actividad de desempeño 1
- 112** Actividad de desempeño 2
- 122** Actividad de desempeño 3
- 132** Actividad de desempeño 4

140 **Módulo Obligatorio 3**

- 140** Visión panorámica
- 142** Actividad de desempeño 1
- 152** Actividad de desempeño 2
- 160** Actividad de desempeño 3
- 167** Actividad de desempeño 4

174 **Módulo Obligatorio 4**

- 174** Visión panorámica
- 176** Actividad de desempeño 1
- 182** Actividad de desempeño 2
- 188** Actividad de desempeño 3
- 193** Actividad de desempeño 4

MÓDULOS ELECTIVOS DE LA ASIGNATURA

202 **Módulo Electivo 1**

- 202** Visión panorámica

210 **Módulo Electivo 2**

- 210** Visión panorámica

ANEXO

- 220** **Bibliografía**
- 222** **Sitios web**
- 223** **Hoja de trabajo**

Presentación

Las Bases Curriculares para la Educación de Personas Jóvenes y Adultas (EPJA) establecen Objetivos de Aprendizaje (OA) de habilidades y actitudes que se integran con conocimientos esenciales, para la comprensión de grandes ideas consideradas relevantes en cada asignatura. El presente programa de estudio es una propuesta de organización curricular que define y desarrolla actividades de desempeño para que los y las estudiantes construyan los aprendizajes establecidos para cada nivel de enseñanza.

Al Ministerio de Educación le corresponde la tarea de elaborar programas de estudio que orienten la implementación de las Bases Curriculares en aquellos establecimientos que no han optado por la elaboración de programas propios. Estos instrumentos constituyen un complemento coherente y alineado con las Bases Curriculares. Además, son una herramienta para apoyar a los y las docentes en el logro de los Objetivos de Aprendizaje y propósitos formativos declarados en cada asignatura y nivel.

Los programas de estudio corresponden a una propuesta que los establecimientos pueden implementar. También pueden ser un referente para aquellas instituciones educativas que deseen elaborar programas de estudio propios, pues son instrumentos que responden a las múltiples realidades educativas provenientes de distintos contextos y que dan origen a una diversidad de aproximaciones didácticas, metodológicas y organizacionales. Estas se expresan en el desarrollo de distintos proyectos educativos, todos válidos mientras permitan el logro de los Objetivos de Aprendizaje.

Los programas de estudio proponen al docente una organización de los Objetivos de Aprendizaje, conocimientos esenciales y grandes ideas de acuerdo con el tiempo disponible dentro del año escolar. Además, son una orientación acerca de cómo desarrollar una comprensión profunda y significativa. Se trata de una estimación temporal aproximada y de carácter propositivo, por lo tanto, puede ser adaptada por las y los docentes de acuerdo con la realidad de sus estudiantes y de su establecimiento.

Para apoyar la implementación de las Bases Curriculares, los programas proporcionan orientaciones disciplinares, didácticas y criterios de evaluación formativa que pueden utilizarse como apoyo para las actividades de desempeño sugeridas. Estas actividades permiten a los y las estudiantes poner en “uso” el conocimiento esencial. Para ello, aplican los procedimientos que definen a las habilidades y actitudes declaradas en los Objetivos de Aprendizaje. En consecuencia, las actividades de desempeño permiten construir aprendizajes y recoger evidencias de comprensión. Además, se enriquecen con recomendaciones de recursos didácticos complementarios y bibliografía para docentes y estudiantes. Se enmarcan en un modelo pedagógico cuyo enfoque es el de la comprensión, lo que implica establecer conexiones desde la experiencia de los y las estudiantes al interior de cada disciplina y también con otras áreas del conocimiento. Las actividades de desempeño de los programas ilustran un modelo para que cada docente, en su establecimiento, pueda construir nuevas actividades acordes con las diversas realidades.

Nociones básicas

Objetivos de Aprendizaje de Habilidades y Actitudes Nucleares

Los **Objetivos de Aprendizaje** de estas Bases Curriculares establecen los desempeños terminales esperables para una asignatura determinada en cada nivel escolar y evidencian, de forma clara y precisa, cuál es el aprendizaje que la o el estudiante debe lograr. Se refieren a las habilidades y actitudes fundamentales de cada asignatura, y se constituyen en el núcleo del aprendizaje.

Las **habilidades** son definidas como procesos estratégicos centrales para realizar tareas y para solucionar problemas con precisión y adaptabilidad. Favorecen la transferencia educativa, es decir, la capacidad para utilizar el conocimiento y aplicarlo a nuevos contextos.

Las **actitudes** son disposiciones frente a objetos, ideas o personas, que incluyen componentes afectivos, cognitivos y valorativos que inclinan a las personas a determinados tipos de acciones. Las actitudes que conforman los Objetivos de Aprendizaje se desprenden de los cuatro ámbitos del marco de Habilidades para el siglo XXI, y su inclusión responde a criterios de pertinencia para ser trabajadas integradamente con las habilidades. En los niveles de Enseñanza Básica, se prioriza el desarrollo de actitudes que fomentan la autonomía y la proactividad, mientras que en los niveles de Enseñanza Media predominan aquellas que fomentan la responsabilidad personal y social de los y las estudiantes.

Las actitudes y habilidades se integran en la construcción de los Objetivos de Aprendizaje nucleares, lo que evidencia su interdependencia y su importancia para una formación integral, que permita a las y los estudiantes contar con una combinación de valores, disposiciones, habilidades y conocimientos para enfrentar los desafíos del futuro¹.

¹ OECD (2020). Op. Cit., pág. 5.

Conocimientos esenciales

Los conocimientos esenciales refieren a una red conceptual coherente y rica en conexiones, que permite construir la comprensión sobre los fenómenos y el mundo. El conocimiento entendido como comprensión, permite a los y las estudiantes refinar, transformar o reemplazar ideas preexistentes que han adquirido en su experiencia cotidiana, y moverse con flexibilidad entre visiones generales, detalles, generalizaciones y ejemplos sobre los fenómenos que estudian.

Los conocimientos esenciales son prioritarios e imprescindibles, pues constituyen una base que permite avanzar de manera progresiva en el aprendizaje de cada asignatura, y construir nuevos conocimientos.

Propósito formativo

Los propósitos formativos de cada asignatura definen las finalidades educativas que se busca desarrollar a partir de los Objetivos de Aprendizaje y conocimientos esenciales en cada nivel. Entregan el para qué del aprendizaje y buscan evidenciar cómo cada área contribuye al logro de los objetivos generales de la Educación Media, definidos en la Ley General de Educación.

En estas Bases Curriculares, las **grandes ideas** operan como propósito formativo de cada nivel y orientan la comprensión y la articulación de los Objetivos de Aprendizaje y los conocimientos esenciales.

Enfoque de la asignatura

Explica los principios, teorías y conceptos disciplinares desde los cuales se han construido los aprendizajes de la asignatura. Se presenta una visión actualizada de dichos elementos de acuerdo con el desarrollo de las disciplinas. Se explicitan también los énfasis teóricos y las perspectivas desde las cuales se espera que los y las docentes y estudiantes aborden los conocimientos, habilidades y actitudes incluidos en los Objetivos de Aprendizaje. Asimismo, en esta sección se explican los enfoques didácticos que permiten orientar la implementación de la asignatura en el aula. Esto último se sustenta en los conceptos, teorías y principios pedagógicos de la enseñanza de cada disciplina.

Habilidades y Actitudes para el siglo XXI

La existencia y el uso de la tecnología en el mundo global, multicultural y en constante cambio ha determinado nuevos modos de acceso al conocimiento, de aplicación de los aprendizajes y de participación en la sociedad. Estas necesidades exigen competencias particulares, las que internacionalmente se conocen como **Habilidades para el siglo XXI** y que responden a los diversos requerimientos del mundo actual, como el aprendizaje de nuevas maneras de pensar, aprender, relacionarse con los demás, comunicarse, usar la tecnología, trabajar, participar en la sociedad, desarrollarse como persona y desarrollar la creatividad, entre otras².

Las Habilidades para el siglo XXI son el foco formativo central que propende a la formación integral de los y las estudiantes. Corresponden a un marco de habilidades y actitudes transversales a todas las asignaturas y a partir de las cuales cada una define sus propios aprendizajes disciplinares. Se presentan organizadas en torno a cuatro ámbitos: Maneras de pensar, Maneras de trabajar, Herramientas para trabajar y Maneras para vivir en el mundo.

DESARROLLO DE ACTITUDES

- ▶ Pensar con **perseverancia** y **proactividad** para encontrar soluciones innovadoras a los problemas.
- ▶ Pensar con **apertura** a distintas perspectivas y contextos, asumiendo riesgos y responsabilidades.
- ▶ Pensar con **consciencia**, reconociendo que los errores ofrecen oportunidades para el aprendizaje.

Continúa →

Maneras de pensar

Desarrollo de la creatividad y la innovación

Las personas creativas poseen habilidades de pensamiento divergente, producción de ideas, fluidez, flexibilidad y originalidad. El pensamiento creativo implica abrirse a diferentes ideas, perspectivas y puntos de vista, ya sea en la exploración personal o en el trabajo en equipo. En este ámbito, la enseñanza implica asumir que este tipo de pensamiento puede desarrollarse en todas las instancias de aprendizaje y en varios niveles: imitación, variación, combinación, transformación y creación original. Para lograr esta última, es importante que las y los docentes consideren que es necesario haber desarrollado varias habilidades y que la creatividad también puede enseñarse mediante actividades más acotadas según los distintos niveles.

2 El conjunto de habilidades seleccionadas para las Bases Curriculares de EPJA corresponden a una adaptación de distintos modelos (Binkley et al., 2012; Fadel et al., 2016). Se han organizado en cuatro categorías: Maneras de pensar, Maneras de trabajar, Herramientas para trabajar y Maneras para vivir en el mundo.

- ▶ Pensar con **flexibilidad** para reelaborar las propias ideas, puntos de vista y creencias.
- ▶ Pensar con **reflexión propia y autonomía** para gestionar el propio aprendizaje, identificando capacidades, fortalezas y aspectos por mejorar.
- ▶ Pensar con consciencia de **que los aprendizajes se desarrollan a lo largo de la vida** y enriquecen la experiencia.
- ▶ Pensar con **apertura hacia otros** para valorar la comunicación como una forma de relacionarse con diversas personas y culturas, compartiendo ideas que favorezcan el desarrollo de la vida en sociedad.

Desarrollo del pensamiento crítico

El pensamiento crítico permite discriminar entre informaciones, declaraciones o argumentos, mediante la evaluación de su contenido y pertinencia; cuestionar la información, tomar decisiones y emitir juicios; y reflexionar críticamente acerca de diferentes puntos de vista, tanto de los propios como de los demás, ya sea para defenderlos o contradecirlos sobre la base de evidencias. Contribuye así, además, a la autorreflexión y corrección de errores, además de favorecer la capacidad de estar abierto a los cambios y de tomar decisiones razonadas. El principal desafío en la enseñanza del pensamiento crítico es la aplicación exitosa de estas habilidades en contextos diferentes de aquellos en que fueron aprendidas.

Desarrollo de la metacognición

Corresponde al concepto de “aprender a aprender”. Se refiere a ser consciente del propio aprendizaje y de los procesos para lograrlo, lo que permite auto-gestionarlo con autonomía, adaptabilidad y flexibilidad. El proceso de pensar acerca del pensar involucra la reflexión propia sobre la posición actual, fijar los objetivos a futuro, diseñar acciones y estrategias potenciales, monitorear el proceso de aprendizaje y evaluar los resultados. Incluye tanto el conocimiento que se tiene sobre sí mismo como estudiante o pensador, como los factores que influyen en el rendimiento. La reflexión acerca del propio aprendizaje favorece su comunicación, por una parte, y la toma de conciencia de las propias capacidades y debilidades, por otra. Desde esta perspectiva, favorece el desarrollo de autoestima, disciplina, capacidad de perseverar y tolerancia a la frustración.

DESARROLLO DE ACTITUDES

- ▶ Trabajar **colaborativamente** en la generación, desarrollo y gestión de proyectos y la resolución de problemas, integrando las diferentes ideas y puntos de vista.
- ▶ Trabajar con **responsabilidad y liderazgo** en la realización de las tareas colaborativas y en función del logro de metas comunes.
- ▶ Trabajar con **empatía y respeto** en el contexto de la diversidad, eliminando toda expresión de prejuicio y discriminación.
- ▶ Trabajar con **autonomía y proactividad** en trabajos colaborativos e individuales para llevar a cabo eficazmente proyectos de diversa índole.

Maneras de trabajar

Desarrollo de la comunicación

La comunicación, ya sea escrita, oral o multimodal, requiere crear estrategias y herramientas que se adecuen a diversas situaciones, propósitos y contextos socioculturales, con el fin de transmitir lo que se desea de manera efectiva. La comunicación permite desarrollar empatía, autoconfianza, valoración de la interculturalidad, así como adaptabilidad, creatividad y rechazo a la discriminación.

Desarrollo de la colaboración

La colaboración entre personas con diferentes habilidades y perspectivas faculta al grupo para tomar mejores decisiones que las que se adoptarían individualmente. Además, el trabajo colaborativo entre pares determina nuevas formas de aprender y de evaluarse a sí mismo y a los demás, lo que permite visibilizar los modos en que se aprende. Esto conlleva nuevas maneras de relacionarse en torno al aprendizaje.

La colaboración compromete, a su vez, actitudes clave para el aprendizaje en el siglo XXI, como el liderazgo, la responsabilidad, perseverancia, apertura de mente hacia lo distinto, aceptación y valoración de las diferencias, autoestima, tolerancia a la frustración y empatía.

DESARROLLO DE ACTITUDES

- ▶ **Aprovechar las herramientas** disponibles para aprender y resolver problemas.
- ▶ **Interesarse** por las posibilidades que ofrece la tecnología para el desarrollo intelectual, personal y social del individuo.
- ▶ **Valorar las TIC** como una oportunidad para informarse, investigar, socializar, comunicarse y participar como ciudadano.
- ▶ **Actuar responsablemente** al gestionar el tiempo para llevar a cabo eficazmente los proyectos personales, académicos y laborales.
- ▶ **Actuar de acuerdo con los principios de la ética** en el uso de la información y de la tecnología, respetando la propiedad intelectual y la privacidad de las personas.

Herramientas para trabajar

Desarrollo de la alfabetización digital

Promueve el desarrollo del pensamiento computacional, la autonomía y el trabajo en equipo, la creatividad, la participación en redes de diversa índole, y el interés por ampliar los propios intereses y horizontes culturales, mediante el uso responsable de la tecnología para hacer frente a nuevos desafíos, como la ciberseguridad y el autocuidado. La utilización de esta como herramienta de trabajo supone dominar las posibilidades que ofrece, así como darle un uso creativo e innovador que, a la vez, promueva el pensamiento crítico. La alfabetización digital apunta también a la resolución de problemas en el marco de la cultura digital que caracteriza al siglo XXI. Para ello, son útiles las herramientas que entrega la programación, el pensamiento computacional, la robótica e internet, entre otros, para desarrollar habilidades que permitan crear contenidos digitales, informarnos y vincularnos con los demás utilizando la tecnología.

Desarrollo del uso de la información

Dice relación con la eficacia y eficiencia en la búsqueda, el acceso, el procesamiento, la clasificación, la integración, la gestión, la evaluación crítica, el uso creativo y ético, y la comunicación de la información. Implica formular preguntas, indagar y generar estrategias para seleccionar, organizar y comunicar. Tiene además siempre en cuenta tanto los aspectos éticos y legales que la regulan, como el respeto a los demás y a su privacidad. Promueve también el acceso, uso responsable, aplicación eficaz y evaluación crítica de Tecnologías de Información y Comunicación (TIC), y su uso creativo de acuerdo con distintos propósitos, atendiendo a las características y convenciones de diversos contextos multiculturales.

DESARROLLO DE ACTITUDES

- ▶ **Perseverar** en torno a metas con miras a la construcción de proyectos de vida y al aporte a la sociedad y al país con autodeterminación, autoconfianza y respeto por uno mismo y por los demás.
- ▶ **Participar** asumiendo posturas razonadas en distintos ámbitos: cultural, social, político, medioambiental, entre otros.
- ▶ **Tomar decisiones razonadas** y que contribuyan al bien común, respetando los derechos humanos, la diversidad y la multiculturalidad.
- ▶ **Actuar con honestidad**, responsabilizándose por las propias acciones y decisiones con conciencia de las implicancias que estas tienen sobre uno mismo y los otros.

Maneras de vivir en el mundo

Desarrollo de la ciudadanía local y global

La ciudadanía se refiere a la participación del individuo en su contexto desde una perspectiva política, social, territorial, cultural, económica, medioambiental, entre otras dimensiones. Por ello, es necesaria la interacción eficaz con las instituciones públicas y con iniciativas que apoyen la cohesión social. La participación también implica reflexionar y tener un juicio crítico acerca de los mensajes de los medios de comunicación masiva, de modo de adoptar una postura razonada ante ellos. La conciencia de ser ciudadano promueve el sentido de pertenencia, la valoración y el ejercicio de los principios democráticos, como los derechos humanos y la igualdad, así como asumir sus responsabilidades como tal. En este sentido, el respeto a los demás, a su privacidad, y a las diferencias valóricas, religiosas y étnicas cobra gran relevancia y se relaciona directamente con una actitud empática, de mentalidad abierta y de adaptabilidad.

Desarrollo del plan de vida y carrera

La construcción y consolidación de un proyecto de vida y de una carrera, oficio u ocupación, requiere la capacidad de adaptarse a los cambios para poder desenvolverse en distintos roles y contextos. Para el logro de objetivos personales, es necesario establecer metas, crear estrategias para conseguirlas, desarrollar la autogestión, actuar con iniciativa y compromiso, tener autonomía para ampliar los aprendizajes, poseer autocritica, reflexionar críticamente y disponerse a integrar las retroalimentaciones recibidas. Por otra parte, para lograr estas metas se requiere interactuar con los demás de manera flexible, con la capacidad de trabajar en equipo y negociar para la búsqueda de soluciones. Esto permite el desarrollo de liderazgo, responsabilidad, ejercicio ético del poder y el respeto a las diferencias en ideas y valores.

Desarrollo de responsabilidad personal y social

La responsabilidad personal y social se interrelacionan constantemente. En lo personal, el respeto por los demás y el rechazo a la discriminación, la conciencia acerca de la propia cultura y las relaciones de esta con las del mundo, el compromiso con la propia vida y el contexto inmediato, y el control de la agresión, la violencia y la autodestrucción permiten que las personas se desarrollen de una manera integral. Por otra parte, el compromiso con la propia persona se traduce, a su vez, en una manera sana y activa de relacionarse con los demás, lo que entrega confianza a los otros y permite comunicarse de una manera asertiva, empática, libre de prejuicios, que acepte los distintos puntos de vista y contribuya a mejorar la sociedad en la que vive. Estas habilidades apuntan a ser consciente de sí mismo y de los otros, además de realizar acciones concretas que den cuenta de la responsabilidad que tiene el individuo con su vida y con su entorno.

Consideraciones Generales

Las consideraciones que se presentan a continuación son relevantes para una óptima implementación de los Programas de Estudio. Se vinculan estrechamente con los enfoques curriculares, y permiten abordar de mejor manera los Objetivos de Aprendizaje de las Bases Curriculares.

Perfil de Egreso

La formación habilita al estudiante para conducir su propia vida en forma autónoma, plena y responsable, de modo que pueda desarrollar planes de vida y proyectos personales, continuar su proceso educativo formal mediante la educación superior o incorporarse a la vida laboral.

Los y las estudiantes que egresan de la modalidad de Educación de Personas Jóvenes y Adultas han desarrollado los conocimientos, habilidades y actitudes definidas en el currículum nacional y transfieren sus aprendizajes a distintos ámbitos: social, cultural, cívico, laboral, intelectual y personal. A partir de dichos aprendizajes, son capaces de alcanzar sus metas académicas y laborales; además de construir un proyecto de vida de acuerdo con sus necesidades e intereses, actuando con autonomía y responsabilidad.

El marco de Habilidades para el siglo XXI, los Objetivos Generales de la Ley General de Educación y las Bases Curriculares para la EPJA definen un conjunto de diez competencias que reúnen habilidades, actitudes y conocimientos que los y las estudiantes han adquirido al finalizar el Segundo Nivel de Educación Media de la modalidad. Estas competencias se organizan según los ámbitos de las Habilidades para el siglo XXI, y su relación de tributación con las habilidades y actitudes nucleares de los Objetivos de Aprendizaje. La competencia 1 se refiere al dominio disciplinar de las asignaturas que los estudiantes deberán alcanzar al finalizar la Educación Media.

Dominio disciplinar

1. Aplica conocimientos y habilidades disciplinares de las áreas de lenguaje, matemáticas, ciencias, historia, geografía y ciencias sociales e idioma extranjero inglés en contextos que impliquen aprendizaje y desarrollo personal.

Maneras de pensar

2. Gestiona el proceso de aprendizaje personal por medio de habilidades de metacognición, reflexión y comunicación, demostrando autonomía, motivación y una sólida autoestima y confianza en las propias capacidades para mejorar y enriquecer su desarrollo personal y cognitivo.
3. Identifica problemas, elabora argumentos, considera nuevas ideas, y propone soluciones creativas e innovadoras ante los desafíos que enfrenta.
4. Piensa de manera crítica y elabora puntos de vista y opiniones propias, utilizando evidencia y con una actitud abierta, dispuesta a cuestionar los supuestos y a reconsiderar las propias visiones.

Maneras de trabajar

5. Trabaja de manera colaborativa con otras personas en la resolución de problemas y en el desarrollo de proyectos. Demuestra habilidades interpersonales de comunicación, gestión y monitoreo del trabajo, la capacidad para asumir roles, reconocer fortalezas y aceptar debilidades, y una actitud perseverante para alcanzar los objetivos propuestos.
6. Se comunica efectivamente con otras personas en lengua materna y en una lengua extranjera, con diferentes propósitos y en diversos contextos, por medio de habilidades de comunicación oral, escrita y no verbal, demostrando capacidad de escuchar y comprender distintos mensajes, y una valoración positiva del lenguaje como fuente de enriquecimiento cultural y personal.

Herramientas para trabajar

7. Utiliza Internet y las herramientas digitales de manera efectiva y eficiente. Demuestra habilidades de búsqueda, selección, manejo y producción de información, y capacidad para resolver tareas, además de reconocer los aspectos éticos y legales involucrados en el acceso y uso de la información en ambientes digitales.
8. Demuestra compromiso y capacidad de autogestionar el aprendizaje en las diversas instancias de formación que enfrenta, por medio de habilidades que le permitan desenvolverse en distintos roles y contextos y planificar un proyecto de vida personal y laboral en el tiempo. Desarrolla una disposición favorable al aprendizaje a lo largo de la vida.

Maneras de vivir en el mundo

9. Se relaciona de manera respetuosa, empática y constructiva con otras personas en las diversas instancias de intercambio y colaboración que enfrenta. Demuestra conciencia y reconocimiento de la propia cultura y la de los demás, y una actitud de rechazo a la violencia, la agresión y la discriminación.
10. Demuestra conciencia de los derechos y responsabilidades ciudadanas al relacionarse con sus pares, con la comunidad y con las instituciones públicas, mediante habilidades de interacción eficaz, de participación y toma de decisiones. Muestra un compromiso con el bien común, la cohesión social, los derechos humanos y los principios de la democracia, a nivel local y global.

Contextualización Curricular

La contextualización curricular es el proceso de apropiación y desarrollo del currículum en una realidad educativa concreta. Este se lleva a cabo considerando las características particulares del contexto escolar, por ejemplo, el medio en que se sitúa el establecimiento educativo, la cultura, el proyecto educativo institucional de la escuela y la comunidad escolar, el tipo de formación diferenciada que se imparte (Humanístico-Científica o Técnico-Profesional), lo que posibilita que el proceso educativo adquiera significatividad para los y las estudiantes desde sus propias realidades y facilita, así, el logro de los Objetivos de Aprendizaje.

El marco de habilidades y actitudes que define esta propuesta permite desarrollar aquellas que facilitan formas de pensar, de vivir en el mundo y de trabajar. Además, entrega herramientas para trabajar que definen el perfil de los y las estudiantes EPJA y que pueden ser utilizadas como estrategias para atender a las necesidades de contextualización de las diferencias que se presentan en las aulas. Los programas de estudio son una propuesta de diseño de clases, de actividades y de evaluaciones flexible, que puede modificarse, ajustarse y transferirse a diferentes realidades y contextos, entre otros:

- ▶ **DIVERSIDAD ETARIA.** Debido a que la edad de los y las estudiantes de Educación de Personas Jóvenes y Adultas puede variar de los 15 a más de 50 años, las actividades propuestas se han diseñado desde un principio de flexibilidad que permita en las aulas ajustarse a las distintas necesidades y posibilidades de estudiantes que no han iniciado o han interrumpido su trayectoria formativa por un corto o un largo periodo de tiempo.
- ▶ **TIPOS DE ESTABLECIMIENTOS.** De acuerdo con las distintas posibilidades originadas por el tipo de establecimiento en las que se implementa la modalidad, como Tercera jornada, Centros de Educación Integrada de Adultos, Establecimientos Educacionales al interior de Recintos Penitenciarios y al interior de Unidades Militares, la implementación didáctica de las actividades propuestas en el programa sugiere el uso de recursos y procedimientos tanto análogos como virtuales.
- ▶ **TRAYECTORIAS FORMATIVAS.** Por diversas razones las trayectorias formativas de los estudiantes EPJA se interrumpen. En consecuencia, la progresión de aprendizajes de las asignaturas que forman parte del Plan de Formación General en sus distintos Niveles de Educación Básica y Educación Media: Lenguaje y Comunicación / Lengua y Literatura, Matemática, Ciencias e Historia, Geografía, Ciencias Sociales y Educación Ciudadana, pueden estar afectadas. Por lo tanto, puede ocurrir que la implementación de los programas de estudio se necesite realizar procesos previos de nivelación, que permitan a las y los estudiantes avanzar en su trayectoria formativa. La implementación del programa se ha diseñado en un tiempo estimativo que, de acuerdo con el plan de estudio, puede ajustarse a las necesidades formativas de los y las estudiantes.

Inclusión y Diversidad

En el trabajo pedagógico, es importante comprender que la diversidad se entiende en términos culturales, sociales, étnicos, religiosos, de género, de estilos de aprendizaje y de niveles de conocimiento y/o de trayectorias escolares. Esta diversidad enriquece los escenarios de aprendizaje y está asociada a los siguientes desafíos:

- Desarrollar aprendizajes significativos que se relacionen con el contexto y la realidad de los estudiantes.
- Crear oportunidades inclusivas para desarrollar el aprendizaje en todos y todas.
- Favorecer y potenciar metodologías integradoras y colaborativas tales como Aprendizaje Basado en Proyectos (ABP) y Aprendizaje Basado en la Resolución de Problemas (ABRP).

Atender a la diversidad de estudiantes en sus contextos implica reconocer las necesidades educativas que poseen para así diseñar experiencias de aprendizaje a partir de tiempos, recursos y estrategias que permitan que cada estudiante logre un aprendizaje de calidad. La experiencia y conocimiento que tengan las y los docentes sobre su asignatura y las estrategias que promuevan un aprendizaje profundo, son herramientas para tomar decisiones pertinentes y oportunas respecto de las necesidades de los y las estudiantes.

Para las y los estudiantes con necesidades educativas especiales, el conocimiento de sus profesores y profesoras, así como el apoyo y las recomendaciones de los especialistas, contribuyen a que todos desarrollen al máximo sus capacidades. Algunas orientaciones para considerar:

- Promover ambientes de aprendizaje inclusivos, donde cada estudiante sienta seguridad para participar, experimentar y contribuir de forma significativa a la clase. Se recomienda destacar positivamente las características particulares y rechazar toda forma de discriminación, agresividad o violencia.
- Proveer igualdad de oportunidades y asegurar que los y las estudiantes puedan participar por igual en todas las actividades. Evitar relacionar el trabajo de aula con estereotipos asociados a género, características físicas o cualquier otro tipo de sesgo que provoque discriminación.
- Utilizar diversos materiales, estrategias didácticas y actividades que se adecuen a las singularidades de los y las estudiantes y sus intereses.
- Promover un trabajo sistemático, con actividades variadas para diferentes estilos de aprendizaje y con ejercitación abundante, procurando que todos tengan acceso a oportunidades de aprendizaje enriquecidas.

Orientaciones Pedagógicas Programas de Estudio EPJA

Todas las actividades siguen los pasos que caracterizan el proceso de aprendizaje para la Educación de Personas Jóvenes y Adultas: identificar la necesidad del aprendizaje; crear una estrategia y recursos para alcanzarlos; desarrollar la estrategia y evaluarla. Para aprender, necesitan saber cuál es el propósito de su aprendizaje, aplicar lo aprendido en la vida y ser agentes de su propio aprendizaje, por medio de su experiencia.

La etapa inicial del aprendizaje es de gran importancia. Si bien el o la estudiante puede no estar siempre consciente de lo que necesita aprender, la motivación y el compromiso por el aprendizaje, como un medio para adquirir autonomía y aprender a aprender, pueden operar como incentivos poderosos para encontrar un sentido al aprendizaje escolar. Asimismo, es relevante que participen en el proceso de diseño del aprendizaje. La literatura señala que, en el estudiantado adulto, compartir el control de las estrategias de aprendizaje lo hace más eficaz³. Hacerlos partícipes como agentes de su aprendizaje, satisface su necesidad de conocer y estimula su autoconcepto como los estudiantes independientes⁴.

3 Knowles, M. S., Holton III, E. F., & Swanson, R. A. (2014). *The adult learner: The definitive classic in adult education and human resource development*. London: Routledge, pág. 148.

4 Ibídem.

Organización Modular del Programa de Estudio

Los programas de estudio correspondientes a las Bases Curriculares de la Educación de Personas Jóvenes y Adultas, proponen una estructura modular que organiza los Objetivos de Aprendizaje de habilidades y actitudes, los conocimientos esenciales y las grandes ideas de cada asignatura de acuerdo con las Bases Curriculares aprobadas para la modalidad.

Los módulos se definen como bloques unitarios de aprendizaje que integran habilidades, actitudes y conocimientos requeridos para adquirir desempeños flexibles en una determinada área o asignatura.

Todas las asignaturas, tanto del plan de Formación General como de Formación Instrumental cuentan con programas de estudio modulares para su implementación. En cuanto a la estructura, cada asignatura se organiza por nivel en cuatro módulos obligatorios y dos módulos electivos. Los primeros organizan los Objetivos de Aprendizaje, conocimientos esenciales y grandes ideas de cada nivel, mientras que los segundos ofrecen oportunidades de profundizar en el desarrollo del OA y en la comprensión de las grandes ideas del nivel, por medio del desarrollo de proyectos o la resolución de problemas.

Módulos Obligatorios

En coherencia con las Bases Curriculares, los módulos obligatorios organizan los Objetivos de Aprendizaje, los conocimientos esenciales y las grandes ideas del nivel. Cada módulo presenta cuatro actividades de aprendizaje y evaluación que desarrollan, como foco principal, las habilidades y actitudes de los Objetivos de Aprendizaje del nivel. En las Bases Curriculares para EPJA, las habilidades son entendidas como conocimientos procedimentales que desarrollan destrezas de pensamiento y hábitos de mente que permiten pensar los contenidos en profundidad. Como señala David Perkins (2017)⁵, desarrollar habilidades permite a las y los estudiantes aprender a pensar sobre el conocimiento, ponerlo “en movimiento para hacer conexiones y predicciones”, además de darle forma “para crear nuevos productos y resultados creativos”.

5 Perkins, D. Prólogo a Swartz, R. et al. (2017). Op. Cit., pág. 8.

Organización del Aprendizaje en los Módulos Obligatorios

Los módulos obligatorios organizan el aprendizaje en torno al desarrollo de una actividad de desempeño y actividades de evaluación que se integran. Los elementos que componen estos módulos son:

Visión panorámica del módulo

La visión panorámica de cada módulo presenta la gran idea, los objetivos de aprendizaje y conocimientos esenciales que se necesitan desarrollar para cumplir el propósito formativo del módulo. Por último, se identifica el tiempo semanal y las horas de clase propuestas para abarcar su implementación.

Propósito del módulo

El propósito del módulo responde a tres interrogantes: ¿qué se espera que los y las estudiantes comprendan?, ¿cómo se evidenciará que han comprendido? y ¿cómo tributa el módulo al marco formativo de las Habilidades y Actitudes para el siglo XXI? Para responder a la primera interrogante se explica brevemente la gran idea que se pretende construir en el módulo. Luego, se relacionan explicativamente habilidades, actitudes y conocimientos esenciales que pondrá en uso el o la estudiante para finalmente detallar cómo estos se integran y tributan al marco de Habilidades y actitudes para el siglo XXI.

Ruta de aprendizaje

Secuencia de cuatro actividades de desempeño que describen sintéticamente qué habilidades –procedimientos estratégicos– y actitudes desarrollará el o la estudiante para poner en uso los conocimientos esenciales declarados en el módulo. Cada desempeño se construye identificando qué hace el estudiante –habilidad o procedimiento aplicado– y el conocimiento esencial que se moviliza. El conjunto de actividades de desempeño se integra coherentemente para dar cuenta del propósito formativo general declarado en el módulo.

Actividades de desempeño

Para organizar el desarrollo de las actividades propuestas se utilizan criterios didácticos transversales que guíen flexiblemente a las y los docentes, de modo que puedan transferir la propuesta a sus diferentes contextos. Los criterios utilizados se distinguen por su función didáctica, es decir, la finalidad formativa que se persigue a través de ellos:

- ▶ **SITUACIÓN EXPERIENCIAL.** Permite enmarcar de forma situada un determinado aprendizaje. Activa y engancha el conocimiento previo con el nuevo conocimiento para desarrollar un aprendizaje significativo.
- ▶ **CONSTRUCCIÓN DEL CONOCIMIENTO.** Permite ilustrar cómo mediar, a través de una propuesta de selección de recursos y estrategias la adquisición y organización de nuevos conocimientos.
- ▶ **PRÁCTICA GUIADA.** Modela paso a paso la mediación que realiza el o la docente, a través de actividades individuales, plenarias o colaborativas que desarrollan los y las estudiantes, para profundizar en la comprensión de un determinado conocimiento.
- ▶ **PRÁCTICA INDEPENDIENTE.** Detalla las actividades individuales y/o colaborativas que desarrollan los y las estudiantes para realizar desempeños flexibles que permitan profundizar y evidenciar su comprensión. Permite monitorear el proceso de aprendizaje.
- ▶ **INTEGRACIÓN.** Corresponde a una actividad de síntesis que realiza el o la estudiante individualmente para evidenciar la comprensión del propósito declarado para la actividad. Por ejemplo, mediante el uso de ticket de salida.
- ▶ **ORIENTACIONES AL DOCENTE.** En esta sección se aclaran y precisan conceptos disciplinares que se han movilizado a lo largo del módulo. Se realizan sugerencias complementarias sobre el trabajo con adultos y/o estrategias didácticas que puedan facilitar su labor. Se sugieren estrategias para guiar la retroalimentación y la evaluación formativa compartiendo criterios, estrategias de retroalimentación y rúbricas.

Módulos Electivos

Los módulos electivos ofrecen oportunidades de profundizar en el desarrollo de las habilidades y actitudes de los Objetivos de Aprendizaje del nivel y en la comprensión de las grandes ideas. Se desarrollan por medio de metodologías de Aprendizaje Basado en Proyectos y Aprendizaje Basado en Resolución de Problemas. Se organizan en torno a un tema que es planteado como problema o desafío y que permite ampliar el conocimiento esencial, profundizar en la comprensión de las grandes ideas y conectar con los intereses y experiencias de los y las estudiantes.

Los problemas y desafíos podrán ser adaptados a los contextos, intereses y experiencias vitales de los y las estudiantes.

Se sugiere considerar los Objetivos de Desarrollo Sostenible de la Agenda para el Desarrollo Sostenible de la Unesco como foco para orientar los problemas y proyectos a desarrollar en los módulos electivos. Estos temas son⁶:

- | | |
|--|---|
| 1. Fin de la pobreza | 10. Reducción de las desigualdades |
| 2. Hambre cero | 11. Ciudades y comunidades sostenibles |
| 3. Salud y Bienestar | 12. Producción y consumo responsable |
| 4. Educación de calidad ⁷ | 13. Acción por el clima |
| 5. Igualdad de género | 14. Vida submarina |
| 6. Agua limpia y saneamiento | 15. Vida de ecosistemas terrestres |
| 7. Energía asequible y no contaminante | 16. Paz, justicia e instituciones sólidas |
| 8. Trabajo decente y crecimiento económico | 17. Alianzas para lograr los objetivos |
| 9. Industria, innovación e infraestructura | |

Esta metodología debe promover un compromiso activo de la o el estudiante con el aprendizaje, lo cual se logrará si es que el proceso de enseñanza cumple con las siguientes características:

- El aprendizaje favorecido se conecta con sus necesidades o inquietudes;
- El o la estudiante sabe de antemano *qué* aprenderá;
- El o la estudiante considera que este aprendizaje es importante (*por qué* aprender);
- El o la estudiante sabe *cómo* ocurrirá este aprendizaje (plan de trabajo) e idealmente participa en su planeamiento.

6 Recuperado de: <https://es.unesco.org/sdgs>

7 Las Bases Curriculares de EPJA se encuentran alineadas con este objetivo N°4, en tanto apuntan al aprendizaje a lo largo de la vida, y a una educación de calidad para todos.

Estructura del Aprendizaje en los Módulos Electivos

Las metodologías de Aprendizaje Basado en Proyectos y Aprendizaje Basado en la Resolución de Problemas son propuestas que permiten desarrollar habilidades y poner en uso el conocimiento, integrar aprendizajes y promover la curiosidad y la búsqueda activa y creativa de respuestas. Estas metodologías buscan que los y las estudiantes puedan transferir el conocimiento a distintas áreas y/o situaciones de la vida real, por medio de aprendizajes significativos y relevantes. En cada nivel se ilustran dos ejemplos, uno de ABP y otro de ABRP, que podrán servir de modelo para que las y los docentes puedan construir nuevos proyectos o problemas.

Tanto en la Resolución de problemas como en ABP se busca conectar los problemas y preguntas con los Objetivos de Desarrollo Sostenible (ODS) de Unesco, con el fin de reforzar su relevancia y transversalidad.

Aprendizaje Basado en Proyectos

Consiste en la organización de las y los estudiantes en torno a una pregunta o desafío originado a partir de un problema real o que sea significativo, que puede ser concreto o abstracto. Mientras el problema sea más complejo, movilizará e integrará diferentes áreas de conocimiento, lo que favorece la interdisciplinariedad. Para su desarrollo, es deseable que las y los docentes se organicen y planifiquen el trabajo de manera conjunta entre los responsables de diferentes asignaturas.

Existe una serie de elementos que son requisitos para que el diseño de un proyecto permita maximizar el aprendizaje y la participación de las y los estudiantes, de manera que aprendan cómo aplicar el conocimiento al mundo real, cómo utilizarlo para resolver problemas, responder preguntas complejas y crear productos de alta calidad⁸. Estos elementos son:

8 Larmer, J., Mergendoller, J., Boss, S. (2015). *Setting the Standard for Project Based Learning: A Proven Approach to Rigorous Classroom Instruction*. United States: ASCD (Adaptación).

Conocimiento esencial, comprensión y habilidades

El proyecto se enfoca en profundizar en la comprensión del conocimiento, ya que permite desarrollar a la vez los Objetivos de Aprendizaje y las habilidades para el siglo XXI que se requieren para realizar el proyecto.

Se basa en un problema significativo para resolver o en una pregunta para responder. Esto en un nivel adecuado de desafío para alumnas y alumnos, que se implementa mediante una pregunta de conducción abierta y atractiva.

Indagación sostenida

El proyecto implica un proceso activo y profundo a lo largo del tiempo, en el que los y las estudiantes crean preguntas, encuentran y utilizan recursos, hacen cuestionamientos adicionales y desarrollan sus propias respuestas.

Autenticidad

El proyecto tiene como contexto el mundo real. Utiliza procesos, herramientas y estándares de calidad conectados con la realidad y tiene un impacto, ya que creará algo que será utilizado o experimentado por otros, y/o está conectado a las propias preocupaciones, intereses e identidades de las y los estudiantes.

Es importante saber en qué contexto del mundo real puede encontrarse un problema como el planteado y por qué el proyecto puede ser significativo.

Voz y elección del estudiante

Guiados por el o la docente, el proyecto permite a los y las estudiantes tomar algunas decisiones sobre los productos que crean, su funcionamiento y el uso del tiempo.

Gestión y autoorganización

El proyecto exige a los y las estudiantes desarrollar el trabajo en equipo, la comunicación y la resolución de problemas; junto con tomar decisiones sobre el diseño y la implementación del proyecto en sus distintas etapas. Esto implica identificar las competencias y procedimientos que son necesarios para desarrollar un plan de trabajo adecuado al proyecto, y una exploración activa de los recursos y actividades con que cuentan para su desarrollo. Asimismo, reconocer las fortalezas y debilidades con que cuenta cada uno de los miembros para su desarrollo.

Evaluación y retroalimentación

El proyecto brinda oportunidades para que los y las estudiantes reflexionen sobre qué y cómo están aprendiendo. Incluye procesos de evaluación formativa y retroalimentación para que los estudiantes den y reciban comentarios sobre su trabajo, para que revisen sus ideas y productos o realicen una investigación adicional.

Producto público

El proyecto requiere que los y las estudiantes demuestren lo que aprenden, mediante la creación de un producto que se presenta u ofrece a personas que se encuentran más allá del aula.

Considerando estos elementos, los programas proponen un diseño de ABP con la siguiente estructura:

- ▶ **PROBLEMA CENTRAL.** Se describe el problema que origina el proyecto. Busca despertar el interés, predisponer a él o la estudiante para el aprendizaje basado en problemas.
- ▶ **PROPÓSITO.** Propósito formativo del proyecto, es decir, qué se espera que aprendan los y las estudiantes con la realización de este.
- ▶ **OBJETIVOS DE APRENDIZAJE.** Identifica y/o registra qué Objetivos de Aprendizaje de la asignatura y de otras asignaturas del nivel se integran para el desarrollo del proyecto.
- ▶ **PREGUNTAS.** Se proponen preguntas orientadoras, que servirán para diseñar las etapas del proyecto. Son preguntas centrales y generales.
- ▶ **TIPO DE PROYECTO.** Identifica el tipo de proyecto de acuerdo con las asignaturas que participan, por ejemplo, STEM o interdisciplinario.
- ▶ **PRODUCTO.** Identifica el producto que se espera construir colaborativamente en el proyecto para dar respuesta concreta al problema.
- ▶ **HABILIDADES Y ACTITUDES PARA EL SIGLO XXI.** Identifica cuáles son las habilidades y actitudes que se desarrollarán, y a qué ámbito pertenecen.
- ▶ **ETAPAS.** Se realiza un cronograma con las distintas etapas del proyecto. En él se establecen las características de cada etapa, qué hará el o la estudiante y cómo lo realizará. Cada etapa se apoya con recursos y/o ilustraciones.
- ▶ **EVALUACIÓN.** Se comparten criterios de evaluación y rúbricas que guíen y permitan monitorear el desarrollo de los aprendizajes durante la realización del proyecto. Los criterios y las rúbricas deben verificar los aprendizajes de los objetivos que se identificaron para el proyecto, de manera descriptiva y por nivel de logro.
- ▶ **DIFUSIÓN FINAL.** Describe cómo se difundirá el producto e incorporará a la comunidad escolar y/o local.
- ▶ **RECURSOS.** Nombra recursos y los clasifica según su tipo.

Aprendizaje Basado en la Resolución de Problemas

El modelo de Aprendizaje Basado en la Resolución de Problemas que presentan los programas de estudio se organiza en torno a un problema o desafío cognitivo para el que se busca encontrar una solución, mediante el uso del conocimiento y el desarrollo de habilidades. En los programas, un problema se define por una situación o pregunta que presenta restricciones y cuya respuesta no es evidente.

Al resolver problemas, las y los estudiantes utilizan procesos y estrategias que se relacionan con análisis crítico, investigación, evaluación y comunicación; planifican su trabajo y reflexionan sobre la solución que mejor responde a las restricciones que presenta el problema o desafío cognitivo. Como resultado, ponen en uso el conocimiento, lo amplían adquiriendo nuevos conceptos, principios e información, y desarrollan nuevas destrezas de pensamiento crítico y creativo⁹.

La resolución de problemas permite motivar y despertar el interés de los y las estudiantes, desarrolla la autonomía y el trabajo en equipo. Sin embargo, esto requiere que las situaciones o problemas sean significativos y relevantes, y que se puedan visualizar las posibles soluciones. Esta metodología requiere que la o el docente adquiriera un rol activo como guía para monitorear el desarrollo del proceso y orientar el trabajo de sus estudiantes.

El modelo que proponen los programas de estudio para el desarrollo del Aprendizaje Basado en la Resolución de Problemas se compone de los siguientes elementos:

- ▶ **TÍTULO.** Se plantea como una afirmación o pregunta que sintetiza el problema o desafío.
- ▶ **PROPÓSITO.** Busca despertar el interés, predisponer a el o la estudiante para el aprendizaje basado en problemas.
- ▶ **PREPARACIÓN.** Busca contextualizar a los y las estudiantes en la situación que se planteará y/o familiarizarlos con la resolución de problemas y su sistema de trabajo.
- ▶ **PRESENTACIÓN DEL PROBLEMA.** Se expone el problema, considerando la contextualización de este en una situación significativa. Se define con claridad y precisión cuál es el problema; se distinguen conceptos centrales y restricciones que constituyen el problema.
- ▶ **POSIBLES SOLUCIONES.** Se describe cómo se regulará el trabajo colaborativo mediante estrategias que promuevan disposiciones actitudinales positivas y que permitan a los y las estudiantes involucrarse con el problema y buscar soluciones (por ejemplo, la perseverancia), y estrategias de mediación para compartir las soluciones. Se ilustran posibles soluciones.
- ▶ **INVESTIGACIÓN.** Describe cómo mediar el trabajo de investigación y el desarrollo de habilidades de indagación y evaluación; se ilustran recursos que se puedan utilizar y conocimientos disciplinares que se movilizan en la solución del problema.
- ▶ **EVALUAR LA SOLUCIÓN DEL PROBLEMA.** Describe cómo mediar estratégicamente las soluciones propuestas al problema, a partir de las habilidades y la evaluación de las posibles soluciones.
- ▶ **COMUNICACIÓN.** Describe cómo se mediará la comunicación individual y/o colaborativa del problema, según códigos de comunicación pertinentes y característicos de las disciplinas.

9 Swartz, R. (2017). *El Aprendizaje basado en el Pensamiento. Cómo desarrollar en los alumnos las competencias del SXXI* (2017). España: SM.
Figura 7-11. Pág. 232. (Adaptación).

Orientaciones para Evaluar los Aprendizajes

La evaluación, como un aspecto intrínseco del proceso de enseñanza-aprendizaje, se plantea en estos programas con un foco formativo al servicio del aprendizaje de los y las estudiantes. Para que esto ocurra, se plantea recoger evidencias que permitan describir con precisión la diversidad existente en el aula para tomar decisiones pedagógicas y retroalimentar a los y las estudiantes. La evaluación desarrollada con foco pedagógico favorece la motivación de los estudiantes a seguir aprendiendo; asimismo, el desarrollo de la autonomía y la autorregulación potencia la reflexión de los y las docentes sobre su práctica y facilita la toma de decisiones pedagógicas pertinentes y oportunas que permitan apoyar de mejor manera los aprendizajes.

Para implementar una evaluación con un foco formativo, se requiere:

- Diseñar experiencias de evaluación que ayuden a los y las estudiantes a poner en práctica lo aprendido en situaciones que muestren la relevancia o utilidad de ese aprendizaje.
- Evaluar solamente aquello que las y los estudiantes efectivamente han tenido la oportunidad de aprender mediante las experiencias de aprendizaje mediadas por el o la docente.
- Procurar que se utilicen diversas formas de evaluar, que consideren las distintas características, ritmos y formas de aprender, necesidades e intereses de los y las estudiantes, junto con evitar posibles sesgos y problemas de accesibilidad para ellos.
- Promover que los y las estudiantes tengan una activa participación en los procesos de evaluación; por ejemplo, al elegir temas sobre los cuales les interese realizar una actividad de evaluación o sugerir la forma en que presentarán a otros un producto; participar en proponer los criterios de evaluación; generar experiencias de auto y coevaluación que les permitan desarrollar su capacidad para reflexionar sobre sus procesos, progresos y logros de aprendizaje.
- Que las evaluaciones sean de la más alta calidad posible; es decir, deben representar de forma precisa los aprendizajes que se busca evaluar. Además, las evidencias que se levantan y fundamentan las interpretaciones respecto de los procesos, progresos o logros de aprendizajes de los y las estudiantes, deben ser suficientes como para sostener de forma consistente esas interpretaciones evaluativas.

El o la docente puede utilizar diferentes métodos para evaluar los Objetivos de Aprendizaje. Para esto, se sugiere emplear una variedad de medios y evidencias, como portafolios, registros anecdóticos, proyectos de investigación grupales e individuales e informes, presentaciones, entre otros. La forma en que se diseñe este tipo de evaluaciones y el modo en que se registre y comunique la información que se obtiene de ellas debe permitir que integren lo formativo y sumativo para retroalimentar tanto la enseñanza como el aprendizaje.

El uso formativo de la evaluación debiera preponderar en las salas de clases, es decir, utilizarse de manera sistemática para reflexionar sobre el aprendizaje y la enseñanza, y para tomar decisiones

pedagógicas pertinentes y oportunas que busquen promover el progreso del aprendizaje de todos y todas, considerando la diversidad como un aspecto inherente a las aulas.

El proceso de evaluación formativa que se propone implica articular el proceso de enseñanza-aprendizaje en función de responder a las siguientes preguntas: **¿A dónde voy?** (qué objetivo de aprendizaje espero lograr), **¿Dónde estoy ahora?** (cuán cerca o lejos me encuentro de lograr ese aprendizaje) y **¿Qué estrategia o estrategias pueden ayudarme a llegar a donde tengo que ir?** (qué pasos tengo que dar para acercarme a ese aprendizaje). Este proceso continuo de establecer un objetivo de aprendizaje, evaluar los niveles actuales y luego trabajar estratégicamente para reducir la distancia entre los dos, es la esencia de la evaluación formativa. Una vez que se alcanza una meta de aprendizaje, se establece una nueva meta y el proceso continúa.

Para promover la motivación para aprender, el nivel de desafío y el nivel de apoyo deben ser los adecuados —en términos de Vygotsky (1978), estar en la zona de desarrollo próximo de los y las estudiantes—, para lo cual se requiere que todas las decisiones que tomen profesores y profesoras y los propios estudiantes se basen en la información o evidencia sobre el aprendizaje recogidas continuamente¹⁰.

Como parte de la evaluación formativa, los programas proponen en algunas de las actividades un conjunto de criterios que permiten evaluar el desempeño de los y las estudiantes en un determinado aprendizaje. Estos criterios permiten identificar el lugar en que se encuentran los estudiantes en el desarrollo de las habilidades y la construcción de conocimientos, junto con entregar información que permita a la o el docente tomar decisiones pedagógicas para avanzar hacia el logro de los aprendizajes propuestos¹¹.

Los criterios de evaluación describen el dominio de conceptos, procedimientos y actitudes en los y las estudiantes. En su conjunto, permiten evaluar la comprensión y disposición o inclinación a actuar de acuerdo con el marco de Habilidades y actitudes para el siglo XXI. Cuando se integran en el desarrollo de la clase, los criterios de evaluación permiten generar un mejoramiento continuo del aprendizaje¹².

10 Mineduc (2017). Presentación de Criterios de evaluación, calificación y promoción al Consejo Nacional de Educación. Fundamentos a la propuesta de actualización de criterios y normas de Evaluación, Calificación y Promoción Escolar de estudiantes de Educación Regular presentada por la Unidad de Currículum y Evaluación al Consejo Nacional de Educación. Santiago, pág. 74.

11 Para la construcción de los criterios, se han considerado las orientaciones de la Mesa Covid Universitaria y la normativa vigente para la atención a la diversidad, la inclusión y la flexibilidad en la respuesta educativa contenida en la Ley N° 20.845 de Inclusión Escolar (Art. 1°, núm. i), y como referente los principios del Decreto N° 83 de 2015 y Decreto N° 67 de 2018.

12 Propuestas Educación Mesa Social Covid-19 (2021). Recomendación para una evaluación pertinente en tiempos de crisis. Santiago de Chile, pág. 65.

Referencias

- KNOWLES, M. S., HOLTON III, E. F., & SWANSON, R. A. (2014). *The adult learner: The definitive classic in adult education and human resource development*. London: Routledge.
- LARMER, J., MERGENDOLLER, J., BOSS, S. (2015). *Setting the Standard for Project Based Learning: A Proven Approach to Rigorous Classroom Instruction*. United States: ASCD.
- LEMOV, D. (2014). *Teach like a champion 2.0: 62 techniques that put students on the path to college*. San Francisco: John Wiley & Sons.
- MINEDUC (2017). *Presentación de Criterios de evaluación, calificación y promoción al Consejo Nacional de Educación*. Fundamentos a la propuesta de actualización de criterios y normas de Evaluación, Calificación y Promoción Escolar de estudiantes de Educación Regular presentada por la Unidad de Currículum y Evaluación al Consejo Nacional de Educación. Recuperado de <https://bibliotecadigital.mineduc.cl/handle/20.500.12365/14655>
- PROPUESTAS EDUCACIÓN MESA SOCIAL COVID-19 (2021). *Recomendación para una evaluación pertinente en tiempos de crisis*. Santiago de Chile.
- SWARTZ, R. (2017). *El Aprendizaje basado en el Pensamiento. Cómo desarrollar en los alumnos las competencias del siglo XXI*. España: SM.
- UNESCO (2015). *La Agenda para el Desarrollo Sostenible*. Recuperado de <https://www.un.org/sustainabledevelopment/es/development-agenda/>
- UNIVERSIDAD DEL DESARROLLO, CENTRO DE INNOVACIÓN. *Aprendizaje Basado en Problemas (ABP)*. Recurso web disponible en: <https://innovaciondocente.udd.cl/metodologias-activas/>

Propósitos Formativos de la Asignatura Pensamiento Computacional

El pensamiento computacional permite a personas jóvenes y adultas adquirir habilidades y conocimientos para resolver problemas, diseñar sistemas y comprender el mundo en que vivimos desde los conceptos fundamentales de la informática (Wing, 2006), competencias imprescindibles de la era digital y del siglo XXI.

Resolver problemas se orienta a que los y las estudiantes sean productores activos de conocimiento, por cuanto utilizan saberes y experiencias ya existentes para dar respuesta a una o más situaciones complejas. Desarrolla también el descubrimiento, al combinar reglas propias de la informática y aplicarlas a situaciones novedosas, auténticas y concretas. Asimismo, por medio de la resolución de problemas, los y las estudiantes desarrollan el pensamiento crítico, al discriminar y evaluar distintos conocimientos de la informática, las tecnologías, los sistemas de la información y el ámbito digital; y establecer condiciones para elaborar juicios informados y tomar decisiones como ciudadanos de un mundo en constante cambio y evolución.

En este escenario se promueve que las y los estudiantes diseñen sistemas de soluciones con impronta creativa, que aborden las situaciones problemáticas desde distintos puntos de vista y reúnan los datos necesarios y disponibles según las restricciones del contexto y los criterios de eficiencia y efectividad. Siguiendo esta lógica, la flexibilidad y perseverancia son actitudes del siglo XXI que potencian la propuesta de soluciones para satisfacer necesidades y deseos de manera sostenible.

En consecuencia, estas Bases favorecen el aprendizaje de conceptos de programación y de las posibilidades que los sistemas digitales y de información ofrecen, no solo desde la perspectiva de usuarios, sino también desde el rol de creadores de tecnologías, con una mirada hacia las crecientes transformaciones sociales que las ciencias de la computación y la inteligencia artificial producen constantemente. El pensamiento computacional se ha transformado progresivamente en la nueva alfabetización digital, por tanto, los conocimientos tecnológicos que poseen los y las estudiantes son la base para potenciar habilidades de resolución de problemas, pensamiento abstracto, pensamiento crítico, razonamiento lógico y analítico, creatividad, colaboración y comunicación.

Las Bases Curriculares de la asignatura Pensamiento Computacional del presente ciclo de formación de Educación Media, buscan contribuir a la educación de personas jóvenes y adultas con dominio de habilidades que les permitan utilizar, expresar y desarrollar sus ideas por medio de la tecnología de la información y la comunicación, a un nivel adecuado para su futuro laboral y sus estudios superiores, y logren ser participantes activos en el espacio digital (Mineduc, 2019).

Enfoque de la Asignatura

La asignatura pone énfasis en el desarrollo de habilidades y formas de razonamiento propias de la informática, aplicadas a la formulación y resolución de problemas de forma flexible, y a la proactividad para encontrar soluciones innovadoras.

Estas lógicas de razonamiento tienen su origen en las ciencias, pero principalmente en los pensamientos ingenieril, científico y matemático, cuyas implicancias se asocian a un modo de pensar, diseñar, probar y usar modelos computacionales, así como a una manera específica de abordar problemas del mundo real mediante la descomposición, la evaluación, la abstracción, la generalización y el pensamiento algorítmico. Con esto se busca conformar una estructura de pensamiento que integre los aprendizajes que los y las estudiantes han adquirido en Educación Básica, entre los que se destacan modelos, funciones o ecuaciones matemáticas; el análisis, la explicación y la representación de las partes de la célula; los sistemas de órganos del ser humano; la clasificación de los seres vivos en distintos grupos taxonómicos; la evaluación de diversos componentes de un circuito eléctrico o el diseño de experimentos.

En este sentido, aprender pensamiento computacional contempla la aplicación de un razonamiento que permite organizar y analizar datos de forma lógica y formular problemas que puedan ser resueltos mediante dispositivos electrónicos u otras herramientas. Así también, implica representar datos de manera abstracta con modelos y simulaciones; identificar y analizar posibles soluciones con el objeto de encontrar la combinación de pasos de manera eficiente y efectiva; implementar soluciones sobre la base de una serie de pasos ordenados y generalizar y transferir ese proceso de solución a nuevos problemas.

Junto con esto, esta asignatura brinda un contexto para el estudio de la programación de gran alcance, con una aplicación que va más allá de la tecnología y la programación en sí misma, entendiendo que esta corresponde a una de las prácticas más visibles del pensamiento computacional. Esta perspectiva posibilita profundizar en el desarrollo de competencias necesarias para la vida laboral, como son las competencias digitales asociadas al manejo de la información, la colaboración en línea, el ejercicio de la ciudadanía digital, la creación de contenidos en línea y la resolución de problemas en y con internet.

Las experiencias cotidianas de las y los estudiantes de la educación de personas jóvenes y adultas permiten situar y dar sentido a lo que sucede en un mundo fuertemente influenciado por las tecnologías. En este contexto, el desarrollo de las habilidades del pensamiento computacional contribuye a propiciar un rol activo, honesto y responsable frente a la sociedad. En el que se releva la importancia de la dimensión ética y de reflexionar en torno a temas controversiales, como las brechas de género que existen en el acceso a la programación y al uso de tecnologías digitales, los desafíos en términos de inclusión, gestión de la diversidad y oportunidades para personas con necesidades especiales que plantea hoy la tecnología. De esta manera, el enfoque de la asignatura favorece una mejor comprensión de la realidad social y promueve una actitud activa en su mejoramiento.

Estructura Curricular

Las Bases Curriculares de la asignatura Pensamiento Computacional se articulan en torno a Objetivos de Aprendizaje de habilidades y actitudes, los cuales fortalecen disposiciones para un saber hacer que integre el respeto a principios éticos y valores fundamentales de la sociedad. Los Objetivos de Aprendizaje se entrelazan con conocimientos esenciales que buscan focalizar la comprensión de las ideas y procesos fundamentales de la asignatura. Las grandes ideas operan como propósito formativo de los niveles 1 y 2 de Educación Media, orientando la comprensión y la articulación de los Objetivos de Aprendizaje y los conocimientos esenciales.

Los Objetivos de Aprendizaje de la asignatura se organizan en los niveles 1 y 2 para desarrollar las habilidades y lógicas de razonamiento del pensamiento computacional, que se fundamentan en las grandes ideas de la informática y en el valor de la abstracción como forma de pensamiento diferenciadora. Estos objetivos integran actitudes de los ámbitos que organizan las Habilidades para el siglo XXI, de acuerdo con un criterio de pertinencia para ser trabajadas integradamente con las habilidades.

Para esta asignatura se han organizado los Objetivos de Aprendizaje en los siguientes ejes¹³:

- Descomponer
- Evaluar
- Modelar
- Generalizar y transferir
- Pensar de forma algorítmica

Descomponer

Pensar en términos de descomponer facilita la comprensión de las tecnologías artefactuales considerando cada una de sus partes o componentes. Esto implica que cada una de las partes debe comprenderse, desarrollarse y evaluarse por separado.

En este eje se desarrollan las actitudes del ámbito Maneras de pensar. En Educación Media se focaliza en promover la actitud de pensar con apertura a distintas perspectivas y contextos, asumiendo riesgos y responsabilidades.

13 Basado en Csizmadia, A. (2015). *Computational Thinking: A Guide for Teachers*. UK: Hodder Education.

Evaluar

Implica desarrollar la habilidad de pensar para asegurar que una solución, ya sea un algoritmo, un sistema o un proceso, cumpla eficientemente con su propósito de acuerdo con el contexto donde se presenta el problema.

En este eje se desarrollan las actitudes del ámbito Maneras de trabajar. Así, en Educación Media se promueve trabajar con honestidad en la realización de tareas colaborativas con miras al logro de una meta común.

Modelar

Modelar es una habilidad que permite hacer más comprensible un sistema u objeto del mundo real. Facilita comprender y estudiar con mayor eficiencia uno o más problemas al extraer los elementos que son más relevantes de una situación.

En este eje se desarrollan las actitudes del ámbito Herramientas para trabajar. Esto implica que en Educación Media se favorece el aprovechar las herramientas disponibles para aprender y resolver problemas.

Generalizar y transferir

La generalización facilita identificar patrones, similitudes y conexiones con la intención de transferir las maneras de solucionar un problema a otras situaciones similares de manera eficaz y eficiente.

En este eje se desarrollan las actitudes del ámbito Maneras de vivir en el mundo. Así, en la educación media se promueve que los estudiantes actúen con honestidad, responsabilizándose por las propias acciones y decisiones con conciencia de las implicancias que estas tienen sobre uno mismo y los demás.

Pensar de forma algorítmica

El pensamiento algorítmico permite llegar a una solución a través de una identificación clara de pasos y reglas, cuando se está frente a problemas que tienen que ser resueltos de forma continua, de manera que no se tengan que pensar una y otra vez los mismos caminos para llegar a resolverlos.

En este eje se desarrollan las actitudes del ámbito de las Herramientas para trabajar. En el caso de Educación Media, implica que los y las estudiantes actúen de acuerdo con los principios de la ética en el uso de la información y de la tecnología, con respeto por la propiedad intelectual y la privacidad de las personas.

Objetivos de Aprendizaje

Se espera que los y las estudiantes sean capaces de:

OA 1

Formular problemas a partir de un contexto o situación específica, con apertura a distintas perspectivas, identificando las partes que lo componen. **(Descomponer)**

OA 2

Organizar datos de una situación problemática estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función. **(Evaluar)**

OA 3

Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponibles. **(Modelar)**

OA 4

Establecer conexiones y semejanzas entre las partes del problema para caracterizar patrones en una situación problemática dada. **(Generalizar y transferir)**

OA 5

Aplicar flexiblemente las soluciones elaboradas e implementadas a otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista. **(Generalizar y transferir)**

OA 6

Elaborar e implementar soluciones a una situación problemática mediante la formulación de algoritmos, actuando de acuerdo con los principios de la ética. **(Pensar de forma algorítmica)**

Conocimientos Esenciales

- Procesos y funcionamiento de las cosas para resolver problemas cotidianos.
- Modelos entidad relación y simulación de sistemas.
- Sistemas computacionales: dispositivos, hardware, software y resolución de problemas.
- Redes e internet: sitios web, aplicaciones móviles, seguridad cibernética y objetos controlados digitalmente.
- Algoritmos y programación: lenguajes, métodos y entornos de programación.
- Representación y análisis de datos.
- Impacto social de la computación: interacciones, cultura, seguridad y ética.

Orientaciones Didácticas y Pedagógicas

DECRETO EN TRÁMITE

Orientaciones Didácticas y Pedagógicas

Las orientaciones que se presentan a continuación destacan elementos que son relevantes al momento de implementar el programa. Estas orientaciones se vinculan estrechamente con el logro de los Objetivos de Aprendizaje. En consecuencia, al organizar el proceso de aprendizaje, se deben considerar los siguientes factores:

► INTEGRACIÓN DE LOS OBJETIVOS DE APRENDIZAJE:

Las actividades modélicas de este programa, en numerosas oportunidades, consideran el desarrollo de los Objetivos de Aprendizaje de manera integrada. De igual manera, las actitudes se deben trabajar conjuntamente con dichos objetivos.

► PLANIFICACIÓN:

Al inicio de cada unidad, es necesario incluir un tiempo para rescatar experiencias y conocimientos previos, pues son el punto de partida para implementar adecuadamente las actividades. Asimismo, se debe considerar un enfoque participativo y colaborativo donde las y los estudiantes sean protagonistas de sus aprendizajes. Durante el desarrollo de las actividades, se debe promover una relación positiva con el aprendizaje entre los y las estudiantes. Para esto, es necesario que se consideren instancias de retroalimentación de los procesos de expresión, creación y apreciación, y aprovechar descubrimientos, errores y dificultades como fuente de nuevos aprendizajes y logros. Para orientar y apoyar al estudiantado en la superación y solución de dificultades, es importante que conozcan y comprendan los criterios de evaluación al inicio de cada actividad. Al finalizar cada módulo, se debe considerar un tiempo adecuado para que expongan, justifiquen y enjuicien críticamente sus proyectos, algoritmos o programas. Esto les permitirá observar sus progresos y valorar sus trabajos y los de sus pares.

► DESARROLLO DEL RAZONAMIENTO LÓGICO:

El razonamiento lógico permite dar sentido a los problemas mediante el análisis y la comprobación de hechos para aprender a pensar de forma efectiva y clara. Esto implica que el o la docente propicie que sus estudiantes pongan en tensión sus propios conocimientos y modelos internos para predecir y sacar conclusiones respecto de una situación o más situaciones que requieran solución. El razonamiento lógico es sumamente relevante en la asignatura, pues permite que el estudiantado depure sus propios códigos para aislar errores e idear mejoras empleando la abstracción, la evaluación y el pensamiento algorítmico.

► TECNOLOGÍAS:

El o la docente debe considerar que las herramientas tecnológicas computacionales actuales varían en su efectividad a través del tiempo. Es imperativo para esta asignatura permitir progresivamente mayor interacción con diferentes componentes artefactuales y también humanos, e impulsar el desarrollo de nuevas herramientas diseñadas y elaboradas expresamente para fomentar el pensamiento computacional con miras a un uso contextualizado y cotidiano. Es necesario que el estudiantado comprenda que los computadores no funcionan a menos que un programa se esté ejecutando y que un amplio rango de dispositivos pueden considerarse computadores. Por lo tanto, es importante establecer la relación entre las personas, la tecnología y los usos, y reconocer que diferentes tecnologías responden a diferentes o iguales necesidades.

Dentro de este uso y la comunicación que existe entre usuarios, computadores, programadores y empresas, la ciberseguridad o seguridad de la información en el mundo digital es compleja. Involucra invasión de la privacidad, aumento de la vigilancia de datos, de las transacciones comerciales y de las cívicas digitales¹⁴. Esto es pertinente para los y las estudiantes en su rol de programadores y usuarios de programas, pues deben actuar con responsabilidad. En esta asignatura y a nivel instrumental, se sugiere la lectura de las políticas de privacidad y la protección de datos personales: tener presente el alcance del uso de los datos y del tipo de datos personales a los que se puede acceder preguntando a los usuarios¹⁵. En cuanto a los consejos de autocuidado, el estudiantado debe tener claro que las personas son las dueñas de los datos personales y no quienes las administran. Esto significa que hay derechos y deberes sobre el uso y los fines que se les da a los datos¹⁶.

► **APRENDIZAJE COLABORATIVO:**

En la elaboración de las múltiples tareas de la asignatura, es importante favorecer la comunicación y la colaboración entre los y las estudiantes, tanto al momento de programar, diseñar o ejecutar un programa. Esto permite profundizar y compartir la construcción de aprendizajes, junto con potenciar habilidades y actitudes transversales como el respeto por la opinión de sus pares y el descubrimiento de la riqueza presente en la diversidad.

► **CIUDADANÍA Y ALFABETIZACIÓN DIGITAL:**

Con la influencia cada vez más creciente de las tecnologías en la sociedad, es sumamente relevante favorecer la comprensión del funcionamiento y los principios que rigen el entorno digital. Asimismo, es necesario que en todo momento se pueda realizar un análisis del rol que juegan las tecnologías en la sociedad, la incidencia en la propia vida cotidiana y la de los demás, así como el papel que ocupan en la construcción del conocimiento. Es necesario que la tecnología esté sujeta a valoración en cuanto a su uso y creación, por lo que es imprescindible la reflexión crítica y ética sobre la misma.

Ciudadanía digital se entiende como el derecho a participar en la sociedad en línea. Esta considera el acceso, la inclusión y las libertades comunicativas. Una vez que se tiene acceso, el o la estudiante debe focalizar su desarrollo como ciudadano digital de acuerdo con ciertas condiciones, como el descifrado de las fuentes de las noticias o bien la construcción de las identidades digitales¹⁷. El estudiantado debe tomar en cuenta las fuentes, la interpretación de la información, los flujos locales y globales, y especialmente anticipar posibles consecuencias de lo expuesto como ciudadano digital.

En este sentido, la alfabetización digital es esencial para formar ciudadanos capaces de desenvolverse en entornos tecnológicos de manera crítica y responsable. Se entiende como un proceso educativo que permite comprender la información, las normas y las convenciones que regulan el uso de tecnologías, así como las formas de evidenciar su apropiación. Esta asignatura se centra en la alfabetización mediante el aprendizaje de la programación, integrando principios y prácticas que favorecen el ejercicio de derechos digitales y la participación activa en la convivencia democrática¹⁸.

14 Pangrazio, L. & Sefton-Green, J. (2021). Digital Rights, digital Citizenship and Digital Literacy: What's the Difference? *Journal of new approach in education research*, 10(1). 15-27.

15 Servicio Nacional del Consumidor. <https://www.sernac.cl/portal/617/w3-article-53061.html>

16 Servicio Nacional del Consumidor. <https://www.sernac.cl/portal/607/w3-article-3389.html>

17 Pangrazio, L. & Sefton-Green, J. (2021). Digital Rights, digital Citizenship and Digital Literacy: What's the Difference? *Journal of new approach in education research*, 10(1). 15-27.

18 Ministerio de Educación de Chile. (2023). Ciudadanía digital para los desafíos de las comunidades educativas. Centro de Innovación MINEDUC.

Orientaciones para la evaluación en Pensamiento Computacional

La evaluación ayuda a determinar las fortalezas y debilidades que surgen durante el proceso de enseñanza y aprendizaje. Con la información proporcionada por las evaluaciones, se pueden tomar decisiones para adecuar la planificación a las características y necesidades del estudiantado. Además, cada estudiante podrá focalizar sus esfuerzos con la confianza de mejorar sus resultados a partir de la información que la evaluación proporciona.

En Pensamiento Computacional, como en todas las asignaturas, es muy importante contar con criterios de evaluación claros y conocidos por los y las estudiantes. En este programa, se proporcionan ejemplos de criterios y pautas de evaluación, los que se recomienda contextualizar según cada situación de aprendizaje.

Para poder evaluar los Objetivos de Aprendizaje, se debe considerar que estos tributan a habilidades y actitudes que buscan configurar de forma dinámica el desarrollo del pensamiento computacional. Esto se realiza mediante propuestas de aprendizaje que refieren a procedimientos individuales y colaborativos, los que requieren ser monitoreados, acompañados y considerados en una lógica de proceso sistemática y rigurosa. Por lo tanto, es necesario favorecer prácticas de evaluación que consideren:

1. Colaborar en torno a la informática.
2. Reconocer y definir problemas computacionales.
3. Desarrollar y usar abstracciones.
4. Crear artefactos computacionales.
5. Probar la tecnología disponible.
6. Comunicar por medio de tecnologías.

En este sentido, para que la evaluación contribuya efectivamente a lograr los aprendizajes, se deben aplicar gradualmente los criterios o pautas de evaluación, tanto durante el desarrollo del proceso como en la presentación de un producto final. En estricto rigor, todas las actividades que se realizan deben ser evaluadas; solo así el o la docente puede asegurar su pertinencia para el logro de un OA. Estas evaluaciones no deben traducirse necesariamente en una calificación; su fin es proporcionar evidencia acerca del grado de avance en el logro del o los Objetivos de Aprendizaje y apoyar o reorientar el trabajo de los y las estudiantes en caso de ser necesario.

Para lograr un proceso de evaluación efectivo es necesario realizar un diseño previo, para lo cual se sugiere responder a preguntas como:

- ¿Cuáles son los Objetivos de Aprendizaje y/o metas que abarcará la evaluación?
- ¿Qué criterios son adecuados a las actividades efectivamente realizadas y los propósitos de la evaluación?
- ¿Qué evidencias en el trabajo de las y los estudiantes permiten demostrar los logros esperados?
- ¿Cuáles son los criterios para definir la calidad en un logro de aprendizaje?
- ¿Qué estrategias y procedimientos son los adecuados?
- ¿Quién o quiénes realizarán la evaluación?
- ¿En qué momentos se hará?
- ¿Cómo se comunicarán los resultados?

Procedimientos de evaluación

Los procedimientos son los métodos o situaciones que se consideran útiles para recoger evidencia en relación con el avance de los aprendizajes. A continuación, se sugieren algunos procedimientos que diversifican y actualizan la forma en que se puede desarrollar este proceso y evaluar el logro de los OA en la asignatura, tanto en instancias formativas como con fines sumativos.

- ▶ **PORTAFOLIO.** Permite evaluar el rendimiento de los y las estudiantes de forma auténtica en el aula. Las tareas y actividades de aprendizaje propuestas pueden flexibilizarse en busca de una solución o respuesta a un problema o pregunta a resolver, lo que permite al estudiantado demostrar sus aprendizajes de diferentes formas. Así se facilita evidenciar la creatividad, la motivación, el razonamiento y las capacidades que van desarrollando progresivamente. El portafolio, concretamente, puede ser una carpeta tradicional o una carpeta digital, donde se guardan y organizan trabajos prácticos, actividades de clase o elementos de un proyecto. A este se puede acceder fácilmente, constituyéndose en evidencia para el o la docente y, a la vez, en una forma de autoevaluación para el o la estudiante. El portafolio es útil para medir el uso de algoritmos, las abstracciones, el modelamiento y la resolución de problemas, que generalmente son difíciles de evaluar por medio de preguntas abiertas u opciones múltiples.
- ▶ **PROYECTOS.** Consiste en plantear un problema o una situación que requiera una solución informática para resolver problemas sencillos o complejos, para efectuar una investigación guiada o para modelar un problema real. Es necesario fijar con precisión los objetivos, considerar resultados abiertos, identificar los recursos necesarios, favorecer el trabajo en equipo y conectar de forma ideal diferentes áreas del conocimiento. Se recomienda que para cada fase del proyecto se tenga una pauta de evaluación que oriente la recolección de evidencia durante el proceso, especialmente cuando los y las estudiantes desarrollen algoritmos.
- ▶ **DESARROLLO DE PROGRAMAS.** Permite evaluar el proceso de diseño de un programa que frecuentemente se repite hasta que la o el estudiante haya dado con la solución necesaria. Con este procedimiento, el estudiantado aprende por qué las personas desarrollan programas, las ventajas y desventajas del diseño de un algoritmo, y cómo las decisiones del programador involucran restricciones, eficiencia, ética y pruebas de usuarios. Como todo procedimiento, debe realizarse usando una pauta de evaluación basada en criterios compartidos previamente con sus estudiantes. Es importante considerar que el diseño del programa, así como los algoritmos necesarios y el lenguaje propiamente tal, pueden ser desarrollados usando un computador o un aparato similar, o bien con recursos análogos.

Una recomendación importante es que, al construir una pauta de evaluación, ya sea basada en ítems de valoración o rúbricas, es necesario tener presente que las diferentes dimensiones de un desempeño (proyecto, código, modelos u otro) se ponderen de acuerdo a su relevancia en el Objetivo de Aprendizaje. Esto significa, por ejemplo, que si el Objetivo de Aprendizaje pone énfasis en la descomposición, no puede asignarse a este aspecto el mismo peso (puntaje máximo) que a aspectos menos relevantes en dicho objetivo, como podría ser la destreza en la ejecución de un procedimiento (ingresar datos).

Orientaciones para la contextualización

El programa propone cuatro módulos alineados con los Objetivos de Aprendizaje para los que se consideran actividades modélicas. Los y las docentes pueden contextualizar y adaptar esta propuesta a su realidad educativa, pues los módulos brindan la oportunidad de ser desarrollados con las herramientas tecnológicas disponibles, como equipos informáticos, o con recursos análogos. Dado este contexto, cada módulo tiene una orientación hacia el uso de las tecnologías digitales disponibles, sin embargo, cada actividad puede ser resuelta utilizando materiales concretos, como bloques, cubos o lápiz y papel. Lo importante es que las actividades siempre estén en directa relación con los Objetivos de Aprendizaje.

Las actividades propuestas promueven el pensamiento algorítmico, el análisis de datos, la creatividad y la resolución de problemas. Consideran las características, intereses y contextos de los y las estudiantes, así como también la infraestructura y recursos disponibles en el establecimiento educacional. Esto significa seleccionar recursos, plataformas, juegos, aplicaciones o simulaciones dentro de un entorno significativo para el estudiantado, que permita, a su vez, recopilar datos, analizar los logros y comunicar resultados.

Es importante que cada docente adapte los diferentes elementos de su planificación (por ejemplo: tiempos o recursos), cuando en su curso tenga estudiantes que presenten algún tipo de necesidad educativa especial. Esto es particularmente necesario en el caso de estudiantes con capacidades visuales, auditivas o de movilidad limitadas o diferentes. El o la docente podrá buscar estrategias y adaptaciones pedagógicas para desarrollar las habilidades propias de la asignatura, pudiendo usar recursos concretos o digitales para modelar problemas, desarrollar códigos, analizar datos o evaluar el impacto de la tecnología en la sociedad.

En cuanto a la diversidad cultural, al planificar e implementar las actividades, es necesario tener presentes las culturas de origen del curso, prestando atención a los problemas que se plantean y a aspectos relacionados con las brechas digitales que puedan existir en cuanto a conectividad o acceso a equipos informáticos. En ambos casos, se debe procurar la participación y apoyo de todos, lo que los preparará para comprender, aceptar y valorar la diversidad.

Visión panorámica*

de los Módulos** de Educación Media

Módulos Obligatorios

Módulo Obligatorio 1 EM

¿Cómo descomponer problemas para evaluarlos usando diferentes fuentes de información, modelos y herramientas?

Módulo Obligatorio 2 EM

¿Cómo se modelan los niveles de interacción entre los sistemas computacionales y los sistemas sociales, ambientales o industriales que enfrenta un usuario?

Módulo Obligatorio 3 EM

¿Cómo transferir la tipología de sistemas y redes para solucionar problemas de seguridad, acceso y control de datos?

Módulo Obligatorio 4 EM

¿Por qué las personas evalúan y seleccionan algoritmos en función del rendimiento, reutilización y la facilidad de implementación para resolver un problema?

Módulos Electivos***

Módulo Electivo 1 EM

¿Cómo crear un programa computacional que genere un impacto amplio a nivel social?

Módulo Electivo 2 EM

¿Cómo el uso de la tecnología impacta en las oportunidades de las personas en la sociedad?

* Los Programas de Estudio de cada asignatura se organizan por nivel en cuatro módulos obligatorios y dos módulos electivos, de los cuales podrá cursarse al menos uno de acuerdo a las necesidades de las trayectorias formativas de las y los estudiantes.

** Se propone organizar flexiblemente los módulos con una duración de seis semanas cada uno para dar cuenta del cumplimiento de las 36 semanas lectivas del tiempo escolar de la modalidad.

*** En el Programa de Estudio de la asignatura se desarrollarán como opciones dos electivos que profundizan las grandes ideas abordadas en los Módulos Obligatorios.

Visión panorámica

de los Objetivos de Aprendizaje y Conocimientos Esenciales

Módulo Obligatorio 1

Gran idea

Los problemas pueden descomponerse en partes más pequeñas y manejables, organizarse siguiendo patrones, y solucionarse en base a reglas, modelos y algoritmos.

Objetivos de aprendizaje

OA 1

Formular problemas a partir de un contexto o situación específica, con apertura a distintas perspectivas, identificando las partes que lo componen. **(Descomponer)**

OA 2

Organizar datos de una situación problemática estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función. **(Evaluar)**

OA 3

Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponibles. **(Modelar)**

Conocimiento esencial

- Procesos y funcionamiento de las cosas para resolver problemas cotidianos.
- Modelos entidad relación y simulación de sistemas.

Tiempo estimado:

6 semanas (12 horas)

Módulo Obligatorio 2

Gran idea

El comportamiento de un sistema computacional (dispositivos, hardware y software) depende de la secuencia de instrucciones dadas por el usuario que automatizarán el desempeño de una tarea para ejecutar una acción o resolver un problema.

Objetivos de aprendizaje

OA 2

Organizar datos de una situación problemática estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función. **(Evaluar)**

OA 3

Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponibles. **(Modelar)**

Conocimiento esencial

- Modelos entidad relación y simulación de sistemas.
- Sistemas computacionales: dispositivos, hardware y software, y resolución de problemas.

Tiempo estimado:

6 semanas (12 horas)

Módulo Obligatorio 3

Gran idea

El comportamiento de un sistema computacional (dispositivos, hardware y software) depende de la secuencia de instrucciones dadas por el usuario que automatizarán el desempeño de una tarea para ejecutar una acción o resolver un problema.

Objetivos de aprendizaje

OA 4

Establecer conexiones y semejanzas entre las partes del problema para caracterizar patrones en una situación problemática dada. **(Generalizar y transferir)**

OA 5

Aplicar flexiblemente las soluciones elaboradas e implementadas a otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista. **(Generalizar y transferir)**

OA 6

Elaborar e implementar soluciones a una situación problemática mediante la formulación de algoritmos, actuando de acuerdo con los principios de la ética. **(Pensar de forma algorítmica)**

Conocimiento esencial

- Redes e Internet: sitios web, aplicaciones móviles, seguridad cibernética y objetos controlados digitalmente.
- Algoritmos y programación: lenguajes, métodos y entornos de programación. Representación y análisis de datos.

Tiempo estimado:

6 semanas (12 horas)

Módulo Obligatorio 4

Gran idea

El pensamiento algorítmico permite organizar pasos y reglas para diseñar soluciones y resolver una diversidad de problemas cuya comprensión se favorece al integrar conceptos matemáticos como variables, funciones, ecuaciones o gráficos, entre otros.

Objetivos de aprendizaje

OA 4

Establecer conexiones y semejanzas entre las partes del problema para caracterizar patrones en una situación problemática dada. **(Generalizar y transferir)**

OA 5

Aplicar flexiblemente las soluciones elaboradas e implementadas a otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista. **(Generalizar y transferir)**

OA 6

Elaborar e implementar soluciones a una situación problemática mediante la formulación de algoritmos, actuando de acuerdo con los principios de la ética. **(Pensar de forma algorítmica)**

Conocimiento esencial

- Algoritmos y programación: lenguajes, métodos y entornos de programación. Representación y análisis de datos.
- Impacto social de la computación: interacciones, cultura, seguridad y ética.

Tiempo estimado:

6 semanas (12 horas)

Módulo Electivo 1 Aprendizaje Basado en Proyectos

Gran idea

El comportamiento de un sistema computacional (dispositivos, *hardware* y *software*) depende de la secuencia de instrucciones dadas por el usuario que automatizarán el desempeño de una tarea para ejecutar una acción o resolver un problema.

Objetivos de aprendizaje

OA 2

Organizar datos de una situación problemática estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función. **(Evaluar)**

OA 3

Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponibles. **(Modelar)**

Conocimientos esenciales

- Modelos entidad-relación y simulación de sistemas.
- Sistemas computacionales: dispositivos, *hardware* y *software*, y resolución de problemas.

Tiempo estimado:

6 semanas (12 horas)

Módulo Electivo 2 Aprendizaje Basado en Resolución de Problemas

Gran idea

El pensamiento algorítmico permite organizar pasos y reglas para diseñar soluciones y resolver una diversidad de problemas cuya comprensión se favorece al integrar conceptos matemáticos como variables, funciones, ecuaciones o gráficos, entre otros.

Objetivos de aprendizaje

OA 4

Establecer conexiones y semejanzas entre las partes del problema para caracterizar patrones en una situación problemática dada. **(Generalizar y transferir)**

OA 5

Aplicar flexiblemente las soluciones elaboradas e implementadas a otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista. **(Generalizar y transferir)**

OA 6

Elaborar e implementar soluciones a una situación problemática mediante la formulación de algoritmos, actuando de acuerdo con los principios de la ética. **(Pensar de forma algorítmica)**

Conocimientos esenciales

- Algoritmos y programación: lenguajes, métodos y entornos de programación. Representación y análisis de datos.
- Impacto social de la computación: interacciones, cultura, seguridad y ética.

Tiempo estimado:

6 semanas (12 horas)

Módulos Obligatorios de la Asignatura

Módulo Obligatorio 1

Módulo Obligatorio 2

Módulo Obligatorio 3

Módulo Obligatorio 4

Visión panorámica

Gran idea

Los problemas pueden descomponerse en partes más pequeñas y manejables, organizarse siguiendo patrones y solucionarse en base a reglas, modelos y algoritmos.

Objetivos de Aprendizaje

OA 1

Formular problemas a partir de un contexto o situación específica, con apertura a distintas perspectivas, identificando las partes que lo componen. **(Descomponer)**

OA 2

Organizar datos de una situación problemática estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función. **(Evaluar)**

OA 3

Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponibles. **(Modelar)**

Conocimientos esenciales

- Procesos y funcionamiento de las cosas para resolver problemas cotidianos.
 - Modelos entidad-relación y simulación de sistemas.
-

Tiempo estimado: 6 semanas (12 horas)

Propósito

En el módulo 1 de la asignatura Pensamiento Computacional de los Niveles 1 y 2 de Educación Media se espera que los y las estudiantes comprendan que, frente a cualquier situación, es necesario entender el problema y hacer un análisis previo de lo que considera el mismo, para poder dar la mejor respuesta posible y satisfacer así las necesidades de forma eficiente y óptima. Asimismo, todo problema puede resolverse por medio de una serie de algoritmos, por ejemplo: los seres humanos en sus rutinas cotidianas, al momento de tener que resolver problemas, siguen una serie de acciones, conscientes o inconscientes, en las que evalúan, organizan o clasifican para llegar a una solución, estas acciones pueden constituir perfectamente un algoritmo. Dicho esto, en el módulo las y los estudiantes distinguirán que la pieza básica del algoritmo es la descomposición y que un problema puede descomponerse en partes pequeñas y abordables que lo conforman en su generalidad.

Los Objetivos de Aprendizaje del módulo 1 buscan desarrollar las habilidades de descomposición y evaluación

a través del modelamiento de base de datos. Para comprender estos conceptos desde las ciencias de la computación, los y las estudiantes pueden organizar la información de manera efectiva y eficiente, pues, al hacer un modelo de base de datos, también descomponen información para darle un sentido. En este módulo podemos entender que el algoritmo y la base de datos no son tan diferentes, pues pueden ser descompuestas, evaluadas y modeladas.

Los Objetivos de Aprendizaje del módulo 1 desarrollan las actitudes del siglo XXI: Maneras de pensar, Maneras de vivir el mundo y las Herramientas para trabajar, promoviendo la utilización de las herramientas y el uso de la información para resolver problemas, diseñar soluciones e interactuar con sistemas en diferentes contextos de la vida diaria. Finalmente, las actividades propuestas son una oportunidad que incentiva la ludificación del espacio de aprendizaje desde contextos reales y cercanos para el uso de las tecnologías y la alfabetización digital.

Ruta de Aprendizaje

¿Cómo descomponer problemas para evaluarlos usando diferentes fuentes de información, modelos y herramientas?

●	Actividad de desempeño 1	Construyen algoritmos en base a la descomposición de los problemas que surgen en la cotidianidad, al tiempo que evalúan las respuestas y toman decisiones respecto de si son las más acertadas.
●	Actividad de desempeño 2	Evalúan las mejores opciones al momento de resolver un problema teniendo en cuenta el algoritmo y pseudocódigo. Además, aprovechan oportunidades para poder utilizar códigos para resolver situaciones-problema.
●	Actividad de desempeño 3	Modelan bases de datos en contextos que plantean la necesidad de soluciones con apertura a diferentes perspectivas, y clasifican y descomponen objetos cotidianos haciendo una relación simple entre ellos.
●	Actividad de desempeño 4	Resuelven problemas en base a la entidad-relación entre tablas de acuerdo con el modelamiento de bases de datos, fundamentando cada atributo en base al dato que almacenará.

Actividad de desempeño

1

Propósito

Esta actividad se dedicará a los fundamentos básicos del funcionamiento del algoritmo. La meta consiste en poder descomponer problemas en base a la lógica computacional de un algoritmo y llevar a cabo esas instrucciones a una codificación centralizada para resolver el problema principal.

Objetivos de aprendizaje

OA 1

Formular problemas a partir de un contexto o situación específica, con apertura a distintas perspectivas, identificando las partes que lo componen. ([Descomponer](#))

OA 2

Organizar datos de una situación problemática, estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función. ([Evaluar](#))

Conocimiento esencial

- Procesos y funcionamiento de las cosas para resolver problemas cotidianos.
-

Tiempo estimado: 3 horas pedagógicas

Desarrollo de la actividad

SITUACIÓN EXPERIENCIAL

El o la docente presenta a sus estudiantes dos casos: el de Carolina, quien estudió mucho para un examen muy importante de Matemática, y el caso de José, quien en vez de estudiar para dicho examen, decidió dormir. Ambos casos se presentan de forma algorítmica donde se muestra el resultado de sus decisiones:



Algunas de las preguntas que pueden promover el proceso de comunicar una argumentación sobre el tipo de conductas de los dos sujetos son:

- ▶ ¿Qué algoritmo representa la aprobación del examen?
- ▶ ¿Cuál es el proceso que hizo Carolina?
- ▶ ¿Cuál es el proceso que hizo José?
- ▶ ¿Cómo se podría hacer un algoritmo donde uno elige si dormir o no?
- ▶ ¿Cuál sería la salida de Carolina y José respectivamente?

CONSTRUCCIÓN DE CONOCIMIENTO

Para construir el conocimiento sobre los pasos del algoritmo (entrada, proceso y salida) y su descomposición, se sugiere comenzar con un acercamiento de la entrada, el proceso y la salida, con uso de un lenguaje verbal o bien un programa en bloques, donde describen qué se tiene por entrada, cuál sería el proceso y qué se obtiene en la salida.



Entrada	La tortuga Kira parte en el punto verde.
Proceso	• La tortuga avanza en LINEA RECTA 100 pasos HACIA LA DERECHA • La tortuga se gira en 90° • La tortuga avanza en LINEA RECTA 100 pasos • La tortuga se gira en 90° • La tortuga avanza en LINEA RECTA 100 pasos • La tortuga se gira en 90° • La tortuga avanza en LINEA RECTA 100 pasos • La tortuga se gira en 90° • La tortuga se para. Los ángulos se entienden en el sentido positivo (contra el reloj).
Salida	La tortuga Kira SE ENCUENTRA EN el punto verde.

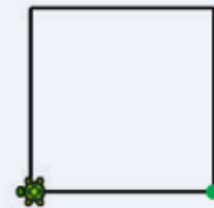
Para motivar la creación y elaboración de un pseudocódigo, se sugiere considerar los movimientos y algunas convenciones del lenguaje, por ejemplo, utilizar solo tres mayúsculas para los movimientos: AVA para avanzar, RET para retroceder, GIR para girar; incluir paréntesis para indicar con números enteros la cantidad de pasos y para indicar una rotación se utiliza una coma para indicar la unidad de medida en grados.

Entrada	La tortuga Kira parte en el punto verde.
Proceso	• AVA (100) • GIR (90, grados) • AVA (100) • GIR (90, grados) • AVA (100) • GIR (90, grados) • AVA (100) • GIR (90, grados) Los ángulos se entienden en el sentido positivo (contra el reloj).
Salida	La tortuga Kira SE ENCUENTRA EN el punto verde.

Se sugiere comparar este pseudocódigo con la programación escrita en inglés en bloques en SCRATCH y ver diferencias o similitudes entre ambos lenguajes.

¿Qué similitudes o diferencias hay entre el pseudocódigo y el programa?

Seudocódigo	Programa en bloques
• AVA (100) • GIR (90, grados) • AVA (100) • GIR (90, grados) • AVA (100) • GIR (90, grados) • AVA (100) • GIR (90, grados) Los ángulos se entienden en el sentido positivo (contra el reloj).	
• Coinciden en los movimientos de avanzar en 4 ocasiones y girar en 90 grados en 4 ocasiones. • Difieren en el sentido del giro, va en el sentido del reloj y la tortuga va en otra dirección. En la imagen se ve que difieren también del punto de partida. En la instrucción del programa, se indica el punto de partida como la instrucción “<i>pen down</i>” que significa “poner la punta del lápiz en el papel”.	



PRÁCTICA GUIADA

Se sugiere elaborar un pseudocódigo y las nociones de entrada, proceso y salida para algunos procedimientos de organización del tiempo que son utilizados cotidianamente. Por ejemplo, la planificación de la lectura de un libro en un periodo determinado.

¿Cuántas páginas debo leer cada día para terminar el libro a tiempo?

Teniendo en cuenta que:

- El libro tiene 120 páginas y quiero terminarlo en 10 días.
- Estos datos se pueden escribir como 120 y 10 para facilitar la lectura.
También puedo incorporar un día extra de repaso o imprevistos, dejando solo 11 días reales de lectura.
- Esto se puede escribir como 11 para facilitar el cálculo ajustado.

Para guiar a sus estudiantes en la elaboración de respuestas, se sugiere que utilicen los términos de entrada, proceso y salida para resolver la problemática planteada, elaborar un pseudocódigo con lo mencionado anteriormente y así poder analizar las diferencias entre escribir los pasos de un algoritmo y un pseudocódigo, el cual podría hacerse mediante PSeint¹⁹, que es un software de escritura de códigos, o bien con una hoja de papel.

Fase 1 – Confeccionar una lista para las fases de un algoritmo

Acción	Resultado esperado	Preguntas de apoyo						
Confeccionar una lista con datos del libro (páginas) y tiempo disponible (días).	Identificar la entrada (total páginas, días) y salida (páginas por día) <table border="1" data-bbox="483 1120 1157 1344"> <thead> <tr> <th data-bbox="483 1120 711 1167">Entrada</th><th data-bbox="711 1120 954 1167">Proceso</th><th data-bbox="954 1120 1157 1167">Salida</th></tr> </thead> <tbody> <tr> <td data-bbox="483 1167 711 1344"> Total páginas - 120 Total días - 10 días </td><td data-bbox="711 1167 954 1344"> Calcular total de páginas por día dividiendo total de páginas por días. </td><td data-bbox="954 1167 1157 1344"> Resultado de lectura planificada de "páginas por días". </td></tr> </tbody> </table>	Entrada	Proceso	Salida	Total páginas - 120 Total días - 10 días	Calcular total de páginas por día dividiendo total de páginas por días.	Resultado de lectura planificada de "páginas por días".	¿Qué datos necesito? ¿En qué unidades están? ¿Cómo represento los días y páginas?
Entrada	Proceso	Salida						
Total páginas - 120 Total días - 10 días	Calcular total de páginas por día dividiendo total de páginas por días.	Resultado de lectura planificada de "páginas por días".						

19 Software para primeros pasos de programación usados en el programa: <http://pseint.sourceforge.net/>

Fase 2 – Elaborar un pseudocódigo

Acción	Resultado esperado	Preguntas de apoyo
Escribir un pseudocódigo que calcule cuántas páginas se deben leer por día.	Algoritmo PlanLectura 1. Definir páginas, días, páginas_por_día como enteros 2. Escribir "Ingrese total de páginas" 3. Leer páginas 4. Escribir "Ingrese cantidad de días" 5. Leer días 6. $\text{páginas_por_día} = \text{páginas} / \text{días}$ 7. Escribir "Debe leer", páginas_por_día, "páginas por día" FinAlgoritmo Ejemplo 1. $\text{páginas} < 120$ 2. $\text{días} < 10$ 3. $\text{páginas_por_día} < \text{páginas} / \text{días}$ 4. $\text{print}(\text{páginas_por_día})$	¿Cómo se representa cada paso? ¿Se puede generalizar para otros libros?

Fase 3 – Evaluar eficiencia del código

Acción	Resultado esperado	Preguntas de apoyo
Permitir que el usuario ingrese diferentes combinaciones de páginas y días.	Probar el algoritmo con distintos casos para validar su utilidad y flexibilidad.	¿Qué pasa si el número de páginas no es múltiplo de los días? ¿Qué pasaría si cambia el tiempo disponible?

Fase 4 – Reflexión crítica sobre el uso de algoritmos

Acción	Resultado esperado	Preguntas de apoyo
Reflexionar sobre para qué sirve programar esto en pseudocódigo si se puede usar calculadora.	Comprender el valor de la programación como herramienta de organización, autonomía y automatización de tareas.	¿Qué ganamos al escribirlo como algoritmo? ¿Cuándo sería útil no depender de una calculadora?

Páginas por día = Total de páginas / Días de lectura

Páginas ajustadas por día = Total de páginas / (Días de lectura – Días de repaso)

PRÁCTICA
INDEPENDIENTE

Se sugiere hacer un juego de roles en parejas, en el que construyen un algoritmo de movimientos para un robot que organiza libros en una biblioteca, siguiendo una cuadrícula.

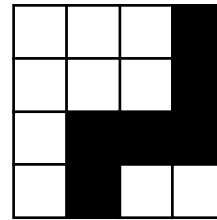
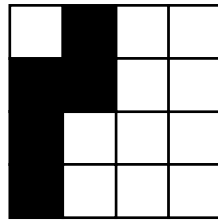
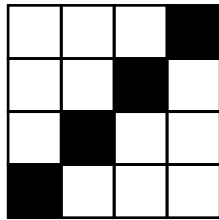
Además, con los conocimientos del módulo, pueden elaborar un seudocódigo que represente los pasos que debe seguir un robot para colocar libros en las zonas de estantería dentro de una biblioteca. Las zonas donde deben ir los libros están representadas por los cuadrados negros.

Cada imagen representa un patrón distinto de organización.
Usa las flechas y lápiz de marcado (→ ← ↑ ↓ ■) en la sección de escritura de código para escribir paso a paso las instrucciones del robot.

Luego, verifica en la cuadrícula si el recorrido del robot coincide con el patrón de estantería que elegiste.

Acción	Resultado esperado					
Confeccionar una lista para las fases de un algoritmo.	Identificar cuadrícula objetivo					
	<table><tr><th>Entrada</th><th>Proceso</th><th>Salida</th></tr><tr><td>Robot debe ordenar la biblioteca.</td><td>Secuencia de movimientos del robot. A través del dibujo de pasos. Comprobar el dibujo del recorrido en la cuadrícula de verificación. Escribir los pasos con seudocódigo en la hoja de escritura.</td><td>Comprobar si se alcanzaron las zonas correctas de organización de libros.</td></tr></table>	Entrada	Proceso	Salida	Robot debe ordenar la biblioteca.	Secuencia de movimientos del robot. A través del dibujo de pasos. Comprobar el dibujo del recorrido en la cuadrícula de verificación. Escribir los pasos con seudocódigo en la hoja de escritura.
Entrada	Proceso	Salida				
Robot debe ordenar la biblioteca.	Secuencia de movimientos del robot. A través del dibujo de pasos. Comprobar el dibujo del recorrido en la cuadrícula de verificación. Escribir los pasos con seudocódigo en la hoja de escritura.	Comprobar si se alcanzaron las zonas correctas de organización de libros.				

1. Elegir una imagen con un patrón de zonas de biblioteca para que el robot vaya a dejar los libros.



2. Escribe el algoritmo del robot. (Usando $\rightarrow \leftarrow \uparrow \downarrow \blacksquare$.)

Pasos 1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18

3. Comprueba tus pasos como si fueras el robot de biblioteca.

Inicio			

4. Preguntas de apoyo

¿Cuál patrón de biblioteca elegiste?
 ¿Dónde debe comenzar el robot?
 ¿Qué camino debe seguir para no desviarse?
 ¿Hay pasos repetidos que podrías simplificar?
 ¿Tu algoritmo sirve para otra imagen también?

Finaliza creando con pseudocódigo en la zona de escritura de código – hoja de trabajo, numerando cada paso. Puedes probar más de una solución e intercambiar la hoja con una compañera o compañero para verificar el recorrido del robot.

Zona de escritura de código – hoja de trabajo

Línea	Instrucción (Código o Bloque)	Tipo de instrucción o comentarios
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		

Evalúa la eficiencia y optimización del código del robot de biblioteca:

Inicio			

**EVALUACIÓN
FORMATIVA**

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Entrada-proceso y salida	Diseña una fase de algoritmo simple.	Diseña fases de algoritmo que son suficientes.	Diseña fases de algoritmo organizadas y entendibles, apreciando cada detalle solicitado.
Elaboración de pseudocódigos	Elabora un pseudocódigo.	Elabora un pseudocódigo que corresponde medianamente a lo solicitado.	Elabora un pseudocódigo que corresponde a lo solicitado.

Orientaciones al docente

Para unificar conceptos disciplinares

Entendemos que evaluar algoritmos en esta actividad requiere que los y las estudiantes cuenten con algunos conocimientos previos y comprendan algunos conceptos principales. Con ello, lograrán entender qué es un algoritmo y cuáles son sus aplicaciones en PSeint. Es recomendable dar explicaciones simples de las fases de un algoritmo; a su vez, explicar con mayor detalle el uso de las condicionales “si” y “según”.

El concepto base en esta actividad corresponde a una serie de instrucciones que siguen patrones o reglamentos que se realizan inconscientemente. A partir de esto, será más fácil poder ampliar las demás funcionalidades, con lo que será una herramienta para expresar algoritmos más complejos que tengan toma de decisiones.

- **Seudocódigo:** es un código falso, es decir, es una idea de cómo sería en base al lenguaje de programación (una base), para ello se utiliza PSeint o Scratch.
- **Código:** es la codificación de un algoritmo en base a un lenguaje de programación profesional como C#, JAVA o Python.

Las nociones de pseudocódigo funcional y óptimo pueden ser trabajadas a medida que los y las estudiantes ejecutan el programa de otro grupo y comparten sus pseudocódigos con otros para que los ejecuten manualmente.

Actitudes

Para apoyar el desarrollo de las actividades es recomendable utilizar, en ciertos periodos, computadores (PSeint) para que los y las estudiantes puedan ver si sus creaciones son correctas y verifiquen si puede haber más alternativas a sus problemas. Sin embargo, siempre está la alternativa de hacer los pseudocódigos mediante una hoja de papel o en un documento Word. También se pueden incluir algunas preguntas que fortalezcan la toma de decisiones del mismo algoritmo y así contribuyan al bien común, por ejemplo: ¿Qué proceso sería el más eficaz? ¿Será la opción más acertada? o ¿es posible hacerlo de otra manera? En cuanto a desarrollar los trabajos colaborativos, se sugiere promover un trabajo simultáneo donde los y las estudiantes se ayuden mutuamente a conseguir el objetivo de ambos.

En la situación experiencial, Carolina estudió y logró sacarse una buena calificación. En cuanto a José, él prefirió quedarse dormido en vez de estudiar y por eso tuvo una mala calificación en el examen de matemáticas.

Orientaciones para organizar e implementar el juego de roles

Se sugieren las siguientes indicaciones para guiar la implementación del trabajo de pares con un juego de roles:

- Elabore tarjetas con dos roles: el o la estudiante que pregunta puede ser considerado el cliente y la persona que responde el informático. Indicar en las tarjetas lo que se espera como tarea en cada rol.
- Ambos tendrán sus tarjetas y empezarán al mismo tiempo. Cada uno dirá el problema y después elaborarán un algoritmo basado en la problemática.
- Determine el tiempo necesario para elaborar un algoritmo simple para comenzar.
- Organice la clase y el espacio para el trabajo en pares.
- Determine el tiempo para que ambos estudiantes revisen mutuamente sus algoritmos. Deben dar una retroalimentación de acuerdo con el seguimiento de la tarjeta, si falta algún detalle pequeño o si se puede hacer de mejor manera.
- Luego, determinar el tiempo para que ambos se ayuden para hacer un pseudocódigo del algoritmo que acaban de crear.
- Determine lo que se hará con los resultados y con la entrega de las respuestas. No es necesario que se haga una presentación de los resultados. Se sugiere conversar sobre las dificultades y obstáculos que surgen a la hora de elaborar un algoritmo y/o pseudocódigo.

Anexo

Elaborar una calculadora en Pseint con las funciones de sumar, restar, multiplicar y dividir con solo dos valores a elegir para realizar el cálculo.

Acción	Resultado esperado		
Confeccionar una lista para las fases de un algoritmo.	Entrada	Proceso	Salida
	Calculadora	1. Seleccionar qué opción usará (sumar, restar, multiplicar o dividir). 2. Ingresar los dos valores que se quieren calcular. "Valor 1 (opción elegida) Valor 2". 3. Calcular la operación matemática. "Valor 1 (opción elegida) Valor 2 = Total".	Resultado de la operación.

Acción	Resultado esperado
Elaborar un código relacionado con lo expresado en las fases de los algoritmos hechos anteriormente.	<pre> Algoritmo Calculadora //definir variables Definir opc, valor1, valor2 Como Entero; Definir total como real; //entrada Escribir "Calculadora"; Escribir "1: Sumar"; Escribir "2: Restar"; Escribir "3: Multiplicar"; Escribir "4: Dividir"; Escribir "Ingrese opcion:"; Leer opc; Si opc ≥ 1 y opc ≤ 4 Entonces Escribir "Ingrese valor 1: "; Leer valor1; Escribir "Ingrese valor 2: "; Leer valor2; Si opc = 1 Entonces //proceso total = valor1 + valor2; //salida Escribir "La suma de " valor1 " + " valor2 " = " total; SiNo Si opc = 2 Entonces //proceso total = valor1 - valor2; //salida Escribir "La resta de " valor1 " - " valor2 " = " total; SiNo Si opc = 3 Entonces //proceso total = valor1 * valor2; //salida Escribir "La multiplicacion de " valor1 " * " valor2 " = " total; SiNo Si opc = 4 Entonces //proceso total = valor1 / valor2; //salida Escribir "La division de " valor1 " / " valor2 " = " total; FinSi FinSi FinSi FinSi //salida Escribir "Opcion invalida"; FinSi FinAlgoritmo </pre>

Acción	Resultado esperado
Evaluar la eficiencia y optimización del código	<pre> Algoritmo Calculadora //definir variables Definir opc, valor1, valor2 Como Entero; Definir total como real; //entrada Escribir "Calculadora"; Escribir "1: Sumar"; Escribir "2: Restar"; Escribir "3: Multiplicar"; Escribir "4: Dividir"; Escribir "Ingrese opcion:"; Leer opc; Si opc ≥ 1 y opc ≤ 4 Entonces Escribir "Ingrese valor 1: "; Leer valor1; Escribir "Ingrese valor 2: "; Leer valor2; Segun opc 1: //proceso total = valor1 + valor2; //salida Escribir "La suma de " valor1 " + " valor2 " = " total; 2: //proceso total = valor1 - valor2; //salida Escribir "La resta de " valor1 " - " valor2 " = " total; 3: //proceso total = valor1 * valor2; //salida Escribir "La multiplicacion de " valor1 " * " valor2 " = " total; 4: //proceso total = valor1 / valor2; //salida Escribir "La division de " valor1 " / " valor2 " = " total; FinSegun SiNo //salida Escribir "Opcion invalida"; FinSi FinAlgoritmo </pre>

Actividad de desempeño

2

Propósito

En esta actividad se pone énfasis en la evaluación para que las y los estudiantes tengan una base sólida de los conceptos principales del algoritmo y así desenvolverse de manera más fluida con los términos ya aprendidos. Además, se añadirán algunos conceptos nuevos que entreguen oportunidades cuando se quieran expresar algoritmos en un código para resolver un problema.

Objetivos de aprendizaje

OA 2

Organizar datos de una situación problemática estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función. [\(Evaluar\)](#)

Conocimiento esencial

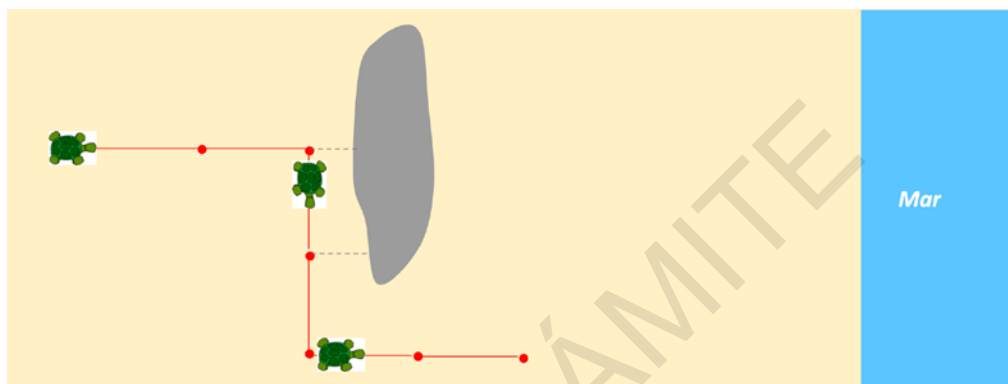
- Procesos y funcionamiento de las cosas para resolver problemas cotidianos.

Tiempo estimado: 3 horas pedagógicas

Desarrollo de la actividad

SITUACIÓN EXPERIENCIAL

El o la docente presenta una situación que puede ser transferida a situaciones reales y a un uso cotidiano de “si pasa esto” y/o de “según esto o lo otro”. Además, introduce las nociones con una situación de decisión entre 2 alternativas para el camino de la tortuga.



La tortuga Kira quiere llegar al mar. En su camino avanza en pasos de 100, pero puede ocurrir que un obstáculo esté en su camino, por ejemplo, una roca. La tortuga puede desviar este obstáculo al girar en 90° a la derecha o a la izquierda. Luego, avanza 100 pasos, gira 90° a la izquierda y mira otra vez para ver si el camino está despejado para dar el próximo avance de 100 pasos hacia el mar. Debe repetir el procedimiento en caso de que “**no**” sea así.

El o la docente puede vincular esta situación con experiencias reales del grupo al preguntar, por ejemplo: ¿Qué decisiones tomaron hoy para sortear un obstáculo en el momento en que se presentó?. Algunas respuestas podrían relacionarse con rutas cerradas al trasladarnos, turnos que cambian a última hora, resolver imprevistos familiares, reorganizar tareas por falta de materiales o ajustar el trayecto si no hubo locomoción. Estos ejemplos cotidianos permiten que los y las estudiantes reconozcan estructuras condicionales simples en su vida diaria antes de formalizarlas enseudocódigo.

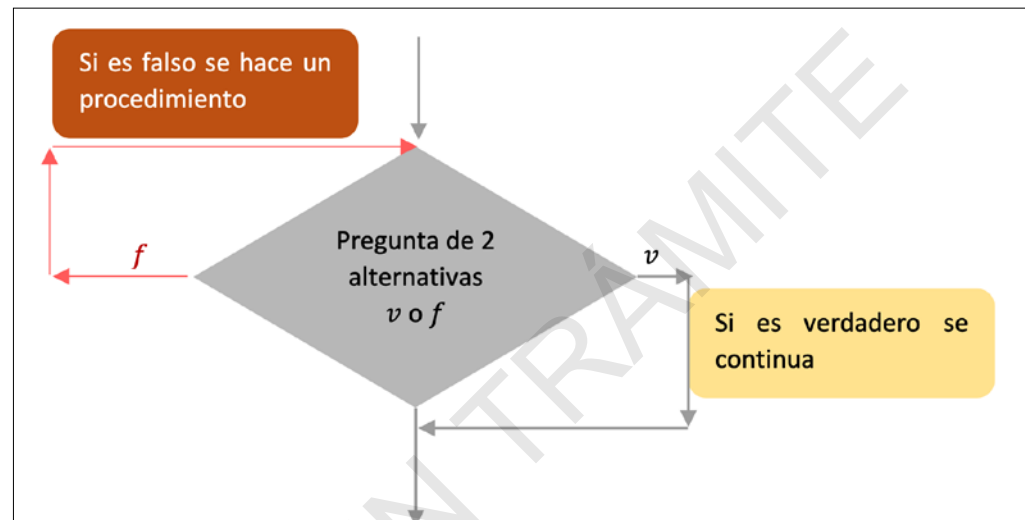
CONSTRUCCIÓN DE CONOCIMIENTO

Para construir el conocimiento, se sugiere resaltar que en programación todos los lenguajes formales disponen de una orden que requiere una decisión entre 2 alternativas, las que pueden ser “sí o no”, o bien “verdadero o falso”. Esta instrucción, en la que se decide entre 2 alternativas, se presenta en el recuadro de abajo con la forma de un rombo.

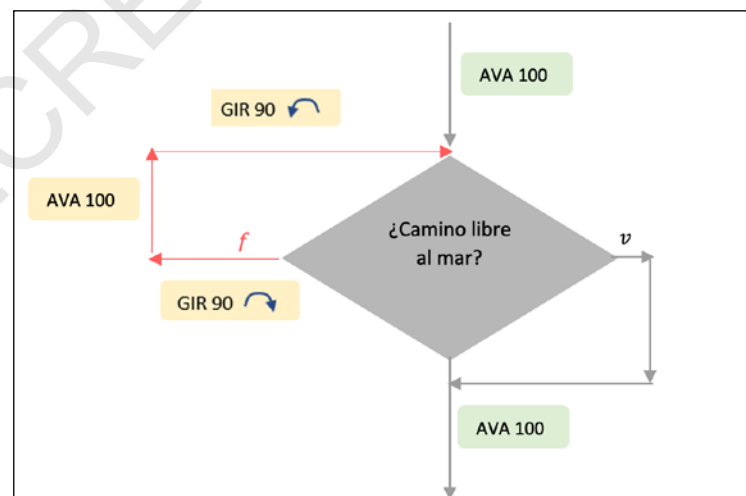
Antes de introducir elseudocódigo o el entorno digital, se propone comenzar con una representación concreta de la situación, mediante el uso de flechas físicas (adelante, izquierda, derecha) y un rombo como símbolo de decisión. Esto permitirá representar con el cuerpo o figuras manipulables el recorrido de una tortuga que debe continuar un circuito y tomar decisiones frente a obstáculos. Cada vez que la tortuga encuentra una roca, el rombo se utiliza como marcador visual para formular la pregunta: ¿El camino está libre?, y decide si avanza o se desvía. Esta experiencia concreta o tangible facilita la comprensión de estructuras condicionales antes de formalizarlas en pseudocódigo.

Para crear flechas físicas puedes descargar:

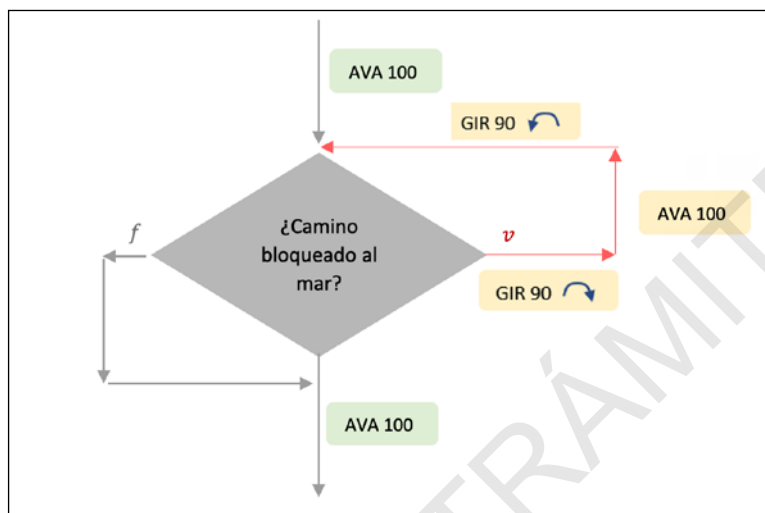
- Flecha a la derecha: https://commons.wikimedia.org/wiki/File:MUTCD_M5-1PR.svg
- Flecha a la izquierda: https://commons.wikimedia.org/wiki/File:MUTCD_M5-1PL.svg
- Flecha dirección (o avanzar): https://commons.wikimedia.org/wiki/File:MUTCD_M6-1PR.svg
- Forma de rombo: <https://commons.wikimedia.org/wiki/File:RhombusWhiteBorder.jpg?uselang=es>



Se sugiere ejemplificar esta idea de 2 alternativas utilizando el camino de la tortuga esquematizando el programa y utilizando el pseudocódigo creado en la actividad de desempeño anterior. Es importante precisar la instrucción de decisión que se refiere a la situación de la tortuga con un rombo.



Se sugiere verificar de forma activa si este programa lleva a la tortuga al mar. Para esto, las y los estudiantes pueden actuar el esquema o bien utilizar fichas o material concreto. También, se sugiere probar con un segundo esquema de programación, donde se cambia la pregunta decisiva en el rombo por: ¿Camino bloqueado al mar?



PRÁCTICA GUIADA

Se sugiere guiar la verificación de que los algoritmos propuestos son, en cierta medida, la mejor opción, o bien presentar la forma de cambiarlos si es necesario y así cumplir con las necesidades que se solicitan. Cada estudiante también podría proponer mejoras para dicho algoritmo y así conseguir satisfacer y dar respuestas a las necesidades.

Con respecto a lo anterior se trabajará con un menú donde habrá dos opciones a elegir:

- Verificar si un número es par o impar.
- Una tabla de multiplicar.

Acción	Resultado esperado	Preguntas de apoyo
Elaborar un código lo más fiel posible a lo establecido en el problema.	<pre>Algoritmo ParImpar Definir aux, num, NumMayor, variables Como Entero Escribir 'Seleccione que tipo de cliente es: '; Escribir '1) Ver si un numero es par o impar'; Escribir '2) Tabla de multiplicar'; Escribir '3) Ver cual de los 10 números es el mayor'; Leer aux; Segun aux Hacer 1: Escribir "Escriba un número: " Leer num Si num % 2 = 0 Entonces Escribir num, " es un número par" SiNo Escribir num, " es un número impar" Fin Si 2: Escribir "Ingrese un Número del 1 - 10 : " Leer Num Escribir 1 " X " Num " = " (1 * Num) Escribir 2 " X " Num " = " (2 * Num) Escribir 3 " X " Num " = " (3 * Num) Escribir 4 " X " Num " = " (4 * Num) Escribir 5 " X " Num " = " (5 * Num) Escribir 6 " X " Num " = " (6 * Num) Escribir 7 " X " Num " = " (7 * Num) Escribir 8 " X " Num " = " (8 * Num) Escribir 9 " X " Num " = " (9 * Num) Escribir 10 " X " Num " = " (10 * Num) FinSegun FinAlgoritmo</pre>	¿Realmente este código cumple con lo prometido de la mejor manera? ¿Existe alguna otra forma de hacerlo? ¿Cómo podemos saber si un número es par o impar en un código?

Acción	Resultado esperado	Preguntas de apoyo
Después del análisis y la nueva metodología aprendida, ¿cómo arreglaría o mejoraría el anterior código?	<pre> Algoritmo ParImpar Definir aux, num, NumMayor, variables Como Entero Escribir 'Seleccione que tipo de cliente es: '; Escribir '1) Ver si un número es par o impar'; Escribir '2) Tabla de multiplicar'; Escribir '3) Ver cual de los 10 números es el mayor'; Leer aux; Segun aux Hacer 1: Escribir "Escriba un número: " Leer num Si num % 2 = 0 Entonces Escribir num, " es un número par" SiNo Escribir num, " es un número impar" Fin Si 2: Escribir "Ingrese un Número del 1 - 12 : " Leer Num Para Cont <-1 Hasta 12 Con Paso 1 Hacer Escribir Cont " X " Num " = " (Cont * Num) Fin Para FinSegun FinAlgoritmo </pre>	¿Cómo sé que esta es la mejor opción? ¿Cuál sería la mayor diferencia entre el código antiguo y el nuevo? ¿De qué otra manera se podría usar la nueva metodología?

PRÁCTICA INDEPENDIENTE

De acuerdo con el ejercicio anterior y al ver que el resultado es satisfactorio, se decide implementar una nueva opción llamada verificar cuál de los 10 números es el mayor. Esta opción servirá para aprender a usar estas nuevas herramientas y así poder expandir la resolución del problema anterior:

Acción	Resultado esperado
Elaborar un código donde se añada la nueva opción ya mencionada en el código hecho anteriormente.	<pre> Algoritmo ParImpar Definir aux, Num, NumMayor, variables Como Entero Escribir 'Seleccione que tipo de cliente es: '; Escribir '1) Ver si un numero es par o impar'; Escribir '2) Tabla de multiplicar'; Escribir '3) Ver cual de los 10 números es el mayor'; Leer aux; Segun aux Hacer 1: Escribir "Escriba un número: " Leer Num Si Num % 2 = 0 Entonces Escribir Num, " es un número par" SiNo Escribir Num, " es un numero impar" Fin Si 2: Escribir "Ingrese un Número del 1 - 12 : " Leer Num Para Cont <- 1 Hasta 12 Con Paso 1 Hacer Escribir Cont " X " Num " = " (Cont * Num) Fin Para 3: Para Cont <- 1 Hasta 10 Con Paso 1 Hacer Escribir "Escriba un número: " Leer Num Si NumMayor < Num Entonces NumMayor = Num Fin Si Fin Para Escribir "El Mayor es : ", NumMayor FinSegun FinAlgoritmo </pre>

EVALUACIÓN FORMATIVA

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Códigos	Elabora un código.	Elabora un código que cumple con lo establecido.	Elabora un código que cumple con lo establecido. Además, lo analizó y mejoró en el proceso.
Metodologías	Usa las metodologías aprendidas.	Usa las nuevas estrategias de forma medianamente efectiva.	Usa las nuevas estrategias de forma efectiva.

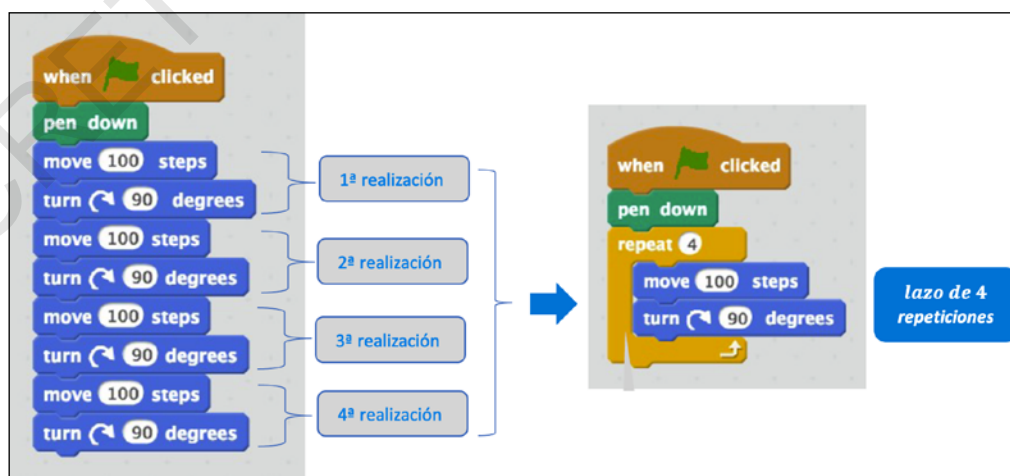
Orientaciones al docente

Para unificar conceptos disciplinares

Podemos entender el funcionamiento de un algoritmo y también cómo se tienen que resolver los problemas para tener la respuesta más aceptable y óptima posible. Se sugiere integrar nuevas estrategias, como “parar” y “repetir”.

- **Parar:** permite medir cuantas veces queremos que se repita la misma acción.
- **Repetir:** permite repetir hasta que una acción en concreto se cumpla.

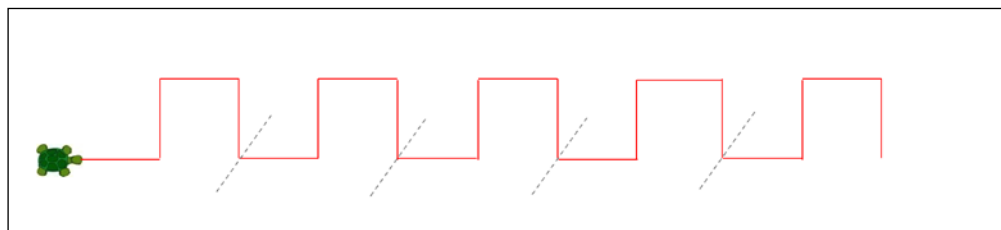
Estas son herramientas para complementar nuestro entendimiento en el funcionamiento del algoritmo y la codificación. El objetivo principal es hacer más sencillo el trabajo de elaboración de programas y la ejecución de ellos. Se sugiere indicar en qué momentos existen pasos iguales y si se puede utilizar la iteración del procedimiento.



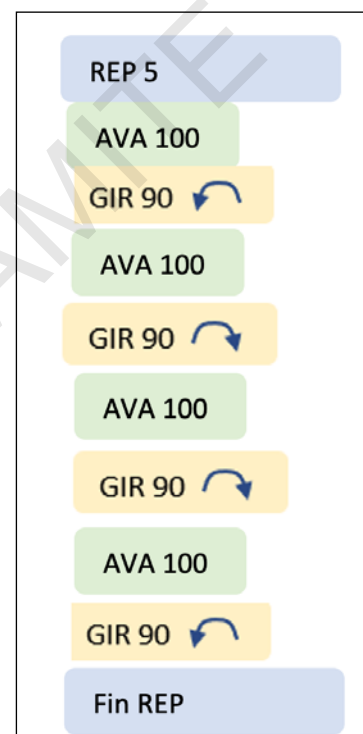
Algunas preguntas guías para esta introducción de los ciclos o bucles podrían ser:

- ¿Cuáles son los pasos de desplazamiento y giro que se repiten?
- ¿Cómo se está abreviando la escritura del programa?

Se recomienda también ver estos ciclos de forma visual, señalar qué es lo que se repite y la cantidad de veces que ocurre. Por ejemplo, en el caso del camino de la tortuga.



Relacionar con la programación en pseudocódigo:



Actitudes

Siempre que se esté usando un computador a la hora de codificar un algoritmo, se recomienda verificar el código de una manera más eficiente. También, poder ejecutar y ver su funcionamiento, aprovechando siempre las herramientas disponibles. Para el desarrollo de los trabajos colaborativos, es recomendable que trabajen en equipo para que se ayuden mutuamente; también es útil para que surjan conversaciones y se tomen decisiones para evaluar de forma correcta el algoritmo mediante el uso de criterios de función y funcionamiento.

Orientaciones para organizar e implementar el trabajo en equipo

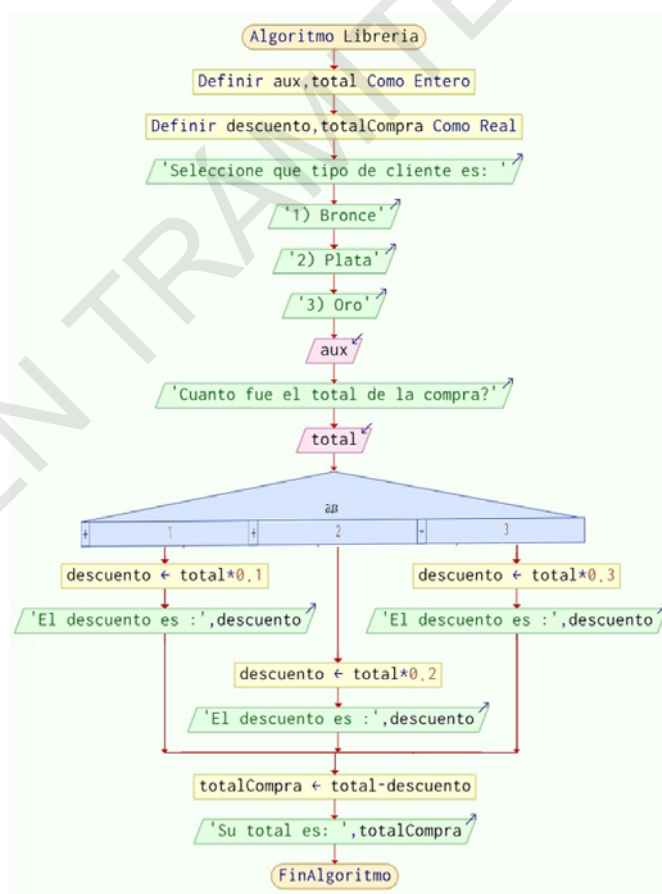
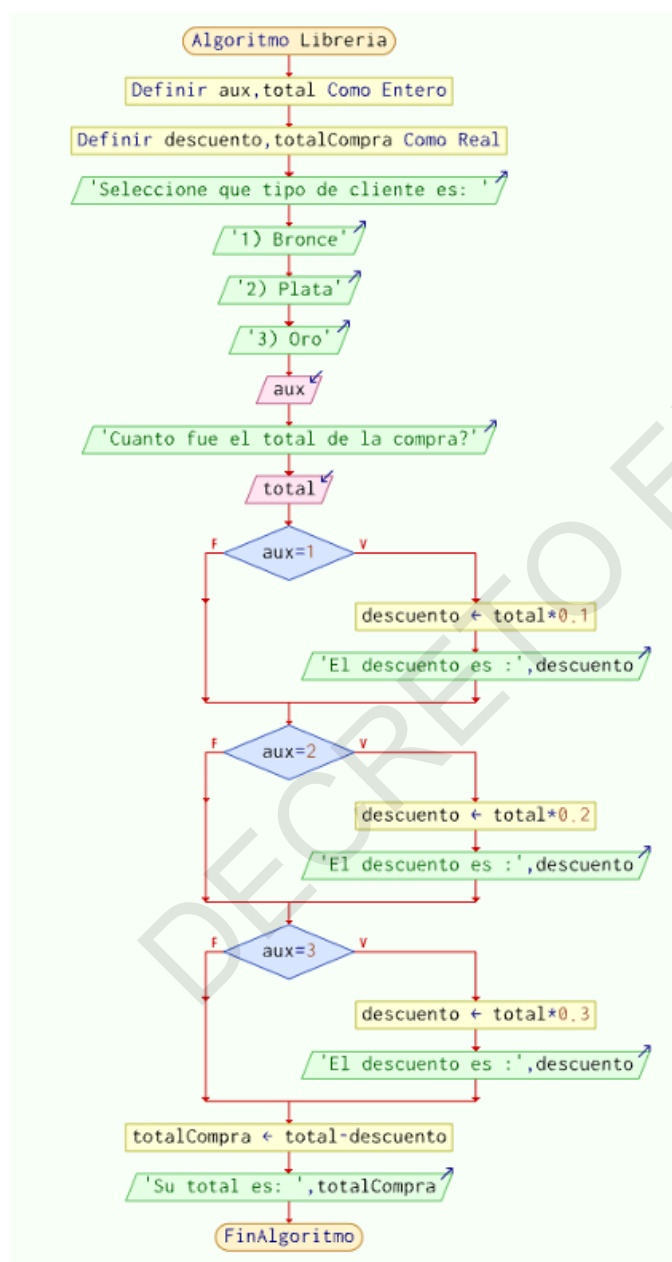
Se sugieren las siguientes indicaciones para guiar la implementación del trabajo en parejas o tríos:

- Elabore ejercicios donde se tengan en cuenta las funcionalidades “paraR” y “repetir”.
- Cada equipo tendrá su ejercicio o desafío a resolver.
- Determine el tiempo necesario de desarrollo del código.
- Los y las estudiantes analizan y verifican si todo está correcto y ven si se le puede hacer una mejora a lo propuesto en el problema.
- Determine el tiempo para el análisis y corrección de código.

Anexo

En esta actividad se muestran dos esquemas de programas donde las y los estudiantes pueden encontrar similitudes y diferencias. Se sugiere guiar la actividad por medio de las siguientes preguntas:

- ¿Cuál es el problema para cada algoritmo?
- ¿Habrá una forma más eficiente de elaborar estos algoritmos?
- ¿Cuáles serían las diferencias entre los dos algoritmos en cuanto a funcionalidad?
- ¿Cómo se pueden mejorar estos algoritmos?



La siguiente actividad complementa la práctica independiente y marca nuevamente los procedimientos que se repiten en el esquema de programación:

El código es eficiente, pero tener que escribir los 10 números en cada momento es agotador. Sería mejor si la máquina hiciera ese trabajo y nos mostrara los números que escogió.

```

3:
Para Cont ← 1 Hasta 10 Con Paso 1 Hacer
    Num ← Aleatorio(0,100)
    Escribir "El numero " Cont " es: " Num
    Si NumMayor < Num Entonces
        NumMayor = Num
    Fin Si
Fin Para

Escribir "El Mayor es : ", NumMayor
FinSegun

```

Mucho mejor, pero a la hora de finalizar el problema, se termina de ejecutar el código. ¿Cómo se podría hacer un bucle para poder elegir todas las opciones y uno decidir cuándo termina de correr el código?

```

1  Algoritmo ParImpar
2  Definir aux, Num, NumMayor, variables Como Entero
3
4  Repetir
5      Escribir 'Seleccione que tipo de cliente es: ';
6      Escribir '1) Ver si un numero es par o impar';
7      Escribir '2) Tabla de multipliar';
8      Escribir '3) Ver cual de los 10 numeros es el mayor';
9      Escribir '4) Salir';
10     Leer aux;
11
12     Segun aux Hacer
13         1:
14             Escribir "Escriba un número: "
15             Leer Num
16
17             Si Num % 2 = 0 Entonces
18                 Escribir Num, " es un numero par"
19             SiNo
20                 Escribir Num, " es un numero impar"
21             Fin Si
22         2:
23             Escribir "Ingrese un Número del 1 - 12 : "
24             Leer Num
25
26             Para Cont ←1 Hasta 12 Con Paso 1 Hacer
27                 Escribir Cont " X " Num " = " (Cont * Num)
28             Fin Para
29         3:
30             Para Cont ← 1 Hasta 10 Con Paso 1 Hacer
31                 Num ← Aleatorio(0,100)
32                 Escribir "El numero " Cont " es: " Num
33                 Si NumMayor < Num Entonces
34                     NumMayor = Num
35                 Fin Si
36             Fin Para
37
38             Escribir "El Mayor es : ", NumMayor
39     FinSegun
40     Hasta Que aux = 4
41 FinAlgoritmo

```

Actividad de desempeño

3

Propósito

Esta actividad, está enfocada en el modelamiento de base de datos y los inicios del sistema de clasificación de estructuración de esta. Se identificarán las entidades y atributos al descomponer puntos clave de los problemas a la vez que aplicarán, a través de actividades concretas, las diferentes clasificaciones de atributos que constituye un modelo de base de datos.

Objetivos de aprendizaje

OA 1

Formular problemas a partir de un contexto o situación específica, con apertura a distintas perspectivas, identificando las partes que lo componen. **(Descomponer)**

OA 3

Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponibles. **(Modelar)**

Conocimiento esencial

- Modelos entidad-relación y simulación de sistemas.

Tiempo estimado: 3 horas pedagógicas

Desarrollo de la actividad

SITUACIÓN
EXPERIENCIAL

El o la docente presenta a sus estudiantes dos objetos para identificar ciertos parámetros que se deben saber y necesitar a la hora de modelar una base de datos.

Se sugiere seleccionar objetos físicos disponibles en el entorno inmediato del estudiante. Por ejemplo, se puede presentar distintos envases (cajas de envases de pasta de dientes o botellas de agua, cajas de alimentos, paquetes de té, etc.) y trabajar con ellos de forma tangible. A partir de esta experiencia concreta, se invita a identificar entidades (producto, marca, proveedor) y atributos (peso, fecha de vencimiento, precio, tipo de empaque, color, entre otros). Esto facilitará la comprensión de conceptos abstractos de modelamiento desde lo cercano y cotidiano.

Algunas de las preguntas que pueden promover el proceso de comunicar una argumentación sobre lo que podemos observar son:

- ¿Cuáles serían las entidades de las siguientes imágenes?
- ¿Qué atributos tendría cada entidad?
- ¿Cómo se relacionarían las dos entidades?

CONSTRUCCIÓN DE
CONOCIMIENTO

Para construir el conocimiento sobre cómo armar un modelo de base de datos, se sugiere comenzar por una experiencia concreta y cercana al contexto de las y los estudiantes. En lugar de partir desde términos abstractos, se propone preguntar cómo organizan sus pertenencias en su espacio personal y vincular esta experiencia con un caso cotidiano, como el registro de préstamo de libros en una biblioteca comunitaria.

Primero, se invita a identificar objetos simples (como libros) para comprender qué es una entidad y qué es un atributo, con sus respectivas descripciones. A partir de esta situación, se puede solicitar que clasifiquen la información necesaria para organizar una colección de libros y así introducir los conceptos de “título”, “autor”, “año” y “disponibilidad” como atributos del libro.

Libro 1	Libro 2	Libro 3
 ▶ Título: Volver a Aprender: Decisiones Cotidianas y Futuro ▶ Autor: Valentina Abarca ▶ Año: 2018 ▶ Disponibilidad: Disponible	 ▶ Título: Matemática para la Vida Diaria a lo largo de la vida ▶ Autor: Héctor Salinas ▶ Año: 2021 ▶ Disponibilidad: No disponible	 ▶ Título: Aprender con Sentido: Trabajo, Comunidad y Autonomía ▶ Autor: María Francisca Gallegos ▶ Año: 2019 ▶ Disponibilidad: Disponible

Para guiar la elaboración de respuestas, se sugiere usar los términos “entidad” y “atributo” y ampliar la lista de atributos considerando si son obligatorios, opcionales o claves primarias. Se recomienda trabajar al menos con tres atributos por entidad. Esta actividad puede realizarse con software como PSeInt o bien en formato desenchufado, mediante el uso de tarjetas o registros físicos para representar cada atributo.

Acción	Resultado esperado	Preguntas de apoyo												
Confeccionar una lista para la clasificación de las entidades.	Identificar la entidad principal "Libro" y listar atributos como título, autor, año de publicación, disponibilidad. Esto por cada libro disponible. <table border="1"> <thead> <tr> <th>Entidad</th><th colspan="2">Atributo</th></tr> </thead> <tbody> <tr> <td rowspan="4">Libro</td><td>Título</td><td></td></tr> <tr> <td>Autoría</td><td></td></tr> <tr> <td>Año de publicación</td><td></td></tr> <tr> <td>Disponibilidad</td><td></td></tr> </tbody> </table>	Entidad	Atributo		Libro	Título		Autoría		Año de publicación		Disponibilidad		¿Qué información permite describir un libro? ¿Cómo identificar cada atributo? ¿Son todos los atributos que se pudieron encontrar? ¿Puede haber datos que se repitan en cada entidad? ¿Existe alguna forma de relacionar todas las entidades? ¿Qué atributos aparecen en todos los casos?
Entidad	Atributo													
Libro	Título													
	Autoría													
	Año de publicación													
	Disponibilidad													
Distinguir las diferentes clasificaciones de atributos y verificar si hay un identificador.	Clasificar los atributos según si son obligatorios, opcionales o claves primarias. Reconocer que el "ID Libro" puede funcionar como clave única.	¿Qué hacer si no le encuentro una clave primaria? ¿Cómo identificar una clave primaria? ¿Cómo saber si un atributo es obligatorio u opcional? ¿Qué dato no se puede repetir? ¿Cuál permitiría buscar o registrar un libro sin confusiones?												
Relacionar las tablas entre ellas.	Proponer relaciones posibles con otras entidades (por ejemplo, "Persona" o "Préstamo") si se extiende la base de datos. Identificar cómo se enlazan por campos comunes.	¿Qué otra información se podría vincular a un libro? ¿Cómo saber si están relacionadas? ¿Cómo sabríamos quién lo pidió prestado o cuándo lo devolvió?												

PRÁCTICA GUIADA

Para guiar la evaluación del modelamiento de base de datos, se debe tener claro cómo definir cada tabla que se vaya generando durante la lectura del mismo problema. Es importante tener presente que no siempre será fácil identificar todas las entidades, atributos y relaciones, pero es necesario tener una base firme para después consultar las dudas que uno pueda tener.

Así, el objetivo de esta práctica será crear un modelo de base de datos relacionado con el registro de insumos y participantes en un taller de carpintería o centro de formación comunitario, considerando aspectos básicos como quién solicita el material, qué herramienta se presta y en qué fecha.

Al completar, la o el docente aclarará que no se está solicitando un modelo extremadamente complejo en ese sentido; solo se solicitan elementos básicos que están relacionados en este registro de cuenta, por ejemplo, nombre, apellido, correo, contraseña y su respectivo inicio de sesión, o bien la publicación y el registro.

Ejemplo de formulario (simulado)
Registro de préstamo de insumos - Taller de Carpintería

- Nombre de quien participa
- Apellido
- Rut o número de identificación
- Herramienta solicitada
- Fecha de préstamo
- Fecha estimada de devolución
- Estado de la herramienta (operativa/no operativa)

Acción	Resultado esperado
Identificar las características que tiene el sistema de registro del taller de carpintería y revisar si son claves primarias, obligatorias u opcionales.	HERRAMIENTA • # id_herramienta • nombre • estado • tipo PRÉSTAMO • # id_prestamo • fecha_salida • fecha_retorno • observaciones USUARIO • # id_usuario • nombre • curso • rut
Elaborar un modelo de base de datos simple.	Relación entre: - USUARIO y PRÉSTAMO (un usuario puede tener varios préstamos). - HERRAMIENTA y PRÉSTAMO (una herramienta puede estar en muchos préstamos a lo largo del tiempo). Modelo relacional simple con llaves primarias y foráneas.
Revisar si es posible que algunos de los atributos ya puestos puedan llegar a ser por sí solos una entidad.	Separación posible de la entidad tipo de herramienta y estado de la herramienta como entidades independientes: - TIPO_HERRAMIENTA (id_tipo, descripción). - ESTADO (id_estado, descripción). Esto permite reutilizar valores y evitar repeticiones innecesarias.

PRÁCTICA INDEPENDIENTE

Se sugieren dos actividades individuales en las que la o el estudiante debe analizar un caso con el objetivo de identificar cuáles son las entidades y atributos presentes en una situación contextualizada. En una de las propuestas, la situación se basa en la organización de una actividad deportiva; en la otra, en la venta de productos. Para ello, se entrega una representación escrita de la situación que describe los elementos relevantes del problema. El o la docente guía la actividad para que el estudiantado disponga de tiempo suficiente para su desarrollo y orienta el análisis, retroalimentando cada uno de los puntos abordados en las respuestas.

Un pequeño ejemplo sería que marcará con **azul** las entidades, con **rojo** los atributos obligatorios y con **verde** los atributos opcionales.

Los productos que se venden en el almacén son comprados a distintos **proveedores** de Valparaíso y Viña del Mar. A algunos proveedores se les llama por **teléfono** y se conoce su **Rut** y **nombre**. Un proveedor puede vender distintos **productos**; es necesario almacenar **el precio de venta** que ofrece cada proveedor y la **fecha de validez** del precio ofrecido.

De los **productos**, se registra el **identificador del código de barra**, **descripción** y **precio unitario de venta sin incluir el IVA y clasificación del producto**; de **algunos productos** se sabe la **fecha de vencimiento y capacidad del envase**; de **todos los productos** se conoce el **nombre de la marca y el identificador de esta**. Un **producto** se relaciona solo con una **marca**.

Sugerencia de práctica alternativa a la anterior manteniendo las marcas con AZUL las entidades, con ROJO los atributos obligatorios y con VERDE los atributos opcionales.

Los **participantes** que se inscriben en **actividades deportivas** provienen de distintos sectores o agrupaciones comunitarias. Algunos son contactados por teléfono, y se conoce **su nombre y RUT**. Un mismo participante puede asistir a diferentes **actividades**, por lo que es necesario registrar la **fecha de inscripción**, la **categoría deportiva** y si requiere algún tipo de acompañamiento o apoyo especial.

De los **participantes**, se registra **el identificador**, **la actividad** a la que se inscriben, **el tipo de inscripción (individual o grupal)**, y en algunos casos, la edad o la comuna de residencia. Cada **actividad** está asociada a una **organización responsable**, de la que se almacena **su nombre y su correo de contacto**.

EVALUACIÓN
FORMATIVA

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Clasificación de tablas	Identifica las tablas.	Identifica la mayoría de las tablas, aunque faltan atributos.	Clasifica todas las tablas con sus respectivos atributos y relaciones.
Diferenciación de atributos	Diferencia solo algunos atributos.	Diferencia medianamente los atributos.	Diferencia todos los atributos (obligatorios y opcionales).
Relacionar tablas	Relaciona algunas tablas.	Relaciona las tablas con algunos errores.	Relaciona todas las tablas sin errores.

Pregunta orientadora para verificar aprendizajes:

¿Cómo decidir qué información registrar (como entidades y atributos) puede ayudarte a organizar mejor las actividades, resolver problemas cotidianos o proponer mejoras en tu comunidad o espacio de participación actual?

Orientaciones al docente

Para unificar conceptos disciplinares

Entendemos que el modelamiento de base de datos es algo más complejo que simplemente hacer un mapa conceptual, pero tampoco está tan alejado de esa realidad. La realización de un modelo de base de datos tiene como objetivo mejorar el control de la información que se guardará dentro de esta base de datos. Se sugiere explicar los elementos básicos primordiales de un modelo de base de datos.

- El modelo contará con una entidad por cada uno de los componentes del proceso de negocio.
- Los atributos, un componente fundamental de cada modelo entidad-relación, nos permiten describir las propiedades que tiene cada entidad.
 - **Obligatorios:** la instancia del atributo debe tener un valor válido.
 - **Opcionales:** la instancia del atributo puede o no tener un valor.
- Con las relaciones se establecen vínculos entre parejas de entidades.
- La clave o llave primaria es un atributo o combinación de atributos que identifican de forma exclusiva una y solo una instancia de una entidad; ningún subconjunto del modelo entidad-relación la posee.

Se sugiere organizar el diagrama entidad-relación (DER) desde sus componentes básicos hasta los más complejos, comenzando de manera intuitiva con una entidad y sus atributos para, luego, avanzar con dos entidades. Según el contexto elegido, se pueden incluir las nociones de atributos compuestos o simples, como también el grado de una relación.

Actitudes

Para apoyar el desarrollo de las actividades en esta etapa inicial, no es necesario el uso de computadores. Se recomienda que, al modelar una base de datos, se utilice una planilla de cálculo y que, al utilizar lenguaje de programación, también se utilicen los computadores. Por otra parte, es recomendable retroalimentar a las y los estudiantes con apertura a diferentes perspectivas y con claridad sobre lo que está bien o mal ejecutado. En cuanto a la actividad individual, es preferible que se favorezca la autonomía, para identificar cada característica de la situación y lograr que cada estudiante actúe con independencia y confianza en el entendimiento de la entidad-relación.

Orientaciones para organizar e implementar el trabajo individual

Se sugieren las siguientes indicaciones para guiar la implementación del trabajo individual:

- Elaborar un ejercicio amplio que esté centrado en algún sistema de negocio o minimercado, para que los y las estudiantes identifiquen las tablas necesarias. Se sugiere pensar también en las relaciones que se dan en la situación para, luego, utilizarlas en las próximas actividades.
- Determinar un tiempo para poder analizar y revisar todo el problema; para identificar las tablas que esconde.
- Pedir a sus estudiantes revisar todas las respuestas que escogieron. Esto beneficiará a quienes no lograron identificar todo, además permitirá aclarar dudas de cómo poder identificar cada caso.

Actividad de desempeño

4

Propósito

En esta actividad se reforzará y ampliará el conocimiento del modelamiento de base de datos, en el que el o la estudiante aprenderá relacionar de forma correcta las tablas entre sí y, a su vez, la normalización de estas. También, aprenderá la estrategia de tres formas comunes para mantener un orden al momento de crear tablas.

Objetivos de aprendizaje

OA 2

Organizar datos de una situación problemática estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función. [\(Evaluar\)](#)

OA 3

Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponibles. [\(Modelar\)](#)

Conocimiento esencial

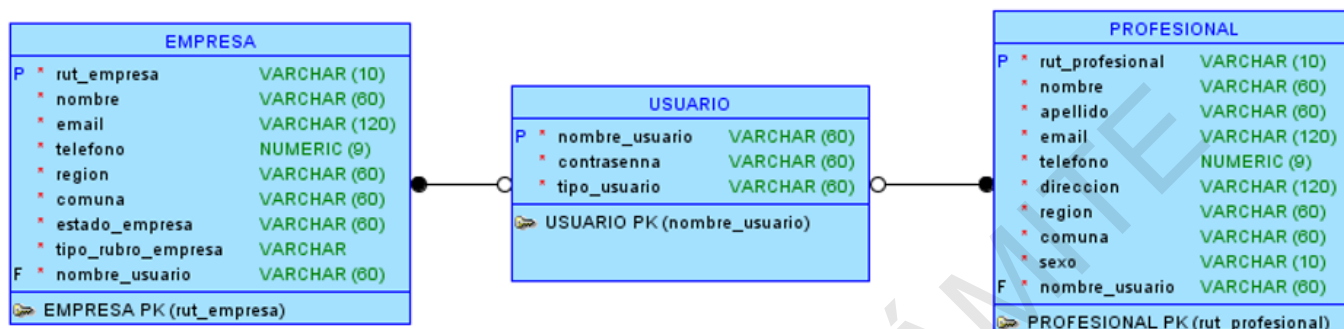
- Modelos entidad-relación y simulación de sistemas.
-

Tiempo estimado: 3 horas pedagógicas

Desarrollo de la actividad

SITUACIÓN EXPERIENCIAL

El o la docente presenta un caso particular de un modelo de base de datos que se ve simple a primera vista, pero tiene sus fallas a la hora de ser algo funcional. No es algo efectivo y eficiente que se esperaría de una base de datos funcional.



Algunas preguntas para motivar a sus estudiantes a que señalen y fundamenten los errores apreciables en este modelo de base de datos son:

- ¿Qué atributos podrían formar su propia tabla? (Entidad y atributos)
- ¿Qué significan los caracteres verdes al lado de los atributos?
- ¿Qué significa la “F” en el último atributo de la tabla “Empresa” y “Profesional”?

CONSTRUCCIÓN DE CONOCIMIENTO

Para construir el conocimiento sobre la forma correcta de hacer un modelo de base de datos, se sugiere leer los problemas cuidadosamente e intuir la información que quizás no esté complementada. El o la estudiante lee con atención y descifra las entidades, los atributos y las relaciones que debe tener el texto.

Marcará de color azul las entidades, de color rojo los atributos obligatorios, de color naranja los atributos opcionales y de color verde los verbos que le permitan reconocer relaciones entre entidades.

Caso biblioteca local o comunitaria

Un grupo de vecinos decide habilitar una biblioteca comunitaria en un salón sin uso del centro social. Lo primero es buscar una o más mesas grandes con superficie firme, altura adecuada y que puedan tener ruedas para moverlas. A un costado se colocarán sillas, cada una con un respaldo resistente y puede tener cojín o estar tapizada.

En las paredes se instalarán estanterías hechas con tablas recogidas, con capacidad para cinco repisas. Algunas pueden tener puertas livianas o estantes móviles. Las estanterías contendrán libros, revistas, cartillas y otros materiales impresos. Cada libro tendrá un título, un autor y una clasificación temática. Además, puede incluir una etiqueta de color o un código de préstamo.

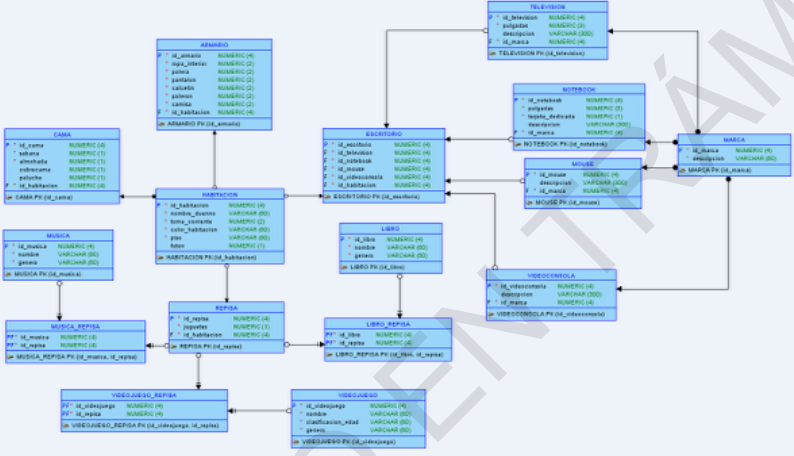
La biblioteca también contará con un pequeño espacio digital donde se encontrará al menos un computador portátil o un teléfono compartido. Allí se ubicarán enchufes,

alargadores y una regleta con interruptor, necesarios para conectar equipos. Cada enchufe tendrá un número y puede contar con protector de voltaje.

Una mesa auxiliar servirá para organizar actividades como lectura guiada o préstamo de libros. Cada actividad tendrá un nombre, un horario definido y puede requerir inscripción o moderación. Las actividades serán coordinadas por un referente comunitario, que puede ser un voluntario, una madre cuidadora o un joven con experiencia.

Acción	Resultado esperado	Preguntas de apoyo
Marcará de color azul las entidades, de color rojo los atributos obligatorios, de color naranja los atributos opcionales y de color verde los verbos que le permitan reconocer relaciones entre entidades.	Un grupo de vecinos decide habilitar una biblioteca comunitaria en un salón sin uso del centro social. Lo primero es buscar una o más mesas grandes con superficie firme, altura adecuada y que puedan tener ruedas para moverlas. A un costado se colocarán sillas, cada una con un respaldo resistente y puede tener cojín o estar tapizada. En las paredes se instalarán estanterías hechas con tablas recogidas, con capacidad para cinco repisas. Algunas pueden tener puertas livianas o estantes móviles. Las estanterías contendrán libros, revistas, cartillas y otros materiales impresos. Cada libro tendrá un título, un autor y una clasificación temática. Además, puede incluir una etiqueta de color o un código de préstamo. La biblioteca también contará con un pequeño espacio digital, donde se encontrará al menos un computador portátil o un teléfono compartido. Allí se ubicarán enchufes, alargadores y una regleta con interruptor, necesarios para conectar equipos. Cada enchufe tendrá un número y puede contar con protector de voltaje. Una mesa auxiliar servirá para organizar actividades como lectura guiada o préstamo de libros. Cada actividad tendrá un nombre, un horario definido y puede requerir inscripción o moderación. Las actividades serán coordinadas por un referente comunitario, que puede ser un voluntario, una madre cuidadora o un joven con experiencia.	¿Cómo determinar qué dato equivale a qué dato? ¿Cómo identificar las relaciones? ¿Qué sucede si hay relaciones que no se mencionan?

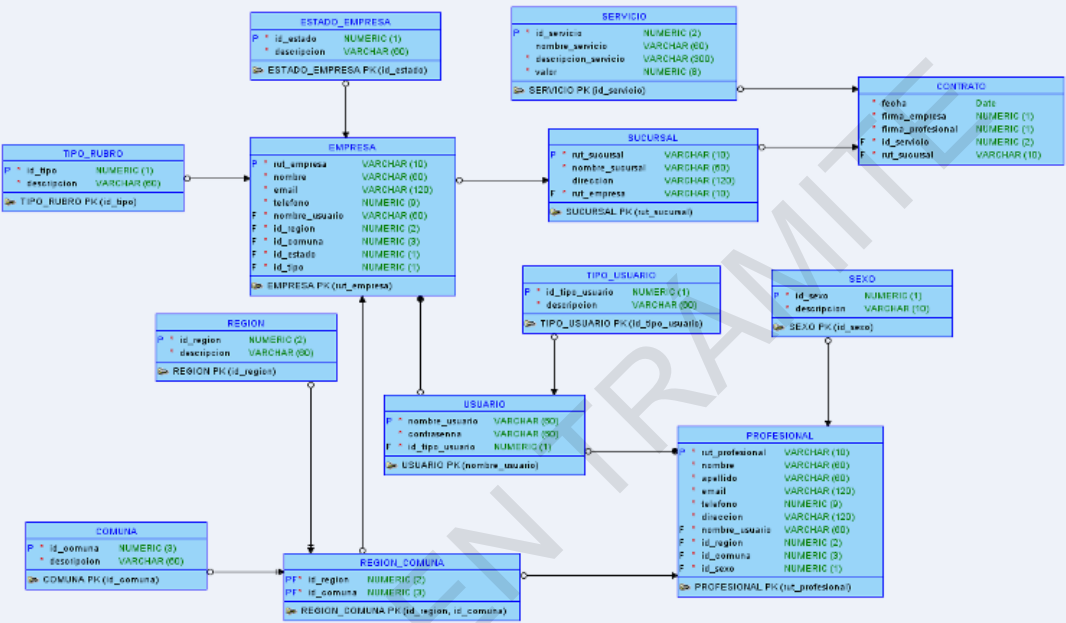
Acción	Resultado esperado	Preguntas de apoyo
Modele la base de datos que acaba de clasificar en el problema.	ENTIDAD: ESPACIO • # id_espacio • nombre_espacio • ubicación • tipo (interior, exterior, techado, etc.) ENTIDAD: MOBILIARIO • # id_mobiliario • nombre • estado • material • origen (donado, fabricado, conseguido) • requiere_reparación (boolean) ENTIDAD: ENCHUFE • # id enchufe • número • ubicación_exacta • tiene_protección (boolean) • tipo_conector ENTIDAD: LIBRO • # id_libro • título • autor • género • código_préstamo • condición (nuevo, usado, frágil) ENTIDAD: USUARIO • # id_usuario • nombre • edad • tipo_usuario (adulto, joven, voluntario) • identificación ENTIDAD: ACTIVIDAD • # id_actividad • nombre • horario • requiere_inscripción (boolean) • capacidad_max ENTIDAD: PRÉSTAMO • # id_prestamo • fecha_salida • fecha_retorno • observaciones • estado (activo, cerrado, vencido)	¿Cómo saber si las relaciones se hicieron bien?

Acción	Resultado esperado	Preguntas de apoyo
Analicé y evalué si se le pueden hacer mejoras al modelo de base de datos (aplicar formas normales).	RELACIONES SUGERIDAS LIBRO — PRÉSTAMO: un libro puede estar asociado a uno o más préstamos. USUARIO — PRÉSTAMO: un usuario puede generar muchos préstamos. ESPACIO — MOBILIARIO: un espacio contiene múltiples tipos de mobiliario. ESPACIO — ENCHUFE: un espacio puede tener varios enchufes instalados. ESPACIO — ACTIVIDAD: las actividades se desarrollan en uno o más espacios. 	¿Cómo saber qué atributo podría llevar su propia tabla? ¿Cómo saber cuándo aplicar las formas normales?

PRÁCTICA GUIADA

Para poder guiar la evaluación del modelamiento de base de datos, es necesario analizar los problemas y dar el mejor resultado, al igual que con los algoritmos. Para ello, la o el docente realizará el ejemplo que utilizamos anteriormente: **Ejemplo de formulario (simulado)** del **Registro de préstamo de insumos - Taller de Carpintería**, pero ahora con una tabla adicional.

Acción	Resultado esperado
Analice el modelo y vea si hay información que no debería estar allí.	Creación de un modelo relacional simple, que incluya entidades como HERRAMIENTA, USUARIO, PRÉSTAMO, y otras relacionadas como ESTADO_HERRAMIENTA, TIPO_HERRAMIENTA. El modelo debe incluir claves primarias (id_), relaciones (uno a muchos o muchos a muchos), y claves foráneas donde corresponda. EMPRESA - P * rut_empresa VARCHAR (10) * nombre VARCHAR (80) * email VARCHAR (120) * telefono NUMERIC (9) - F * nombre_usuario VARCHAR (80) - EMPRESA PK (rut_empresa) USUARIO - P * nombre_usuario VARCHAR (80) * contrasenna VARCHAR (80) - USUARIO PK (nombre_usuario) PROFESIONAL - P * rut_profesional VARCHAR (10) * nombre VARCHAR (80) * apellido VARCHAR (80) * email VARCHAR (120) * telefono NUMERIC (9) * direccion VARCHAR (120) - F * nombre_usuario VARCHAR (80) - PROFESIONAL PK (rut_profesional) SUCURSAL - P * rut_sucursal VARCHAR (10) * nombre_sucursal VARCHAR (80) * direccion VARCHAR (120) - SUCURSAL PK (rut_sucursal) CONTRATO * fecha Date * firma_empresa NUMERIC (1) * firma_profesional NUMERIC (1) SEXO - P * idsexo NUMERIC (1) * descripcion VARCHAR (10) - SEXO PK (idsexo) TIPO_USUARIO - P * id_tipo_usuario NUMERIC (1) * descripcion VARCHAR (80) - TIPO_USUARIO PK (id_tipo_usuario) ESTADO_EMPRESA - P * id_estado NUMERIC (1) * descripcion VARCHAR (80) - ESTADO_EMPRESA PK (id_estado) REGION - P * id_region NUMERIC (2) * descripcion VARCHAR (80) - REGION PK (id_region) SERVICIO - P * id_servicio NUMERIC (2) * nombre_servicio VARCHAR (80) * descripcion_servicio VARCHAR (300) * valor NUMERIC (8) - SERVICIO PK (id_servicio) COMUNA - P * id_comuna NUMERIC (3) * descripcion VARCHAR (80) - COMUNA PK (id_comuna)

Acción	Resultado esperado
Relacione las tablas de acuerdo con la lógica de un negocio (aplicar formas normales).	Aplicación de formas normales para reducir redundancias: - Separar atributos repetitivos (como tipos o estados) en entidades propias. - Detectar y eliminar dependencias funcionales incorrectas. - Optimizar el modelo relacional al mantener la consistencia de los datos y al facilitar futuras consultas. 

PRÁCTICA INDEPENDIENTE

Se sugiere una actividad individual en la que tengan que analizar un caso y ver cuáles son las entidades, los atributos y las relaciones de esta. Se puede utilizar el ejercicio de la práctica independiente anterior o uno nuevo. De cualquier modo, lo ideal es siempre analizar punto por punto junto a la clase y que después hagan el respectivo modelo de base de datos individualmente.

Pregunta reflexiva sobre eficiencia y mejora (criterio de toma de decisiones):

¿Cómo ayuda el simplificar y ordenar la información (a través de la normalización de datos) a tomar mejores decisiones, evitar errores y hacer más eficiente el uso de recursos en los proyectos o actividades de tu comunidad o espacio de vida actual?

EVALUACIÓN FORMATIVA

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Analizar problemas	Analiza el problema.	Analiza el problema y consigue obtener la mayoría de los datos.	Analiza el problema y logra identificar todos los datos de la información.
Relacionar tablas	Relaciona las tablas con errores.	Relaciona las tablas con algunos errores.	Logra relacionar las tablas sin errores.
Formas normales	Aplica las formas normales con errores.	Aplica las formas normales con algunos desaciertos.	Aplica las formas normales de buena manera, además de saber cuándo ya es suficiente.

Orientaciones al docente

Para unificar conceptos disciplinares

Entenderemos que la normalización de la base de datos es una técnica de modelado que consiste en designar y aplicar una serie de reglas a las relaciones obtenidas del paso del diagrama entidad-relación al modelo relacional. Las relaciones y normalizar un modelo es primordial en esta actividad. Se sugiere explicar y dar ejemplos para poder entender de mejor manera las tres formas principales de relaciones (1:1 – 1:M – M:N). Además, es necesario entender el concepto de las tres formas normales y en qué circunstancias se utilizan. Se puede utilizar el siguiente ejemplo:

- Primera forma normal: los atributos contienen valores simples.

En la siguiente tabla hay un atributo con doble valor, no es simple. Entonces, esto se puede solucionar al eliminar el atributo o se incluye un nuevo atributo como segundo correo.

ID	NOMBRE	PUESTO	SALARIO	correo electrónico
12	Fabiola Reyes	Secretaria	200	fr@empresa.com
13	Elsa Fredes	Secretaria	200	elf@empresa.com; f@empresa.gma x
14	Ani Lindo	Administrativa	150	al@empresa.com

- Segunda forma normal: los atributos contienen valores simples y todos los atributos que no son clave primaria tienen dependencia funcional completa con respecto a todas las claves existentes en el esquema. Para recuperar un atributo, se accede por la clave completa.

ID	NOMBRE	PUESTO	SALARIO	correo electrónico
12	Fabiola Reyes	Secretaria	200	fr@empresa.com
13	Elsa Fredes	Secretaria	200	elf@empresa.com
13	Elsa Fredes	Secretaria	200	f@empresa.gma
14	Ani Lindo	Administrativa	150	al@empresa.com

Para hacer una segunda forma normal, se tienen dos tablas posibles ID-nombre-puesto-salario:

ID	NOMBRE	PUESTO	SALARIO
12	Fabiola Reyes	Secretaria	200
13	Elsa Fredes	Secretaria	200
14	Ani Lindo	Administrativa	150

La otra es ID-correo:

ID	correo electrónico
12	fr@empresa.com
13	elf@empresa.com
13	f@empresa.gma
14	al@empresa.com

- Tercera forma normal: está en primera y segunda forma normal. Además, los atributos que no son clave no dependen transitivamente.

ID	NOMBRE	PUESTO	SALARIO
12	Fabiola Reyes	Secretaria	200
13	Elsa Fredes	Secretaria	200
14	Ani Lindo	Administrativa	150

La tabla tiene dependencias transitivas ID-puesto y puesto-salario, entonces ID-salario. Para normalizar esta tabla hay dos posibilidades para separar: la primera tabla es puesto y salario:

PUESTO	SALARIO
Secretaria	200
Administrativa	150

Y la segunda tabla es ID-nombre-puesto, considerando el puesto como una clave foránea.

ID	NOMBRE	PUESTO
12	Fabiola Reyes	Secretaria
13	Elsa Fredes	Secretaria
14	Ani Lindo	Administrativa

Actitudes

Para apoyar el desarrollo de las actividades y el modelado de la base de datos, es recomendable la utilización de computadores, para así entregar una retroalimentación más visual de lo que realizan de manera correcta e incorrecta. Además, se debe ser constante al hacer un modelo de base de datos. Las bases aprendidas en la actividad anterior deben estar solucionadas y ser comprendidas por parte del estudiantado. Se sugiere motivar a sus estudiantes para que puedan resolver sus dudas y así comprender los elementos faltantes para lograr resolver la actividad. En cuanto al desarrollo de los trabajos individuales, se recomienda promover la autonomía y el pensamiento de cada estudiante, así como la confianza en sus conocimientos previos.

Orientaciones para organizar e implementar trabajo individual

Se sugieren las siguientes indicaciones para guiar la implementación del trabajo individual:

- Elaborar un ejercicio amplio que esté centrado en algún sistema de negocio o minimercado, para que los y las estudiantes identifiquen las tablas necesarias. Se sugiere pensar en las relaciones que se dan en la situación para, luego, utilizarlas en las próximas actividades.
- Determinar un tiempo para poder analizar y revisar todo el problema, y para identificar las tablas que esconde.
- Invitar a los y las estudiantes a revisar entre todas las respuestas que escogieron. Esto beneficiará a quienes no lograron identificarlas todas. Además, se podrán aclarar dudas sobre cómo identificar cada caso. Una vez finalizada la retroalimentación, deberán trabajar en las mejoras del modelamiento de la situación e incluir aquellas relaciones o pasos que son necesarios. Puede ser en una hoja, pero se recomienda siempre el uso de computadores para estos casos.
- Determinar el tiempo para desarrollar el modelo de base de datos con sus respectivas entidades, atributos y relaciones, y aplicar las formas normales.
- Dar retroalimentación de la tarea. Para ello, se sugiere pedir a alguien del curso que exponga o al final de la clase mostrar cómo debería ser el modelo y por qué debería quedar así. Explicar cómo lo analizó y lo desarrolló.

Visión panorámica

Gran idea

El comportamiento de un sistema computacional (dispositivos, *hardware* y *software*) depende de la secuencia de instrucciones dadas por el usuario que automatizarán el desempeño de una tarea para ejecutar una acción o resolver un problema.

Objetivos de aprendizaje

OA 2

Organizar datos de una situación problemática, estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función. [\(Evaluar\)](#)

OA 3

Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponibles. [\(Modelar\)](#)

Conocimientos esenciales

- Modelos entidad-relación y simulación de sistemas.
 - Sistemas computacionales: dispositivos, hardware y software, y resolución de problemas.
-

Tiempo estimado: 6 semanas (12 horas)

Propósito

En el módulo 2 de la asignatura Pensamiento Computacional de los Niveles 1 y 2 de Educación Media, se espera que las y los estudiantes refuercen aprendizajes previos y que, de forma inicial, comprendan que organizar y modelar datos está asociado a las instrucciones que dan a un sistema computacional. Esto con el objetivo de recibir una respuesta deseada. Además, se incorpora la idea de, que a través del uso de dispositivos, hardware y software diferentes, se automatizan desempeños de acuerdo con las tareas que deseen realizar. La idea de la relación entre el ser humano y la máquina se refuerza desde los niveles de interacción que tengan los y las estudiantes con herramientas digitales en cada actividad de desempeño en la que tengan que diseñar una solución y resolver un problema.

Las y los estudiantes desarrollan nuevos conocimientos y profundizan habilidades para resolver problemas cada vez más complejos, que requieren una organización y modelado de datos más precisos y óptimos. En el proceso la identificación de fallos, se debe considerar el uso de

criterios de usabilidad y simulaciones simples, además de evaluar las tecnologías digitales que tengan disponibles para entregar soluciones acordes a las necesidades que se plantean. El énfasis del módulo está en que el estudiantado confíen en sus capacidades para interactuar con uno o más sistemas computacionales. La evaluación es una habilidad que necesitan seguir trabajando para asegurar que una solución, ya sea un algoritmo, un sistema o un proceso, cumpla eficientemente con su propósito de conseguir una respuesta de acuerdo con lo requerido.

Los Objetivos de Aprendizaje del módulo 2 desarrollan las actitudes del siglo XXI del ámbito de las Maneras de pensar y Herramientas para trabajar, promoviendo el uso provechoso de las herramientas para resolver problemas o diseñar soluciones, con apertura en distintas perspectivas y contextos. Finalmente, las actividades de desempeño propuestas esperan ser una oportunidad para comprender el uso de las tecnologías y cómo las personas interactúan con ellas de forma cotidiana para satisfacer necesidades.

Ruta de Aprendizaje

¿Cómo se modelan los niveles de interacción entre los sistemas computacionales y los sistemas sociales, ambientales o industriales que enfrenta un usuario?

●	Actividad de desempeño 1	Construyen modelos de base de datos mediante el uso de entidades, atributos y relaciones, para obtener diferentes formas de resolver problemas.
●	Actividad de desempeño 2	Resuelven problemas utilizando los condicionales, mientras, repetir y para, aprovechando las herramientas para analizar y evaluar un problema.
●	Actividad de desempeño 3	Modelan y evalúan subprocesos, implementando e implementan las funcionalidades ya conocidas de acuerdo con las necesidades del problema.
●	Actividad de desempeño 4	Modelan y evalúan formas para programar mediante el uso 1, 2 y 3 dimensiones, y la resolución de problemas de algoritmos.

Actividad de desempeño

1

Propósito

Esta actividad está orientada a consolidar los conocimientos relacionados con el modelo de base de datos y, a su vez, conocer la nueva entidad de relación extendida, para tener un mayor número de herramientas y formas de modelar, y así realizar una base de datos de forma correcta. Estas nuevas formas son elementos extras que pueden cambiar significativamente el resultado final de un proceso para resolver un problema.

Objetivos de Aprendizaje

OA 3

Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponible. (Modelar)

Conocimiento esencial

- Modelos entidad-relación y simulación de sistemas.

Tiempo estimado: 3 horas pedagógicas

Desarrollo de la actividad

SITUACIÓN EXPERIENCIAL

El o la docente presenta diferentes modelos entidad-relación y una progresión de la inclusión de símbolos para agregar más información:

Modelo entidad-relación	Descripción
	El jefe asigna tareas a trabajadores y trabajadoras, quienes finalizan las tareas.
	1: n 1 persona puede dirigir varios (n) proyectos.
	Relaciona dos entidades y expresa 1: n Cada proyecto puede ser ejecutado por una sola empresa. Así, se utiliza la llave primaria "Nombre" de la tabla "Banco" como una llave extranjera para muchos proyectos.

Algunas de las preguntas que pueden motivar a sus estudiantes a indicar las diferencias entre estos tres modelos de base de datos son:

- ¿Cuáles son las diferencias?
- ¿Qué es lo nuevo?
- ¿Qué semejanzas existen entre los modelos?
- ¿Para qué se puede usar cada una?

CONSTRUCCIÓN DE CONOCIMIENTO

Para que las y los estudiantes sigan consolidando la habilidad de modelar una base de datos, se sugiere explicar que la lectura cuidadosa de textos es esencial para inferir información que no está explícita, es decir, cada estudiante debería leer con atención para identificar las entidades, los atributos y las relaciones que están en el texto propuesto.

Caso red de bibliotecas comunitarias “Conecta-Lecturas”

Una red de bibliotecas comunitarias “Conecta-Lecturas” necesita organizar un sistema que facilite el préstamo y la gestión de libros, recursos y espacios de trabajo en distintos territorios. Las normas de funcionamiento fueron compartidas por la encargada de la coordinación territorial. Cada biblioteca lleva registro de usuarios que pueden acceder a los servicios si se inscriben, para lo cual deben entregar su RUT, nombre completo, dirección, comuna, correo electrónico, número de contacto y si participan en algún programa educativo o taller local. Si la persona declara pertenecer a un taller, debe mencionar el nombre del facilitador o monitora a cargo. Si no participa de talleres, se solicita indicar el tipo de espacio que más utiliza (lectura, trabajo grupal, conexión digital, etc.) y con qué frecuencia. No se puede declarar ambas situaciones.

Cada persona puede inscribirse solo una vez por biblioteca. La inscripción indica los datos personales, la fecha de inscripción, firma digitalizada y foto del carnet. También se registra si necesita adaptaciones por accesibilidad (opcional).

Luego, la persona puede solicitar préstamo de libros o uso de espacios. En la solicitud se ingresan los datos del recurso o espacio requerido: nombre del libro o tipo de sala (reunión, lectura, conectividad), el horario solicitado y si se requiere uso de enchufe, red Wi-Fi o apoyo de facilitador. Además, se asocia un código de solicitud, la fecha de creación de la solicitud y se puede agregar un comentario u observación sobre el uso. Cada usuario puede ingresar hasta tres solicitudes simultáneas para diferentes recursos.

Las monitoras de cada sede deben registrar el estado del recurso antes y después de cada uso. Pueden evaluar el estado del libro o espacio, describir la condición del material (buen estado, deterioro leve, etc.), registrar si se respetaron los horarios y asignar un nivel de uso responsable (escala del 1 al 5). Cada solicitud puede ser evaluada, pero solo una vez. Además, pueden trabajar en parejas rotativas entre sedes, por lo que se debe registrar quién se asignó a qué sede y qué tareas realizó. Es necesario registrar su RUT, nombre completo, correo institucional y código de monitora para facilitar la trazabilidad.

Acción	Resultado esperado															
Marcar de color azul las entidades, de color rojo los atributos obligatorios, de color naranja los atributos opcionales y de color verde los verbos que le permitan reconocer las relaciones entre entidades.	<table><tr><th>Color</th><th>Clasificación</th><th>Ejemplo del texto</th></tr><tr><td>Azul</td><td>Entidades</td><td>USUARIO, MONITORA, INSCRIPCIÓN, SOLICITUD, RECURSO, EVALUACIÓN, BIBLIOTECA.</td></tr><tr><td>Rojo</td><td>Atributos obligatorios</td><td>RUT, nombre completo, dirección, fecha de inscripción, tipo de recurso, estado, evaluación.</td></tr><tr><td>Naranja</td><td>Atributos opcionales</td><td>Participación en taller, comentarios, accesibilidad, uso de enchufe, Wi-Fi, apoyo facilitador.</td></tr><tr><td>Verde</td><td>Verbos que indican relación</td><td>Inscribirse, solicitar, registrar, evaluar, asociar, trabajar, utilizar.</td></tr></table>	Color	Clasificación	Ejemplo del texto	Azul	Entidades	USUARIO, MONITORA, INSCRIPCIÓN, SOLICITUD, RECURSO, EVALUACIÓN, BIBLIOTECA.	Rojo	Atributos obligatorios	RUT, nombre completo, dirección, fecha de inscripción, tipo de recurso, estado, evaluación.	Naranja	Atributos opcionales	Participación en taller, comentarios, accesibilidad, uso de enchufe, Wi-Fi, apoyo facilitador.	Verde	Verbos que indican relación	Inscribirse, solicitar, registrar, evaluar, asociar, trabajar, utilizar.
	Color	Clasificación	Ejemplo del texto													
	Azul	Entidades	USUARIO, MONITORA, INSCRIPCIÓN, SOLICITUD, RECURSO, EVALUACIÓN, BIBLIOTECA.													
	Rojo	Atributos obligatorios	RUT, nombre completo, dirección, fecha de inscripción, tipo de recurso, estado, evaluación.													
	Naranja	Atributos opcionales	Participación en taller, comentarios, accesibilidad, uso de enchufe, Wi-Fi, apoyo facilitador.													
Verde	Verbos que indican relación	Inscribirse, solicitar, registrar, evaluar, asociar, trabajar, utilizar.														
Modelar la base de datos que acaba de clasificar en el problema.	USUARIO • # rut • nombre_completo • dirección • comuna • correo • teléfono • tipo_uso_preferido (si no participa en talleres) INSCRIPCIÓN • # id_inscripcion • fecha_inscripcion • firma • foto_carnet • requiere_adaptaciones (opcional) • id_biblioteca (FK a BIBLIOTECA) • rut_usuario (FK a USUARIO) MONITORA • # id_monitora • rut • nombre_completo • correo_institucional BIBLIOTECA • # id_biblioteca • nombre • dirección • comuna RECURSO • # id_recurso • nombre_recurso • tipo (ejemplo: libro, sala, dispositivo) • requiere enchufe (booleano) • requiere_wifi (booleano) • requiere_apoyo (booleano)															

Acción	Resultado esperado
Modelar la base de datos que acaba de clasificar en el problema.	SOLICITUD • # id_solicitud • fecha_creacion • comentario (opcional) • id_recurso (FK a RECURSO) • rut_usuario (FK a USUARIO) EVALUACIÓN • # id_evaluacion • estado_recurso (texto) • condicion_material (ejemplo: bueno, regular, deteriorado) • puntaje_uso_responsable (1 a 5) • id_recurso (FK a RECURSO) • id_monitora (FK a MONITORA) Este modelo permite construir relaciones entre usuarios, monitoras, solicitudes y recursos disponibles en las bibliotecas, lo que permite un control distribuido en red.
Analizar y evaluar si se pueden hacer mejoras al modelo de base de datos.	El o la estudiante revisa el modelo de base de datos que incluye las entidades USUARIO, INSCRIPCIÓN, MONITORA, BIBLIOTECA, RECURSO, SOLICITUD y EVALUACIÓN, e identifica aspectos que se pueden mejorar para que el sistema funcione mejor y sea más fácil de entender y usar. A partir del análisis, cada estudiante propone cambios, como crear una tabla para las comunas en lugar de escribir el nombre directamente, usar listas cerradas para ciertos campos (como el estado del recurso), o permitir registrar varias evaluaciones por recurso. También señala si hay datos que se repiten o relaciones que faltan. Finalmente, explica con sus palabras por qué esos cambios harían que el sistema sea más ordenado, más fácil de mantener y más útil para quienes lo usan en una biblioteca o centro comunitario.

PRÁCTICA GUIADA

Para guiar la evaluación de modelar una base de datos con las nuevas relaciones, es importante considerar que la mayor información debe obtenerse a través de la documentación, pero también puede observarse en ejemplos de la interfaz que acompañan al modelo. A continuación, se muestran pantallas de diseño de inicio de sesión, registro de personas y selección de piezas básicas de computador, para pensar cómo podrían implementarse las tablas.

Antes de comenzar, se recomienda escribir y dibujar una plantilla de diseño con casilleros, también llamada mockup. Este corresponde a un dibujo o esquema que representa cómo se verían las pantallas de un sistema digital, incluyendo los casilleros para ingresar datos, botones y listas. Sirve para planificar el sistema antes de programarlo o crear la base de datos.

En RearmaCom, un centro comunitario de recuperación de computadores, las empresas locales donan equipos antiguos para ser reacondicionados y distribuidos a personas o instituciones que los necesiten. Para mantener el orden, el registro de PERSONAS usuarias y la gestión de CONTRATOS solo puede ser realizada por un PROFESIONAL autorizado del centro.

Después de iniciar sesión como USUARIO, la profesional puede registrar una persona nueva (PERSONA), con sus datos personales y contacto. También puede gestionar los contratos de donación (CONTRATO), donde se indica la empresa (EMPRESA_LOCAL),

la cantidad de equipos donados, la sucursal involucrada (SUCURSAL) y las firmas digitales requeridas.

Cada computador está compuesto por piezas (PIEZA_COMPUTADOR) que deben registrarse individualmente, especificando su tipo (TIPO_PIEZA), sus características técnicas y su precio estimado. Esto ayuda a llevar el control sobre qué piezas pueden ser reutilizadas o deben ser recicladas.

Cada persona registrada pertenece a una comuna (COMUNA), que a su vez se vincula a una región (REGION), y tiene un sexo (SEXO). Según su rol, el tipo de usuario (TIPO_USUARIO) puede ser beneficiario, técnico, voluntario, etcétera.

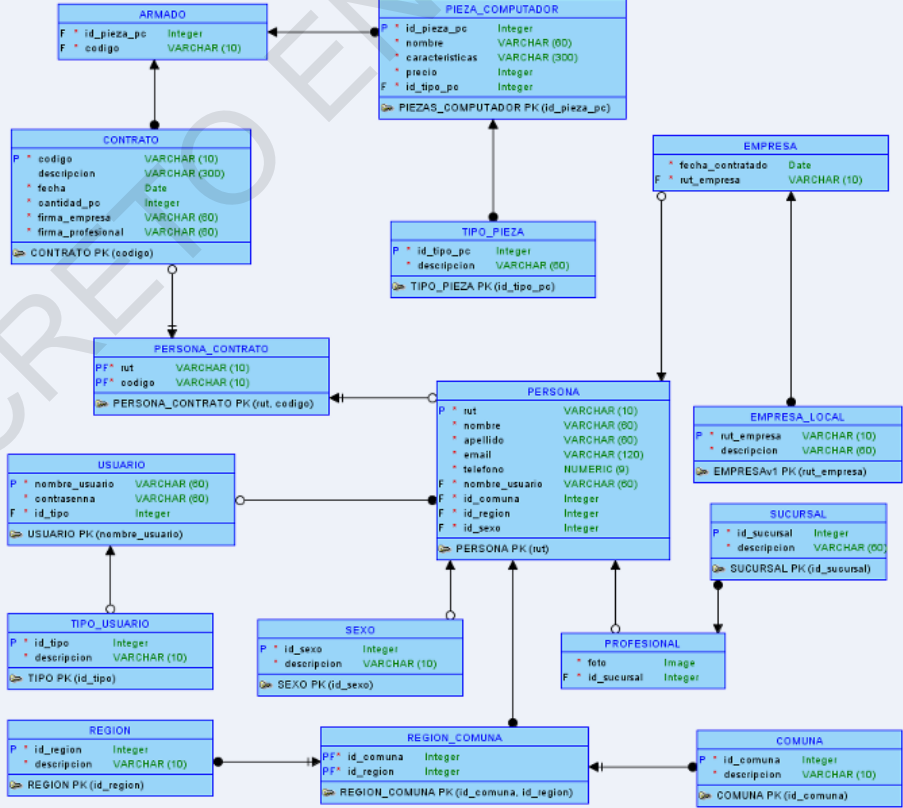
Finalmente, todos los datos quedan vinculados en el sistema para asegurar trazabilidad y responsabilidad en la entrega, reutilización o descarte de los computadores RearmaCom.

A continuación, se muestran plantillas de diseños con casilleros para completar, o mockups de inicio de sesión, del formulario de registro y de selección de piezas básicas.

The image displays three distinct user interface mockups for the RearmaCom system, arranged horizontally. Each mockup is enclosed in a thin black border and features a light gray background with various form elements.

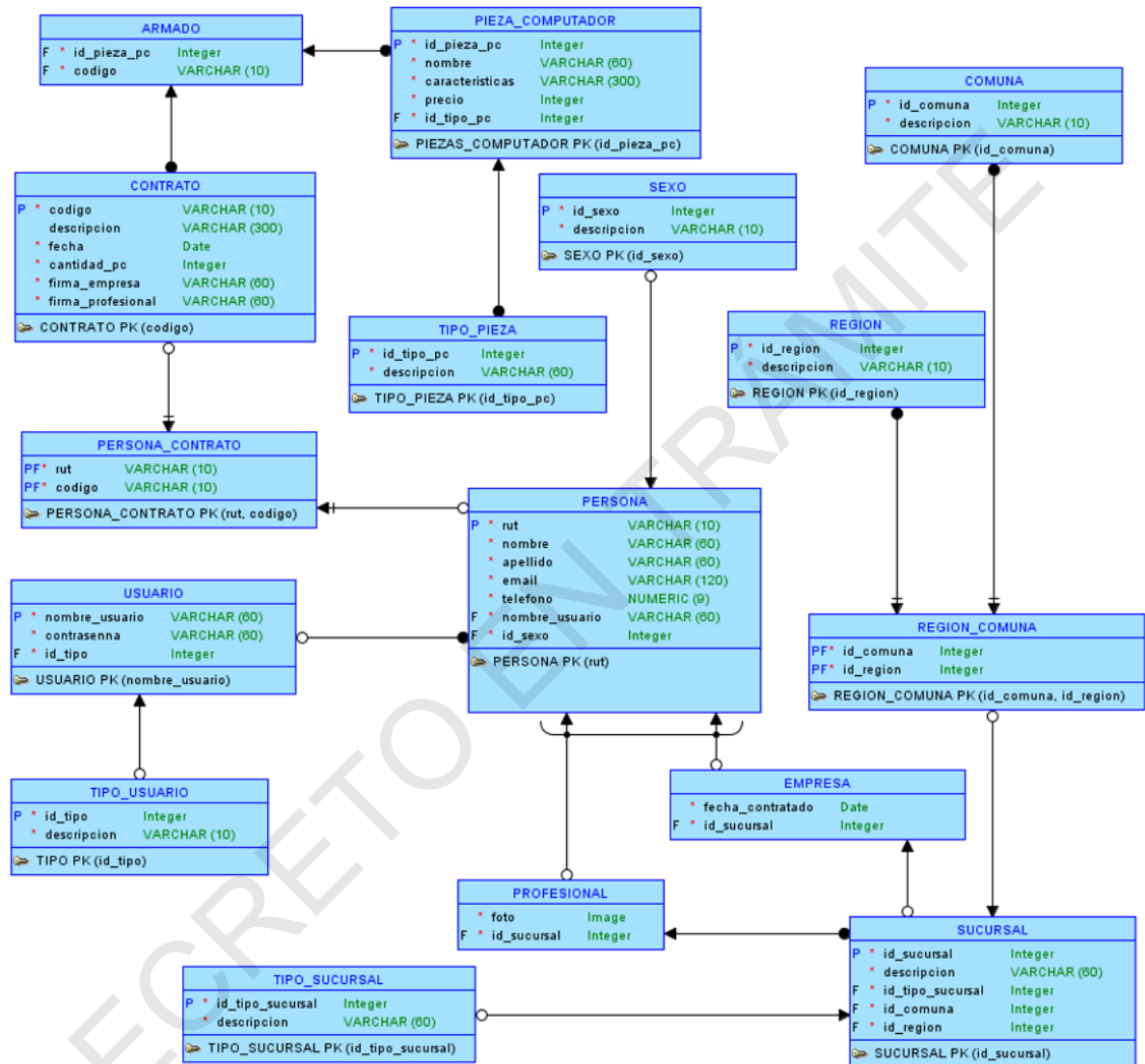
- Mockup 1 (Left): 'Iniciar sesion'**
 - Header: 'Iniciar sesion' with a user icon.
 - Fields: Two input boxes labeled 'NOMBRE USUARIO' and 'CONTRASEÑA'.
 - Action: A blue button labeled 'INICIAR SESION'.
- Mockup 2 (Middle): 'FORMULARIO DE REGISTRO'**
 - Header: 'FORMULARIO DE REGISTRO' with a user icon.
 - Fields: A grid of input boxes for 'RUT', 'CORREO ELECTRONICO', 'NOMBRE', 'APELLIDO', 'NOMBRE USUARIO', 'CONTRASEÑA', 'REGION' (dropdown), 'COMUNA' (dropdown), 'DIRECCION', 'TELEFONO', and 'SELECCIONE SEXO' (dropdown).
 - Options: Two radio buttons for 'EMPRESA' and 'PROFESIONAL'.
 - Fields: 'SELECCIONE EMPRESA' (dropdown), 'SELECCIONE SUCURSAL' (dropdown), 'FECHA CONTRATADO', and 'SUBIR FOTO PERFIL' (with a camera icon).
 - Action: A blue button labeled 'REGISTRARSE'.
- Mockup 3 (Right): 'SELECCIONAR PIEZAS BASICAS'**
 - Header: 'SELECCIONAR PIEZAS BASICAS' with a computer icon.
 - Section 1: 'Procesador (CPU) *' with a dropdown menu showing '(3985) Ryzen 5 - 5600X (AM4) \$300'. Below it, a small product image, 'AMD CPU RYZEN - 5600X (AM4)', 'ID: 3985', a link 'VER DETALLE', and the price '\$300'.
 - Section 2: 'PLACA MADRE *' with a dropdown menu showing '(4165) ASUS M/B AMD X570 TUF GAMING W/ FI (AM4)'. Below it, a small product image, 'ASUS M/B AMD X570 TUF GAMING W/ FI (AM4)', 'ID: 4165', a link 'VER DETALLE', and the price '\$200'.

Al hacer modelos, se sugiere considerar elementos que sus estudiantes conozcan, que puedan ser tabulados y que su significado sea directo. Es decir, datos que no tengan alguna connotación diferente para el usuario, como lo es el uso de la clave "sexo". Por otra parte, se acostumbra siempre a tener las mismas claves como "región" y "comuna". Se sugiere promover diferentes claves según los trabajos o las aspiraciones del curso y tener en cuenta que los productos, las herramientas y las pertenencias también tienen atributos que pueden ser consultados.

Acción	Resultado esperado
Identificar las tablas que debería llevar este modelo de base de datos.	
Desarrollar el modelo de base de datos con sus relaciones.	

PRÁCTICA INDEPENDIENTE

Se sugiere una actividad individual de comparación de dos modelos entidad-relación. Para esto, se puede considerar el modelo obtenido en la práctica guiada y un modelo al que se le han realizado algunos cambios. Se sugiere revisar punto por punto para ver en qué parte se hacen los cambios y cómo estos contribuyen a la optimización del modelo.



Darle importancia a que en la selección de región y comuna junto con la entidad sucursal y observar que, en vez de haber una sucursal y una empresa local, se puede dejar solo la sucursal, como también las diferencias entre sucursales de la empresa o de las tiendas de RearmaCom (Centro comunitario ejemplificado al inicio de la práctica guiada).

EVALUACIÓN FORMATIVA

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Problemática	Analiza el problema.	Analiza el problema y consigue obtener la mayoría de los datos.	Analiza el problema y logra identificar todos los datos de la información.
Tablas	Relaciona las tablas con errores.	Relaciona las tablas con algunos errores.	Relaciona las tablas sin errores.
Formas normales	Aplica las formas normales con errores.	Aplica las formas normales con algunos desaciertos.	Aplica las formas normales de buena manera, además de saber cuándo ya es suficiente.

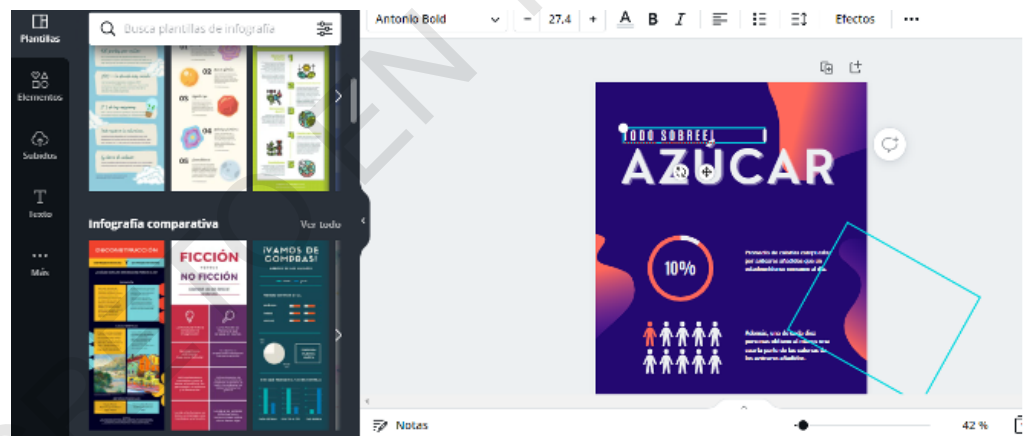
Orientaciones al docente

Para unificar conceptos disciplinares

Hay que considerar que cada aspecto de la base de datos incide en que el resultado final sea óptimo y funcione de la mejor manera al guardar información y complementar alguna aplicación para la cual se crea dicha base de datos. Es importante priorizar que los y las estudiantes analicen cada aspecto de los problemas que se trabajan en las actividades, ya que es fundamental contar con una buena base de descomposición para llegar al resultado esperado. Los nuevos conceptos se pueden incorporar de manera sencilla, pero es necesario tener presente que corresponden a la 3ª forma normal. Si bien esta se utiliza con poca frecuencia, cuando se requiera será una nueva forma de implementar alternativas a la hora de resolver problemas.

- Los nuevos símbolos permiten explicar jerarquías, herencia, exclusividad de relaciones y relaciones intransferibles.
- La jerarquía es el proceso de subdividir una entidad en varias subentidades, relacionándolas con la entidad a la que se refieren o a la entidad PADRE, por lo tanto, las subentidades son HIJOS.
- La herencia, por su parte, es la implementación de la jerarquía de subtipos y supertipos, dado que los subtipos heredan todos los atributos de la entidad supertipo.
- El arco exclusivo es una restricción sobre dos o más relaciones de una entidad. Un arco indica que cualquier instancia de una entidad puede tener solo una relación válida respecto a las relaciones que están rodeadas por el arco. Por lo tanto, el arco representa una relación exclusiva de las relaciones entre el arco y puede usarse también en relaciones de subtipo.

- Reglas del subtipo:
 - ▶ Un subtipo hereda todos los atributos del supertipo.
 - ▶ Un subtipo hereda todas las relaciones del supertipo.
 - ▶ Un subtipo normalmente tiene sus propios atributos o relaciones.
 - ▶ Un subtipo nunca existe solo, requiere del supertipo.
 - ▶ Un subtipo también se conoce como subentidad.
 - ▶ Un subtipo puede tener sus propios subtipos, estos se denominan subtipos anidados.
- Se sugiere utilizar una aplicación para el modelamiento de bases de datos, como Oracle Data Modeler 20 u otra alternativa gratuita, emergente o accesible que permita representar entidades y relaciones de forma clara. Un mockup es una plantilla que viene con un diseño y sirve para replicar u obtener información. En estas plantillas se tienen casilleros que pueden ser cambiados con nueva información. Se pueden encontrar plantillas de libre disposición en fuentes como www.canva.com o slidego y, en caso de no tener acceso a internet, se pueden utilizar plantillas manuales. Estas se elaboran con cartón, marcando en él los espacios o casilleros. En caso de que se haga en forma manual, se sugiere practicar el tipo de letra, tener plantillas con tipos de letras o trabajar a modo de collage en la edición de estos elementos.



Actitudes

Para apoyar el desarrollo de las actividades, es recomendable el uso de los computadores para modelar la base de datos, así podrán obtener una retroalimentación más visual y ver los aciertos y desaciertos que logran al resolver los problemas. En cuanto al desarrollo del trabajo individual, se sugiere promover la independencia y pensamiento propio de cada estudiante.

**Orientaciones para
organizar e implementar
el juego de roles**

Se sugieren las siguientes indicaciones para guiar la implementación del trabajo individual:

- Elaborar un ejercicio amplio que esté centrado en algún sistema de negocio o minimercado, para que los y las estudiantes identifiquen las tablas necesarias. Se sugiere pensar también en las relaciones que se dan en la situación, para utilizarlas en las próximas actividades.
- Determinar un tiempo para poder analizar y revisar todo el problema; para identificar las tablas que esconde.
- Invitar a las y los estudiantes a revisar entre todas las respuestas que escogieron. Esto beneficiará a quienes no lograron identificarlas todas. Además, se podrán aclarar dudas de cómo identificar cada caso. Una vez finalizada la retroalimentación, los y las estudiantes deberán trabajar en las mejoras del modelamiento de la situación e incluir las relaciones o los pasos necesarios. Aunque puede ser en una hoja, se recomienda siempre el uso de computadores.
- Determinar el tiempo para poder desarrollar el modelo de base de datos con sus respectivas entidades, atributos, relaciones y aplicar las formas normales.
- Dar retroalimentación de la tarea. Para ello, se sugiere pedir a alguien del curso que exponga o al final de la clase mostrar cómo debería ser el modelo y por qué debería quedar así. Explicar cómo lo analizó y lo desarrolló.
- Adaptar la complejidad de las actividades al nivel de conocimiento de quienes participan.
- Utilizar ayudas visuales como diagramas y fichas para facilitar la comprensión.
- Fomentar la participación activa y el trabajo en equipo.
- Dedicar tiempo a la reflexión sobre la importancia de la organización de la información.

Actividad de desempeño

2

Propósito

En esta actividad se retomará el funcionamiento del algoritmo y se incorporará una nueva función de la estructura de los repetidores o ciclos. Esto con el fin de concluir esta estructura y, a su vez, fortalecer el conocimiento con más práctica, al tiempo que se promueve el análisis de los problemas y la búsqueda de las mejores soluciones posibles.

Objetivos de aprendizaje

OA 2

Organizar datos de una situación problemática estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función. [\(Evaluar\)](#)

Conocimiento esencial

- Sistemas computacionales: dispositivos, *hardware* y *software*, y resolución de problemas.
-

Tiempo estimado: 3 horas pedagógicas

Desarrollo de la actividad

SITUACIÓN EXPERIENCIAL

El o la docente presenta una situación cotidiana, como llegar a un servicio público o a un centro de salud donde hay que sacar un número para recibir atención. Explica, como si lo estuviera relatando en voz alta, que uno se acerca a una máquina y esta le pide en la pantalla que ingrese su RUT o un código para recibir su ticket. ¿Qué ocurre si uno se equivoca al digitarlo? La máquina nos mostrará un error y nos pedirá ingresarlo nuevamente hasta que sea correcto.

Se sugiere conversar, en términos de la programación, qué ocurre en esta situación. Algunas preguntas que pueden fomentar las nociones de las funciones **mientras** y **para** son:

- ¿Cuáles son las preguntas o instrucciones del sistema de atención automático?
- ¿Qué sucede si nos equivocamos en la cifra?
- ¿Qué consideraciones hay que tener para elaborar un algoritmo?

CONSTRUCCIÓN DE CONOCIMIENTO

En esta nueva forma de implementar se requiere de un código para un tótem de atención. El sistema debe solicitar el RUT del usuario y darle 3 oportunidades para ingresarlo correctamente. Si se equivoca las 3 veces, la máquina se bloquea y le pide al usuario que solicite ayuda en un mesón. Si acierta, podrá ver las opciones de trámites.

El menú tendrá las siguientes opciones:

```
*****
* ***** TÓTEM DE ATENCIÓN ***** *
*
* * INGRESE SU RUT (sin puntos ni guión)*
*          * INTENTO NRO 1
*
* RUT: >
*
* OPCIÓN: >
*****

*****
* ***** MENÚ DE OPCIONES ***** *
*
* 1. INFORMACIONES *
* 2. ATENCIÓN COMERCIAL *
* 3. REALIZAR UN TRÁMITE *
* 4. SALIR *
*****
```

- Informaciones: el usuario recibe un número para ser atendido en el mesón de informaciones generales y resolver dudas rápidas.
- Atención comercial: se le asigna un número para ser atendido por un ejecutivo de cuentas o para contratar un nuevo servicio.
- Realizar un trámite: se entrega un ticket para las ventanillas donde se realizan pagos, se entregan o retiran documentos específicos.
- Salir: la operación se cancela y la pantalla vuelve al inicio.

Acción	Resultado esperado	Preguntas de apoyo
Realizar el menú de las opciones del cajero automático.	<pre> Definir saldo, monto como Real; Definir contador, opc, salir como Entero; Definir usuario, clave como Caracter; //SALDO DEL CLIENTE saldo ← 100000; // CLAVES DE ACCESO usuario ← '5555'; Mientras (contador ≠ 4) Hacer Si (contador ≠ 3) Entonces Escribir "***** CAJERO AUTOMÁTICO *****"; Escribir "* INTENTO NRO ", contador + 1, " *"; Escribir "*****"; Escribir " "; Escribir Sin Saltar "CLAVE : "; Leer clave; Escribir ""; Limpiar Pantalla; Si (clave == usuario) Entonces Mientras (salir ≠ 4) Hacer Limpiar Pantalla; Escribir "* BIENVENIDOS AL BANCO *"; Escribir "MENU DE OPCIONES"; Escribir "1. VER SALDO"; Escribir "2. DEPOSITO"; Escribir "3. RETIRO"; Escribir "4. SALIR"; Escribir Sin Saltar "OPCIÓN : "; Leer opc; </pre>	¿Cómo verificar si el usuario es correcto o incorrecto? ¿Qué ciclo o repetidor usar?
Realizar las operaciones que hará el cajero automático.	<pre> Segun (opc) Hacer 1: Escribir "SALDO ACTUAL : ", saldo; Esperar Tecla; 2: Escribir Sin Saltar "MONTO A DEPOSITAR : "; Leer monto; saldo ← saldo + monto; 3: Escribir Sin Saltar "MONTO A RETIRAR : "; Leer monto; Si (monto > saldo) Entonces Escribir "¡Error no dispone de ese Saldo!.."; Esperar Tecla; SiNo saldo ← saldo - monto; FinSi 4: Escribir "Gracias por su visita, regrese pronto..."; salir ← 4; contador ← 4; FinSegun FinMientras </pre>	

Acción	Resultado esperado	Preguntas de apoyo
Finalizar las operaciones del cajero automático.	<pre>SiNo contador ← contador + 1; FinSi SiNo Escribir "*****"; Escribir "*"; Escribir "* TARJETA BLOQUEADA *"; Escribir "*"; Escribir "*****"; contador ← 4; FinSi FinMientras</pre>	

PRÁCTICA GUIADA Recordando la actividad de RearmaCom, ahora las y los estudiantes asumirán la perspectiva del programador. La forma de hacerlo no es la mejor, pero se debería entender que debido a que no se trabaja con una base de datos y mucho menos con dimensiones, hay limitaciones a la hora de crear esto en un pseudocódigo.

Acción	Resultado esperado
Realizar el menú.	<pre>Algoritmo PCmasters Repetir Limpiar Pantalla; Escribir "===== "; Escribir " BIENVENIDOS AL PC-Masters "; Escribir "===== "; Escribir ""; Escribir "MENU DE OPCIONES"; Escribir "1. Identificar usuario"; Escribir "2. Pedir armado de pc"; Escribir "3. Firmar contrato"; Escribir "4. SALIR"; Escribir ""; Escribir Sin Saltar "OPCIÓN : "; Leer aux; Escribir "";</pre>

Acción	Resultado esperado
Confeccionar las opciones dentro de un “según”.	<pre>Segun aux 1: Escribir Sin Saltar "Ingrese su nombre: " Leer Nombre;</pre>
Seguir con la siguiente opción y confeccionar un guardado de información más explícito.	<pre>2: Escribir "SELECCIONE PROCESADOR"; Escribir "1. Intel core i5 11600K \$259.990"; Escribir "2. Intel core i9 10900k \$519.990"; Escribir "3. AMD Ryzen 5 5600x \$279.990"; Escribir "4. AMD Ryzen 9 5950x \$799.990"; Leer aux; Segun aux 1: Procesador = "Intel core i5 11600K"; ProcesadorV = 259990; 2: Procesador = "Intel core i9 10900k"; ProcesadorV = 519990; 3: Procesador = "AMD Ryzen 5 5600x"; ProcesadorV = 279990; 4: Procesador = "AMD Ryzen 9 5950x"; ProcesadorV = 799990; FinSegun PrecioTotal = PrecioTotal + ProcesadorV;</pre>

Acción	Resultado esperado
Dependiendo de la opción que elija, hacer opciones específicas para cada usuario (Intel o AMD). Hay que resaltar que también deberá existir una opción para la tarjeta gráfica, la memoria RAM, el disco duro, el gabinete y la fuente de poder.	<pre> Si aux = 1 o aux = 2 Entonces Escribir "SELECCIONE PLACA MADRE"; Escribir "1. Intel - ASUS ROG STRIX B365-F GAMING \$159.990"; Escribir "2. Intel - Gigabyte Z390 AORUS ELITE \$179.990"; Leer aux; Segun aux 1: PlacaMadre = "Intel - ASUS ROG STRIX B365-F GAMING"; PlacaMadreV = 159990; 2: PlacaMadre = "Intel - Gigabyte Z390 AORUS ELITE"; PlacaMadreV = 179990; FinSegun SiNo Escribir "SELECCIONE PLACA MADRE"; Escribir "1. AMD - Gigabyte B450 I AORUS PRO WIFI \$139.990"; Escribir "2. AMD - ASUS ROG Crosshair VIII Dark Hero WI-FI \$459.990"; Leer aux; Segun aux 1: PlacaMadre = "AMD - Gigabyte B450 I AORUS PRO WIFI"; PlacaMadreV = 139990; 2: PlacaMadre = "AMD - ASUS ROG Crosshair VIII Dark Hero WI-FI"; PlacaMadreV = 459990; FinSegun FinSi PrecioTotal = PrecioTotal + PlacaMadreV; </pre>
Finalizar con la última opción y dar un resumen de lo seleccionado anteriormente. Preguntar cuántos computadores serán y, luego, confirmar la orden.	<pre> 3: Escribir "PIEZAS DE COMPUTADOR"; Escribir "Nombre del comprador: ", Nombre; Escribir "Procesador: ", Procesador, " - \$", ProcesadorV; Escribir "Placa Madre: ", PlacaMadre, " - \$", PlacaMadreV; Escribir "Tarjeta Video: ", TarjetaGrafica, " - \$", TarjetaGraficaV; Escribir "Memoria RAM: ", MemoriaRam, " - \$", MemoriaRamV; Escribir "Disco Duro: ", DiscoDuro, " - \$", DiscoDuroV; Escribir "Gabinete: ", Gabinete, " - \$", GabineteV; Escribir "Fuente de Poder: ", FuentePoder, " - \$", FuentePoderV; Escribir "Precio TOTAL: \$", PrecioTotal; Repetir Escribir "Cuántos computadores serán?" Leer cant; PrecioTotalTotal = PrecioTotal * cant; Escribir "Precio total por la cantidad de ",cant," computadores: \$",PrecioTotalTotal; Escribir "Desea confirmar la orden?" Escribir "1. Confirmar"; Escribir "2. Desconfirmar"; Escribir ""; Escribir Sin Saltar "OPCIÓN : "; Leer aux; Hasta Que aux = 1 FinSegun Hasta Que aux = 4 FinAlgoritmo </pre>

PRÁCTICA
INDEPENDIENTE

Se sugiere que el o la docente proponga un problema que se resuelva individualmente. Los problemas deben ser hechos específicamente para utilizar algún ciclo o repetidor. Por ejemplo: en un nuevo videojuego donde se enfrenta un jugador contra un robot, se necesita determinar quién gana el juego. Cuando el juego comienza, quien juega se enfrenta al robot y ambos tendrán 50 energía de daño. La persona que juega y el robot comienzan con 500 de energía. Ambos se enfrentan y se determina cuál de los dos golpea primero mediante un número al azar; quien golpea primero gana energía.

Para construir el modelo de resolución de este problema, se recomienda utilizar el programa disponible en el contexto, idealmente alguno que permita explorar la programación con bloques, como Scratch, OctoStudio u otro equivalente. Si no se dispone de acceso a computadoras, una alternativa desconectada viable es la creación de **un tablero de decisiones**, que consiste en una superficie o plantilla donde se organizan tarjetas con instrucciones o acciones en un orden lógico, se representa visualmente la secuencia, la repetición o la toma de decisiones que permitiría resolver el problema como si se estuviera programando.

EVALUACIÓN
FORMATIVA

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Seudocódigo	Elabora un pseudocódigo.	Elabora un pseudocódigo que corresponde medianamente a lo solicitado.	Elabora un pseudocódigo que corresponde a lo solicitado.
Condicionales	Elabora las estructuras condicionales con varios errores.	Elabora las estructuras condicionales con algunos errores.	Elabora las estructuras condicionales sin ningún error.
Repetidores o ciclos	Elabora mal las estructuras cíclicas.	Elabora de manera moderada las estructuras cíclicas.	Elabora las estructuras cíclicas de manera correcta y sin errores.
Código	Elabora un código ordenado.	Elabora un código ordenado y medianamente funcional (falta corregir detalles).	Elabora un código ordenado y funcional (funciona en cada momento de su ejecución).

Revisión por pares con enfoque en la rúbrica:

Los y las estudiantes trabajan en parejas para revisar el pseudocódigo y el código de sus compañeros, para lo cual utilizan la rúbrica como guía. Cada estudiante proporciona retroalimentación específica sobre el pseudocódigo, las estructuras condicionales, los ciclos y el código en general.

Autoevaluación con preguntas guía:

Seudocódigo: ¿Logré expresar claramente la lógica de mi programa en el pseudocódigo? ¿Qué aspectos podría mejorar?

Condicionales: ¿Utilicé las estructuras condicionales de manera correcta? ¿Qué errores cometí y cómo los puedo corregir?

Ciclos: ¿Implementé los ciclos de forma eficiente? ¿Podría haber utilizado una estructura diferente?

Código: ¿Mi código es ordenado y fácil de entender? ¿Funciona correctamente en todas las situaciones?

DECRETO EN TRÁMITE

Orientaciones al docente

Para unificar conceptos disciplinares

Entendemos que el algoritmo es algo más que instrucciones que siguen un orden. Con las nuevas herramientas, tendrá más de una dirección a la hora de trabajarlo. Por esto, la función “mientras” nos permite llevar a cabo algoritmos más complejos y cercanos a la realidad. Además, permite complementar la categoría de los ciclos y funciona de la siguiente manera:

- **Mientras:** este ciclo nos permite repetir durante el tiempo en que se cumple una acción en concreto. Se podría decir que este ciclo en particular es la contraparte del ya mencionado “repetir”.

Con la estructura condicional los algoritmos seguían siendo lineales, pero con la estructura de ciclos se abren nuevas brechas que permiten repetir, en más de una ocasión, ciertas actividades que son cruciales al resolver los problemas que aparecen en etapas posteriores del mismo proceso.

Se sugiere una tabla para organizar conocimientos nuevos y anteriores, y comparar una misma situación con diferentes funciones de programación: una donde funciona “mientras” y otra con “para”, por ejemplo:

Función: Mientras

Elaborar un programa que entregue, después de haber ingresado todas las notas, el promedio de las notas

Programa promedio de notas

Variable: contador de notas i (número entero)

Variable: suma de notas s (número racional)

Variable: promedio de notas p (número racional)

$i \leftarrow 0$

$s \leftarrow 0$

Mientras $i \leq 10$

Entrar número s

Leer número s

Suma $s = s + s$

$i \leftarrow i + 1$

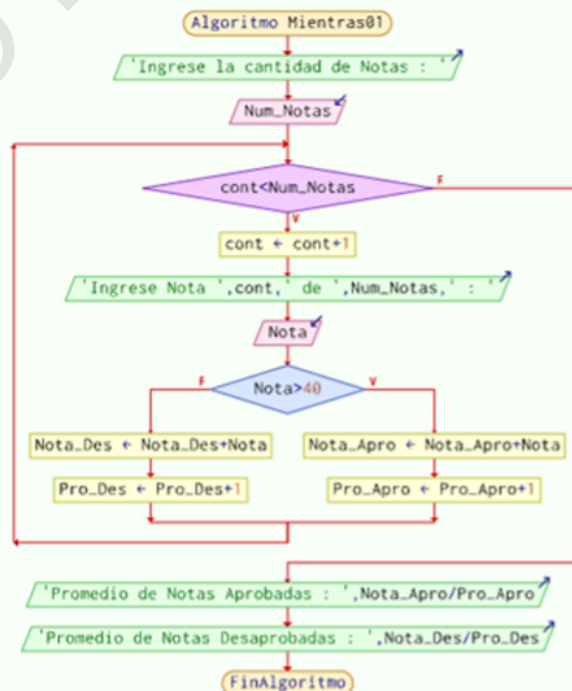
Fin Mientras

Escribir: "Ya ingresaron 10 notas"

Promedio: $p = \frac{s}{i}$

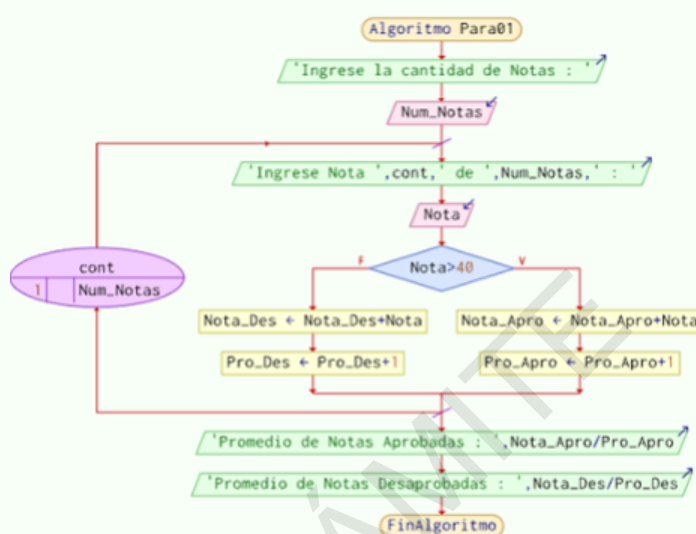
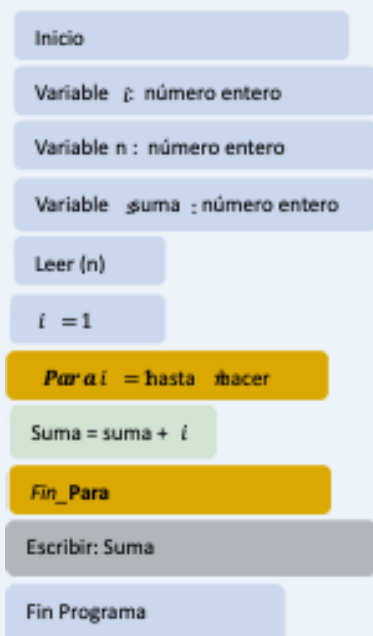
Escribir: "El promedio es p "

Fin Programa



Función: **Para**

Calcular la suma de los números naturales desde 1 hasta n



Las delimitaciones de “para” y “hasta” se pueden hacer en relación con otras variables, por ejemplo, con “k” y “j”. Para ello se escribe la instrucción “Para i = k hasta i = j” que permite programar estas delimitaciones.

Actitudes

Para apoyar el desarrollo de las actividades, se sugiere utilizar computadores, ya que estos problemas son más complejos y más largos, por lo que es necesario tener una retroalimentación en el momento. Esto entregará más confianza en lo que se hace al programar. Por otra parte, en cuanto al desarrollo de las actividades individuales, estas son necesarias para que el o la estudiante confíe en sus conocimientos y en su forma de pensar.

**Orientaciones para
organizar e implementar
la actividad**

Se sugieren las siguientes indicaciones para guiar la implementación del trabajo individual del videojuego:

- Determinar un tiempo para así resolver cualquier duda del problema.
- Determinar el tiempo que se cree necesario para terminar o, al menos, llevar algo del desarrollo del ejercicio.
- Determinar el tiempo para retroalimentar y guía de acuerdo con las necesidades que se presenten.
- Favorecer que alguien presente su resultado. Puede ser por voluntad propia o elegido al azar. Esto permitirá retroalimentar o reforzar si está todo correcto, aclarar dudas finales acerca del desarrollo de la actividad y poner un nuevo reto, por ejemplo, en vez de ser de solo un jugador, ahora el videojuego será entre dos.

Actividad de desempeño

3

Propósito

En esta actividad, se pone énfasis en el uso de los subprocesos y el cómo estos facilitan la organización del código. Esto sirve para que las y los estudiantes tengan clara la transferencia de datos/información entre proceso y proceso. También, se reforzará lo aprendido anteriormente en los contextos de nuevos conocimientos, ya que en todo momento se utilizan las metodologías que se aprenden progresivamente en los procesos.

Objetivos de aprendizaje

OA 2

Organizar datos de una situación problemática estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función. **(Evaluar)**

OA 3

Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponibles. **(Modelar)**

Conocimiento esencial

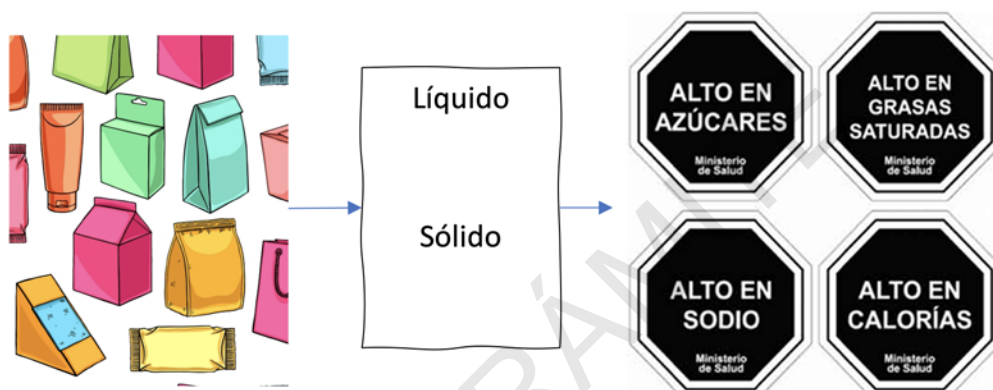
- Modelos entidad-relación y simulación de sistemas.
- Sistemas computacionales: dispositivos, *hardware* y *software* y resolución de problemas.

Tiempo estimado: 3 horas pedagógicas

Desarrollo de la actividad

SITUACIÓN EXPERIENCIAL

El o la docente presenta una situación de clasificación de productos, donde se requiere hacer primero una clasificación del estado de la materia (líquido o sólido) y, a partir de esta clasificación, se dan condiciones para asignar el logo correspondiente de alto en azúcares, alto en grasas saturadas, alto en sodio o alto en calorías.



Para motivar la descomposición de la programación en partes o en subprocesos se sugieren las siguientes preguntas:

- ¿Cómo describirías el proceso de asignación de logos a un producto alimenticio?
- ¿Qué partes tiene el proceso?
- ¿Cómo podría ser la programación completa?

Pueden registrar el procedimiento de instrucciones en una hoja de trabajo de códigos. En el anexo podrán encontrar la hoja para escribir en modo de zona de escritura de código.

CONSTRUCCIÓN DE CONOCIMIENTO

Para construir el conocimiento, se sugiere comenzar con la metodología del subproceso y el problema de etiquetar los alimentos con las etiquetas definidas por la ley. El alimento puede ser sólido o líquido y, según el tipo, estos tienen un valor límite para las 4 etiquetas que existen: alto en azúcares, alto en grasas saturadas, alto en sodio y alto en calorías.

Tabla N1 Límites de alimentos sólidos		Tabla N1 Límites de alimentos líquidos	
Nutriente o energía	Datos	Nutriente o energía	Datos
Azúcares totales g/100g	20	Azúcares totales g/100g	2
Grasas saturadas g/100g	5	Grasas saturadas g/100g	4
Sodio mg/100g	700	Sodio mg/100g	80
Energía kcal/100g	300	Energía kcal/100g	80



La actividad se desarrolla con los subprocesos presentes.

Acción	Resultado esperado	Preguntas de apoyo
Elaborar primeros pasos del código.	<pre>SubProceso SolicitarDatos(menu) FinSubProceso Algoritmo Etiquetas Definir opc Como Entero; Repetir Escribir "Ingrese tipo de alimento"; Escribir "1) Solido"; Escribir "2) Liquido"; Escribir "3) Salir"; Leer opc; Segun opc 1: SolicitarDatos(1); 2: SolicitarDatos(2); 3: Escribir "Vuelva pronto!"; De Otro Modo: Escribir "Ingrese opcion de menu correcta"; FinSegun Hasta Que opc = 3 FinAlgoritmo</pre>	¿Qué se espera del subproceso?
Elaborar dentro del subproceso el ingreso de los datos que se compararán.	<pre>SubProceso SolicitarDatos(menu) Definir calorías,sodio,grasas,azucar Como Entero Escribir "Ingrese Azucares totales g/100g"; Leer azucar; Escribir "Ingrese Grasas saturadas g/100g"; Leer grasas; Escribir "Ingrese Sodio mg/100g"; Leer sodio; Escribir "Ingrese Energia kcal/100g"; Leer calorías;</pre>	

Acción	Resultado esperado	Preguntas de apoyo
Según la opción elegida, comparar los datos ingresados y finalizar el código.	<pre> Segun menu 1: Si azucar > 20 Entonces Escribir "Alto en azucares totales"; FinSi Si grasas > 5 Entonces Escribir "Alto en grasas saturadas"; FinSi Si sodio > 700 Entonces Escribir "Alto en sodio"; FinSi Si calorías > 300 Entonces Escribir "Alto en calorías"; FinSi 2: Si azucar > 2 Entonces Escribir "Alto en azucares totales"; FinSi Si grasas > 4 Entonces Escribir "Alto en grasas saturadas"; FinSi Si sodio > 80 Entonces Escribir "Alto en sodio"; FinSi Si calorías > 80 Entonces Escribir "Alto en calorías"; FinSi FinSegun FinSubProceso </pre>	

PRÁCTICA GUIADA Se sugiere proponer a los y las estudiantes la siguiente situación: una clínica requiere para el área de nutrición un programa que permita realizar un cálculo del índice de masa corporal. Para esto se requiere la información de peso en kilogramos (kg) y de altura en metros (m).

La fórmula para sacar el cálculo es la siguiente:

$$IMC = \frac{\text{Peso (kg)}}{\text{Altura}^2 \text{ (m)}}$$

Diagnóstico	IMC
PESO INSUFICIENTE	<18.5
PESO NORMAL	18.5 – 24.9
SOBREPESO	25 – 29.9
OBESIDAD GRADO 1	30 – 34.9
OBESIDAD GRADO 2	35 – 39.9
OBESIDAD GRADO 3	≥ 40

Se requiere obtener la siguiente información:

- Categorización por paciente.
- La cantidad de pacientes por categoría.
- La categoría que tiene mayor cantidad de pacientes.

Para cada requerimiento se requiere su propio procedimiento.

Acción	Resultado esperado
Crear un algoritmo listo para incorporar los tres subprocesos que tendrá la resolución del problema. No es necesario crearlos todos de una vez; se puede trabajar primero desde el proceso mismo y después pasarlo a un subproceso.	Proceso ProcesoIMC <pre> Definir cantidadClientes, i, ipe,pno,sg,ob1,ob2,ob3 Como Entero; Definir peso, talla Como Real; Escribir "Bienvenido a IMC"; Escribir "Indique la cantidad de clientes"; Leer cantidadClientes; Para i = 1 Hasta cantidadClientes Hacer Escribir "-----"; Escribir "Ingrese peso del cliente N°",i; Leer peso; Escribir "Ingrese altura en metros del cliente N°",i; Leer talla; Escribir "El cliente ingresado esta en la categoria:"; Escribir "-----"; categoriaIMC(peso,talla,ipe,pno,sg,ob1,ob2,ob3); //Llamar otro subproceso para que cuente la categoria. FinPara Escribir "-----"; Escribir "Cantidad de clientes por categoria:"; Escribir "-----"; cantCliente(ipe,pno,sg,ob1,ob2,ob3); Escribir "-----"; Escribir "La categoria con mas clientes es:"; Escribir "-----"; cantIMC(ipe,pno,sg,ob1,ob2,ob3); Escribir "-----"; FinProceso </pre>

Acción	Resultado esperado
Implementar el primer subproceso donde se hará el cálculo y la respectiva cuenta de los pacientes para cada categoría.	<pre> SubProceso categoriaIMC(peso,talla,ipe Por Referencia Definir imc Como Real; imc = peso / talla ↑ 2; Si imc < 18.5 Entonces Escribir "Peso Insuficiente"; ipe = ipe + 1; FinSi Si imc ≥ 18.5 y imc ≤ 24.9 Entonces Escribir "Peso Normal"; pno = pno + 1; FinSi Si imc ≥ 25 y imc ≤ 29.9 Entonces Escribir "SobrePeso"; sg = sg + 1; FinSi Si imc ≥ 30 y imc ≤ 34.9 Entonces Escribir "Obesidad Grado I"; ob1 = ob1 + 1; FinSi Si imc ≥ 35 y imc ≤ 39.9 Entonces Escribir "Obesidad Grado II"; ob2 = ob2 + 1; FinSi Si imc ≥ 40 Entonces Escribir "Obesidad Grado III"; ob3 = ob3 + 1; FinSi FinSubProceso </pre>

Acción	Resultado esperado
El segundo subproceso determinará el total de pacientes en cada categoría y solo se mostrarán si son mayores a cero.	<pre> SubProceso cantIMC(ipe,pno,sg,ob1,ob2,ob3) Si ipe ≥ pno y ipe ≥ sg y ipe ≥ ob1 y ipe ≥ ob2 y ipe ≥ ob3 Entonces Escribir "Mas Clientes en Insuficiente Peso"; FinSi Si pno ≥ ipe y pno ≥ sg y pno ≥ ob1 y pno ≥ ob2 y pno ≥ ob3 Entonces Escribir "Mas Clientes en Peso Normal"; FinSi Si sg ≥ ipe y sg ≥ pno y sg ≥ ob1 y sg ≥ ob2 y sg ≥ ob3 Entonces Escribir "Mas Clientes en SobrePeso Grado I"; FinSi Si ob1 ≥ ipe y ob1 ≥ pno y ob1 ≥ sg y ob1 ≥ ob2 y ob1 ≥ ob3 Entonces Escribir "Mas Clientes en Obesidad Grado I"; FinSi Si ob2 ≥ ipe y ob2 ≥ pno y ob2 ≥ sg y ob2 ≥ ob1 y ob2 ≥ ob3 Entonces Escribir "Mas Clientes en Obesidad Grado II"; FinSi Si ob3 ≥ ipe y ob3 ≥ pno y ob3 ≥ sg y ob3 ≥ ob1 y ob3 ≥ ob2 Entonces Escribir "Mas Clientes en Obesidad Grado III"; FinSi FinSubProceso </pre>

PRÁCTICA INDEPENDIENTE

Se sugiere un trabajo en parejas o individual para escribir el tercer subproceso y describir los resultados de la situación de peso en kilogramos (kg) y de altura en metros (m) de los pacientes de la clínica, mediante el uso de la fórmula para sacar el índice de masa corporal:

$$IMC = \frac{\text{Peso (kg)}}{\text{Altura}^2 \text{ (m)}}$$

Y así poder determinar la categoría del paciente: PESO INSUFICIENTE, PESO NORMAL, SOBREPESO, OBESIDAD GRADO 1, OBESIDAD GRADO 2 y OBESIDAD GRADO 3.

Para lograrlo, se sugiere hacer una lectura comprensiva de lo realizado en la práctica guiada y rehacer si fuera necesario los subprocesos para comprender lo que falta del algoritmo.

Acción	Resultado esperado
Por último, el tercer subproceso, el cual determinará en que categoría hay más clientes.	<pre> SubProceso cantIMC(ipe,pno,sg,ob1,ob2,ob3) Si ipe ≥ pno y ipe ≥ sg y ipe ≥ ob1 y ipe ≥ ob2 y ipe ≥ ob3 Entonces Escribir " Mas Clientes en Insuficiente Peso"; FinSi Si pno ≥ ipe y pno ≥ sg y pno ≥ ob1 y pno ≥ ob2 y pno ≥ ob3 Entonces Escribir "Mas Clientes en Peso Normal"; FinSi Si sg ≥ ipe y sg ≥ pno y sg ≥ ob1 y sg ≥ ob2 y sg ≥ ob3 Entonces Escribir "Mas Clientes en SobrePeso Grado I"; FinSi Si ob1 ≥ ipe y ob1 ≥ pno y ob1 ≥ sg y ob1 ≥ ob2 y ob1 ≥ ob3 Entonces Escribir "Mas Clientes en Obesidad Grado I", FinSi Si ob2 ≥ ipe y ob2 ≥ pno y ob2 ≥ sg y ob2 ≥ ob1 y ob2 ≥ ob3 Entonces Escribir "Mas Clientes en Obesidad Grado II"; FinSi Si ob3 ≥ ipe y ob3 ≥ pno y ob3 ≥ sg y ob3 ≥ ob1 y ob3 ≥ ob2 Entonces Escribir "Mas Clientes en Obesidad Grado III"; FinSi FinSubProceso </pre>
Resultado	<pre> ----- Cantidad de clientes por categoria: ----- Cantidad de Clientes con Insuficiente Peso: 1 Cantidad de Clientes con Obesidad Grado I: 2 ----- La categoria con mas clientes es: ----- Mas Clientes en Obesidad Grado I ----- *** Ejecución Finalizada. *** </pre>

EVALUACIÓN FORMATIVA

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Códigos	Elabora un código.	Elabora un código que cumple con lo establecido.	Elabora un código que cumple con lo establecido y también lo analiza y mejora en el proceso.
Problemática	Analiza el problema.	Analiza el problema y da una opinión al respecto.	Analiza el problema y da un argumento de cómo mejorarlo y optimizarlo.
Condicionales	Elabora las estructuras condicionales con errores.	Elabora las estructuras condicionales con algunos errores.	Elabora las estructuras condicionales sin errores.
Repetidores o ciclos	Elabora las estructuras cíclicas con errores.	Elabora las estructuras cíclicas con algunos errores.	Aplica las estructuras cíclicas sin errores.
Subprocesos	Elabora un subproceso.	Elabora de manera moderada un subproceso	Elabora y aplica sin errores el uso de subproceso.

Orientaciones al docente

Para unificar conceptos disciplinares

Con la nueva metodología, es posible estar de acuerdo que su función principal es dejar todo más organizado. Las personas al programar dividen las tareas en capas que van llamándolas cuando las necesite. Los subprocesos son eso, pero de una manera más simplificada, ya que este lenguaje no maneja el sistema donde se ven los programas de programación en otros ámbitos más allá de la educación. Debemos entender que los subprocesos son una complejidad agregada a lo que ya se ha visto anteriormente y que se puede implementar en todos los ejemplos en los que tenga sentido utilizar esta función.

- Un subproceso o función no es más que un bloque de código con un determinado fin o propósito. Podemos definir más de uno y nos permiten **ahorrar líneas de código** si nuestros programas son muy extensos o hay codificación redundante.

Actitudes

Es recomendable el uso de computadores a la hora de codificar nuestro código. Más allá del papel, el computador se utiliza para verificar o evaluar de una manera más eficiente el código y también poder ejecutar y ver el funcionamiento de nuestro algoritmo. Para el desarrollo de los trabajos individuales, es necesario tener claro que esto es para que el o la estudiante tenga confianza de sus conocimientos y su forma de pensar. Por último, siempre es bueno ser críticos y honestos al momento de crear nuestros algoritmos, y siempre resolver de forma creativa los problemas que vayan apareciendo.

Orientaciones para organizar e implementar la actividad:

Se sugieren las siguientes indicaciones para guiar la implementación del trabajo individual:

- Determinar un tiempo para así resolver cualquier duda del problema.
- Determinar el tiempo que cree necesario para haber terminado o al menos llevar algo de desarrollo de la actividad.
- Determinar el tiempo para retroalimentar a sus estudiantes y guiar a quiénes presenten mayor dificultad.
- Favorecer que alguien presente su resultado. Puede ser por voluntad propia o elegido al azar para poder dar una retroalimentación con refuerzo positivo.

Actividad de desempeño

4

Propósito

En esta actividad se busca que los y las estudiantes puedan evaluar y modelar algoritmos complejos mediante la metodología de los arreglos. Su función primordial es guardar información y lo que más se asemeja a esto es el procesamiento de una hoja de cálculo tradicional.

Objetivos de aprendizaje

OA 2

Organizar datos de una situación problemática, estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función. **(Evaluar)**

OA 3

Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponibles. **(Modelar)**

Conocimiento esencial

- Modelos entidad-relación y simulación de sistemas.
 - Sistemas computacionales: dispositivos, *hardware* y *software*, y resolución de problemas.
-

Tiempo estimado: 3 horas pedagógicas

Desarrollo de la actividad

SITUACIÓN EXPERIENCIAL

Se sugiere presentar los arreglos como un nuevo tipo de dato para programar la organización de materiales en un taller comunitario, diferenciando entre el ingreso de un listado y la lectura de los datos desde una tabla estructurada.

Material	Cantidad	Disponible	Ubicación 1	Ubicación 2	Ubicación 3
Madera	12	Sí	Caja 1	Estante 3	-
Clavos	200	No	Caja 2	-	-

Este arreglo tiene 3 filas, solo dos son materiales disponibles.

Este arreglo tiene 6 columnas, algunas son texto y otras números.

Esta tabla es un nuevo tipo de dato. Se llama “arreglo” y se puede usar para programar, por ejemplo, la organización de materiales y la búsqueda de ubicaciones.

A partir de esta tabla, se sugiere realizar algunas preguntas que permitan reconocer el uso y lectura de una tabla:

- ¿Cómo se puede saber dónde están los materiales?
- ¿Cómo identificar los que están disponibles?
- ¿Qué decisiones se pueden tomar con esta tabla?

CONSTRUCCIÓN DE CONOCIMIENTO

Para construir el conocimiento sobre la lectura de arreglos, se sugiere introducir los conceptos de filas y columnas a partir de la situación anterior.

En un pseudocódigo se anota primero el lugar de la fila y luego la posición de la columna **m[filas][columna]**

La numeración de las filas y de las columnas empieza en 0.

$m[0][0] = \text{"Madera"}$

$m[1][2] = \text{"No"}$

$m[0][1] + m[1][1] = 12 + 200 = 212$

La lectura de materiales y su disponibilidad puede hacerse al menos de dos maneras. Una de ellas es acceder a cada fila completa y otra es revisar solo una columna (por ejemplo, "disponible").

En estos casos, es necesario incluir el largo del arreglo para recorrerlo, por ejemplo, al sumar todas las cantidades:

$d[1] = 200$

$d[0] = 12$

$d[0] + d[1] = 212$

$\text{largo}[d] = 2$

PRÁCTICA GUIADA

Elaborar un programa en pseudocódigo que incluya las funciones “posiciones” y “largo”, y que permita revisar cuáles materiales están disponibles y cuántos hay. Se sugiere estructurar el procedimiento en 4 partes. En la primera, se hace la definición del tipo de datos y las variables que serán consideradas. La segunda parte, es el ingreso de datos utilizando la función posiciones. La tercera parte de la elaboración, indica cómo se recorren los datos según la posición. La última parte es la obtención del resultado y se sugiere evaluar al compartir el programa con otros y al ingresar otros valores para ver el resultado.

Acción	Resultado esperado	Preguntas de apoyo
Elaborar las definiciones.	Nombre programa: materiales disponibles d[posiciones]: textos/números i: enteros disponibles: texto largo[d]: enteros	¿Cómo podría empezar?
Elaborar el ingreso de datos.	d = {"Madera", 12, "Sí", "Caja 1", "Estante 3", "-", "Clavos", 200, "No", "Caja 2", "-", "-"} Escribir d[i]	¿Qué puedo hacer para guardar los arreglos?
Iniciar el programa	Para i = 0 hasta i < largo[d] Si d[i][2] == "Sí" entonces Escribir d[i]	¿Cómo puedo llamar a cada uno de los arreglos y ordenarlos?
Resultado esperado.	Material disponible: Madera, cantidad: 12, ubicación: Caja 1	¿Qué resultado obtuve y cómo lo puedo usar?

PRÁCTICA INDEPENDIENTE

Se sugiere un trabajo de pares, donde resuelvan un ejercicio con arreglos. Para esto, se sugiere entregar una tabla para que cada estudiante programe con un arreglo. En este caso, entregar información a la tabla de la base de datos.

CÓDIGO	NOMBRE	DIRECCIÓN	SEXO	SUELDO
COD01	ANA	CONCHALÍ	F	350000
COD02	LUIS	QUILICURA	M	375000
COD03	MARCELA	VITACURA	F	325000

El algoritmo debe ser capaz de agregar estos 3 datos y mostrarlos de la misma manera. Después de haber ingresado el primer dato, saldrá una ventana que preguntará si desea ingresar otro registro. Al confirmarlo, se repetirá la secuencia y se volverá a ingresar. Al rechazar la opción se mostrarán los registros que logró ingresar y finalizará el programa.

Es necesario considerar la lectura de la posición del dato según la fila y la columna para la programación, utilizando un corchete cuadrado y anotando como par ordenado, donde la fila va en primer lugar y se comienza desde el cero. Se sugiere retomar lo trabajado en la construcción del conocimiento.

Acción	Resultado esperado	Preguntas de apoyo
Elaborar las definiciones de manera correcta.	Algoritmo agregar_registros Definir continuar, datos Como Caracter; Definir f, c, fila como numero; Dimension datos[10,5]; continua ← "S"; f ← 0; c ← 0; fila ← 1;	¿Cómo podría empezar?
Elaborar el ingreso de datos.	<pre>//Registrar Datos Mientras MAYUSCULAS(continua) == "S" Hacer Escribir Sin Saltar "CODIGO "; Leer datos(fila,1); Escribir Sin Saltar "NOMBRE "; Leer datos(fila,2); Escribir Sin Saltar "DIRECCION "; Leer datos(fila,3); Escribir Sin Saltar "SEXO "; Leer datos(fila,4); Escribir Sin Saltar "SUELDO "; Leer datos(fila,5); fila ← fila + 1; Escribir Sin Saltar "Desea Ingresar Otro Registro (S/N)? "; Leer continuar; Escribir " "; FinMientras</pre>	¿Qué puedo hacer para que se guarde en los arreglos?
Listar los datos ingresados de manera ordenada.	<pre>//Listar Datos Escribir "REGISTROS: "; Para f ← 1 hasta fila-1 con paso 1 hacer Para c ← 1 hasta 5 con paso 1 hacer Escribir Sin Saltar datos[f,c], " "; FinPara Escribir " "; FinPara FinAlgoritmo</pre>	¿Cómo puedo llamar a cada arreglo y ordenarlos?

Acción	Resultado esperado	Preguntas de apoyo
Resultado esperado.	<pre> *** Ejecución Iniciada. *** CODIGO :> COD01 NOMBRE :> ANA DIRECCION :> CONCHALI SEXO :> F SUELDO :> 350000 Desea Ingresar Otro Registro (S/N)? > S CODIGO :> COD02 NOMBRE :> LUIS DIRECCION :> QUILICURA SEXO :> M SUELDO :> 375000 Desea Ingresar Otro Registro (S/N)? > S CODIGO :> COD03 NOMBRE :> MARCELA DIRECCION :> VITACURA SEXO :> F SUELDO :> 325000 Desea Ingresar Otro Registro (S/N)? > N REGISTROS: COD01 ANA CONCHALI F 350000 COD02 LUIS QUILICURA M 375000 COD03 MARCELA VITACURA F 325000 *** Ejecución Finalizada. *** </pre>	

EVALUACIÓN FORMATIVA

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Códigos	Elabora un código.	Elabora un código que cumple con lo establecido.	Elabora un código que cumple con lo establecido y lo analiza y mejora en el proceso.
Problemática	Analiza el problema.	Analiza el problema y da una opinión al respecto.	Analiza el problema y da un argumento de cómo mejorarlo y optimizarlo.
Arreglos	Elabora las estructuras de arreglos con varios errores.	Elabora las estructuras de arreglos con algunos errores.	Elabora las estructuras de arreglos sin errores.

Orientaciones al docente

Para unificar conceptos disciplinares

Los arreglos son, en general, la parte más difícil de entender, pero si el conocimiento previo ha sido comprendido por sus estudiantes, tendrán una buena base, pues, como es usual, siempre hay que practicarlo mediante ejercicios, ya que es muy importante a la hora de analizar algún problema que pueda tener una de estas formas de solucionarse. Hay que recordar que los arreglos son cajas que guardan información y cuando uno quiera ver esa información guardada, simplemente debe llamarla mediante el comando preciso para saber dicha información. En cuanto a las actividades, es necesario siempre estar pendiente de las dudas que puedan surgir y utilizar las herramientas aprendidas en los módulos anteriores.

- Un arreglo se define como una colección finita, homogénea y ordenada de elementos.
- Finita: porque todo arreglo tiene un límite, es decir, se debe determinar cuál será el número máximo de elementos que podrán formar parte del arreglo.
- Homogénea: porque todos los elementos del arreglo son del mismo tipo, por ejemplo: arreglo de enteros, de booleanos, pero nunca una combinación de distintos tipos.
- Ordenada: se puede determinar cuál es el primer elemento, el segundo elemento, el enésimo elemento.

El uso de la palabra sexo en este contexto es utilizado para identificar solo dos atributos, se sugiere ampliar esta categoría agregando otros atributos y utilizando la palabra género.

Actitudes

Es recomendable apoyar el desarrollo de estas actividades con el uso de un computador para que sus estudiantes puedan ver sus algoritmos en tiempo real y ver el funcionamiento de estos. En cuanto al desarrollo de la actividad individual, es necesario siempre responder las consultas y guiar, pero sobre todo favorecer el trabajo en equipo y que se ayuden mutuamente para resolver los problemas que tengan por delante.

Orientaciones para organizar e implementar actividades

Se sugieren las siguientes indicaciones para guiar la implementación de las actividades:

- Elaborar actividades donde se utilicen los arreglos.
- Determinar el tiempo necesario para elaborar el análisis y el desarrollo del algoritmo.
- Concluir mostrando algunos de los algoritmos de los y las estudiantes para que sean revisados por el curso (esto con el fin de mostrar la forma en la que lo hicieron y ver si alguien lo hizo de otra manera).

Visión panorámica

Gran idea

El comportamiento de un sistema computacional (dispositivos, *hardware* y *software*) depende de la secuencia de instrucciones dadas por el usuario que automatizarán el desempeño de una tarea para ejecutar una acción o resolver un problema.

Objetivos de aprendizaje

OA 4

Establecer conexiones y semejanzas entre las partes del problema para caracterizar patrones en una situación problemática dada. ([Generalizar y Transferir](#))

OA 5

Aplicar flexiblemente las soluciones elaboradas e implementadas a otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista. ([Generalizar y Transferir](#))

OA 6

Elaborar e implementar soluciones a una situación problemática mediante la formulación de algoritmos, actuando de acuerdo con los principios de la ética. ([Pensar de forma algorítmica](#))

Conocimientos esenciales

- Redes e Internet: sitios web, aplicaciones móviles, seguridad cibernética y objetos controlados digitalmente.
 - Algoritmos y programación: Lenguajes, métodos y entornos de programación.
-

Tiempo estimado: 6 semanas (12 horas)

Propósito

En el módulo 3 de la asignatura Pensamiento Computacional de los Niveles 1 y 2 de Educación Media, se espera que los y las estudiantes puedan aplicar, en diferentes escenarios, los conocimientos asociados al comportamiento de los sistemas computacionales. Además de distinguir el rol que les cabe como usuarios al automatizar procesos lógicos para desempeñar o ejecutar tareas que resuelvan problemas de seguridad, acceso y manejo de datos en un ambiente digital. La adquisición de conocimientos asociados al manejo de sistemas de información cobra fuerza en este módulo, al ser los y las estudiantes quienes comienzan a transferir sus aprendizajes a estos ámbitos de las ciencias de la computación.

Sobre este escenario, el módulo pone énfasis en desarrollar las habilidades de generalización y transferencia a través del pensamiento algorítmico y la programación propiamente tal. Al tener conocimientos sobre tablas y modelado de datos, desde la descomposición, el modelado, la abstracción y la evaluación, las y los estudiantes podrán trabajar con información en distintos lenguajes y manejarlos para poder utilizar de forma eficiente distintos datos y fuentes relacionadas con problemas de

seguridad, acceso y control. Asimismo, a través de las actividades de desempeño del módulo, comprenderán que los sistemas computacionales pueden estar vinculados a redes e Internet, y distinguirán que el uso de sitios web, aplicaciones móviles y objetos controlados digitalmente siempre tiene a la informática y sus diferentes clasificaciones para determinar o evaluar su funcionamiento.

Los Objetivos de Aprendizaje del módulo desarrollan las actitudes del siglo XXI del ámbito de las Maneras de Trabajar y Herramientas para trabajar, promoviendo el aprovechamiento de las herramientas para resolver problemas o diseñar de forma colaborativa soluciones para el logro de metas comunes. Asimismo, al aprovechar las herramientas disponibles, debe enfocarse en que estas no son siempre limitantes para poder crear soluciones a problemas cotidianos, pues ofrecen distintas oportunidades de aprendizaje. Finalmente, se espera que las actividades de desempeño propuestas sean una oportunidad para comprender que las interacciones del estudiantado con las tecnologías tienen efectos en las personas y en la sociedad, pues estas son objeto de valoración y nunca neutras.

Ruta de Aprendizaje

¿Cómo distinguir la tipología de sistemas y redes para solucionar problemas de seguridad, acceso y control de datos?

	Actividad de desempeño 1	Resolver problemas computacionales relacionados con la funcionalidad de un objeto relevando la función de programación “hasta que”.
	Actividad de desempeño 2	Resolver problemas computacionales relacionados con la funcionalidad de un objeto relevando la función de programación “hasta que”.
	Actividad de desempeño 3	Utilizar las oportunidades que pueden ofrecer las bases de datos en ámbitos de desempeño profesional o cotidiano.
	Actividad de desempeño 4	Aplicar soluciones computacionales de forma flexible a diferentes problemas, transfiriendo los conocimientos asociados al uso de base de datos en estructuras informacionales más complejas.

Actividad de desempeño

1

Propósito

Esta actividad desafía a los y las estudiantes a volver a los orígenes de la formulación de algoritmos. Los aborda desde un enfoque más técnico, secuencial y abstracto, y favorece el desarrollo del pensamiento computacional al definir los pasos que se desencadenan para la consecución de una acción a partir de una instrucción. Además, integra y moviliza los aprendizajes adquiridos previamente, y busca que los y las estudiantes puedan resolver progresivamente situaciones complejas a nivel computacional, en cuanto al nivel de transferencia y generalización.

Objetivos de aprendizaje

OA 5

Aplicar flexiblemente las soluciones elaboradas e implementadas a otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista.
(Generalizar y transferir)

OA6

Elaborar e implementar soluciones a una situación problemática mediante la formulación de algoritmos, actuando de acuerdo con los principios de la ética.
(Pensar de forma algorítmica)

Conocimiento esencial

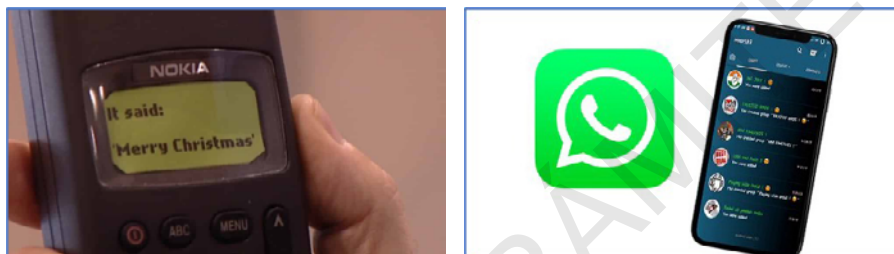
- Algoritmos y programación: lenguajes, métodos y entornos de programación.

Tiempo estimado: 3 horas pedagógicas

Desarrollo de la actividad

SITUACIÓN EXPERIENCIAL

Se sugiere que el o la docente explique la importancia de la tecnología y lo inherente que es al desarrollo del ser humano. Enfatiza en la velocidad con la cual ésta ha avanzado durante los últimos años. Es posible considerar como contexto que, por ejemplo, el primer mensaje de texto enviado en el mundo fue el 3 de diciembre de 1992 desde un computador a un teléfono móvil y que, en la actualidad, una gran cantidad de personas en el mundo utiliza WhatsApp, una aplicación de mensajería instantánea, como principal medio de comunicación.



Algunas de las preguntas que podrían promover procesos de pensamiento asociados a la argumentación y reflexión sobre avances tecnológicos asociados al uso de Internet y redes a través del tiempo, podrían ser:

- ¿Qué piensan sobre los avances de Internet durante los últimos 10 años?
- ¿Cuáles son las evidencias que tienen para indicar que la tecnología ha avanzado?
- ¿Cómo serían las comunicaciones en el presente si no se hubiese inventado internet?
- ¿De qué manera imaginas el avance de la tecnología durante los próximos 10 años?

CONSTRUCCIÓN DE CONOCIMIENTO

Para seguir profundizando en el desarrollo del pensamiento algorítmico, se sugiere recuperar los fundamentos básicos aprendidos y cómo estos ayudan a resolver cualquier problema que se presente a un usuario o creador de tecnología. Para esta actividad, el o la docente puede dar la instrucción de crear un algoritmo en base a necesidades cotidianas o la funcionalidad de un objeto, por ejemplo, lavarse los dientes, ir a trabajar por la mañana, funcionamiento de un reloj, entre otras.

Para el caso de un reloj, se creará un cronómetro que funcione con “hora: minuto: segundo”. La idea principal es que se ingrese qué hora, minuto y segundo deba contar el cronómetro hasta llegar a 0. Después de eso, se le preguntará al usuario si desea repetir la operación.

Después de haber hecho el pseudocódigo, será necesario analizarlo y evaluarlo de acuerdo con su funcionamiento.

Acción	Resultado esperado	Preguntas de apoyo
Elaborar el inicio del pseudocódigo y los datos de cuánto durará el cronómetro.	Algoritmo Cronometro Definir H, M, S, CH, CM, Cs como numero; Definir continuar como caracter; continuar ← "s"; Mientras continuar ="s" Hacer Repetir Escribir "Ingrese la hora : "; Leer H; Hasta Que H ≤ 24; Repetir Escribir "Ingrese el minuto : "; Leer M; Hasta Que M ≤ 60; Repetir Escribir "Ingrese el segundo : "; Leer S; Hasta Que S ≤ 60;	¿Qué algoritmo podría hacer?
Realizar el algoritmo complejo para completar la función de cronómetro.	Escribir "TIEMPO INGRESADO : ", H, ":", M, ":", S; Para CH ← 0 hasta H con paso 1 hacer para CM ← 0 hasta M con paso 1 hacer para CS ← 0 hasta S con paso 1 hacer Escribir (H-CH), ":", M, ":", (S-CS); Esperar 1 Segundo; FinPara S ← 60; M ← M -1; FinPara M ← 60; S ← 60; FinPara Escribir "FIN DE CRONÓMETRO"; Escribir "Volvera a Reiniciar la cuenta? (s/n)"; Leer continuar; FinMientras FinAlgoritmo	¿El algoritmo debe ser complejo?

PRÁCTICA GUIADA

Para guiar el proceso de Evaluar, se sugiere que la o el docente retome el ejercicio "RearmaCom" (el ejemplo de programación se llamará "PCMasters") del módulo anterior. Pero esta vez utilizar los arreglos para guardar los artículos que se introducirán, como máximo 10 y, al finalizar, se hará una factura donde además se sumará el IVA total de los productos comprados. Se trabaja con el IVA de un 18%.

Acción	Resultado esperado
Primero se crean las dimensiones para guardar las diferentes categorías que guardará el producto. Esto debido a que solo guardan una clase de datos (carácter o numérico, pero no ambos).	Algoritmo PCMastersV2 <pre> Dimension articulo[10,1], cantidad[10,1], precio[10,1], total[10,1] Empresa = "PCMasters" continuar ← "S"; fila ← 1; //Inicio Escribir "INGRESE DATOS DEL CLIENTE" Escribir "===== Escribir Sin Saltar "Nombre : " Leer Nombre Escribir Sin Saltar "RUT : " Leer Rut Escribir Sin Saltar "Direccion : " Leer Direccion </pre>
Luego, se crea un mientras se mantenga el carácter "s", para poder repetir en cada momento la acción y así agregar los nuevos artículos.	<pre> //Desarrollo Escribir "INGRESE ARTICULOS" Escribir "===== Mientras MAYUSCULAS(continuar) == "S" Hacer Escribir Sin Saltar "Articulo Nro " fila " : "; Leer articulo(fila,1); Escribir Sin Saltar "Cantidad : "; Leer cantidad(fila,1); Escribir Sin Saltar "Pre. Unitario \$: "; Leer precio(fila,1); fila ← fila + 1; Escribir Sin Saltar "OTRO ARTICULO (S/N)?"; Leer continuar; Escribir " "; FinMientras </pre>
Crear el menú de la factura con todos los datos del comprador ingresados anteriormente, y utilizar la función para y hasta con las diferentes dimensiones. Hacer el respectivo cálculo del precio y guardar el total como una suma.	<pre> Escribir "FACTURA Nro 000-1" Escribir "===== Escribir "Empresa : " Empresa Escribir "Cliente : " Nombre " RUT : " Rut Escribir "Direccion : " Direccion Escribir "===== Escribir "CANTIDAD DESCRIPCION PRE. UNIT. PRE. TOTAL" Para f ← 1 hasta fila-1 con paso 1 hacer Para c ← 1 hasta 1 con paso 1 hacer Escribir Sin Saltar cantidad[f,c], " "; Escribir Sin Saltar " " articulo[f,c], " "; Escribir Sin Saltar " \$" precio[f,c], " "; total[f,c] = cantidad[f,c] * precio[f,c] TotalT = TotalT + total[f,c] Escribir Sin Saltar " \$" total[f,c], " "; FinPara Escribir " "; FinPara </pre>

Acción	Resultado esperado
Al finalizar, mostrar el costo total y el IVA, además de la suma de estos dos para dar el fin de la factura con el total a pagar.	<pre>Escribir "-----" IVA = TotalT * 0.18 Escribir " \$. " TotalT Escribir " IMPUERTO IVA 18% \$. " IVA TotalTT = TotalT + IVA Escribir " -----" Escribir " TOTAL A PAGAR \$. " TotalTT FinAlgoritmo</pre>
Resultado de la operación.	<pre>INGRESE DATOS DEL CLIENTE ===== Nombre :> Harry Potter RUT :> 168457899 Direccion :> Av Siempre Viva 1234 INGRESE ARTICULOS ===== Articulo Nro 1 :> Intel core i5 Cantidad :> 2 Pre. Unitario \$:> 189990 OTRO ARTICULO (S/N)?> s Articulo Nro 2 :> RTX 3080Ti Cantidad :> 3 Pre. Unitario \$:> 899990 OTRO ARTICULO (S/N)?> s Articulo Nro 3 :> ROG Helios Cantidad :> 1 Pre. Unitario \$:> 299990 OTRO ARTICULO (S/N)?> n FACTURA Nro 000-1 ===== Empresa : PCmasters Cliente : Harry Potter Direccion : Av Siempre Viva 1234 RUT : 168457899 ===== CANTIDAD DESCRIPCION PRE. UNIT. PRE.TOTAL 2 Intel core i5 \$.189990 \$.379980 3 RTX 3080Ti \$.899990 \$.2699970 1 ROG Helios \$.299990 \$.299990 ===== \$.3379940 IMPUERTO IVA 18% \$.608389.2 ----- TOTAL A PAGAR \$.3988329.2000000002</pre>

**PRÁCTICA
INDEPENDIENTE**

Se recomienda un trabajo entre pares, donde cada estudiante proponga un cambio en la situación de la factura con el IVA, y realizar la programación en pseudocódigo con el mismo esquema anterior. Algunos de los cambios que se sugieren son: variar la cantidad de artículos, el porcentaje de IVA, no incluir la dirección del cliente o agregar un nuevo dato del cliente, o bien calcular primero el IVA de cada artículo y luego el IVA total.

Entrada	Salida
Cantidad de artículos.	Algoritmo PCMastersV2 Dimension articulo[10,1], cantidad[10,1], precio[10,1], total[10,1] Empresa = "PCMasters" continuar ← "S"; fila ← 1;
Sin dirección.	Escribir "INGRESE DATOS DEL CLIENTE" Escribir "===== Escribir Sin Saltar "Nombre :" Leer Nombre Escribir Sin Saltar "RUT :" Leer Rut Escribir Sin Saltar "Direccion :" Leer Direccion
Cambio en el porcentaje.	Escribir "===== IVA = TotalT * 0.18 Escribir " Escribir " \$. " TotalT IMPUESTO IVA 18% \$. " IVA TotalTT = TotalT + IVA Escribir " Escribir " ----- TOTAL A PAGAR \$. " TotalTT

EVALUACIÓN FORMATIVA

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Pensamiento algorítmico	Formula una problemática algorítmica simple.	Formula un problema algorítmico medianamente complejo.	Formula un problema algorítmico complejo.
Seudocódigo	Elabora un pseudocódigo con dificultad para ejecutarse.	Elabora un pseudocódigo que logra ejecutarse de forma parcial.	Elabora un pseudocódigo que logra ejecutarse de forma completa.

Revisión por pares con enfoque en la rúbrica:

Los y las estudiantes trabajan en parejas para revisar el pseudocódigo y el código de sus pares, utilizando la rúbrica como guía. Cada estudiante proporciona retroalimentación específica sobre el pseudocódigo, las estructuras condicionales, los ciclos y el código en general.

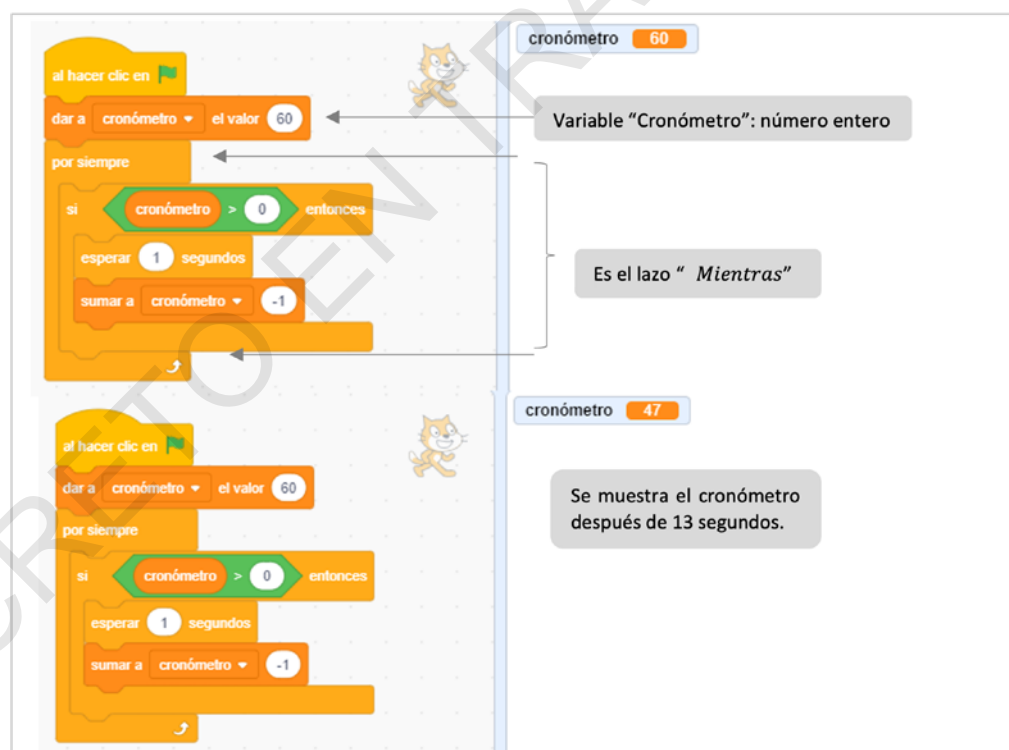
Orientaciones al docente

Para unificar conceptos disciplinares

En cuanto a pensar de forma algorítmica, se refiere a cómo poder ver las acciones cotidianas y convertirlas en algoritmos y así dar una serie de instrucciones para interpretarlo de esta manera. Además, es necesario tener claros todos los conceptos que conlleva crear un algoritmo para poder crear los propios, con el fin de resolver problemas y así comprender mejor lo que conlleva pensar de esta forma.

Pensar de forma algorítmica corresponde a la capacidad para expresar soluciones compuestas de una serie de pasos finitos, no ambiguos y ordenados que permitan que una persona o un procesador de información pueda llevarlo adelante.

De ser posible, se sugiere incluir algunos elementos de la programación en bloques, para ver alcances y limitantes en la creación de un cronómetro de cuenta regresiva. En este ejemplo, se ha utilizado el programa **Scratch**.



La función lazo "mientras" da la condición de "correr el programa", que termina automáticamente cuando la condición no se cumple más. En este caso el lazo no necesita una condición de "si no".

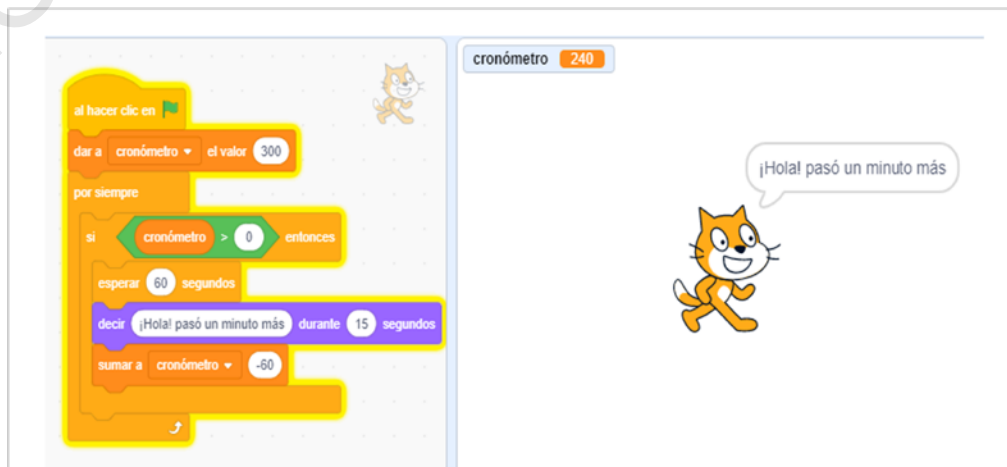


Se sugiere incluir algunas preguntas que permitan agudizar la lectura de programas en pseudocódigo y en bloques, por ejemplo, cuando se detecta que el cronómetro indica a partir de 60 segundos y cada 5 segundos el tiempo restante.

¿En qué modo está funcionando el cronómetro que describe el programa en **Scratch**?



También se pueden incluir nuevas instrucciones, como elaborar un programa en **Scratch** que muestre un cronómetro que cuenta hacia atrás a partir de 300 segundos y que muestre cada 60 segundos la frase “¡Hola! Pasó un minuto más”. Y solicitar que este mensaje aparezca por 15 segundos.



Actitudes Para apoyar el desarrollo de las actividades y cuando se requiera de un nivel complejo de codificación, es recomendable el uso de computadores que faciliten el desarrollo de los ejercicios. Se puede utilizar una hoja de papel o un documento Word para los problemas más simples de pensamiento algorítmico.

**Orientaciones para
organizar e implementar
el trabajo colaborativo**

Se sugieren las siguientes indicaciones para guiar la implementación del trabajo entre pares:

- Dejar que los y las estudiantes elijan sus pares o que la o el docente elija las parejas y explicar el concepto del desafío (estudiante 1 hace su propio problema para que estudiante 2 lo resuelva y viceversa).
- Los y las estudiantes elaborarán sus desafíos y los explicarán para poder resolverlo de manera algorítmica.
- Determinar el tiempo necesario para elaborar un algoritmo complejo.
- Organizar el espacio para el trabajo entre pares.
- Determinar el tiempo para que revisen el algoritmo de su par y retroalimenten a partir de una rúbrica previamente conocida.

DECRETO EN TRÁMITE

Actividad de desempeño

2

Propósito

Para esta actividad se llevará a cabo la ampliación del conocimiento en base al funcionamiento de los algoritmos. Esto implica desafiar a los y las estudiantes a tomar conciencia del nivel de dificultad que llegan a tener los problemas y la magnitud de las soluciones. Significa pensar y poner en uso habilidades y conocimientos posibles de seguir transfiriendo a otras realidades. En la medida que manejan más herramientas para plantear algoritmos, se suman más procedimientos y acciones que aumentan la eficiencia al pensar en soluciones frente a un problema.

Objetivos de aprendizaje

OA 4

Establecer conexiones y semejanzas entre las partes del problema para caracterizar patrones en una situación problemática dada. ([Generalizar y transferir](#))

OA 5

Aplicar flexiblemente las soluciones elaboradas e implementadas a otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista. ([Generalizar y Transferir](#))

Conocimiento esencial

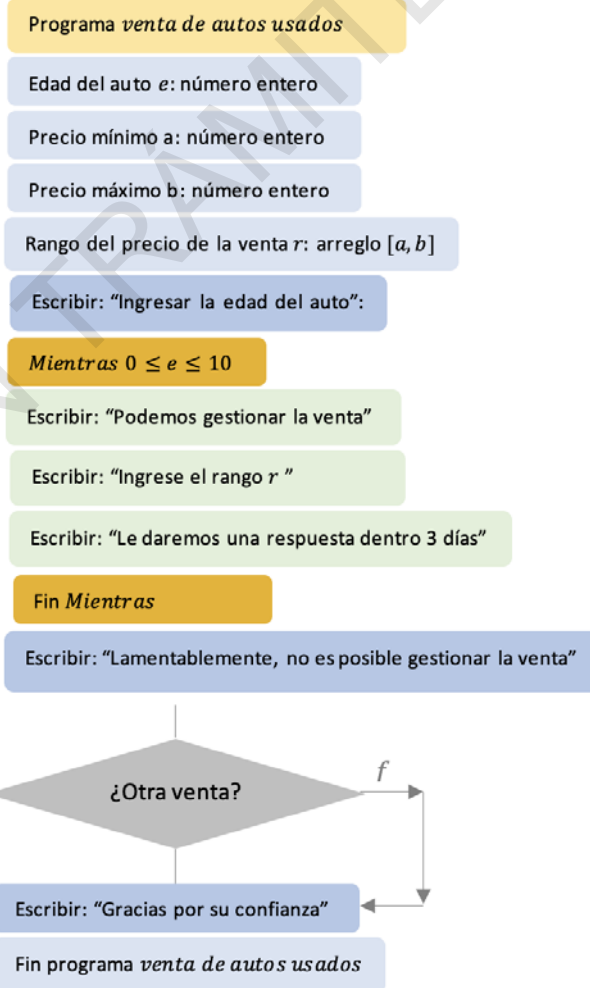
- Algoritmos y programación: lenguajes, métodos y entornos de programación.

Tiempo estimado: 3 horas pedagógicas

Desarrollo de la actividad

SITUACIÓN EXPERIENCIAL

El o la docente presenta una situación de venta de autos usados y la creación de un programa en pseudocódigo que puede servir como plataforma virtual. El objetivo es que, quienes accedan a la plataforma, puedan ingresar la edad del auto y el programa aceptará gestionar la venta solo si no se supera la edad de 10 años. Después, los usuarios pueden ingresar los datos del auto en venta. Se puede ingresar más de un auto en venta, pero si alguno supera los 10 años de antigüedad, aparecerá el mensaje “Lamentablemente no podemos gestionar la venta de su auto”.



Algunas de las preguntas que podrían promover la conversación en torno a los lazos y funciones de programación son:

- ¿Cómo describirías este programa?
- ¿Cuál es el problema al cuál se quiere responder?
- ¿Cómo funciona este programa?

CONSTRUCCIÓN DE CONOCIMIENTO

Para construir el conocimiento, se sugiere comparar dos programas diferentes, pero con el mismo objetivo, en este caso: elaborar una calculadora. En ambos programas se necesita diferenciar los lazos y las funciones de programación que se han utilizado.

Antes de iniciar, pueden crear con pseudocódigo en la hoja de anexo, en la zona de escritura de código –hoja de trabajo– y numerar cada paso.

Acción	Resultado esperado	Preguntas de apoyo
Elaborar las definiciones de manera correcta.	Algoritmo Calculadora Definir opc, valor1, valor2 Como Entero ; Definir total como real ; Escribir "Calculadora"; Escribir "1: Sumar"; Escribir "2: Restar"; Escribir "3: Multiplicar"; Escribir "4: Dividir"; Escribir "Ingrese opcion:"; Leer opc;	¿Cuál es el nombre del programa? ¿De qué tipo son las variables? ¿Cómo lee el programa estas palabras?
Reconocer los subprocesos, lazos y funciones.	<pre> Si opc ≥ 1 y opc ≤ 4 Entonces Escribir "Ingrese valor 1: "; Leer valor1; Escribir "Ingrese valor 2: "; Leer valor2; Segun opc 1: total = valor1 + valor2; Escribir "La suma de " valor1 " + " valor2 " = " total; 2: total = valor1 - valor2; Escribir "La resta de " valor1 " - " valor2 " = " total; 3: total = valor1 * valor2; Escribir "La multiplicacion de " valor1 " * " valor2 " = " total; 4: total = valor1 / valor2; Escribir "La division de " valor1 " / " valor2 " = " total; FinSegun SiNo Escribir "Opcion invalida"; FinSi FinAlgoritmo </pre>	¿Dónde se utiliza la función "si...entonces"? ¿Dónde comienza la función "según"? ¿Dónde termina la función "si... entonces"? ¿Está incluida la función "para"? ¿Qué tipos de subprocesos están incluidos?

Para comparar, se sugiere utilizar las mismas preguntas de apoyo y anotar las discrepancias para, luego, ver las diferencias en el funcionamiento del programa.

Acción	Resultado esperado	Preguntas de apoyo
Elaborar las definiciones de manera correcta.	Algoritmo CalculadoraV2 Dimension Num[1,10], Signos[1,10]	¿Cuál es el nombre del programa? ¿De qué tipo son las variables? ¿Cómo lee el programa estas palabras?
Reconocer los subprocesos, lazos y funciones.	<pre> Repetir Limpiar Pantalla TotalT = 0; Escribir "Calculadora"; Escribir Sin Saltar "Ingrese el 1 numero: "; Leer Num[1,1]; TotalT = TotalT + Num[1,1]; Escribir Sin Saltar "Cuántas operaciones planea hacer? (Maximo 10)"; Leer f; Escribir ""; Para c ← 2 hasta f con paso 1 hacer Escribir "1: Sumar - 2: Restar - 3: Multiplicar - 4: Dividir"; Escribir Sin Saltar "Ingrese opcion:"; Leer opc; Segun opc 1: Signos[1,c] = " + "; 2: Signos[1,c] = " - "; 3: Signos[1,c] = " * "; 4: Signos[1,c] = " / "; FinSegun Escribir Sin Saltar "Ingrese el " c " numero: "; Leer Num[1,c]; Segun opc 1: TotalT = TotalT + Num[1,c]; 2: TotalT = TotalT - Num[1,c]; 3: TotalT = TotalT * Num[1,c]; 4: TotalT = TotalT / Num[1,c]; FinSegun FinPara Para c ← 1 hasta f con paso 1 hacer Escribir Sin Saltar Signos[1,c]; Escribir Sin Saltar Num[1,c]; FinPara Escribir Sin Saltar " = " TotalT; Escribir ""; Escribir Sin Saltar "OTRA OPERACION (S/N)?"; Leer aux; Escribir "Precione una tecla" Esperar Tecla Hasta Que Mayusculas(aux) == "N" FinAlgoritmo </pre>	¿Dónde se utiliza la función "si... entonces"? ¿Dónde comienza la función "según"? ¿Dónde termina la función "si... entonces"? ¿Está incluida la función "para"? ¿Qué tipos de subprocesos están incluidos?

PRÁCTICA GUIADA

Se sugiere proponer a las y los estudiantes volver a recrear el algoritmo para calcular el promedio final. Para esto deben tener claro que se preguntará el nombre del estudiante y sus cuatro respectivas notas (tres notas parciales y una nota final). Después de haber ingresado las notas, se le preguntará si desea agregar otro estudiante. Al finalizar, se pondrá el nombre con el promedio final.

$$\text{Promedio Nota} = \frac{\text{suma de todas las notas}}{\text{cantidad de notas}}$$

$$\text{Promedio final} = (\text{Promedio Nota} \cdot 0,6) + (\text{Final} \cdot 0,4)$$

Acción	Resultado esperado
Elaborar el ingreso de datos mediante el guardado de información en los arreglos.	<pre>Algoritmo CalcularPromedioFinalV2 Dimension Alumno[10,1], Nota[10,3], promedio[10,1], NT[10,1] continuar ← "S"; fila ← 1; Mientras MAYUSCULAS(continuar) == "S" Hacer Escribir Sin Saltar "Nombre Alumno " fila ":"; Leer Alumno(fila,1); Para c ← 1 hasta 3 con paso 1 hacer Escribir Sin Saltar "Nota " c ":"; Leer Nota(fila,c); FinPara Escribir Sin Saltar "Nota Examen Final: "; Leer NT(fila,1); fila ← fila + 1; Escribir Sin Saltar "OTRO ALUMNO (S/N)?"; Leer continuar; Escribir ""; FinMientras</pre>
Utilizar los arreglos guardados para hacer los respectivos cálculos y proporcionar la respectiva visualización del nombre y el promedio final.	<pre> Escribir "NOMBRE PROMEDIO"; Para f ← 1 hasta fila-1 con paso 1 hacer Para c ← 1 hasta 3 con paso 1 hacer promedio[f,1] = promedio[f,1] + Nota[f,c]; FinPara promedio[f,1] = ((promedio[f,1] / 3) * 0.6 + (NT[f,1] * 0.4)); Escribir Sin Saltar Alumno[f,1] " "; Escribir Sin Saltar promedio[f,1]; Escribir " "; FinPara FinAlgoritmo</pre>

Acción	Resultado esperado
Resultado esperado tras la ejecución del algoritmo.	<pre> Nombre Alumno 1:> Wolverine Nombre Alumno 2:> Peter Parker Nota 1:> 20 Nota 1:> 60 Nota 2:> 55 Nota 2:> 65 Nota 3:> 40 Nota 3:> 60 Nota Examen Final: > 45 Nota Examen Final: > 70 OTRO ALUMNO (S/N)?> s OTRO ALUMNO (S/N)?> n NOMBRE PROMEDIO Wolverine 41 Peter Parker 65 </pre>

Se sugiere que el o la docente favorezca situar el foco en las mejoras respecto de los aprendizajes para elaborar un pseudocódigo mediante el uso de las bases del algoritmo. Destaque que esto permite elaborar soluciones mucho más complejas y así llevar a cabo soluciones que abarcan un mayor número de posibilidades.

Luego, podría señalar que, en otras circunstancias, las formas de formular un algoritmo pueden ser ilimitadas. Algunas se parecerán en mayor o menor medida entre otras, pero siempre se intentará resolver el mismo problema, por eso es necesario que los y las estudiantes puedan tener claro que lo importante es siempre resolver el problema.

PRÁCTICA
INDEPENDIENTE

Se sugiere un trabajo individual de evaluación y la transferencia de algoritmos. Para esto, se pueden considerar dos programas en pseudocódigo y código: el pseudocódigo mediante el uso del software (ya antes trabajado) PSeint u otro formato y, como código, el lenguaje de programación Python, R, o en aplicaciones de programación (Scratch, Octostudio), en las que se hace una agregación de datos a una lista o arreglo y se pone información como nombre, apellido y dirección.

Acción	Resultado esperado
Ver que es un ejercicio donde se guarda información en un arreglo y después se muestra la información ingresada.	<pre>Algoritmo Listado Definir datos como Caracter; Definir f, c como Entero; Dimension datos[3,3]; Para f ← 1 hasta 2 con paso 1 hacer Escribir Sin Saltar "NOMBRE : "; Leer datos[f,1]; Escribir Sin Saltar "APELLIDO : "; Leer datos[f,2]; Escribir Sin Saltar "DIRECCIÓN : "; Leer datos[f,3]; Escribir ""; FinPara Escribir ""; Escribir "NOMBRE APELLIDO DIRECCION" Para f ← 1 hasta 2 con paso 1 hacer Escribir datos[f,1]," ",datos[f,2]," ",datos[f,3]; FinPara FinAlgoritmo</pre>
Esto es una versión más formal de lo que está arriba. Entonces apreciar aquí que, a pesar de tener diferencias, no se ven muy diferentes, ya que hacen exactamente lo mismo.	<pre>if __name__ == '__main__': datos = [[str() for ind0 in range(3)] for ind1 in range(3)] for f in range(3): print("NOMBRE : ") datos[f][0] = input() print("APELLIDO : ") datos[f][1] = input() print("DIRECCIÓN : ") datos[f][2] = input() print("") print("") for f in range(3): print(datos[f][0]," ",datos[f][1]," ",datos[f][2])</pre>

EVALUACIÓN FORMATIVA

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Seudocódigo	Elabora un pseudocódigo.	Elabora un código que cumple con lo establecido.	Elabora un código que cumple con lo establecido e incorpora mejoras.
Solución	Mejora el pseudocódigo.	Mejora la solución incorporando una opción no descrita en el problema.	Mejora la solución incorporando opciones que mejoran lo establecido en el problema y la solución.
Análisis del problema	Analiza el problema sin mejoras.	Analiza el problema y entrega parcialmente mejoras.	Analiza el problema y entrega mejoras para optimizarlo.

Orientaciones al docente

Para unificar conceptos disciplinares

En este módulo se proponen desafíos o ejercicios más complejos. Motive a las y los estudiantes a pensar más allá del problema para poder resolver. Además, es necesario revisar lo que es un lenguaje de programación computacional para trabajar, de preferencia, con Python porque es un lenguaje sencillo y muy recomendable para empezar a introducir al estudiantado en lenguajes usados en ámbitos profesionales y de trabajo.

La CalculadoraV1 es un ejercicio que se hizo en el módulo 1. Primero se explica a los y las estudiantes que se desea usar “sumas, restar, multiplicar o dividir” y después se ingresan los datos a calcular con su respectiva visualización y el resultado final.

La CalculadoraV2 es una mejora que permite utilizar todas las herramientas necesarias para poder expresar una mayor magnitud de resultados, con el fin de no solo elegir simplemente dos valores a calcular, sino que para que el usuario decida cuántos valores planea calcular. En este caso, el máximo es de 10 y termina haciendo el cálculo esperado.

Actitudes

En esta actividad se recomienda el uso de computadores para codificar el pseudocódigo, para verificar de una manera eficiente lo realizado, ejecutar y ver el funcionamiento de este. Para el desarrollo del trabajo individual es necesario tener claro que esto es para que cada estudiante sienta seguridad de lo aprendido y tenga procesos metacognitivos que le permitan evaluar su forma de pensar algorítmicamente.

Orientaciones para organizar e implementar el juego de roles

Se sugieren las siguientes indicaciones para guiar la implementación del trabajo individual:

- Elaborar un ejercicio simple que pueda ser mejorado de diferentes maneras.
- Determinar un tiempo para así resolver cualquier duda del problema.
- Determinar el tiempo que cree necesario para haber terminado o al menos llevar algo de desarrollo del ejercicio.
- Determinar el tiempo para retroalimentar a sus estudiantes.

Actividad de desempeño

3

Propósito

Esta actividad se enfocará en las reglas básicas del funcionamiento de “consulta de base de datos” y en aprovechar oportunidades cotidianas para la creación de tablas, sus restricciones y su respectivo ingreso de información. Asimismo, se seguirá favoreciendo que las soluciones a problemas estén en los dominios del uso de redes e Internet, así como en la relación que tienen las personas con los computadores para obtener diferentes respuestas o resultados.

Objetivos de aprendizaje

OA 5

Aplicar flexiblemente las soluciones elaboradas e implementadas a otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista. [\(Generalizar y transferir\)](#)

Conocimiento esencial

- Redes e Internet: sitios web, aplicaciones móviles, seguridad cibernética y objetos controlados digitalmente.

Tiempo estimado: 3 horas pedagógicas

Desarrollo de la actividad

SITUACIÓN EXPERIENCIAL

Se sugiere que el o la docente presente a sus estudiantes una situación laboral de registro de datos de empleadores según una tabla con ciertas condiciones de entrada. Esta creación de tablas permite organizar los datos de los empleados de forma ordenada y de fácil acceso. También, con estos programas es posible acceder a parte de la información según se requiera.

En la oficina
de
Recursos Humanos



Hola, bienvenido a la empresa. Le asignaremos un código de 4 caracteres que usaremos de forma interna.

Por favor, dígame su nombre completo....

¿Cuál es su dirección?

En su contrato, el sueldo indicado es...

¿En qué sección trabajará?

Algunas de las preguntas que pueden motivar la introducción de las nuevas funciones de creación “CREATE”, de variación “Alter”, de condiciones “CONSTRAINT” y de ingresar “INSERT”, pueden ser:

- ¿Cómo se pueden organizar estos datos?
- ¿Cómo podemos programar la creación de una tabla?
- ¿Qué ocurre cuando se tienen varios empleados?

CONSTRUCCIÓN DE CONOCIMIENTO

Se sugiere ampliar el conocimiento sobre las funciones de programación, incluyendo la creación, la variación, el condicional y el ingreso. Para esto, continuar con la misma situación y explicar la transferencia desde el modelo relación-entidad a la programación en pseudocódigo y cada vez señalar lo que hace cada función.

EMPLEADO	
P *	codigo VARCHAR (4)
*	nombre VARCHAR (30)
*	direccion VARCHAR (30)
*	sexo VARCHAR (1)
*	sueldo NUMERIC (8)
EMPLEADO PK (codigo)	

```
-- CREACION DE LA TABLA EMPLEADO
CREATE TABLE empleado (
  codigo VARCHAR2(4) NOT NULL,
  nombre VARCHAR2(30) NOT NULL,
  direccion VARCHAR2(30) NOT NULL,
  sexo NUMBER(1) NOT NULL,
  sueldo NUMBER(8) NOT NULL
);
ALTER TABLE empleado ADD CONSTRAINT empleado_pk PRIMARY KEY ( codigo );
-- INSERTAR LOS DATOS A EMPLEADO EL 1 = FEMENINO Y EL 2 = MASCULINO
INSERT INTO empleado VALUES ( 'AN01', 'ANA', 'CONCHALI', 1, 350000 );
INSERT INTO empleado VALUES ( 'LU01', 'LUIS', 'QUILICURA', 2, 375000 );
INSERT INTO empleado VALUES ( 'MA01', 'MARCELA', 'VITACURA', 1, 325000 );
```

PRÁCTICA GUIADA Se sugiere realizar en conjunto el desarrollo del siguiente modelo de base de datos y agregar datos a cada tabla para, luego, revisar el posible orden de eliminación de datos.

Modo entidad-relación

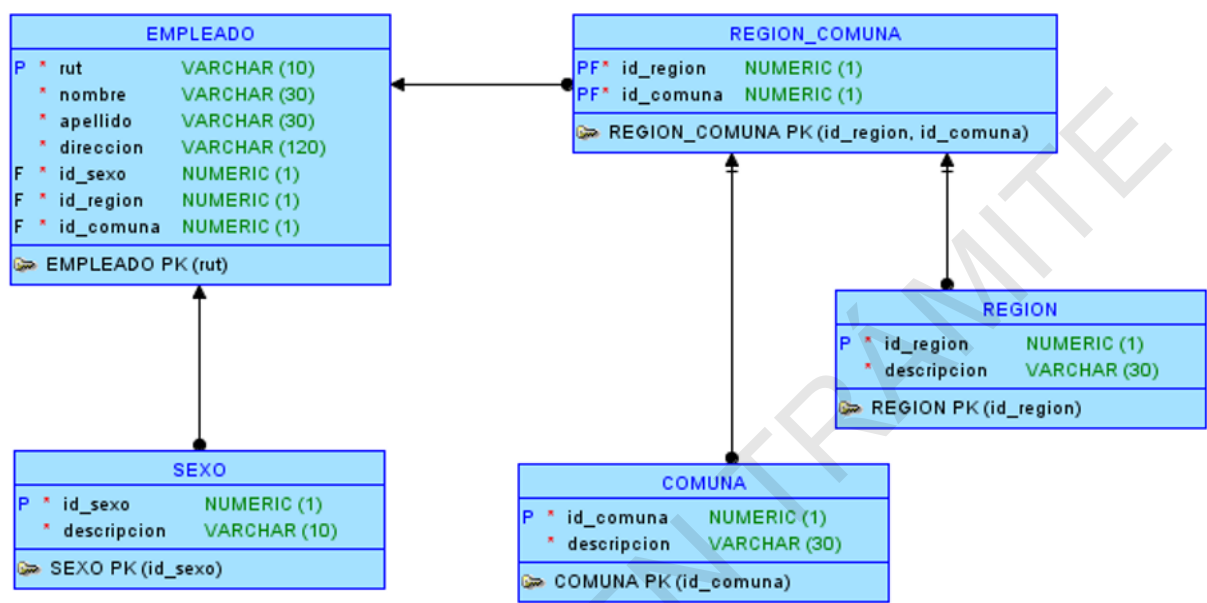


Tabla de datos para ingresar

RUT	NOMBRE	APELLIDO	DIRECCION	ID_SEXO	ID_REGION	ID_COMUNA
18789456-K	Alan	Albornoz	Av Siempre Vida #1234	2	1	1
12321987-9	Xavier	Rojas	Lo Ovalle #659	2	2	4
14852963-7	Macarena	Diaz	Camilo Henriquez #8492	1	3	7

ID_COMUNA	DESCRIPCION
1	Camarones
2	Putre
3	Conchali
4	Quilicura
5	Vitacura
6	Los Lagos
7	Paillaco

ID_REGION	DESCRIPCION
1	ARICA Y PARINACOTA
2	METROPOLITANA DE SANTIAGO
3	LOS RÍOS

ID_SEXO	DESCRIPCION
1	Femenino
2	Masculino
3	Otros...

ID_REGION	ID_COMUNA
1	1
1	2
2	3
2	4
2	5
3	6
3	7

Acción	Resultado esperado	Preguntas de apoyo
Crear tablas con sus respectivas claves primarias y claves foráneas.	<pre>CREATE TABLE comuna (id_comuna NUMBER(1) NOT NULL CONSTRAINT comuna_pk PRIMARY KEY, descripcion VARCHAR2(30) NOT NULL); CREATE TABLE region (id_region NUMBER(1) NOT NULL CONSTRAINT region_pk PRIMARY KEY, descripcion VARCHAR2(30) NOT NULL); CREATE TABLE region_comuna (id_region NUMBER(1) NOT NULL, id_comuna NUMBER(1) NOT NULL, CONSTRAINT region_comuna_pk PRIMARY KEY (id_region, id_comuna), CONSTRAINT reg_com_comuna_fk FOREIGN KEY (id_comuna) REFERENCES comuna (id_comuna), CONSTRAINT reg_com_region_fk FOREIGN KEY (id_region) REFERENCES region (id_region)); CREATE TABLE sexo (id_sexo NUMBER(1) NOT NULL CONSTRAINT sexo_pk PRIMARY KEY, descripcion VARCHAR2(10) NOT NULL); CREATE TABLE empleado (rut VARCHAR2(10) NOT NULL CONSTRAINT empleado_pk PRIMARY KEY, nombre VARCHAR2(30) NOT NULL, apellido VARCHAR2(30) NOT NULL, direccion VARCHAR2(120) NOT NULL, id_sexo NUMBER(1) NOT NULL, id_region NUMBER(1) NOT NULL, id_comuna NUMBER(1) NOT NULL, CONSTRAINT empleado_reg_com_fk FOREIGN KEY (id_region,id_comuna) REFERENCES region_comuna (id_region,id_comuna), CONSTRAINT empleado_sexo_fk FOREIGN KEY (id_sexo) REFERENCES sexo (id_sexo));</pre>	¿Cómo saber con cuál empezar?
Elaborar el ingreso de datos a las tablas anteriormente ya creadas.	<pre>INSERT INTO sexo VALUES (1,'Femenino'); INSERT INTO sexo VALUES (2,'Masculino'); INSERT INTO sexo VALUES (3,'Otros...'); INSERT INTO comuna VALUES (1,'Camarones'); INSERT INTO comuna VALUES (2,'Putre'); INSERT INTO comuna VALUES (3,'Conchali'); INSERT INTO comuna VALUES (4,'Quilicura'); INSERT INTO comuna VALUES (5,'Vitacura'); INSERT INTO comuna VALUES (6,'Los Lagos'); INSERT INTO comuna VALUES (7,'Paillaco'); INSERT INTO region VALUES (1,'ARICA Y PARINACOTA'); INSERT INTO region VALUES (2,'METROPOLITANA DE SANTIAGO'); INSERT INTO region VALUES (3,'LOS RÍOS'); INSERT INTO region_comuna VALUES (1,1); INSERT INTO region_comuna VALUES (1,2); INSERT INTO region_comuna VALUES (2,3); INSERT INTO region_comuna VALUES (2,4); INSERT INTO region_comuna VALUES (2,5); INSERT INTO region_comuna VALUES (3,6); INSERT INTO region_comuna VALUES (3,7); INSERT INTO empleado VALUES ('18789456-K','Alan','Albornoz', 'Av Siempre Vida #1234',2,1,1); INSERT INTO empleado VALUES ('12321987-9','Xavier','Rojas', 'Lo Ovalle #659', 2,2,4); INSERT INTO empleado VALUES ('14852963-7','Macarena','Diaz', 'Camilo Henriquez #8492',1,3,7);</pre>	¿En qué orden debería insertar los datos?

Acción	Resultado esperado	Preguntas de apoyo
Elaborar la eliminación de las tablas.	<pre>DROP TABLE empleado; DROP TABLE sexo; DROP TABLE region_comuna; DROP TABLE comuna; DROP TABLE region;</pre>	¿En qué orden deberían eliminarse?

PRÁCTICA INDEPENDIENTE

Se sugiere un trabajo de pares y continuar con el ejercicio anterior y priorizando una tarea particular, por ejemplo: la gerencia de recursos humanos requiere contar con una lista de empleados donde esté toda la información necesaria y ordenada alfabéticamente. La lista debe tener el formato y orden que se aprecia en la imagen.

RUT	NOMBRE COMPLETO	DIRECCION	SEXO	COMUNA	REGION
18789456-K	Alan Albornoz	Av Siempre Vida #1234	Masculino	Camarones	ARICA Y PARINACOTA
14852963-7	Macarena Diaz	Camilo Henriquez #8492	Femenino	Paillaco	LOS RÍOS
12321987-9	Xavier Rojas	Lo Ovalle #659	Masculino	Quilicura	METROPOLITANA DE SANTIAGO

Acción	Resultado esperado																												
Realizar un SELECT FROM para poder visualizar los datos de EMPLEADO.	<pre>SELECT RUT, NOMBRE, APELLIDO, DIRECCION, ID_SEXO, ID_COMUNA, ID_REGION FROM EMPLEADO;</pre> <table><tr><th>RUT</th><th>NOMBRE</th><th>APELLIDO</th><th>DIRECCION</th><th>ID_SEXO</th><th>ID_COMUNA</th><th>ID_REGION</th></tr><tr><td>18789456-K</td><td>Alan</td><td>Albornoz</td><td>Av Siempre Vida #1234</td><td>2</td><td>1</td><td>1</td></tr><tr><td>12321987-9</td><td>Xavier</td><td>Rojas</td><td>Lo Ovalle #659</td><td>2</td><td>4</td><td>2</td></tr><tr><td>14852963-7</td><td>Macarena</td><td>Diaz</td><td>Camilo Henriquez #8492</td><td>1</td><td>7</td><td>3</td></tr></table>	RUT	NOMBRE	APELLIDO	DIRECCION	ID_SEXO	ID_COMUNA	ID_REGION	18789456-K	Alan	Albornoz	Av Siempre Vida #1234	2	1	1	12321987-9	Xavier	Rojas	Lo Ovalle #659	2	4	2	14852963-7	Macarena	Diaz	Camilo Henriquez #8492	1	7	3
RUT	NOMBRE	APELLIDO	DIRECCION	ID_SEXO	ID_COMUNA	ID_REGION																							
18789456-K	Alan	Albornoz	Av Siempre Vida #1234	2	1	1																							
12321987-9	Xavier	Rojas	Lo Ovalle #659	2	4	2																							
14852963-7	Macarena	Diaz	Camilo Henriquez #8492	1	7	3																							
Realizar un SELECT FROM + JOIN para llamar las tablas que están relacionadas con la tabla EMPLEADO.	<pre>SELECT E.RUT, E.NOMBRE, E.APELLIDO, E.DIRECCION, S.ID_SEXO, C.ID_COMUNA, R.ID_REGION FROM EMPLEADO E JOIN COMUNA C ON C.ID_COMUNA = E.ID_COMUNA JOIN REGION R ON R.ID_REGION = E.ID_REGION JOIN SEXO S ON S.ID_SEXO = E.ID_SEXO;</pre>																												
Realizar los ajustes finales (llamar la descripción de las tablas para poner el nombre y no el ID) y darles un orden dependiendo del nombre del empleado.	<pre>SELECT RUT, NOMBRE ' ' APELLIDO "NOMBRE COMPLETO", DIRECCION, S.DESCRIPCION SEXO, C.DESCRIPCION COMUNA, R.DESCRIPCION REGION FROM EMPLEADO E JOIN COMUNA C ON C.ID_COMUNA = E.ID_COMUNA JOIN REGION R ON R.ID_REGION = E.ID_REGION JOIN SEXO S ON S.ID_SEXO = E.ID_SEXO ORDER BY E.NOMBRE;</pre>																												

EVALUACIÓN FORMATIVA

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Creación de tablas	Crea tablas, pero no se ejecuta la secuencia.	Crea tablas, pero debe repetirse el proceso para su ejecución.	Crea tablas que ejecutan la secuencia sin errores.
Inserción de tablas	Inserta tablas, pero no se ejecuta la secuencia	Inserta tablas, pero debe repetirse el proceso para su ejecución.	Inserta tablas que ejecutan la secuencia sin errores.
Eliminación de tablas	Elimina tablas, pero no se ejecuta la secuencia	Elimina tablas, pero debe repetirse el proceso para su ejecución.	Elimina tablas que ejecutan la secuencia sin errores.

Revisión por pares con enfoque en la rúbrica:

Los y las estudiantes trabajan en parejas para revisar el pseudocódigo y el código de sus pares, con uso de la rúbrica como guía. Cada estudiante proporciona retroalimentación específica sobre el pseudocódigo, las estructuras condicionales, los ciclos y el código en general.

Orientaciones al docente

Para unificar conceptos disciplinares

En esta actividad se deja de lado el modelamiento de base de datos, se pone énfasis en la consulta y creación de base de datos, las aplicaciones utilizadas, como Oracle DataModeler y se utiliza además Oracle Database Express Edition (XE) Release (18c) y Oracle SQL Developer 21.2.1.

Los comandos tienen como objetivo permitir la manipulación de los datos existentes en las tablas de la base de datos:

- **CREATE:** permite crear tablas.
- **ALTER:** permite alterar tablas
- **DROP:** permite eliminar tablas.
- **INSERT:** permite añadir filas a una tabla.
- **UPDATE:** permite actualizar valores de una o más columnas para un subconjunto de las filas de una tabla.
- **DELETE:** borra una o más filas de una tabla, según la condición WHERE o las dependencias de restricciones de clave foránea.
- **SELECT:** permite la selección de datos sobre la tabla. Consiste en elegir un subconjunto de filas que cumplan con algunas condiciones determinadas. También se pueden seleccionar todas las filas de una tabla.
- **COMMIT:** guarda los cambios de la transacción en curso (para guardar los datos insertados).
- **ROLLBACK:** deshace los cambios de la transacción en curso (para cancelar la inserción de datos.)

Las condiciones o CONSTRAINTS que afectan a una tabla son las siguientes:

- **NOT NULL:** especifica que la columna no puede almacenar valores nulos (vacíos).
- **UNIQUE:** especifica que el valor de la columna debe ser único para todas las filas en la tabla. No puede ser referenciada como foránea.
- **PRIMARY KEY:** identifica en forma única cada fila de la tabla y puede ser referenciada como foránea.
- **FOREIGN KEY (FK):** establece y hace cumplir una relación de clave foránea entre la columna de la tabla a crear y una columna de la tabla referenciada.
- **CHECK:** especifica una condición que debe ser verdadera para los valores que almacenará la columna.

Se sugiere comenzar con ejemplos sencillos de cada nueva función. Un ejemplo de condiciones podría ser:

Vendedor				Cliente		
N° (VNR)	Nombre (Vnombre)	Posición	Sueldo (U\$)	N° (CNR)	Nombre (Cnombre)	Supervisor
2011	Reyes	Junior	1 000	1	Astudillo	2011
2012	Espinoza	Senior	2 000	2	Martí	2012

Marco para programar en pseudocódigo:

- El número del cliente es una llave primaria.
- El cliente tiene nombre.
- El supervisor es una llave foránea para conectar la tabla del vendedor.

Posible programación para completar en pseudocódigo:

```
CREATE TABLE Cliente (
    CNR NÚMERO,
    Nombre VARCHAR (30)
    CONSTRAINT Cnombre NOT NULL,
    Supervisor NÚMERO,
    PRIMARY KEY (CNR),
    CONSTRAINT FK_Cliente
    FOREIGN KEY (Supervisor)
    REFERENCES Vendedor (VNR)
```

Actitudes

Para apoyar el desarrollo de las actividades es recomendable el uso de computadores, pues hay que agregar tablas con información a nivel informático, por tanto, es recomendable tener el *hardware* y *software* necesario para poder trabajar de manera óptima. En cuanto al desarrollo de la actividad en pares, se sugiere promover la ayuda entre estudiantes para conseguir resolver los problemas.

Orientaciones para organizar e implementar el trabajo entre pares

Se sugieren las siguientes indicaciones para guiar la implementación del trabajo entre pares:

- Elaborar un modelo de base de datos de entre 10 y 20 tablas.
- Cada estudiante deberá crear esas tablas mediante un Script en base al modelo que tendrán.
- Determinar el tiempo necesario para elaborar el Script.
- Determinar lo que se hará con los resultados. Si es necesario, se puede mostrar el Script de un o una estudiante y revisarlo con la clase para analizar posibles problemas y aciertos. Se sugiere conversar sobre las dudas que puedan haber con respecto a la creación de estos Scripts.

Actividad de desempeño 4

Propósito

Esta actividad pone el foco en el mundo de la consulta de base de datos. Transfiere los conocimientos y las habilidades para el manejo de las herramientas que se pueden utilizar al trabajar con estructuras informacionales complejas y precisas que permiten conseguir y aplicar soluciones de forma flexible a diferentes problemas.

Objetivos de aprendizaje

OA 4

Establecer conexiones y semejanzas entre las partes del problema para caracterizar patrones en una situación problemática dada. [\(Generalizar y transferir\)](#)

OA 5

Aplicar flexiblemente las soluciones elaboradas e implementadas a otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista. [\(Generalizar y transferir\)](#)

Conocimiento esencial

- Redes e Internet: sitios web, aplicaciones móviles, seguridad cibernética y objetos controlados digitalmente.
-

Tiempo estimado: 3 horas pedagógicas

Desarrollo de la actividad

SITUACIÓN EXPERIENCIAL

Se sugiere que la o el docente presente dos tablas iguales: una sin ninguna modificación y otra que está consultando datos precisos, complementarios y organizados.

EMPLOYEE_ID	NAME	EMAIL	JOB_TITLE	SALARY	MANAGER_ID	DEPARTMENT_NAME
1	100 Steven King	SKING	President	\$24,000.00	Sin Jefe	Executive
2	101 Neena Kochhar	NKOCHHAR	Administration Vice President	\$17,000.00	100	Executive
3	102 Lex De Haan	LDEHAAN	Administration Vice President	\$17,000.00	100	Executive
4	103 Alexander Hunold	AHUNOLD	Programmer	\$9,000.00	102	IT
5	104 Bruce Ernst	BERNST	Programmer	\$6,000.00	103	IT
6	105 David Austin	DAUSTIN	Programmer	\$4,800.00	103	IT
7	106 Valli Pataballa	VPATABAL	Programmer	\$4,800.00	103	IT
8	107 Diana Lorentz	DLORENTZ	Programmer	\$4,200.00	103	IT
9	108 Nancy Greenberg	NGREENBE	Finance Manager	\$12,008.00	101	Finance
10	109 Daniel Faviat	DFAVIET	Accountant	\$9,000.00	108	Finance

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	JOB_ID	SALARY	MANAGER_ID	DEPARTMENT_ID
1	100 Steven	King	SKING	AD_PRES	24000	(null)	90
2	101 Neena	Kochhar	NKOCHHAR	AD_VP	17000	100	90
3	102 Lex	De Haan	LDEHAAN	AD_VP	17000	100	90
4	103 Alexander	Hunold	AHUNOLD	IT_PROG	9000	102	60
5	104 Bruce	Ernst	BERNST	IT_PROG	6000	103	60
6	105 David	Austin	DAUSTIN	IT_PROG	4800	103	60
7	106 Valli	Pataballa	VPATABAL	IT_PROG	4800	103	60
8	107 Diana	Lorentz	DLORENTZ	IT_PROG	4200	103	60
9	108 Nancy	Greenberg	NGREENBE	FI_MGR	12008	101	100
10	109 Daniel	Faviat	DFAVIET	FI_ACCOUNT	9000	108	100

Algunas de las preguntas que podrían ayudar a distinguir la consulta de bases de datos podrían ser:

- ¿Para qué sirve la consulta de base de datos?
- ¿Cuál es la funcionalidad de ofrecer datos más específicos al usuario?

CONSTRUCCIÓN DE CONOCIMIENTO

Se sugiere que el o la docente presente a sus estudiantes un desafío en un contexto real, donde al utilizar la base de datos “HR” se necesiten las siguientes especificaciones para la consulta de base de datos, por ejemplo:

- Ordenar los datos según el nuevo sueldo de forma descendente.
- Mostrar el nombre y el apellido juntos.
- Hacer un nuevo salario dependiendo de los siguientes datos:
 - Si trabaja en “PR_REP” su salario será aumentado el 5%
 - Si trabaja en “SA_MAN” su salario será aumentado el 3%
 - Si trabaja en “IT_PROG” su salario será aumentado un 1%
- Hacer un ranking dependiendo del nuevo salario:
 - Si salario es inferior a 5 000 dólares, es un salario bajo.
 - Si salario está entre 5 000 y 10 000 dólares, es un salario medio.
 - Si salario es superior a 10 000 dólares, es un salario alto.
- La fecha debe tener el formato que se ve en la imagen.

EMPLOYEE_ID	NAME	JOB_ID	SALARY	NEW_SALARY	RANKING	MANAGER_ID	HIRE_DATE
1	145 John Russell	SA_MAN	\$14,000.00	\$14,420.00	ALTO	100	01 de octubre del 2004
2	146 Karen Partners	SA_MAN	\$13,500.00	\$13,905.00	ALTO	100	05 de enero del 2005
3	147 Alberto Errazuriz	SA_MAN	\$12,000.00	\$12,360.00	ALTO	100	10 de marzo del 2005
4	148 Gerald Cambrault	SA_MAN	\$11,000.00	\$11,330.00	ALTO	100	15 de octubre del 2007
5	149 Eleni Zlotkey	SA_MAN	\$10,500.00	\$10,815.00	ALTO	100	29 de enero del 2008
6	204 Hermann Baer	PR_REP	\$10,000.00	\$10,500.00	ALTO	101	07 de junio del 2002
7	103 Alexander Hunold	IT_PROG	\$9,000.00	\$9,900.00	MEDIO	102	03 de enero del 2006
8	104 Bruce Ernst	IT_PROG	\$6,000.00	\$6,600.00	MEDIO	103	21 de mayo del 2007
9	106 Valli Pataballa	IT_PROG	\$4,800.00	\$5,280.00	MEDIO	103	05 de febrero del 2006
10	105 David Austin	IT_PROG	\$4,800.00	\$5,280.00	MEDIO	103	25 de junio del 2005
11	107 Diana Lorentz	IT_PROG	\$4,200.00	\$4,620.00	BAJO	103	07 de febrero del 2007

Acción	Resultado esperado	Preguntas de apoyo
Organizar datos y primer avance de la consulta.	<pre>SELECT employee_id, first_name, last_name, job_ID, salary, manager_id, department_ID, hire_date FROM EMPLOYEES WHERE JOB_ID = 'PR_REP' OR JOB_ID = 'SA_MAN' OR JOB_ID = 'IT_PROG';</pre>	¿Por dónde empezar?

Acción	Resultado esperado	Preguntas de apoyo
Desarrollar el “WHEN - CASE” relacionado con la dependencia del trabajo para identificar el aumento del trabajo.	<pre>SELECT employee_id, first_name, last_name, job_ID, salary, CASE JOB_ID WHEN 'PR_REP' THEN SALARY+SALARY*0.05 WHEN 'SA_MAN' THEN SALARY+SALARY*0.03 WHEN 'IT_PROG' THEN SALARY+SALARY*0.1 ELSE SALARY END NEW_SALARY, manager_id, department_ID, hire_date FROM EMPLOYEES WHERE JOB_ID = 'PR_REP' OR JOB_ID = 'SA_MAN' OR JOB_ID = 'IT_PROG';</pre>	
Desarrollar el segundo “WHEN - CASE” para la clasificación del nuevo salario.	<pre>SELECT employee_id, first_name, last_name, job_ID, salary, CASE JOB_ID WHEN 'PR_REP' THEN SALARY+SALARY*0.05 WHEN 'SA_MAN' THEN SALARY+SALARY*0.03 WHEN 'IT_PROG' THEN SALARY+SALARY*0.1 ELSE SALARY END NEW_SALARY, CASE WHEN DECODE(JOB_ID, 'PR_REP', SALARY+SALARY*0.05, 'SA_MAN', SALARY+SALARY*0.03, 'IT_PROG', SALARY+SALARY*0.1, SALARY) < 5000 THEN 'BAJO' WHEN DECODE(JOB_ID, 'PR_REP', SALARY+SALARY*0.05, 'SA_MAN', SALARY+SALARY*0.03, 'IT_PROG', SALARY+SALARY*0.1, SALARY) BETWEEN 5000 AND 10000 THEN 'MEDIO' WHEN DECODE(JOB_ID, 'PR_REP', SALARY+SALARY*0.05, 'SA_MAN', SALARY+SALARY*0.03, 'IT_PROG', SALARY+SALARY*0.1, SALARY) > 10000 THEN 'ALTO' ELSE ' ' END RANKING, manager_id, department_ID, hire_date FROM EMPLOYEES WHERE JOB_ID = 'PR_REP' OR JOB_ID = 'SA_MAN' OR JOB_ID = 'IT_PROG';</pre>	¿Cómo clasifico el nuevo salario?
Últimos detalles y finalización de la consulta.	<pre>SELECT employee_id, first_name ' ' last_name NAME, JOB_ID, TO_CHAR (SALARY, '\$99,999.00') SALARY, CASE JOB_ID WHEN 'PR_REP' THEN TO_CHAR (SALARY+SALARY*0.05, '\$99,999.00') WHEN 'SA_MAN' THEN TO_CHAR (SALARY+SALARY*0.03, '\$99,999.00') WHEN 'IT_PROG' THEN TO_CHAR (SALARY+SALARY*0.1, '\$99,999.00') ELSE TO_CHAR (SALARY, '\$99,999.00') END NEW_SALARY, CASE WHEN DECODE(JOB_ID, 'PR_REP', SALARY+SALARY*0.05, 'SA_MAN', SALARY+SALARY*0.03, 'IT_PROG', SALARY+SALARY*0.1, SALARY) < 5000 THEN 'BAJO' WHEN DECODE(JOB_ID, 'PR_REP', SALARY+SALARY*0.05, 'SA_MAN', SALARY+SALARY*0.03, 'IT_PROG', SALARY+SALARY*0.1, SALARY) BETWEEN 5000 AND 10000 THEN 'MEDIO' WHEN DECODE(JOB_ID, 'PR_REP', SALARY+SALARY*0.05, 'SA_MAN', SALARY+SALARY*0.03, 'IT_PROG', SALARY+SALARY*0.1, SALARY) > 10000 THEN 'ALTO' ELSE ' ' END RANKING, NVL (TO_CHAR(MANAGER_ID), 'SIN JEFE') MANAGER_ID, TO_CHAR (HIRE_DATE, 'dd "de" month "del" yyyy') HIRE_DATE FROM EMPLOYEES WHERE JOB_ID = 'PR_REP' OR JOB_ID = 'SA_MAN' OR JOB_ID = 'IT_PROG' ORDER BY NEW_SALARY DESC;</pre>	

PRÁCTICA GUIADA

En esta práctica, se sugiere que la o el docente haga varias demostraciones de cómo usar ciertos elementos de la consulta de base de datos en los cuales priorizan el uso del JOIN:

Acción	Resultado esperado																																																
Mostrar para cada empleado en qué área de trabajo se desempeña, el departamento correspondiente y la ciudad en la que se encuentra.	<pre>SELECT FIRST_NAME, LAST_NAME, D.DEPARTMENT_NAME, J.JOB_TITLE, J.MAX_SALARY, E.SALARY, L.CITY, L.COUNTRY_ID FROM EMPLOYEES E JOIN DEPARTMENTS D ON D.DEPARTMENT_ID=E.DEPARTMENT_ID JOIN JOBS J ON J.JOB_ID=E.JOB_ID JOIN LOCATIONS L ON L.LOCATION_ID=D.LOCATION_ID;</pre>																																																
Resultado esperado.	<table><tr><th>FIRST_NAME</th><th>LAST_NAME</th><th>DEPARTMENT_NAME</th><th>JOB_TITLE</th><th>MAX_SALARY</th><th>SALARY</th><th>CITY</th><th>COUNTRY_ID</th></tr><tr><td>Steven</td><td>King</td><td>Executive</td><td>President</td><td>40000</td><td>24000</td><td>Seattle</td><td>US</td></tr><tr><td>Neena</td><td>Kochhar</td><td>Executive</td><td>Administration Vice President</td><td>30000</td><td>17000</td><td>Seattle</td><td>US</td></tr><tr><td>Lex</td><td>De Haan</td><td>Executive</td><td>Administration Vice President</td><td>30000</td><td>17000</td><td>Seattle</td><td>US</td></tr><tr><td>Jennifer</td><td>Whalen</td><td>Administration</td><td>Administration Assistant</td><td>6000</td><td>4400</td><td>Seattle</td><td>US</td></tr><tr><td>Nancy</td><td>Greenberg</td><td>Finance</td><td>Finance Manager</td><td>16000</td><td>12008</td><td>Seattle</td><td>US</td></tr></table>	FIRST_NAME	LAST_NAME	DEPARTMENT_NAME	JOB_TITLE	MAX_SALARY	SALARY	CITY	COUNTRY_ID	Steven	King	Executive	President	40000	24000	Seattle	US	Neena	Kochhar	Executive	Administration Vice President	30000	17000	Seattle	US	Lex	De Haan	Executive	Administration Vice President	30000	17000	Seattle	US	Jennifer	Whalen	Administration	Administration Assistant	6000	4400	Seattle	US	Nancy	Greenberg	Finance	Finance Manager	16000	12008	Seattle	US
FIRST_NAME	LAST_NAME	DEPARTMENT_NAME	JOB_TITLE	MAX_SALARY	SALARY	CITY	COUNTRY_ID																																										
Steven	King	Executive	President	40000	24000	Seattle	US																																										
Neena	Kochhar	Executive	Administration Vice President	30000	17000	Seattle	US																																										
Lex	De Haan	Executive	Administration Vice President	30000	17000	Seattle	US																																										
Jennifer	Whalen	Administration	Administration Assistant	6000	4400	Seattle	US																																										
Nancy	Greenberg	Finance	Finance Manager	16000	12008	Seattle	US																																										
Mostrar los departamentos que no tienen empleados.	<pre>SELECT FIRST_NAME, LAST_NAME, D.DEPARTMENT_NAME FROM EMPLOYEES E RIGHT OUTER JOIN DEPARTMENTS D ON D.DEPARTMENT_ID=E.DEPARTMENT_ID WHERE E.EMPLOYEE_ID IS NULL;</pre>																																																
Resultado esperado.	<table><tr><th>FIRST_NAME</th><th>LAST_NAME</th><th>DEPARTMENT_NAME</th></tr><tr><td>(null)</td><td>(null)</td><td>Treasury</td></tr><tr><td>(null)</td><td>(null)</td><td>Corporate Tax</td></tr><tr><td>(null)</td><td>(null)</td><td>Control And Credit</td></tr><tr><td>(null)</td><td>(null)</td><td>Shareholder Services</td></tr></table>	FIRST_NAME	LAST_NAME	DEPARTMENT_NAME	(null)	(null)	Treasury	(null)	(null)	Corporate Tax	(null)	(null)	Control And Credit	(null)	(null)	Shareholder Services																																	
FIRST_NAME	LAST_NAME	DEPARTMENT_NAME																																															
(null)	(null)	Treasury																																															
(null)	(null)	Corporate Tax																																															
(null)	(null)	Control And Credit																																															
(null)	(null)	Shareholder Services																																															
Agregar la tabla rango de sueldo y trabajar mediante estos datos.	<pre>CREATE TABLE RANGO_SUELDO(ID_RS INTEGER NOT NULL PRIMARY KEY, RANGO_MIN NUMBER, RANGO_MAX NUMBER, DESCRIPCION VARCHAR2(50)); INSERT INTO RANGO_SUELDO VALUES (1,0,3999,'BAJO'); INSERT INTO RANGO_SUELDO VALUES (2,4000,6999,'MEDIO BAJO'); INSERT INTO RANGO_SUELDO VALUES (3,7000,9999,'MEDIO'); INSERT INTO RANGO_SUELDO VALUES (4,10000,40000,'ALTO'); SELECT FIRST_NAME, LAST_NAME, SALARY, RS.DESCRIPCION FROM EMPLOYEES E JOIN RANGO_SUELDO RS ON E.SALARY BETWEEN RS.RANGO_MIN AND RS.RANGO_MAX ORDER BY E.EMPLOYEE_ID;</pre>																																																

Acción	Resultado esperado																																
Resultado esperado.	<table><tr><th>FIRST_NAME</th><th>LAST_NAME</th><th>SALARY</th><th>DESCRIPCION</th></tr><tr><td>Steven</td><td>King</td><td>24000</td><td>ALTO</td></tr><tr><td>Neena</td><td>Kochhar</td><td>17000</td><td>ALTO</td></tr><tr><td>Lex</td><td>De Haan</td><td>17000</td><td>ALTO</td></tr><tr><td>Alexander</td><td>Hunold</td><td>9000</td><td>MEDIO</td></tr><tr><td>Bruce</td><td>Ernst</td><td>6000</td><td>MEDIO BAJO</td></tr><tr><td>David</td><td>Austin</td><td>4800</td><td>MEDIO BAJO</td></tr><tr><td>Valli</td><td>Pataballa</td><td>4800</td><td>MEDIO BAJO</td></tr></table>	FIRST_NAME	LAST_NAME	SALARY	DESCRIPCION	Steven	King	24000	ALTO	Neena	Kochhar	17000	ALTO	Lex	De Haan	17000	ALTO	Alexander	Hunold	9000	MEDIO	Bruce	Ernst	6000	MEDIO BAJO	David	Austin	4800	MEDIO BAJO	Valli	Pataballa	4800	MEDIO BAJO
FIRST_NAME	LAST_NAME	SALARY	DESCRIPCION																														
Steven	King	24000	ALTO																														
Neena	Kochhar	17000	ALTO																														
Lex	De Haan	17000	ALTO																														
Alexander	Hunold	9000	MEDIO																														
Bruce	Ernst	6000	MEDIO BAJO																														
David	Austin	4800	MEDIO BAJO																														
Valli	Pataballa	4800	MEDIO BAJO																														

**PRÁCTICA
INDEPENDIENTE**

Se recomienda favorecer un trabajo individual, en el que se proponga un desafío en un contexto real, donde el uso de una base de datos requiera definir especificaciones para realizar una consulta. También, se puede ocupar en muchos trabajos donde se deba hacer una o dos cosas para completar la consulta (se sugiere trabajar en una base de datos nueva, con nuevas tablas e información).

Por ejemplo, se puede trabajar en mostrar solamente datos de quiénes nacieron en algún mes o año, dar un formato numérico en específico o agregar nuevas tablas, etc.

Pueden registrar el procedimiento de instrucciones en una hoja de trabajo de códigos. En el anexo podrán encontrar la hoja para escribir en modo de zona de escritura de código.

**EVALUACIÓN
FORMATIVA**

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Consulta de base de datos	Genera Script.	Genera Script de manera correcta, pero algunos datos no se muestran cómo deberían.	Genera Script con todo lo solicitado dentro del problema.
Uso de herramientas	Usa herramientas que no son necesarias.	Usa las herramientas que son necesarias, pero de manera ineficiente.	Usa las herramientas necesarias y consigue generar un Script eficiente.

Orientaciones al docente

Para unificar conceptos disciplinares

Como la consulta de base de datos es muy específica al momento de hacer ejercicios, siempre se debe enseñar cómo funcionan los comandos. De esta manera, el estudiantado entenderá el concepto principal y el uso que se les puede dar. Se está trabajando con la base de datos de Oracle, específicamente la de recursos humanos “HR”, pues es una base de datos que se puede desbloquear y probar siempre que se use Oracle. A partir de estas herramientas, se puede tener una base sólida para trabajar con las demás herramientas que necesitan más conceptualización y, a su vez, son de una dificultad elevada para entender todas las funcionalidades u ocasiones en donde se podrían utilizar.

Actitudes

Para apoyar el desarrollo de las actividades, se recomienda el uso de computadores, ya que trabajos con información precisa deben ser evaluados constantemente para poder apreciar los cambios y errores que pudieran ocurrir. La actividad individual tiene como objetivo favorecer que estudiantado tenga autonomía y seguridad para resolver los ejercicios en contextos de aplicación real.

Orientaciones para organizar e implementar el trabajo individual

Se sugieren las siguientes indicaciones para guiar el trabajo individual.

- Elaborar un ejercicio amplio, como en la construcción de conocimiento, o implementar varios ejercicios con dificultad moderada donde se solicite entre uno a dos cambios.
- Determinar un tiempo específico para poder realizar las actividades.
- Dar retroalimentación del ejercicio.
- Favorecer la reflexión, evaluación y generalizar lo aprendido durante la actividad y módulo.

Visión panorámica

Gran idea

El pensamiento algorítmico permite organizar pasos y reglas para diseñar soluciones y resolver una diversidad de problemas cuya comprensión se favorece al integrar conceptos matemáticos como variables, funciones, ecuaciones o gráficos, entre otros.

Objetivos de aprendizaje

OA 4

Establecer conexiones y semejanzas entre las partes del problema para caracterizar patrones en una situación problemática dada. [\(Generalizar y Transferir\)](#)

OA 5

Aplicar flexiblemente las soluciones elaboradas e implementadas en otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista. [\(Generalizar y Transferir\)](#)

OA 6

Elaborar e implementar soluciones a una situación problemática mediante la formulación de algoritmos, actuando de acuerdo con los principios de la ética. [\(Pensar de forma algorítmica\)](#)

Conocimientos esenciales

- Algoritmos y programación: lenguajes, métodos y entornos de programación.
 - Representación y análisis de datos.
 - Impacto social de la computación: interacciones, cultura, seguridad y ética.
-

Tiempo estimado: 6 semanas (12 horas)

Propósito

En el módulo 4 de la asignatura Pensamiento Computacional de los Niveles 1 y 2 de Educación Media, se espera que las y los estudiantes puedan pensar de forma algorítmica para organizar los pasos que les permitan resolver uno o más problemas, e integrar conceptos matemáticos como variables, funciones, ecuaciones o gráficos. Asimismo, favorece la capacidad de evaluar y seleccionar algoritmos en función de su rendimiento, reutilización y la facilidad de implementarlos, pues un algoritmo óptimo es aquel que es eficiente en trabajar, economiza tiempo y reduce la redundancia, indistintamente del lenguaje de programación que se utilice.

Sobre este escenario, el módulo 4 pone mayor énfasis en el desarrollo del pensamiento algorítmico a través de procesos de generalización y transferencia de soluciones a problemas que requieran el uso de algoritmos eficientes o necesiten una mayor eficiencia y optimización en su estructura. Esto permitirá a los y las estudiantes utilizar las bases del algoritmo para crear códigos funcionales dentro de un lenguaje de programación. Se espera que tengan experiencias en diferentes entornos de programación al

momento de analizar y representar datos en un ambiente digital, y avancen hacia la comprensión del impacto social que tiene la computación en las interacciones de las personas a nivel local y global, en la cultura, en la seguridad y en el uso ético de la misma.

Los Objetivos de Aprendizaje del módulo desarrollan las actitudes del siglo XXI del ámbito Maneras de Trabajar y Herramientas para trabajar, promoviendo el uso provechoso de las herramientas para dar soluciones a problemáticas cotidianas en ambientes digitales, acorde con principios éticos. Asimismo, el conocimiento asociado a la alfabetización en tecnologías digitales, desde la perspectiva de la programación, permitirá a los y las estudiantes acceder y manejar información desde distintas fuentes, para poder decidir cuál y cómo usarla. Finalmente, las actividades de desempeño propuestas esperan consolidar procesos metacognitivos en cuanto al pensamiento computacional, el procesamiento de datos y la entrega de soluciones a problemas que existen en el entorno cotidiano.

Ruta de Aprendizaje

¿Por qué las personas evalúan y seleccionan algoritmos en función del rendimiento, la reutilización y la facilidad de implementación para resolver un problema?

	Actividad de desempeño 1	Evaluar las herramientas necesarias para poder trabajar con información requerida de forma eficiente y precisa.
	Actividad de desempeño 2	Implementar algoritmos que permitan trabajar con los datos necesarios para lograr obtener una solución a una problemática específica.
	Actividad de desempeño 3	Aplicar un proceso lógico que vincule el sistema de un algoritmo con un lenguaje de programación determinando.
	Actividad de desempeño 4	Elaborar e implementar una variedad de algoritmos para resolver un problema con un lenguaje de programación profesional.

Actividad de desempeño

1

Propósito

Esta actividad desafía a los y las estudiantes a trabajar mediante la conexión o relación que existen entre las tablas que conforman una base de datos con el fin de extender las herramientas, evaluarlas y aumentar la comprensión de la información almacenada en ellas. Asimismo, deberán trabajar con información requerida de forma eficiente y precisa, y aplicar funcionalidades de consulta y representación de datos.

Objetivos de aprendizaje

OA 4

Establecer conexiones y semejanzas entre las partes del problema para caracterizar patrones en una situación problemática dada. ([Generalizar y transferir](#))

Conocimiento esencial

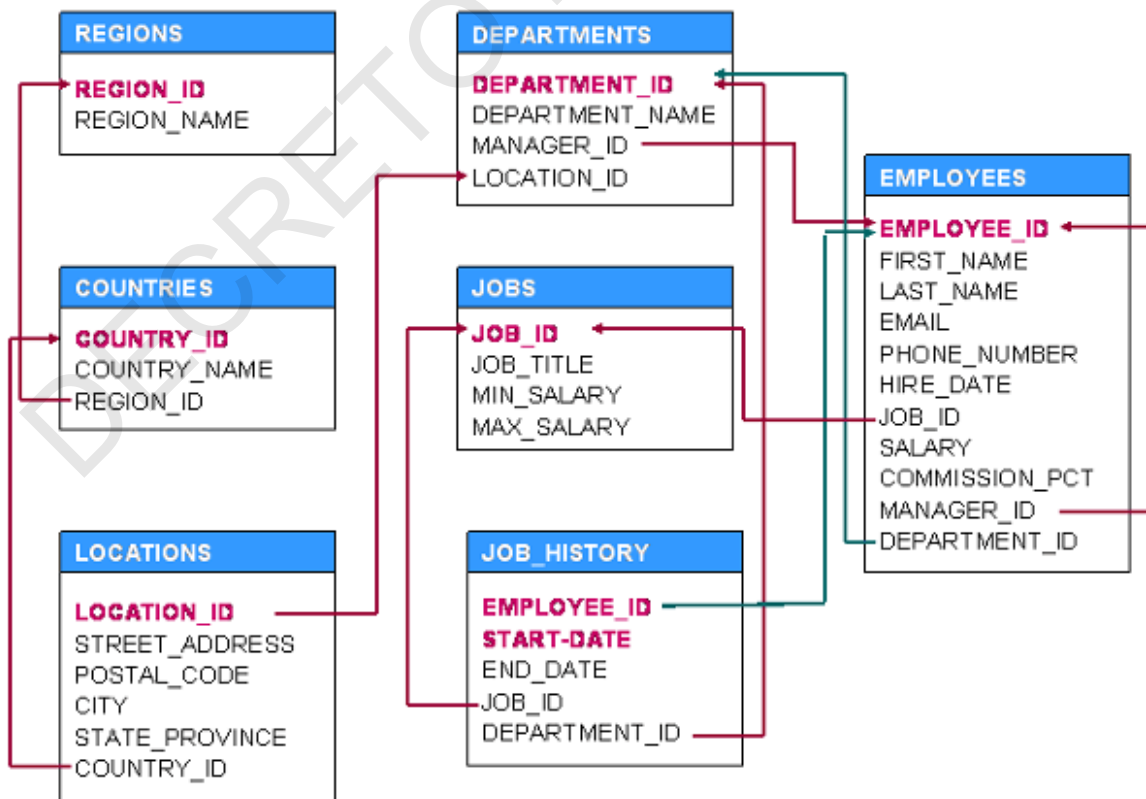
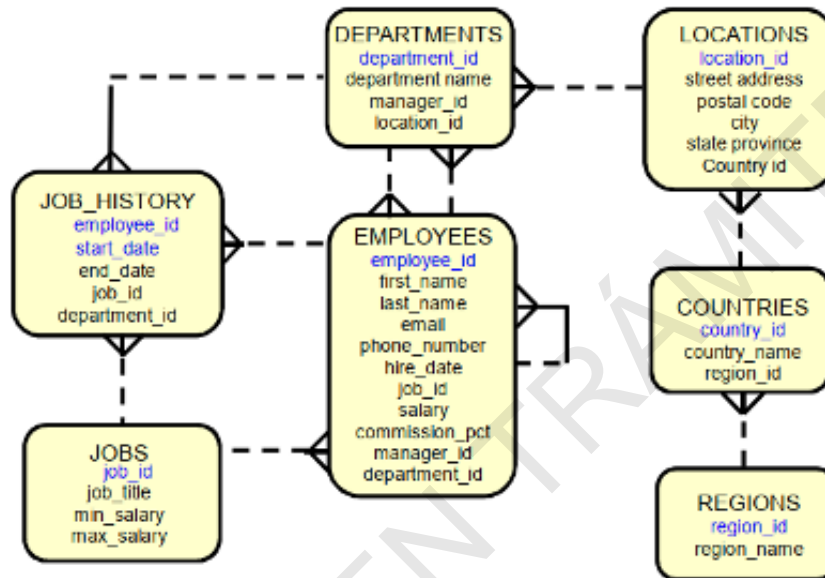
- Representación y análisis de datos.

Tiempo estimado: 3 horas pedagógicas

Desarrollo de la actividad

SITUACIÓN EXPERIENCIAL

Se sugiere que la o el docente presente dos modelos de datos entidad-relación que den cuenta de las diferentes formas de explicar un modelo de base de datos que proporciona Oracle en su servicio, llamada “HR” (Recursos Humanos).



Algunas de las preguntas que podrían promover procesos de pensamiento asociados a comunicar la importancia de las relaciones entre tablas podrían ser:

- ¿Qué es lo que explica cada modelo?
- ¿Cuál es la importancia de las relaciones en la consulta de base de datos?
- ¿Por qué es útil tener claros los datos de una base de datos?

CONSTRUCCIÓN DE CONOCIMIENTO

Para profundizar en el objetivo de la actividad, se sugiere proponer a los y las estudiantes utilizar la base de datos “HR” y dar las siguientes especificaciones para esta consulta de base de datos:

- Mostrar el nombre y el apellido juntos.
- Hacer una descripción de la comisión y darle las opciones:
 - ▶ Si es “nulo”, la opción será “Solo sueldo”.
 - ▶ Si tiene un valor, la opción será “Sueldo + Comisión”.
- Se debe mostrar la fecha de contratación y del término del contrato.
- Crear una categoría donde se vean los días, meses y años trabajados.
- El formato de las fechas y el sueldo deben ser el mismo que el de la imagen.
- El resultado final debe mostrar la misma cantidad de datos e información.

ID EMPLEADO	EMPLEADO	SUELDO	COMISION	COMISION DESCRIPCION	CARGO	FECHA DE CONTRATACION	FECHA DE TERMINO	DÍAS / MESES / AÑOS CONTRATADOS
1	101 Neena Kochhar	\$17,000	0	Solo Sueldo	Public Accountant	21 de 09 del 1997	27 de 10 del 2001	1497 / 49 / 4
2	101 Neena Kochhar	\$17,000	0	Solo Sueldo	Accounting Manager	28 de 10 del 2001	15 de 03 del 2005	1234 / 41 / 3
3	102 Lex De Haan	\$17,000	0	Solo Sueldo	Programmer	13 de 01 del 2001	24 de 07 del 2006	2018 / 66 / 6
4	114 Den Raphaely	\$11,000	0	Solo Sueldo	Stock Clerk	24 de 03 del 2006	31 de 12 del 2007	647 / 21 / 2
5	122 Payam Kaufling	\$7,900	0	Solo Sueldo	Stock Clerk	01 de 01 del 2007	31 de 12 del 2007	364 / 12 / 1
6	176 Jonathon Taylor	\$8,600	0,2	Sueldo + Comision	Sales Manager	01 de 01 del 2007	31 de 12 del 2007	364 / 12 / 1
7	176 Jonathon Taylor	\$8,600	0,2	Sueldo + Comision	Sales Representative	24 de 03 del 2006	31 de 12 del 2006	282 / 9 / 1
8	200 Jennifer Whalen	\$4,400	0	Solo Sueldo	Public Accountant	01 de 07 del 2002	31 de 12 del 2006	1644 / 54 / 5
9	200 Jennifer Whalen	\$4,400	0	Solo Sueldo	Administration Assistant	17 de 09 del 1995	17 de 06 del 2001	2100 / 69 / 6
10	201 Michael Hartstein	\$13,000	0	Solo Sueldo	Marketing Representative	17 de 02 del 2004	19 de 12 del 2007	1401 / 46 / 4

Acción	Resultado esperado
Verificar datos a utilizar.	<pre> SELECT * FROM EMPLOYEES; SELECT * FROM JOBS; SELECT * FROM JOB_HISTORY; </pre>
Armar el primer paso.	<pre> SELECT E.EMPLOYEE_ID "ID EMPLEADO", E.FIRST_NAME ' ' E.LAST_NAME EMPLEADO, E.SALARY SUELDO, JH.START_DATE "FECHA DE CONTRATACION", JH.END_DATE "FECHA DE TERMINO" FROM EMPLOYEES E RIGHT JOIN JOB_HISTORY JH ON JH.EMPLOYEE_ID = E.EMPLOYEE_ID; </pre>
Agregar los datos faltantes.	<pre> SELECT E.EMPLOYEE_ID "ID EMPLEADO", E.FIRST_NAME ' ' E.LAST_NAME EMPLEADO, E.SALARY SUELDO, NVL(E.COMMISSION_PCT,0) COMISION, NVL2(E.COMMISSION_PCT,'Sueldo + Comision','Solo Sueldo') "COMISION DESCRIPCION", J.JOB_TITLE CARGO, JH.START_DATE "FECHA DE CONTRATACION", JH.END_DATE "FECHA DE TERMINO", ROUND(JH.END_DATE-JH.START_DATE) ' / ' ROUND(MONTHS_BETWEEN(JH.END_DATE,JH.START_DATE)) ' / ' ROUND((JH.END_DATE-JH.START_DATE)/365) "DÍAS / MESES / AÑOS CONTRATADOS" FROM EMPLOYEES E RIGHT JOIN JOB_HISTORY JH ON JH.EMPLOYEE_ID = E.EMPLOYEE_ID JOIN JOBS J ON J.JOB_ID=JH.JOB_ID; </pre>
Revisar los últimos detalles.	<pre> SELECT E.EMPLOYEE_ID "ID EMPLEADO", E.FIRST_NAME ' ' E.LAST_NAME EMPLEADO, TO_CHAR(E.SALARY,'\$99,999') SUELDO, NVL(E.COMMISSION_PCT,0) COMISION, NVL2(E.COMMISSION_PCT,'Sueldo + Comision','Solo Sueldo') "COMISION DESCRIPCION", J.JOB_TITLE CARGO, TO_CHAR (JH.START_DATE, 'dd "de" mm "del" yyyy') "FECHA DE CONTRATACION", TO_CHAR (JH.END_DATE, 'dd "de" mm "del" yyyy') "FECHA DE TERMINO", ROUND(JH.END_DATE-JH.START_DATE) ' / ' ROUND(MONTHS_BETWEEN(JH.END_DATE,JH.START_DATE)) ' / ' ROUND((JH.END_DATE-JH.START_DATE)/365) "DÍAS / MESES / AÑOS CONTRATADOS" FROM EMPLOYEES E RIGHT JOIN JOB_HISTORY JH ON JH.EMPLOYEE_ID = E.EMPLOYEE_ID JOIN JOBS J ON J.JOB_ID=JH.JOB_ID; </pre>

PRÁCTICA GUIADA

Para guiar el proceso de incorporar nuevos conocimientos al proceso de consulta de datos, se sugiere guiar a las y los estudiantes para trabajar con JOIN y GROUP BY.

Acción	Resultado esperado																																																																		
Mostrar en qué departamentos trabajan los empleados y se muestre organizado por la dirección del departamento.	<pre>SELECT E.EMPLOYEE_ID, E.FIRST_NAME ' ' E.LAST_NAME NAME, D.DEPARTMENT_NAME, L.CITY, L.STREET_ADDRESS, C.COUNTRY_NAME FROM EMPLOYEES E JOIN DEPARTMENTS D ON D.DEPARTMENT_ID = E.DEPARTMENT_ID JOIN LOCATIONS L ON L.LOCATION_ID = D.LOCATION_ID JOIN COUNTRIES C ON C.COUNTRY_ID = L.COUNTRY_ID ORDER BY L.STREET_ADDRESS;</pre>																																																																		
Resultado esperado.	<table><thead><tr><th>EMPLOYEE_ID</th><th>NAME</th><th>DEPARTMENT_NAME</th><th>CITY</th><th>STREET_ADDRESS</th><th>COUNTRY_NAME</th></tr></thead><tbody><tr><td>1</td><td>145 John Russell</td><td>Sales</td><td>Oxford</td><td>Magdalen Centre, The Oxford Science Park</td><td>United Kingdom</td></tr><tr><td>2</td><td>146 Karen Partners</td><td>Sales</td><td>Oxford</td><td>Magdalen Centre, The Oxford Science Park</td><td>United Kingdom</td></tr><tr><td>3</td><td>179 Charles Johnson</td><td>Sales</td><td>Oxford</td><td>Magdalen Centre, The Oxford Science Park</td><td>United Kingdom</td></tr><tr><td>4</td><td>177 Jack Livingston</td><td>Sales</td><td>Oxford</td><td>Magdalen Centre, The Oxford Science Park</td><td>United Kingdom</td></tr><tr><td>5</td><td>147 Alberto Errazuriz</td><td>Sales</td><td>Oxford</td><td>Magdalen Centre, The Oxford Science Park</td><td>United Kingdom</td></tr><tr><td>6</td><td>148 Gerald Cambrault</td><td>Sales</td><td>Oxford</td><td>Magdalen Centre, The Oxford Science Park</td><td>United Kingdom</td></tr><tr><td>7</td><td>149 Eleni Zlotkey</td><td>Sales</td><td>Oxford</td><td>Magdalen Centre, The Oxford Science Park</td><td>United Kingdom</td></tr><tr><td>8</td><td>150 Peter Tucker</td><td>Sales</td><td>Oxford</td><td>Magdalen Centre, The Oxford Science Park</td><td>United Kingdom</td></tr><tr><td>9</td><td>151 David Bernstein</td><td>Sales</td><td>Oxford</td><td>Magdalen Centre, The Oxford Science Park</td><td>United Kingdom</td></tr><tr><td>10</td><td>152 Peter Hall</td><td>Sales</td><td>Oxford</td><td>Magdalen Centre, The Oxford Science Park</td><td>United Kingdom</td></tr></tbody></table>	EMPLOYEE_ID	NAME	DEPARTMENT_NAME	CITY	STREET_ADDRESS	COUNTRY_NAME	1	145 John Russell	Sales	Oxford	Magdalen Centre, The Oxford Science Park	United Kingdom	2	146 Karen Partners	Sales	Oxford	Magdalen Centre, The Oxford Science Park	United Kingdom	3	179 Charles Johnson	Sales	Oxford	Magdalen Centre, The Oxford Science Park	United Kingdom	4	177 Jack Livingston	Sales	Oxford	Magdalen Centre, The Oxford Science Park	United Kingdom	5	147 Alberto Errazuriz	Sales	Oxford	Magdalen Centre, The Oxford Science Park	United Kingdom	6	148 Gerald Cambrault	Sales	Oxford	Magdalen Centre, The Oxford Science Park	United Kingdom	7	149 Eleni Zlotkey	Sales	Oxford	Magdalen Centre, The Oxford Science Park	United Kingdom	8	150 Peter Tucker	Sales	Oxford	Magdalen Centre, The Oxford Science Park	United Kingdom	9	151 David Bernstein	Sales	Oxford	Magdalen Centre, The Oxford Science Park	United Kingdom	10	152 Peter Hall	Sales	Oxford	Magdalen Centre, The Oxford Science Park	United Kingdom
EMPLOYEE_ID	NAME	DEPARTMENT_NAME	CITY	STREET_ADDRESS	COUNTRY_NAME																																																														
1	145 John Russell	Sales	Oxford	Magdalen Centre, The Oxford Science Park	United Kingdom																																																														
2	146 Karen Partners	Sales	Oxford	Magdalen Centre, The Oxford Science Park	United Kingdom																																																														
3	179 Charles Johnson	Sales	Oxford	Magdalen Centre, The Oxford Science Park	United Kingdom																																																														
4	177 Jack Livingston	Sales	Oxford	Magdalen Centre, The Oxford Science Park	United Kingdom																																																														
5	147 Alberto Errazuriz	Sales	Oxford	Magdalen Centre, The Oxford Science Park	United Kingdom																																																														
6	148 Gerald Cambrault	Sales	Oxford	Magdalen Centre, The Oxford Science Park	United Kingdom																																																														
7	149 Eleni Zlotkey	Sales	Oxford	Magdalen Centre, The Oxford Science Park	United Kingdom																																																														
8	150 Peter Tucker	Sales	Oxford	Magdalen Centre, The Oxford Science Park	United Kingdom																																																														
9	151 David Bernstein	Sales	Oxford	Magdalen Centre, The Oxford Science Park	United Kingdom																																																														
10	152 Peter Hall	Sales	Oxford	Magdalen Centre, The Oxford Science Park	United Kingdom																																																														
Mostrar cuántos empleados tiene cada departamento.	<pre>SELECT E.DEPARTMENT_ID,D.DEPARTMENT_NAME,COUNT(*) FROM EMPLOYEES E JOIN DEPARTMENTS D ON D.DEPARTMENT_ID = E.DEPARTMENT_ID GROUP BY E.DEPARTMENT_ID, D.DEPARTMENT_NAME ORDER BY E.DEPARTMENT_ID;</pre>																																																																		
Resultado esperado.	<table><thead><tr><th>DEPARTMENT_ID</th><th>DEPARTMENT_NAME</th><th>COUNT(*)</th></tr></thead><tbody><tr><td>1</td><td>10 Administration</td><td>1</td></tr><tr><td>2</td><td>20 Marketing</td><td>2</td></tr><tr><td>3</td><td>30 Purchasing</td><td>6</td></tr><tr><td>4</td><td>40 Human Resources</td><td>1</td></tr><tr><td>5</td><td>50 Shipping</td><td>45</td></tr><tr><td>6</td><td>60 IT</td><td>5</td></tr><tr><td>7</td><td>70 Public Relations</td><td>1</td></tr><tr><td>8</td><td>80 Sales</td><td>34</td></tr><tr><td>9</td><td>90 Executive</td><td>3</td></tr><tr><td>10</td><td>100 Finance</td><td>6</td></tr><tr><td>11</td><td>110 Accounting</td><td>2</td></tr></tbody></table>	DEPARTMENT_ID	DEPARTMENT_NAME	COUNT(*)	1	10 Administration	1	2	20 Marketing	2	3	30 Purchasing	6	4	40 Human Resources	1	5	50 Shipping	45	6	60 IT	5	7	70 Public Relations	1	8	80 Sales	34	9	90 Executive	3	10	100 Finance	6	11	110 Accounting	2																														
DEPARTMENT_ID	DEPARTMENT_NAME	COUNT(*)																																																																	
1	10 Administration	1																																																																	
2	20 Marketing	2																																																																	
3	30 Purchasing	6																																																																	
4	40 Human Resources	1																																																																	
5	50 Shipping	45																																																																	
6	60 IT	5																																																																	
7	70 Public Relations	1																																																																	
8	80 Sales	34																																																																	
9	90 Executive	3																																																																	
10	100 Finance	6																																																																	
11	110 Accounting	2																																																																	

PRÁCTICA
INDEPENDIENTE

Se recomienda un trabajo individual en el que cada estudiante trabaje con ejercicios (máximo 5) donde aplique las herramientas aprendidas en los modelos. Pero esta vez con el desafío de incorporar JOIN en las bases de datos creadas por ellos mismos y considerar situaciones cotidianas en las que el uso de las bases de datos es un facilitador de tareas en diferentes ámbitos de actividad humana, como el transporte, la salud o las comunicaciones. Para guiar al estudiantado, se puede incorporar la resolución de problemas de manera algorítmica a partir de bases de datos.

Pueden registrar el procedimiento de instrucciones en una hoja de trabajo de códigos. En el anexo podrán encontrar la hoja para escribir en modo de zona de escritura de código.

EVALUACIÓN FORMATIVA

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Consulta de base de datos	Genera Script con errores.	Genera Script de manera correcta, pero no se visualizan algunos datos.	Genera Script de manera correcta, visualizando todos los datos.
Herramienta JOIN	Usa la herramienta JOIN con errores.	Usa la herramienta JOIN, pero de manera ineficiente.	Usa la herramienta JOIN de manera comprensible y eficiente.

Orientaciones al docente

Para unificar conceptos disciplinares

Para poder fortalecer los conocimientos de los y las estudiantes, lo primordial es la práctica mediante ejercicios. Se recomienda el uso de nuevas bases de datos y no solo utilizar la anterior mencionada "HR", pues las consultas de base de datos nos sirven para poder conectar diferentes elementos que proporciona la base de datos, estos son la comunicación entre tablas para mantener la información lo más fiel que se pueda.

El concepto inicial del JOIN es la comunicación. Los datos guardados en una tabla a veces pueden ser códigos que tengan su significado en otra tabla. El usuario preferirá ver la información real antes de un código y tener que ir a revisar qué significa ese código como tal. Para eso sirve primordialmente esta herramienta.

Una cláusula JOIN se utiliza para combinar filas de dos o más tablas, según una columna relacionada entre ellas.

Actitudes

Para apoyar el desarrollo de las actividades, se recomienda el uso de computadores, pues los trabajos que se realizan requieren de una conexión directa a una base de datos con la información que se trabajará. En cuanto al desarrollo de la actividad individual, es necesario el computador para que cada estudiante tenga autonomía y confianza a la hora de querer resolver problemas.

Orientaciones para organizar e implementar el trabajo individual

Se sugieren las siguientes indicaciones para guiar el trabajo individual:

- Elaborar un ejercicio amplio, como el de la construcción de conocimiento, o implementar varios ejercicios con dificultad moderada en los que se les solicite entre uno a dos cambios.
- Se puede trabajar con una base de datos ya hecha con anterioridad o una nueva con máximo 5 tablas.
- Determinar un tiempo específico para poder realizar las actividades.
- Dar retroalimentación del ejercicio.

Actividad de desempeño 2

Propósito

En esta actividad, se trabajará cómo algunas “consultas” no se pueden obtener con un solo intento, por lo que es necesario agrupar un número de datos para lograr conseguir el resultado esperado al implementar un algoritmo. En la medida que incorporen este desempeño, las y los estudiantes podrán comprender que, al momento de hacer una consulta de base de datos, se requiere tener claridad en la especificidad de la información que realmente se necesita conseguir para aplicar soluciones algorítmicas a diferentes problemáticas.

Objetivos de aprendizaje

OA 4

Establecer conexiones y semejanzas entre las partes del problema para caracterizar patrones en una situación problemática dada. [\(Generalizar y transferir\)](#)

OA 5

Aplicar flexiblemente las soluciones elaboradas e implementadas a otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista. [\(Generalizar y Transferir\)](#)

Conocimiento esencial

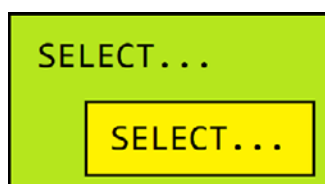
- Representación y análisis de datos.
-

Tiempo estimado: 3 horas pedagógicas

Desarrollo de la actividad

SITUACIÓN EXPERIENCIAL

Se sugiere explicar que la consulta de base de datos se basa en la recolección de datos específicos y que, para obtener esos datos aún más precisos, hacen falta más herramientas. Luego, se propone presentar dos casos diferentes, pero que en la práctica se utilizan bastante a menudo simultáneamente en contextos, por ejemplo, de sucursales comerciales que necesitan saber cuántos trabajadores tienen. La primera funciona como ejemplificación del concepto del anidado y la otra detalla con exactitud lo que ocurre con los “GROUP BY”:



GROUP BY		NOMBRE	SUCURSAL
		José	Norte
		Juan	Norte
		Pedro	Sur
		María	Norte

SUCURSAL	CANTIDAD
Norte	3
Sur	1

Algunas de las preguntas que podrían promover procesos de pensamiento asociados al análisis de los casos presentados podrían ser:

- ¿Qué es lo que se representa en la primera imagen?
- ¿Qué se está agrupando en la segunda imagen?
- ¿Cuál es la similitud de las imágenes?

CONSTRUCCIÓN DE CONOCIMIENTO

Para construir el conocimiento con las herramientas (GROUP BY/ SUB-CONSULTAS), se sugiere que el o la docente las pueda trabajar con sus estudiantes mediante ejercicios separados y después unirlos en un solo ejercicio:

- Se necesita obtener a los empleados que tengan el mismo salario que tiene “Hermann Baer” y los datos deben mostrarse de la siguiente manera:

	IDENTIFICACION EMPLEADO	EMPLEADO	JOB_ID	SALARIO
1	150	Peter Tucker	SA_REP	\$10,000
2	156	Janette King	SA_REP	\$10,000
3	169	Harrison Bloom	SA_REP	\$10,000
4	204	Hermann Baer	PR_REP	\$10,000

- Se necesita saber el promedio de empleados que hay en por departamento:

	PROMEDIO DE EMPLEADOS POR DEPARTAMENTO
1	8,91666666666666666666666666666667

- Se necesita saber los empleados que tienen un salario menor al promedio general de los empleados:

EMPLEADO	SALARIO
1	132 \$2,100
2	128 \$2,200
3	136 \$2,200
4	135 \$2,400
5	127 \$2,400
6	182 \$2,500
7	191 \$2,500

Acción	Resultado esperado	Preguntas de apoyo
Ejercicio 1.	<pre>SELECT EMPLOYEE_ID "IDENTIFICACION EMPLEADO", FIRST_NAME ' ' LAST_NAME EMPLEADO, JOB_ID,TO_CHAR(SALARY,'\$99,999') SALARIO FROM EMPLOYEES WHERE SALARY = (SELECT SALARY FROM EMPLOYEES WHERE EMPLOYEE_ID = 204);</pre>	
Ejercicio 2	<pre>SELECT AVG(E.ID1) "PROMEDIO DE EMPLEADOS POR DEPARTAMENTO" FROM (SELECT COUNT(*) AS ID1 FROM EMPLOYEES R GROUP BY R.DEPARTMENT ID) E;</pre>	¿Cómo hacer el promedio de un contador?
Ejercicio 3	<pre>SELECT EMPLOYEE_ID EMPLEADO, TO_CHAR(SALARY,'\$99,999') SALARIO FROM EMPLOYEES WHERE SALARY < (SELECT AVG(SALARY) FROM EMPLOYEES) ORDER BY SALARY;</pre>	

PRÁCTICA GUIADA

Para implementar algoritmos que permitan trabajar con datos, se propone guiar a las y los estudiantes mostrándoles nuevos conceptos en cuanto a consultas de base de datos, en los cuales se trabajará con JOIN, GROUP BY y SUB-CONSULTAS:

Acción	Resultado esperado																																		
Elaborar el ingreso de datos mediante el guardado de información en los arreglos.	<pre> SELECT DEPARTMENT_ID "ID DEPARTAMENTO", DEPARTMENT_NAME "NOMBRE DEPARTAMENTO" FROM DEPARTMENTS WHERE DEPARTMENT_ID <> ALL (SELECT DEPARTMENT_ID FROM EMPLOYEES WHERE DEPARTMENT_ID IS NOT NULL GROUP BY DEPARTMENT_ID) ORDER BY DEPARTMENT_ID; </pre>																																		
Resultado esperado.	<table> <thead> <tr> <th>ID DEPARTAMENTO</th><th>NOMBRE DEPARTAMENTO</th></tr> </thead> <tbody> <tr><td>1</td><td>120 Treasury</td></tr> <tr><td>2</td><td>130 Corporate Tax</td></tr> <tr><td>3</td><td>140 Control And Credit</td></tr> <tr><td>4</td><td>150 Shareholder Services</td></tr> <tr><td>5</td><td>160 Benefits</td></tr> <tr><td>6</td><td>170 Manufacturing</td></tr> <tr><td>7</td><td>180 Construction</td></tr> <tr><td>8</td><td>190 Contracting</td></tr> <tr><td>9</td><td>200 Operations</td></tr> <tr><td>10</td><td>210 IT Support</td></tr> <tr><td>11</td><td>220 NOC</td></tr> <tr><td>12</td><td>230 IT Helpdesk</td></tr> <tr><td>13</td><td>240 Government Sales</td></tr> <tr><td>14</td><td>250 Retail Sales</td></tr> <tr><td>15</td><td>260 Recruiting</td></tr> <tr><td>16</td><td>270 Payroll</td></tr> </tbody> </table>	ID DEPARTAMENTO	NOMBRE DEPARTAMENTO	1	120 Treasury	2	130 Corporate Tax	3	140 Control And Credit	4	150 Shareholder Services	5	160 Benefits	6	170 Manufacturing	7	180 Construction	8	190 Contracting	9	200 Operations	10	210 IT Support	11	220 NOC	12	230 IT Helpdesk	13	240 Government Sales	14	250 Retail Sales	15	260 Recruiting	16	270 Payroll
ID DEPARTAMENTO	NOMBRE DEPARTAMENTO																																		
1	120 Treasury																																		
2	130 Corporate Tax																																		
3	140 Control And Credit																																		
4	150 Shareholder Services																																		
5	160 Benefits																																		
6	170 Manufacturing																																		
7	180 Construction																																		
8	190 Contracting																																		
9	200 Operations																																		
10	210 IT Support																																		
11	220 NOC																																		
12	230 IT Helpdesk																																		
13	240 Government Sales																																		
14	250 Retail Sales																																		
15	260 Recruiting																																		
16	270 Payroll																																		
Conseguir el departamento con el mayor número de empleados.	<pre> SELECT D.DEPARTMENT_NAME DEPARTAMENTO, COUNT(*) "TOTAL DE EMPLEADOS" FROM EMPLOYEES E JOIN DEPARTMENTS D ON E.DEPARTMENT_ID=D.DEPARTMENT_ID GROUP BY D.DEPARTMENT_NAME HAVING COUNT(*) = (SELECT MAX(COUNT(*)) FROM EMPLOYEES GROUP BY DEPARTMENT_ID); </pre>																																		
Resultado esperado.	<table> <thead> <tr> <th>DEPARTAMENTO</th><th>TOTAL DE EMPLEADOS</th></tr> </thead> <tbody> <tr> <td>1 Shipping</td><td>45</td></tr> </tbody> </table>	DEPARTAMENTO	TOTAL DE EMPLEADOS	1 Shipping	45																														
DEPARTAMENTO	TOTAL DE EMPLEADOS																																		
1 Shipping	45																																		

PRÁCTICA INDEPENDIENTE

Se recomienda un trabajo individual donde el o la estudiante trabaje con ejercicios (máximo 5) en los que aplique las herramientas aprendidas en los modelos. Pero esta vez con el desafío de incorporar GROUP BY y SUB-CONSULTAS en bases de datos creadas por el estudiantado y donde consideraron situaciones cotidianas, donde el uso de las bases de datos es un facilitador de tareas en diferentes ámbitos, como el transporte, la salud o las comunicaciones. Para guiarles, se puede incorporar la resolución de problemas de manera algorítmica a partir de bases de datos.

EVALUACIÓN FORMATIVA

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Consulta de base de datos	Genera Script con errores.	Genera Script de manera correcta, pero no se visualizan algunos datos.	Genera Script de manera correcta y se visualizan todos los datos.
Herramientas GROUP BY / SUBQUERIES	Usa las herramientas GROUP BY/ SUBQUERIES con errores.	Usa las herramientas GROUP BY/ SUBQUERIES, pero de manera ineficiente.	Usa las herramientas GROUP BY/SUBQUERIES de manera comprensible y eficiente.

Orientaciones al docente

Para unificar conceptos disciplinares

Trabajar con dos consultas. Debe ser algo confuso en un inicio y puede provocar ciertos enredos al momento de comprender y planear lo que se debe hacer para resolver la problemática a nivel inicial.

Términos del GROUP BY:

- La declaración **GROUP BY** agrupa las filas que tienen los mismos valores en filas de resumen, como "encontrar el número de clientes en cada país".
- La instrucción **GROUP BY** se utiliza a menudo con funciones de agregado (COUNT(), MAX(), MIN(), SUM(), AVG()) agrupa el conjunto de resultados de una o más columnas.

Términos de las SUB-CONSULTAS:

- Una **subconsulta** o consulta interna o una consulta anidada es una consulta dentro de otra consulta SQL e incrustada dentro de la cláusula WHERE.
- Una **subconsulta** se utiliza para devolver datos que se utilizarán en la consulta principal como condición para restringir aún más los datos que se recuperarán.
- Las **subconsultas** se pueden usar con las instrucciones SELECT, INSERT, UPDATE y DELETE junto con los operadores como =, <, >, <=, IN, BETWEEN, etc.

Hay algunas reglas que las subconsultas deben seguir:

- Las subconsultas deben ir entre paréntesis.
- Una subconsulta solo puede tener una columna en la cláusula *SELECT*, a menos que haya varias columnas en la consulta principal para que la subconsulta compare sus columnas seleccionadas.
- Un comando *ORDER BY* no se puede usar en una subconsulta, aunque la consulta principal puede usar un *ORDER BY*. El comando *GROUP BY* se puede utilizar para realizar la misma función que *ORDER BY* en una subconsulta.
- Las subconsultas que devuelven más de una fila solo se pueden usar con operadores de valor múltiple, como el operador *IN*.
- La lista *SELECT* no puede incluir referencias a valores que se evalúen como *BLOB*, *ARRAY*, *CLOB* o *NCLOB*.
- Una subconsulta no puede incluirse inmediatamente en una función establecida.
- El operador *BETWEEN* no se puede utilizar con una subconsulta. Sin embargo, el operador *BETWEEN* se puede utilizar dentro de la subconsulta.

Actitudes

Para apoyar el desarrollo de las actividades se recomienda el uso de computadores, pues los trabajos que realizarán requieren de una conexión directa a la base de datos que cuenta con la información que se trabajará. En cuanto al desarrollo de la actividad individual, el computador es necesario para dar autonomía y confianza al resolver problemas.

Orientaciones para organizar e implementar el trabajo individual

Se sugieren las siguientes indicaciones para guiar al estudiante en el trabajo individual:

- Elaborar un ejercicio amplio, como en la construcción de conocimiento o implementar varios ejercicios con dificultad moderada, en los que se les solicite dos consultas.
- Trabajar con una base de datos nueva con máximo 10 tablas.
- Dar retroalimentación del ejercicio.

Actividad de desempeño

3

Propósito

Esta actividad está enfocada en el conocimiento y uso del lenguaje de programación Python, con la aplicación de un proceso lógico en el que se vincule la implementación de un algoritmo con dicho lenguaje. Al mismo tiempo, se propicia la búsqueda de soluciones a situaciones problemáticas mediante la elaboración de algoritmos y la programación, y se visualizan las diferencias en cuanto a hacer un pseudocódigo y un código.

Objetivos de aprendizaje

OA 6

Elaborar e implementar soluciones a una situación problemática mediante la formulación de algoritmos, actuando de acuerdo con los principios de la ética.
(Pensar de forma algorítmica)

Conocimiento esencial

- Algoritmos y programación: lenguajes, métodos y entornos de programación.
- Representación y análisis de datos.

Tiempo estimado: 3 horas pedagógicas

Desarrollo de la actividad

SITUACIÓN EXPERIENCIAL

Se sugiere presentar un pseudocódigo y un código, y hacer notar que en ambos se utiliza el mismo algoritmo. Este resuelve detectar si un número ingresado es par o impar ante, por ejemplo, la necesidad de configurar una máquina que detecte la apertadura de llantas de un auto:

<pre> Escribir "Escriba un número: " Leer Num Si Num % 2 = 0 Entonces Escribir Num, " es un numero par" SiNo Escribir Num, " es un numero impar" Fin Si </pre>	<pre> print("Ingrese Número : ") num = int(input()) cociente = round(num / 2) residuo = cociente * 2 resultado = num - residuo if resultado == 0: print("NÚMERO PAR") else: print("NÚMERO IMPAR") </pre>
---	--

Los y las estudiantes tendrán que analizar los algoritmos, establecer las diferencias entre ellos y comentar guiados por las siguientes preguntas:

- ¿Cómo están estructurados los algoritmos?
- ¿Qué diferencias hay entre un pseudocódigo y un código?
- ¿Hay alguna semejanza en cuanto a funcionalidad?

CONSTRUCCIÓN DE CONOCIMIENTO

Para construir el conocimiento con las bases del algoritmo y crear pseudocódigos, se sugiere recordar a los y las estudiantes las medidas aprendidas con anterioridad, de manera que funcione para un código en un programa de programación profesional. Como ejercicio de práctica, los y las estudiantes pueden elaborar las nuevas sintaxis de la forma más realista posible, para así verlas en un nuevo lenguaje de programación, como Python. Para esto, se trabajará con problemas ya antes vistos en PSeint, de forma que el proceso de transferencia tenga mayor significado.

Pueden registrar el procedimiento de instrucciones en una hoja de trabajo de códigos. En el anexo podrán encontrar la hoja para escribir en modo de zona de escritura de código

Como ejercicio, se realizará una calculadora con las funciones de “sumar, restar, multiplicar y dividir” con solo dos valores a elegir para realizar el cálculo.

Acción	Resultado esperado
Definir los valores a utilizar .	<pre>opc = int valor1 = int valor2 = int total = float</pre>
Crear menú e ingresar valores a calcular.	<pre>print("Calculadora") print("""1: Sumar 2: Restar 3: Multiplicar 4: Dividir""") opc = int(input("Ingrese opcion: ")) valor1 = int(input("Ingrese valor1: ")) valor2 = int(input("Ingrese valor2: "))</pre>
Hacer el cálculo dependiendo de la opción que ingreso.	<pre>if opc == 1: total = valor1 + valor2 print("La suma de ",valor1," + ",valor2," = ",total) elif opc == 2: total = valor1 - valor2 print("La resta de ",valor1," - ",valor2," = ",total) elif opc == 3: total = valor1 * valor2 print("La multiplicacion de ",valor1," * ",valor2," = ",total) elif opc == 4: total = valor1 / valor2 print("La division de ",valor1," / ",valor2," = ",total)</pre>

PRÁCTICA GUIADA

Para guiar a sus estudiantes, se sugiere un ejercicio de la misma categoría, pero con un concepto más real que se podría ver reflejado en la cotidianidad. Por ejemplo, se necesita determinar el cálculo del impuesto en la renta de unos departamentos ubicados en alguna parte del país. Los datos son los siguientes:

CATEGORÍA	RENTA	IMPUESTO
Bronce	Menos de \$1 440 000	3%
Plata	Entre \$1 440 000 y \$6 000 000	5%
Oro	Entre \$6 000 000 y \$11 400 000	10%
Diamante	Mas de \$11 400 000	20%

Se sugiere como desafío que escriban un programa que pregunte al usuario su renta anual y muestre en el terminal lo que debe pagar de impuesto en total:

Acción	Resultado esperado
Programación Python esperada para el desarrollo del ejercicio..	<pre>1 # Preguntar al usuario por la renta 2 renta = float(input("¿Cuál es tu renta anual? ")) 3 # Condicional para determinar el tipo impuesto 4 if renta < 1440000: 5 tipo = 3 6 elif renta >= 1440000 and renta < 6000000: 7 tipo = 5 8 elif renta >= 6000000 and renta < 11400000: 9 tipo = 10 10 elif renta >= 11400000: 11 tipo = 20 12 13 # Mostrar por pantalla el impuesto a pagar 14 print("El impuesto a pagar es: \$", renta * tipo / 100, sep='')</pre>

PRÁCTICA
INDEPENDIENTE

Se sugiere que las y los estudiantes realicen un trabajo individual con al menos 5 ejercicios con condicionales, para así poder usar el nuevo lenguaje de programación aplicando un proceso lógico. Por ejemplo, con la siguiente situación:

Una pizzería llamada Luigi's ofrece pizzas vegetarianas y tradicionales. Los ingredientes para cada tipo de pizza son los siguientes:

- Vegetariano: choclo, pimentón y tofu.
- Tradicional: jamón, pepperoni y salchichas.

Se debe escribir un programa que pregunte al usuario si quiere una pizza vegetariana o no, y en función de su respuesta le mostrara un menú en el cual elegir los ingredientes por categoría. Solo se puede optar por un ingrediente además de la mozzarella y el tomate, que están en todas las pizzas. Al finalizar, debe mostrar en pantalla los detalles de la pizza elegida por el cliente.

EVALUACIÓN
FORMATIVA

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Código	Elabora un código, varios con errores.	Elabora un código con algunos errores.	Elabora un código sin errores.
Sintaxis	Usa la sintaxis con varios errores.	Usa la sintaxis con algunos errores.	Usa la sintaxis sin errores.

Orientaciones al docente

Para unificar conceptos disciplinares

En el lenguaje de la programación, el algoritmo es la base principal de todo entendimiento lógico a nivel computacional. La aplicación utilizada para desarrollar los códigos de los ejercicios es Microsoft Studio Code, con uso del lenguaje de programación “Python” que, a pesar de ser un lenguaje estricto en cuanto a escritura se refiere, es uno de los más sencillos para aprender lo que es codificación profesional:

En PSeint fue posible ver que en cuanto a condicionales existen el Si y el Según, en este lenguaje solo existe la derivación del Si (If, elif, else).

“Microsoft Studio Code” es un editor de texto que a su vez facilita la implementación de sugerencias a la hora de codificar en el lenguaje impuesto, también se puede utilizar “Sublime Text” como medio para realizar los ejercicios.

Ejemplificar con demostraciones en vivo, proyecciones de ejemplo u otros ejercicios digitales en aplicaciones (como OctoStudio u otras), plataformas de formación (Code.org, Khan Academy, entre otras) y software de programación (Scratch, PSeint, entre otros), permite reforzar los contenidos de manera visual y práctica. Estas herramientas ofrecen entornos interactivos donde las y los estudiantes pueden experimentar con algoritmos, estructuras condicionales y ciclos, y observar en tiempo real cómo sus decisiones afectan el comportamiento del programa.

Actitudes

Para apoyar el desarrollo de las actividades, se recomienda utilizar en todo momento computadores configurados para la ocasión, con sus respectivos programas y controladores para poder trabajar en los problemas que vayan surgiendo a lo largo del conocimiento de este lenguaje. En cuanto al desarrollo de la actividad individual, esta se hará así para que cada estudiante experimente autonomía y utilice los conocimientos previos relacionados con el funcionamiento del algoritmo y así pueda resolver los problemas.

Orientaciones para organizar e implementar el trabajo individual

Para guiar al estudiantado en el trabajo individual se sugiere:

- Elaborar cinco ejercicios como en la construcción de conocimiento.
- Determinar un tiempo específico para poder realizar las actividades.
- Evaluar el impacto de su resultado a nivel social o ético de acuerdo con el problema que resuelve.
- Dar retroalimentación del ejercicio.

Actividad de desempeño

4

Propósito

Esta actividad desafía a aprender cómo no repetir código y cómo poder trabajar las herramientas de los bucles en el lenguaje de programación Python. La idea es que, luego, puedan elaborar e implementar algoritmos con un nuevo lenguaje de programación, generalizando y transfiriendo los aprendizajes asociados a la programación de código y pseudocódigo a realidades o necesidades cotidianas propias o de otros.

Objetivos de aprendizaje

OA 5

Aplicar flexiblemente las soluciones elaboradas e implementadas a otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista.
(Generalizar y transferir)

OA 6

Elaborar e implementar soluciones a una situación problemática mediante la formulación de algoritmos, actuando de acuerdo con los principios de la ética.
(Pensar de forma algorítmica)

Conocimiento esencial

- Algoritmos y programación: lenguajes, métodos y entornos de programación.
- Representación y análisis de datos.
- Impacto social de la computación: interacciones, cultura, seguridad y ética.
- Lenguaje de programación: Python.
- Sintaxis básicas (tipos, operadores, variables, funciones, listas, tuplas, diccionarios).
- Condicionales (if y else).

Tiempo estimado: 3 horas pedagógicas

Desarrollo de la actividad

SITUACIÓN EXPERIENCIAL

Se sugiere presentar a las y los estudiantes una imagen comparativa con respecto a los bucles entre PSeint y Python. Se recomienda plantear un contexto donde se requiera un menú para albergar dos opciones y que una de ellas sea calcular el promedio de personas en un restaurant, de acuerdo a una cantidad seleccionable de personas a disposición del usuario. Por ejemplo, calcular el promedio de personas que pueden ingresar dado el aforo de lugar:



Luego, los y las estudiantes tendrán que analizar y comentar los bucles en base a las siguientes preguntas:

- ¿Por qué no existe el repetir en el lenguaje de programación Python?
- ¿Cuáles son las diferencias fundamentales entre Python y PSeint?
- ¿Qué otra aplicación o programa debiésemos tener presente para comparar?

CONSTRUCCIÓN DE CONOCIMIENTO

Al momento de programar, los algoritmos son una pieza fundamental en cuanto a establecer un orden en los códigos, por lo que para la siguiente actividad se sugiere trabajar con los menús que se trabajaron en módulos anteriores con PSeint.

Con respecto a lo anterior se trabajará con un menú donde habrá dos opciones a elegir:

- Calcular el promedio dependiendo de la cantidad de notas que ingrese.
- Una tabla de multiplicar.

Acción	Resultado esperado
Elaborar el menú con las dos opciones previamente señaladas.	<pre> 1 opc = 1 2 while opc == 1 or opc == 2: 3 opc = int(input("""INGRESE UNA OPCION 4 1- Calcular promedio 5 2- Tabla de multiplicar 6 3- Salir 7 """)) </pre>
Elaborar la primera opción, ingresando al ingresar el total de notas, entrar a un bucle “for” para así no repetir código y repetir la cantidad previamente ingresada por el usuario, para terminar con el promedio final calculado.	<pre> 9 if opc == 1: 10 suma = 0 11 n = int(input("INGRESE LA CANTIDAD DE NOTAS : ")) 12 for cont in range(0,n): 13 nota = int(input("NOTA : ")) 14 suma = suma + nota 15 print("PROMEDIO : {:.2f}".format(suma/n)) </pre>
Elaborar la segunda opción, al ingresar el número para el que se harán sus respectivas multiplicaciones.	<pre> 17 if opc == 2: 18 print("Ingrese un Número: ") 19 num = int(input()) 20 for cont in range(1,13): 21 print(cont," X ",num," = ",(cont*num)) </pre>

PRÁCTICA GUIADA

Siguiendo el ejercicio anterior, se sugiere que el o la docente sitúe a sus estudiantes en un contexto real cercano o cotidiano, por ejemplo:

Para el área de nutrición, un hospital requiere de un programa que permita realizar un cálculo de índice de masa corporal. Para esto se necesita la información de su peso (kg) y altura (metros).

Diagnóstico	IMC
PESO INSUFICIENTE	<18.5
PESO NORMAL	18.5 – 12.9
SOBREPESO	25 – 29.9
OBESIDAD GRADO 1	30 – 34.9
OBESIDAD GRADO 2	35 – 39.9
OBESIDAD GRADO 3	≥ 40

La fórmula para sacar el cálculo es el siguiente:

$$IMC = \frac{\text{Peso (kg)}}{\text{Altura}^2 \text{ (m)}}$$

Acción	Resultado esperado
Iniciar las variables, dar a elegir cuántos clientes se harán el diagnóstico del IMC. Se hace el cálculo con la fórmula y se verifica en qué categoría se encuentra el cliente.	<pre> 30 ipe = 0 31 pno = 0 32 sg = 0 33 ob1 = 0 34 ob2 = 0 35 ob3 = 0 36 37 print("Bienvenido a IMC") 38 cant_cli = int(input("Indique la cantidad de clientes: ")) 39 40 for i in rango(0,cant_cli): 41 print("-----") 42 print("Ingrese peso del cliente N°",i+1) 43 peso = int(input()) 44 print("Ingrese altura en metros del cliente N°",i+1) 45 talla = int(input()) 46 print("El cliente ingresado esta en la categoria:") 47 imc = peso / talla ** 2 48 49 if imc < 18.5: 50 print("Peso Insuficiente") 51 ipe = ipe + 1 52 elif imc >= 18.5 and imc <= 24.9: 53 print("Peso Normal") 54 pno = pno + 1 55 elif imc >= 25 and imc <= 29.9: 56 print("SobrePeso") 57 sg = sg + 1 58 elif imc >= 30 and imc <= 34.9: 59 print("Obesidad Grado I") 60 ob1 = ob1 + 1 61 elif imc >= 35 and imc <= 39.9: 62 print("Obesidad Grado II") 63 ob2 = ob2 + 1 64 elif imc >= 40 : 65 print("Obesidad Grado III") 66 ob3 = ob3 + 1 67 print("-----") 68 69 print("-----") 70 print("Cantidad de clientes por categoria:") 71 cant_cliente(ipe,pno,sg,ob1,ob2,ob3) 72 print("-----") 73 print("La cantidad con mas clientes es:") 74 cant_imc(ipe,pno,sg,ob1,ob2,ob3) 75 print("-----") </pre>
Elaborar una función para verificar la cantidad de clientes por categoría.	<pre> 1 def cant_cliente(ipe,pno,sg,ob1,ob2,ob3): 2 if ipe > 0: 3 print("Cantidad de Clientes con Insuficiente Peso: ",ipe) 4 if pno > 0: 5 print("Cantidad de Clientes con Peso Normal: ",pno) 6 if sg > 0: 7 print("Cantidad de Clientes con SobrePeso Grado I: ",sg) 8 if ob1 > 0: 9 print("Cantidad de Clientes con Obesidad Grado I: ",ob1) 10 if ob2 > 0: 11 print("Cantidad de Clientes con Obesidad Grado II: ",ob2) 12 if ob3 > 0: 13 print("Cantidad de Clientes con Obesidad Grado III: ",ob3) </pre>
Elaborar una segunda función para ver cuál categoría es la que tiene la mayor cantidad de clientes.	<pre> 15 def cant_imc(ipe,pno,sg,ob1,ob2,ob3): 16 if ipe >= pno and ipe >= sg and ipe >= ob1 and ipe >= ob2 and ipe >= ob3: 17 print("Mas Clientes en Insuficiente Peso") 18 if pno >= ipe and pno >= sg and pno >= ob1 and pno >= ob2 and pno >= ob3: 19 print("Mas Clientes en Peso Normal") 20 if sg >= ipe and sg >= pno and sg >= ob1 and sg >= ob2 and sg >= ob3: 21 print("Mas Clientes en SobrePeso Grado I") 22 if ob1 >= ipe and ob1 >= pno and ob1 >= sg and ob1 >= ob2 and ob1 >= ob3: 23 print("Mas Clientes en Obesidad Grado I") 24 if ob2 >= ipe and ob2 >= pno and ob2 >= sg and ob2 >= ob1 and ob2 >= ob3: 25 print("Mas Clientes en Obesidad Grado II") 26 if ob3 >= ipe and ob3 >= pno and ob3 >= sg and ob3 >= ob1 and ob3 >= ob2: 27 print("Mas Clientes en Obesidad Grado III") </pre>

PRÁCTICA INDEPENDIENTE

Se sugiere que los y las estudiantes realicen un trabajo individual con al menos 3 ejercicios con bucles, para así poder usar el nuevo lenguaje de programación, por ejemplo: elaborar un programa que permita almacenar los nombres y promedios de la cantidad estudiantes de un curso, las notas deben estar en un rango de 10 y 70. Mostrar al estudiante con la mayor nota.

Ejercicio complementario

Como segundo ejercicio, se propone una actividad que relacione la lógica de programación con el funcionamiento de un taller de carpintería, donde cada herramienta cumple una función específica y debe estar organizada para facilitar el trabajo. El estudiante deberá crear un programa que simule el control de herramientas en dicho taller. El programa debe permitir:

- Ingresar el nombre de cada herramienta.
- Registrar su estado (disponible, en uso, o en reparación).
- Indicar cuántas unidades hay de cada una.

Al usar bucles, el programa debe permitir ingresar múltiples herramientas y, al finalizar, mostrar un resumen que incluya:

- El total de herramientas registradas.
- Cuántas están disponibles para trabajar.
- Cuál es la herramienta con mayor cantidad disponible.

Este ejercicio permite reforzar el uso de bucles, condicionales y listas, al tiempo que conecta con un contexto práctico y cotidiano. Se recomienda el uso de plataformas gratuitas que permite programar en Python directamente desde el navegador, sin necesidad de instalar software adicional. También se pueden utilizar aplicaciones para repasar conceptos clave de forma interactiva o Code.org para quienes aún están en transición desde lenguajes visuales.

EVALUACIÓN FORMATIVA

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Código	Elabora un código con varios errores.	Elabora un código con algunos errores.	Elabora un código sin errores.
Sintaxis	Usa la sintaxis con varios errores.	Usa la sintaxis con algunos errores.	Usa la sintaxis sin errores.

Orientaciones al docente

Para unificar conceptos disciplinares

Con el conocimiento obtenido para diseñar un algoritmo, todo se hace mucho más fácil al implementarlo en un lenguaje de programación. Pero eso no significa que las y los estudiantes deban dejar de aprender cosas nuevas con respecto al lenguaje de programación, ya que muchos aspectos mínimos cambian a favor del lenguaje utilizado para trabajar.

Por estas razones en PSeint se tenía el *Repetir*, *Para* y *Mientras*, pero en el nuevo lenguaje se tiene las variantes Para y Mientras (For y While).

Actitudes

Para apoyar el desarrollo de las actividades, se recomienda utilizar en todo momento computadores. En cuanto al desarrollo de la actividad individual, esta se hará así para que cada estudiante experimente autonomía y utilice los conocimientos previos que posee respecto del funcionamiento del algoritmo y pueda resolver los problemas.

Orientaciones para organizar e implementar

Se sugieren las siguientes indicaciones para guiar al estudiante en el trabajo individual.

- Elaborar tres ejercicios, como en la construcción de conocimiento.
- Determinar un tiempo específico para poder realizar las actividades.
- Evaluar el impacto de su resultado a nivel social o ético de acuerdo con el problema que resuelve.
- Dar retroalimentación del ejercicio.

Módulos Electivos de la Asignatura

Módulo Electivo 1
**Aprendizaje Basado
en Proyectos**

Módulo Electivo 2
**Aprendizaje Basado en
Resolución de Problemas**

Módulo Electivo 1

Aprendizaje Basado en Proyectos

Visión panorámica

Gran idea

El comportamiento de un sistema computacional (dispositivos, *hardware* y *software*) depende de la secuencia de instrucciones dadas por el usuario que automatizarán el desempeño de una tarea para ejecutar una acción o resolver un problema.

Objetivos de aprendizaje

OA 2

Organizar datos de una situación problemática estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función. ([Evaluar](#))

OA 3

Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponibles. ([Modelar](#))

Conocimientos esenciales

- Modelos entidad-relación y simulación de sistemas.
 - Sistemas computacionales: dispositivos, *hardware* y *software*, y resolución de problemas.
-

Tiempo estimado: 6 semanas (12 horas)

Propósito

Este proyecto tiene como propósito la creación de una base de datos orientada a un diseño computacional profesional, que incorpore la formación de datos de acuerdo con el problema que sea necesario solucionar.

Para iniciar, los y las estudiantes analizan los elementos esenciales del problema. En equipos de trabajo deberán distinguir la información específica que conforma la base de datos y sus respectivas relaciones. También, podrán contrastar con otros equipos sus resultados y retroalimentar el proceso que siguieron.

Luego, a partir del análisis y la obtención de la información del problema planteado, las y los estudiantes modelan la base de datos para verificar si sus datos tienen coherencia estructural para resolver el problema en cuanto a lo esencial del mismo.

Por último, crean la base de datos en un contenedor (Oracle, SQL Server o SQL Lite) y trabajan en la creación de entidades para ella. Para esto, desarrollan una consulta que pueda crear las tablas e insertar los datos que puedan tener significado en el problema; exponen la base de datos creada y buscan la posibilidad de concretar en la realidad sus ideas, dando contexto técnico y funcional a sus creaciones.

Problema central

En Chile, una empresa de alimentos requiere una base de datos centralizada donde almacenar todos sus productos, según el stock de cada tienda y las personas que ahí trabajan. Estos productos vienen principalmente de proveedores que conocen su información personal. Por otra parte, los proveedores pueden estar o no registrados en la empresa: si están registrados se les solicita su RUT, nombre, dirección, sexo y opcionalmente el teléfono y correo electrónico.

La empresa tiene al menos una tienda en cada región de Chile. Estas se manejan en base a un identificador, nombre y su respectiva dirección descrita con región y comuna. Cada tienda tiene su propio stock de productos. Cada producto tienen un identificador, nombre, descripción, fecha de llegada, precio y categoría. En la boleta debe ir el empleado que realizó la venta junto a su respectivo cliente, además de contener la fecha del día en el que se factura. La boleta debe tener el detalle de la compra, con los productos que se seleccionaron, su cantidad y el precio total. Además, debe indicar en qué tienda se realizó la compra.

Las categorías que tienen los productos alimenticios son las siguientes:

- *Grupo I: Leche y derivados. Son alimentos plásticos. En ellos predominan las proteínas.*
- *Grupo II: Carnes, pescados y huevos. Alimentos plásticos. En ellos predominan las proteínas.*
- *Grupo III: Legumbres, frutos secos y patatas. Alimentos energéticos, plásticos y reguladores. En ellos predominan los glúcidos, pero también poseen cantidades importantes de proteínas, vitaminas y minerales.*

- Grupo IV: Hortalizas. Alimentos reguladores. En ellos predominan las vitaminas y minerales.
- Grupo V: Frutas. Alimentos reguladores. En ellos predominan las vitaminas y minerales.
- Grupo VI: Cereales. Alimentos energéticos. En ellos predominan los glúcidos.
- Grupo VII: Mantecas y aceites. Alimentos energéticos. En ellos predominan los lípidos.

Los trabajadores se identifican mediante su RUT, nombre, apellido, dirección, sexo y su cargo. Asimismo, ellos pueden recibir muchos pedidos en el día.

Los cargos que tienen los trabajadores son los siguientes:

- Gerente.
- Cajero.
- Reponedor.
- Bodeguero.
- Especialista de sección, como frutería, carnicería o pescadería, panadería entre otros.

Propósito Crear una base de datos centralizada y funcional, orientada a un diseño computacional profesional, que resuelva un problema en un ámbito de la actividad humana que sea cercano o significativo para los y las estudiantes. A su vez, modelar la base de datos y elaborar un informe descriptivo del proceso.

Objetivos de Aprendizaje

OA 2

Organizar datos de una situación problemática estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función. **(Evaluar)**

OA 3

Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponibles. **(Modelar)**

Preguntas

- ▶ ¿Cuál es el propósito de la base de datos solicitada?
- ▶ ¿Cómo la base de datos ayuda en el manejo de datos críticos?
- ▶ ¿Cuáles son los elementos necesarios para diseñar una base de datos?
- ▶ ¿Qué medios permiten modelar la base de datos?
- ▶ ¿Cómo se pueden identificar los datos correspondientes dentro de un problema?
- ▶ ¿Qué debo hacer cuando no logro encontrar las relaciones correspondientes en el problema?
- ▶ ¿Cómo se puede crear la base de datos centralizada para responder al problema?
- ▶ ¿Cómo se organiza la información en lugares donde no se usan computadoras?
- ▶ ¿Qué problemas podrían solucionarse con bases de datos en tu comunidad o trabajo?

Tipo de proyecto Proyecto intradisciplinario

Producto Base de datos centralizada, que resuelve un problema de control de stock de una empresa de distribución de alimentos a nivel nacional, y un informe descriptivo del proceso de solución que permita generalizar y transferir la solución a otros contextos que requieran del control de datos. Para desarrollar su propuesta, los y las estudiantes eligen entre distintos tipos de prototipos o productos: un modelo digital programado o una planilla en Excel, que simule el funcionamiento de una base de datos; o bien, un sistema de registro manual estructurado en papel, que represente de forma lógica y ordenada el flujo de información. Esta flexibilidad fomenta la autonomía, la toma de decisiones y la adaptación de soluciones a distintos entornos y niveles de complejidad.

Habilidades y actitudes para el siglo XXI

Habilidades:

- Maneras de Pensar: Creatividad e Innovación.
- Maneras de Trabajar: Comunicación y colaboración.

Actitudes:

- Pensar con perseverancia y proactividad para encontrar soluciones innovadoras a los problemas.
- Trabajar con honestidad en la realización de tareas colaborativas con miras al logro de una meta común.

Etapas

ETAPA 1
(Análisis del problema y modelado inicial)

En esta etapa, las y los estudiantes descomponen y analizan el problema principal, y aplican el conocimiento necesario para identificar tablas, columnas y relaciones. Asimismo, idean el modelo de base de datos a través del diseño inicial. Se sugiere incluir ejercicios en papel y tarjetas físicas donde las y los estudiantes organicen los datos antes de modelarlos en software, pues esto permite visualizar de forma concreta la estructura de la información. Además, se debe permitir que elijan entre prototipos y productos: un modelo digital de programación, una planilla en Excel o un sistema de registro manual estructurado en papel.

ETAPA 2
(Modelado de la base de datos)

En esta etapa los y las estudiantes modelan la base de datos a partir del análisis realizado en la etapa anterior. El objetivo es ordenar y vincular correctamente los elementos para que la base de datos cumpla con los requerimientos del problema. Como apoyo, se puede crear una maqueta física con tarjetas de colores que representen entidades y relaciones. Esto facilitará la comprensión de la estructura lógica antes de la implementación digital.

ETAPA 3
(Implementación y carga de datos)

En esta etapa los y las estudiantes crean la base de datos, ejecutan consultas básicas y cargan datos coherentes con el problema planteado. Para diversificar el contexto, se propone que, en lugar de una empresa de alimentos, modelen una base de datos para la gestión de asistencia en un centro de capacitación para personas adultas. Esta variación permite aplicar los mismos principios a un entorno distinto y promueve la transferencia del aprendizaje.

ETAPA 4
(Informe y reflexión
sobre el proyecto)

Finalmente, los y las estudiantes elaboran un informe descriptivo del proceso de solución. Detallan los pasos seguidos, los inconvenientes encontrados y las decisiones tomadas durante el desarrollo. Se promueve la reflexión sobre el proceso creativo y se agrega un componente de transferencia del aprendizaje. Se invita a las y los estudiantes a pensar cómo podrían aplicar lo aprendido en otros problemas cotidianos o laborales. Además, se sugiere compartir el producto final con sus pares, para recibir retroalimentación y fortalecer la comunicación técnica.

Cronograma semanal

ETAPA 1 En esta etapa se conforman equipos de trabajo, idealmente de tres integrantes. Descomponen y analizan el problema principal, para que encontrar los datos necesarios para la creación y modelamiento de la base de datos:

- Leen el problema, lo descomponen para encontrar cada dato y analizan cada aspecto para identificar lo esencial. Además, idean el modelo de base de datos que resolverá el problema y proponen una solución individual para, luego, decidir en equipo la mejor opción.
- A partir del análisis en equipos, identifican las entidades y relaciones que tiene el problema. Con estos datos se crean los primeros diseños que tendrá el modelo. Cada integrante debería realizar un diseño.
- Analizan los primeros diseños y verifican los datos para encontrar otros nuevos que puedan servir y complementar al problema.
- Explican detalladamente por qué se escogió el diseño final en equipo.

ETAPA 2 En esta etapa, los y las estudiantes desarrollan su modelo de base de datos de acuerdo con el mejor diseño seleccionado con anterioridad por el equipo. Con esto, los y las estudiantes en conjunto logran seleccionar uno de sus diseños para resolver la problemática principal:

- Retoman sus análisis previos para poder trabajar en un modelado de la base de datos de forma limpia.
- Plantean ideas para incorporar nuevas entidades y relaciones que podrían servir para complementar el desarrollo del modelo que trabajan como equipo. En este paso, se sugiere que un o una integrante incorpore entidades y relaciones y las demás personas del equipo las evalúen.

ETAPA 3 En esta etapa, los y las estudiantes crean la base de datos con sus conocimientos sobre la consulta de base de datos. Elaboran lo que esté modelado de acuerdo con una consulta para crear las entidades y las relaciones definitivas. Además, se insertan los datos que tengan coherencia con el problema y así se concluye con la consulta:

- En equipo crean la base de datos e incorporan los conocimientos de la consulta de base de datos.
- Definen los parámetros necesarios e insertan los datos correspondientes que tengan significado para la resolución del problema. Se recomienda que solo una persona inserte datos y las demás evalúen el procedimiento.

ETAPA 4 En esta etapa, desde sus respectivas experiencias, las y los estudiantes harán una descripción de todo lo sucedido durante el desarrollo del proyecto, con diálogo entre ellos y solicitando que sus apuntes, notas o comentarios sean considerados al hacer un informe que detalle todo el proceso. Esto con el objetivo de hacerlos conscientes del trabajo que realizaron y que lo puedan utilizar en otras situaciones a través de la generalización y o la transferencia de la experiencia de aprendizaje. Esto se refiere a la documentación posterior a la producción de la base de datos, para dar una retroalimentación de lo que se hizo y así explicar el proceso creativo para obtener la forma más óptima para resolver el problema principal del proyecto.

También pueden responder algunas preguntas en el informe descriptivo, por ejemplo:

- ¿Cómo llegaron a resolver el problema?
- ¿Cómo fue el proceso de asignar roles en el equipo?
- ¿Qué retos afrontaron a lo largo del proyecto?
- ¿Qué aspectos del proyecto fueron significativos para ustedes?
- ¿Qué podrían mejorar de la solución obtenida para utilizarla en otros contextos?

Finalmente, exponen y comparte sus soluciones con el curso o con la comunidad a través de medios digitales o de forma presencial.

Evaluación formativa

Se recomienda que las y los estudiantes realicen apuntes de los análisis y bocetos de los posibles modelos que surjan mientras se diseña. El registro de cada una de las fases del proyecto es la evidencia necesaria para realizar evaluaciones formativas por parte del profesorado. Acompañado de los apuntes realizados para los desempeños específicos de cada dato o evidencia, se recomienda utilizar pautas de evaluación basadas en escalas de apreciación y/o rúbricas que permitan constatar avances, fortalezas y elementos por mejorar.

Evaluación sumativa

Para evaluar sumativamente se sugiere considerar tanto los resultados finales, como cada una de las fases del proyecto. Para esto último, podrán utilizar las pautas aplicadas en la evaluación formativa y una pauta de evaluación de los productos finales del proyecto, como el ejemplo que se propone en Anexos.

Difusión final

Se propone realizar un informe que aborde todo lo sucedido en el proyecto, como las evidencias del análisis, los modelados y bocetos previos al resultado final del problema. Es necesario abarcar todos los detalles encontrados en la problemática principal y reflexionar sobre toda la experiencia y los procesos que permitieron crear la base de datos centralizada para una empresa de distribución de alimentos, además de evaluar cómo sus aprendizajes sobre el control de stock podría ser útiles en situaciones similares de su vida.

Recursos

- RECURSOS MATERIALES**
- Cuaderno o un block de notas para anotar datos o hacer bocetos de los modelos.
 - Computadores para trabajar con el modelamiento y posterior consulta de datos.
 - Computadores o celulares para investigar.

- RECURSOS**
- Aplicaciones:**
- Octostudio, Sololearn u otra.

Plataformas:

- Code.org

Programas:

- Oracle Data Modeler:
<https://www.oracle.com/cl/database/technologies/appdev/datamodeler.html>
- Oracle SQL Developer:
<https://www.oracle.com/tools/downloads/sqldev-downloads.html>
- Oracle Database 18c Express Edition:
<https://www.oracle.com/database/technologies/xe-downloads.html>
- Scratch
- PSeint

Anexo

Pauta de evaluación de proyecto

Como el o la docente tendrá que evaluar tanto el proceso como el producto, se recomienda aplicar esta pauta y tener a la vista la bitácora y/o los apuntes del estudiantado.

PAUTA PARA EVALUAR PROYECTOS				
	3	2	1	0
PLANIFICACIÓN DEL PROYECTO				
Descomponen el problema obteniendo la mayoría de los datos.				
Analizan el problema distinguiendo problemática principal.				
DESARROLLO DEL PROYECTO				
Modela base de datos que responde a la problemática.				
Prueban el funcionamiento de la consulta para la creación de tablas y la inserción de datos de forma efectiva.				
Integran los datos de manera correcta y son coherentes para el tipo de problema.				
Cumplen con las etapas del proyecto y completan la mayoría de las tareas a tiempo.				
INFORME DEL PROYECTO				
Describen detalladamente el proceso de obtención del producto final.				
Explican logros y desafíos del proyecto.				
Comunican la solución al problema.				
TOTAL				

Módulo Electivo 2

Aprendizaje Basado en Resolución de Problemas

Visión panorámica

Gran idea

El pensamiento algorítmico permite organizar pasos y reglas para diseñar soluciones y resolver una diversidad de problemas cuya comprensión se favorece al integrar conceptos matemáticos como variables, funciones, ecuaciones o gráficos, entre otros.

Objetivos de aprendizaje

OA4

Establecer conexiones y semejanzas entre las partes del problema para caracterizar patrones en una situación problemática dada. ([Generalizar y transferir](#))

OA5

Aplicar flexiblemente las soluciones elaboradas e implementadas a otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista. ([Generalizar y transferir](#))

OA6

Elaborar e implementar soluciones a una situación problemática mediante la formulación de algoritmos, actuando de acuerdo con los principios de la ética. ([Pensar de forma algorítmica](#))

Conocimientos esenciales

- Algoritmos y programación: lenguajes, métodos y entornos de programación.
 - Representación y análisis de datos.
 - Impacto social de la computación: interacciones, cultura, seguridad y ética.
-

Tiempo estimado: 6 semanas (12 horas)

Propósito del módulo electivo 2

En el módulo electivo basado en el aprendizaje a través de la resolución de problemas de la asignatura Pensamiento Computacional de los Niveles 1 y 2 de Educación Media, se espera que las y los estudiantes logren distinguir patrones con un lenguaje de programación y puedan resolver diferentes situaciones problema que pueden ocurrir en la creación de programas con interfaz. Asimismo, se espera que comprendan que los algoritmos permiten resolver una infinidad de problemas.

Esta comprensión permitirá a sus estudiantes internalizar de mejor manera las fases que puede llegar a tener un desarrollo informático y su correspondiente proceso de investigación, análisis y producción. También, experimentarán con diferentes algoritmos que pudieran llegar a necesitar alguna modificación para adaptarlos a lo que ellos necesiten.

Los Objetivos de Aprendizaje del módulo electivo 2 buscan desarrollar el pensar de forma algorítmica, generalizar y transferir. En particular, los y las estudiantes deberán aplicar sus aprendizajes en los diferentes módulos para cumplir con los desafíos propuestos, esto debido a que los retos que lleva resolver problemas y usar tecnologías son variables y responden a necesidades específicas de cada contexto.

Los Objetivos de Aprendizaje del módulo electivo 2 desarrollan las actitudes del siglo XXI del ámbito de las Maneras de pensar, Maneras de vivir el mundo y las Herramientas para trabajar promoviendo el aprovechar las herramientas y uso de la información para resolver problemas, diseñar soluciones e interactuar con sistemas en diferentes contextos de la vida diaria. Finalmente, las actividades propuestas son una oportunidad para integrar aprendizajes desde contextos reales y cercanos para el uso de las tecnologías y la alfabetización digital.

Problema

Una calculadora con interfaz gráfica

Propósito

En este módulo electivo, se busca desarrollar las habilidades de pensar de forma algorítmica, generalizar y transferir, mediante la metodología del aprendizaje basado en problemas. Para esto, se propone el contexto relacionado con la creación de una calculadora con interfaz gráfica, seleccionar los objetos que podría llegar a utilizar y la respectiva construcción en base a un lenguaje de programación determinado. Se favorecerá el desarrollo del pensamiento algorítmico para descubrir su funcionamiento, modelar una interfaz coherente con el problema e identificar los objetos a utilizar.

Preparación

Para que los y las estudiantes se preparen para el problema, se sugiere plantear el contexto de P.O.O. (Programación Orientada a Objetos) en el lenguaje de programación Python. Se recomienda presentar una definición y un contexto del tema para poder guiar a los y las estudiantes al conocimiento previo de lo que son las interfaces gráficas en los programas:

- ¿Qué es una calculadora con interfaz gráfica?
- ¿Para qué podrían utilizarla?
- ¿Cuál es la función principal y la función secundaria de una calculadora con interfaz gráfica?

En el sentido de investigar desde el problema establecido, se sugiere anotar todas las respuestas y, a partir de ellas, generar categorías para las diferentes características. Por ejemplo, se puede analizar si podría necesitarse algo extra en cuanto a lenguaje de programación, si se trata de algo diferente a lo ya antes establecido o qué forma podría tener la interfaz. Estas categorías, en particular la forma y la percepción de que se necesita algo más concreto para crear en Python, dan origen a la presentación del problema y favorecen formas de pensar más complejas, que luego se pueden generalizar y transferir, especialmente en la fase de prueba de la solución.

Presentación del problema

Para presentar el problema del módulo, se sugiere entregar el problema impreso y por medio de la escucha activa y la lectura comprensiva, promover que los y las estudiantes registren detalles de la situación general y de su propia situación con respecto al diseño y elaboración de una calculadora con interfaz gráfica:

Para poder calcular problemas matemáticos simples, muchas compañías tienen creadas sus propias calculadoras, incluso hay aplicaciones (App) incorporadas en teléfonos móviles. Sin embargo, tengo la necesidad de diseñar y elaborar una calculadora digital con interfaz gráfica para problemas matemáticos más complejos y poner atención al proceso necesario para su programación y los requisitos para su funcionamiento. ¿Qué debemos hacer para crear una interfaz gráfica? ¿Cómo me ayudan los algoritmos a enfrentarme a la creación de un objeto? ¿Qué necesito saber para obtener un modelo visual?

Algunas consideraciones para enfrentar el problema

- Diagnóstico de los recursos con los que se cuentan para poder diseñar y programar, de manera que sea posible responder al problema.
- Apoyos en información, por ejemplo, información de cómo se trabaja la programación orientada a objetos y cómo se utilizan, para que se pueda trabajar con objetos y no solo con información secuencial.
- Preguntas previas que permiten acercarse al problema, como: ¿Qué se necesita para crear una interfaz en Python? ¿Cómo puedo trabajar en base a objetos? ¿Qué tipo de algoritmo me permitiría diseñar la calculadora?
- ¿Cómo podríamos usar una interfaz gráfica para registrar préstamos de libros o herramientas?
- ¿Qué otros problemas podrían resolverse con un sistema visual interactivo?
- Organización para enfrentar el problema y las posibles variaciones que se pueden hacer.

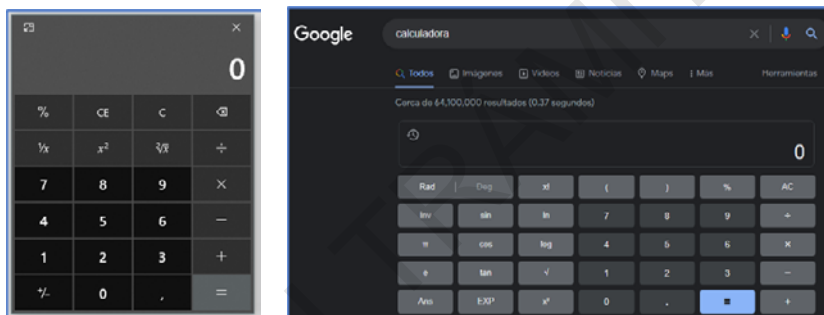
Los y las estudiantes asocian la situación con necesidades que pudieran darse en un contexto cotidiano o laboral, donde si bien tengo una solución computacional para una tarea, no satisface una necesidad específica. Esta etapa es importante, porque se trata esencialmente de saber cuál es la pregunta que se quiere responder y la motivación para responderla, así surgirán de manera natural los pasos hacia la solución y la necesidad del conocimiento.

Posibles soluciones al problema

Los y las estudiantes experimentan con diferentes formas para trabajar en el problema. Profundizan en el significado de trabajar con objetos en la programación, proponen posibles soluciones y establecen cómo empezar a desarrollar su algoritmo para que sea satisfactorio. También, observan otras calculadoras ya creadas que se pueden ver mediante internet o en el mismo sistema operativo utilizado (Windows, Android, Mac, etc.), para así poder identificar cómo funcionan y qué es lo que se requiere.

Desarrollo de estrategias de acción concreta

- Pantallazo de dos calculadoras en diferentes plataformas y probar usos, secuencias, funcionamientos, tipo de datos y forma:



Investigación

Los y las estudiantes indagan respecto del funcionamiento del P.O.O. (Programación Orientada a Objetos) y cómo se pueden crear interfaces en el lenguaje de programación Python. Luego, en caso de ser necesario, comparten la información con sus pares para generar mayor conocimiento común. Organizan la información en un informe descriptivo y aplican lo aprendido en una codificación de prueba para asegurar que la información adquirida servirá para resolver el problema.

La investigación que realiza cada estudiante enfatiza en la búsqueda de conocimientos que le ayuden a comprender la programación orientada a objetos, los algoritmos más comunes y a generalizar las relaciones que existen entre la programación, el lenguaje y la interfaz.

Las nuevas preguntas que pueden organizar esta etapa podrían ser:

- ¿Dónde sería necesario buscar datos de P.O.O e interfaces?
- ¿Para qué sirve la Programación Orientada a Objetos?
- ¿Qué herramientas servirán para lograr hacer una interfaz gráfica?
- ¿Cómo funcionan las calculadoras mediante software?
- ¿Cómo los aprendizajes en la asignatura ayudarán a resolver el problema?
- ¿Qué otros ejemplos puedo encontrar de programación orientada a objetos?

Estas preguntas ayudan a orientar la investigación y recopilar información necesaria para crear o modificar algoritmos posibles de usar en la programación orientada a objetos. Los apoyos de información encontrados pueden ser incorporados en el informe del proceso de obtención de la solución de manera que se genere una fuente de conocimiento individual o de equipos posibles de alimentar a través del tiempo.

Evaluar una solución del problema

Los estudiantes evalúan la solución al problema y transfieren la misma a otras problemáticas para validarla. Se sugiere utilizar la información obtenida en el proceso de investigación para analizar la solución y, a su vez, considerar si es necesario obtener más para, luego, dar paso a un proceso de comparación (semejanzas y diferencias) y lo factible que puede llegar a ser usar el modelo o una modificación de este para resolver el problema.

Se recomienda presentar el desarrollo de la situación, por ejemplo: qué método se utiliza para poder obtener una interfaz gráfica y su funcionamiento. El método para utilizar se puede obtener de la página web <https://docs.python.org/es/3/library/tk.html>. Aquí se ven algunos ejemplos de las funcionalidades que puede llegar a tener el incorporar **un método** en Python.

Métodos	Descripción
Grid	El método grid permite posicionar los widgets en una celda en específico. Indicamos la celda usando el índice de fila y columna correspondiente; el ancho y la altura de cada celda son configurables. Además, un widget puede ocupar varias celdas si lo deseamos. Al usar grid podemos crear fácilmente interfaces gráficas de usuario tipo formulario.
Label	El método label son textos estáticos, de ahí que se llame label o etiqueta de texto.
Entry	Una caja de texto que permite al usuario ingresar cualquier texto de una línea.
Text	Una caja de texto que permite al usuario ingresar cualquier texto en varias líneas.

Además de describir las funcionalidades que pueden llegar a ejercer los métodos antes mencionados, es requerido investigar **los modificadores u opciones** que pueden llegar a tener.

Opciones	Descripción
bg	El color de fondo predeterminado del widget de texto.
bd	El ancho del borde alrededor del widget de texto. El valor predeterminado es 2 píxeles.
font	La fuente predeterminada para el texto insertado en el widget.

Se sugiere presentar la definición de las opciones en un cuadro y describir los métodos en orden, para así inscribir cuáles opciones sirven en cada método. Además, se pedirá una investigación completa de las funcionalidades P.O.O. y para qué sirven en el uso de la programación. Por lo tanto, es necesario dar ejemplos y descripciones de lo que se necesita para trabajar con dicha información.

Para finalizar, se requiere un análisis de lo propuesto en la investigación, es decir, se requiere una conclusión y especificación de lo que se utilizará para desarrollar la solución al problema: calculadora con interfaz gráfica, para representar un pensamiento algorítmico previo a la producción del programa.

Elaborar y ajustar la investigación

Los y las estudiantes describen diferentes funcionalidades, métodos, opciones y widgets capaces de proporcionar lo necesario para crear una calculadora en el lenguaje de programación de Python. Para esto, se elabora una investigación y descripción de los diferentes métodos y opciones que pueden incorporarse y adaptarse al problema principal:

- La consideración de diferentes fuentes para poder crear interfaces gráficas debería quedar expresada según la compatibilidad y facilidad de uso que se podría llegar a tener.
- Las definiciones obtenidas se evalúan bajo la pregunta: ¿Sirve esta información para cumplir con la solución? ¿Sirve esta información y sus definiciones para lograr lo requerido?
- La evaluación de la investigación debe describir una serie de funcionalidades que se puedan utilizar en el lenguaje de programación correspondiente y así tener una selección de lo que se podría usar.

Comunicar

Los resultados obtenidos son comunicados y comparados entre ellos, por ejemplo: los diferentes métodos que se pueden utilizar para diseñar interfaces gráficas en Python, los modificadores u opciones que estos pueden tener, cómo se podría diseñar una calculadora con interfaz gráfica, entre otros. También se puede reflexionar sobre las estrategias utilizadas en el proceso de investigación, sobre los algoritmos que se podrían utilizar en el proceso creativo, la diferencia de los datos obtenidos, o la posibilidad de transferir los conocimientos obtenidos a otras situaciones.

Algunas de las preguntas que pueden guiar la conversación son:

- ¿Qué tiene en común los métodos y funciones?
- ¿A qué pregunta nueva o diferente responde el modelo creado respecto de los investigados?
- ¿Qué métodos son utilizados en mi modelo en comparación con otros?

Se sugiere comunicar la información y las sugerencias mediante alguna de estas posibilidades:

- Presentación
- Informe
- Video
- Simulación

Orientaciones al docente

Para unificar conceptos disciplinares

Se sugiere precisar las opciones que pueden variar requiriendo de cada información adquirida por cada estudiante. Cabe resaltar que el método de “Tkinter” no es el único conocido para crear interfaces gráficas dentro de Python, también existe “PySimpleGUI”. Además, se puede usar un sistema de capas junto a HTML para crear una interfaz web junto a “Django”. El límite puede variar dependiendo de lo que se busque y se quiera hacer dentro del lenguaje, pero es seguro que el método más adecuado para esta experiencia es “Tkinter” por su fácil acceso y sus funcionalidades.

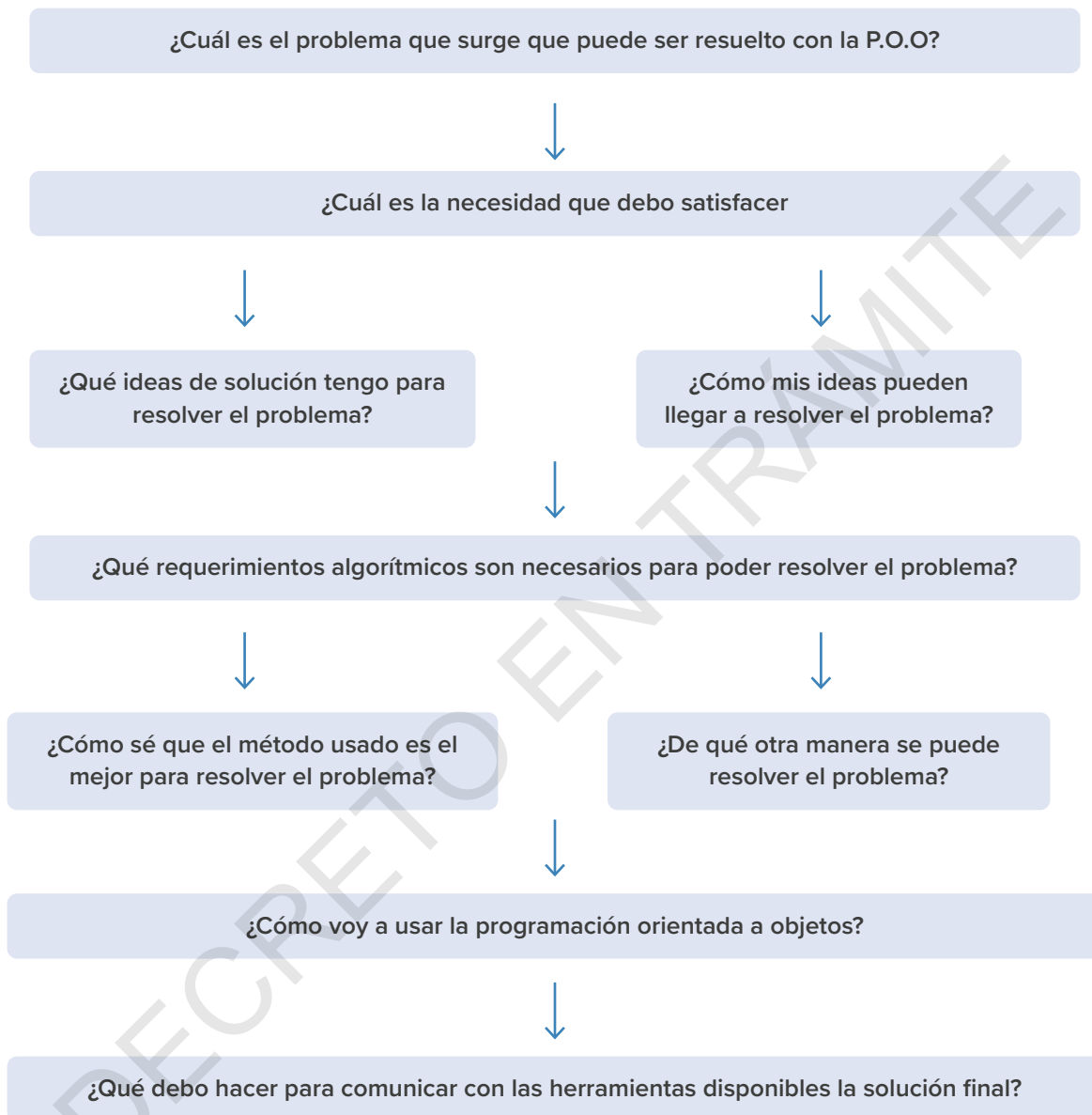
El uso que se le puede dar a cada una de ellas dependerá de la investigación que se haga al respecto y, con ello, la comprensión que se pueda dar a la información en general y particular. Es esencial que la o el docente haga su propia investigación para orientar a sus estudiantes.

Para focalizar el desarrollo de habilidades

Aunque la actividad desarrolla varias habilidades del lenguaje de programación y cómo se pueden incorporar métodos externos que fueron diseñados para este sistema, se sugiere focalizar el desarrollo en solo una de ellas. Para esto se puede considerar la siguiente estrategia interrogativa, para enfocar el desarrollo de la habilidad pensar de forma algorítmica.

- ¿Cuál es el problema que surge y que puede ser resuelto con la P.O.O?
- ¿Cuál es la necesidad que debo satisfacer?
- ¿Qué ideas de solución tengo para resolver el problema?
- ¿Cómo mis ideas pueden llegar a resolver el problema?
- ¿Qué requerimientos algorítmicos son necesarios para poder resolver el problema?
- ¿Cómo sé que el método usado es el mejor para resolver el problema?
- ¿De qué otra manera se puede resolver el problema?
- ¿Cómo voy a usar la programación orientada a objetos?
- ¿Qué debo hacer para comunicar con las herramientas disponibles la solución final?

Además, poner a disposición de los y las estudiantes un organizador gráfico que puede ser utilizado como bitácora para el proceso de solución al problema:



Anexo

Bibliografía

Sitios web

Hoja de trabajo

Bibliografía

- ARAYA, R., ISODA, M., GONZÁLEZ, O., E INPRASITHA, M. (2019). *A Framework for Computational Thinking for the Transition to a Super Smart Society*. APEC InMside Report.
- ARTECONA, F., BONETTI, E., DARINO, C., MELLO, F., ROSÁ, M., Y SCÓPISE, M. (2017). *Pensamiento Computacional*. Un Aporte para la Educación de Hoy. Fundación Telefónica. <https://www.fundaciontelefonica.cl/arte-y-cultura-digital/publicaciones-listado/pagina-item-publicaciones/itempubli/618/>
- BASOGAIN, X., OLABE, J., Y DEL ROSARIO, R. (2016). *PC-01: Introduction to computational thinking: Educational technology in primary and secondary education - IEEE Conference Publication*. <https://ieeexplore.ieee.org/document/7751816/citations?tabFilter=papers>
- CAS (2015). *Pensamiento Computacional*. Guía para profesores. Computing At School. <http://www.codemas.org/wp-content/uploads/2016/04/Pensamiento-computacional-Guía-para-profesores.pdf>
- FÁBREGA, R., JARA, M., CARREÑO, E., Y OSORIO, M. (2016). *Enseñanza de Lenguajes de Programación en la Escuela ¿Qué están haciendo en otros países?* Fundación Telefónica. <http://www.fundaciontelefonica.cl/wp-content/uploads/descargas/1504030044-Ense%C3%B1anza%20de%20Lenguajes%20de%20Programaci%C3%B3n%20en%20otros%20pa%C3%ADses.pdf>
- GROVER, S. & PEA, R. (2013). *Computational Thinking in K–12: A Review of the State of the Field*. Educational Researcher, 42(1), 38-43.
- ISTE (2011). *Operational definition of computational thinking for K-12 education*. NSF, CSTA, ISTE. https://cdn.iste.org/www-root/Computational_Thinking_Operational_Definition_ISTE.pdf?_ga=2.79189782.305816076.1642012397-617136009.1642012397
- JARA, M., Y FÁBREGA, R. (2017). *Bases para la implementación de la Enseñanza de la Programación en el Sistema Escolar: Exploración de Estrategias Públicas y Privadas*. Fundación Telefónica. <https://www.fundaciontelefonica.cl/arte-y-cultura-digital/publicaciones-listado/pagina-item-publicaciones/itempubli/641/>
- LYE, S. Y., Y KOH, J. H. L. (2014). *Review on teaching and learning of computational thinking through programming: What is next for K-12?* Computers in Human Behavior, 41, 51–61. <https://doi.org/10.1016/j.chb.2014.09.012>
- MINISTERIO DE EDUCACIÓN DE CHILE (2019). *Bases curriculares 3° y 4° medio: Plan de formación general, plan de formación diferenciada humanístico-científico*. https://www.curriculumnacional.cl/614/articles-91414_bases.pdf
- MINISTERIO DE EDUCACIÓN, UCE (2019). *¿Qué es ABP?* <https://www.curriculumnacional.cl/porta/Secciones/ABP-Chile-aprende-por-proyectos/134607:Que-es-ABP>

- MINISTERIO DE EDUCACIÓN, UCE (2019). *Metodología de aprendizaje basado en proyectos*. <https://www.curriculumnacional.cl/portal/Curso/Tecnico-Profesional/3-Medio-TP/140166:Metodologia-de-aprendizaje-basado-en-proyectos>
- MINISTERIO DE EDUCACIÓN DE CHILE. (2023). *Ciudadanía digital para los desafíos de las comunidades educativas*. Centro de Innovación MINEDUC.
- MINISTRY OF EDUCATION (2017). *The New Zealand curriculum*. Technology Learning Area. <http://nzcurriculum.tki.org.nz/The-New-Zealand-Curriculum>
- RESNICK, M. (2007). *All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten*. C&C '07 Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition. Washington D. C.
- RICO LUGO, M. J. Y BOSAGAIN OLABE, X. (2018). *Pensamiento computacional: rompiendo brechas digitales y educativas*. EDMETIC, Revista de Educación Mediática y TIC, 7(1), 26-42. <https://doi.org/10.21071/edmetic.v7i1.10039>
- TEDRE, M. Y DENNING, P. J. (2016). *The Long Quest for Computational Thinking*. Proceedings of the 16th Koli Calling Conference on Computing Education Research, November 24-27, 2016, Koli, Finland: pp. 120-129.
- VIVIAN, R., FALKNER, K., & FALKNER, N. (2014). *Addressing the challenges of a new digital technologies curriculum: MOOCs as a scalable solution for teacher professional development*. Research in Learning Technology, 22 (Special Issue: Riding Giants: How to innovate and educate ahead of the wave), 1–19.
- WING, J. (2006). *Computational Thinking*. It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use. Communications of the ACM, 49(3), 33-35. <https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>
- WING, J. (2011). *Computational thinking - What and why?* *The Link*, (6), 20-23. https://www.cs.cmu.edu/sites/default/files/11-399_The_Link_Newsletter-3.pdf
- ZAPATA, M. (2015). *Pensamiento computacional: Una nueva alfabetización digital*. RED-Revista de Educación a Distancia, 46(4). <https://revistas.um.es/red/article/view/240321/183001>

Sitios web

Ejercicio Imc:

<https://eligevivirsano.gob.cl/calculadora-de-indice-de-masa-corporal/>

Funciones de Pseint:

<https://pseint.sourceforge.net/index.php?page=documentacion.php>

Programación por bloques:

<https://code.org/es/teachers>

<https://scratch.mit.edu/ideas>

Programación y diseño:

<https://www.tinkercad.com/>

<https://www.rstudio.com/>

Fundaciones para el desarrollo tecnológico:

<https://Cuantrix.mx/>

<https://www.curriculumnacional.cl/estudiantes/Aprendo-en-linea/Programa-CODE-ORG/238620:CODE-ORG>

<https://iste.org/>

<https://csteachers.org/>

Hoja de trabajo

Zona de escritura de código

Línea	Instrucción (Código o Bloque)	Tipo de instrucción o comentarios
1		DECRETO EN TRÁMITE
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		

DECRETO EN TRÁMITE