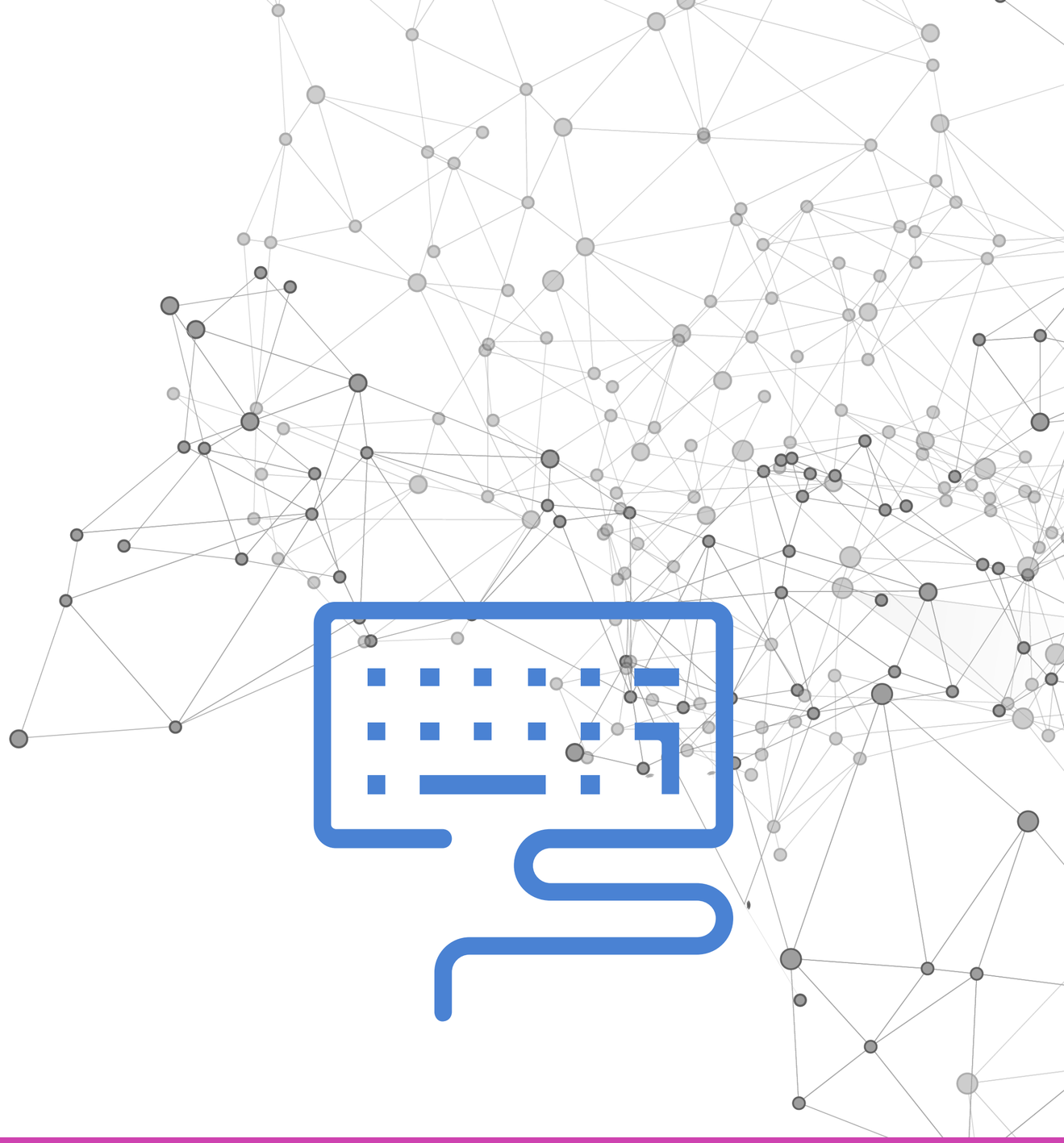
Ministerio de Educación
Gobierno de Chile

UNIVERSIDAD CATÓLICA DE TEMUCO

Diccionarios y programabilidad de las redes



Módulo 3: Instalación y configuración de redes.

Telecomunicaciones

Perfil de Egreso - Objetivos de Aprendizaje de la Especialdad

Módulo 1

OA1 Leer y utilizar esquemas, proyectos y en general todo el lenguaje simbólico asociado a las operaciones de montaje y mantenimiento de redes de telecomunicaciones.

Módulo 2

OA6 Realizar mantenimiento y reparaciones menores en equipos ysistemas de telecomunicaciones, utilizando herramientas y pautas de mantención establecidas por el fabricante.

OA7 Aplicar la normativa y los implementos de seguridad y protección relativos almontaje y el mantenimiento de las instalaciones de telecomunicaciones y la normativa del medio ambiente.

Módulo 3

OA2 Instalar equipos y sistemas de telecomunicaciones de generación, transmisión, repetición, amplificación, recepción, y distribución de señal de voz, imagen y datos, según solicitud de trabajo y especificaciones técnicas del proyecto.

OA10 Determinar los equipos y sistemas de comunicación necesarios para una conectividad efectiva y eficiente, de acuerdo a los requerimientos de los usuarios.

Módulo 4

OA9 Detectar y corregir fallas en circuitos de corriente continua de acuerdo a los requerimientos técnicos y de seguridad establecidos.

Módulo 5

OA2 Instalar equipos y sistemas de telecomunicaciones de generación, transmisión, repetición, amplificación, recepción y distribución de señal de voz, imagen y datos, según solicitud de trabajo y especificaciones técnicas del proyecto.

OA4 Realizar medidas y pruebas de conexión y de continuidad de señal eléctrica, de voz, imagen y datos- en equipos, sistemas y de redes de telecomunicaciones, utilizando instrumentos de medición y certificación de calidad de la señal autorizada por la normativa vigente.

Módulo 6

OA8 Instalar y configurar una red inálambrica según tecnologías y protocolos establecidos.

OA7 Aplicar la normativa y los implementos de seguridad y protección relativos al montaje y el mantenimiento de las instalaciones de telecomunicaciones y la normativa del medio ambiente.

Módulo 7

OA5 Instalar y configurar una red de telefonía (tradicional o IP) en una organización según los parámetros técnicos establecidos.

Módulo 8

OA3 Instalar y/o configurar sistemas operativos en computadores o servidores con el fin de incorporarlos a una red laN, cumpliendo con los estándares de calidad y seguridad establecidos.

Módulo 9

OA10 Determinar los equipos y sistemas de comunicación necesarios para una conectividad efectiva y eficiente, de acuerdo, a los requerimientos de los usuarios.

OA6 Realizar el mantenimiento y reparaciones menores en equipos y sistemas de telecomunicaciones, utilizando herramientas y pautas de mantención establecidas por el fabricante.

Módulo 10

No está asociado a Objetivos de Aprendizaje de la Especialidad (AOE), sino a genéricos. No obstante, puede asociarse a un OAE como estrategia didáctica.

Perfil de Egreso – Objetivos de Aprendizaje Genéricos

A- Comunicarse oralmente y por escrito con claridad, utilizando registros de habla y de escritura pertinentes a la situación laboral y a la relación con los interlocutores.	**B-** Leer y utilizar distintos tipos de textos relacionados con el trabajo, tales como especificaciones técnicas, normativas diversas, legislación laboral, así como noticias y artículos que enriquezcan su experiencia laboral.	**C-** Realizar las tareas de manera prolija, cumpliendo plazos establecidos y estándares de calidad, y buscando alternativas y soluciones cuando se presentan problemas pertinentes a las funciones desempeñadas.
D- Trabajar eficazmente en equipo, coordinando acciones con otros in situ o a distancia, solicitando y prestando cooperación para el buen cumplimiento de sus tareas habituales o emergentes.	**E-** Tratar con respeto a subordinados, superiores, colegas, clientes, personas con discapacidades, sin hacer distinciones de género, de clase social, de etnias u otras.	**F-** Respetar y solicitar respeto de deberes y derechos laborales establecidos, así como de aquellas normas culturales internas de la organización que influyen positivamente en el sentido de pertenencia y en la motivación laboral.
G- Participar en diversas situaciones de aprendizaje, formales e informales, y calificarse para desarrollar mejor su trabajo actual o bien para asumir nuevas tareas o puestos de trabajo, en una perspectiva de formación permanente.	**H-** Manejar tecnologías de la información y comunicación para obtener y procesar información pertinente al trabajo, así como para comunicar resultados, instrucciones e ideas.	**I-** Utilizar eficientemente los insumos para los procesos productivos y disponer cuidadosamente los desechos, en una perspectiva de eficiencia energética y cuidado ambiental.
J- Emprender iniciativas útiles en los lugares de trabajo y/o proyectos propios, aplicando principios básicos de gestión financiera y administración para generarles viabilidad.	**K-** Prevenir situaciones de riesgo y enfermedades ocupasionales, evaluando las condiciones del entorno del trabajo y utilizando los elementos de protección personal según la normativa correspondiente.	**L-** Tomar decisiones financieras bien informadas, con proyección a mediano y largo plazo, respecto del ahorro, especialmente del ahorro previsional, de los seguros, y de los riesgos y oportunidades del endeudamiento crediticio así como de la inversión.

Marco de Cualificaciones Técnico Profesional (MCTP) Nivel 3 y su relación con los OAG

HABILIDADES

1. Información
1. Analiza y utiliza información de acuerdo a parámetros establecidos para responder a las necesidades propias de sus actividades y funciones.
2. Identifica y analiza información para fundamentar y responder a las necesidades propias de sus actividades.

2. Resolución de problemas
1. Reconoce y previene problemas de acuerdo a parámetros establecidos en contextos conocidos propios de su actividad o funcion.
2. Detecta las causas que originan problemas en contextos conocidos de acuerdo a parámetros establecidos.
3. Aplica soluciones a problemas de acuerdo a parámetros establecidos en contextos conocidos propios de una función.

3. Uso de recursos
1. Selecciona y utiliza materiales, herramientas y equipamiento para responder a una necesidad propia de una actividad o función especializada en contextos conocidos.
2. Organiza y comprueba la disponibilidad de los materiales, herramientas y equipamiento.
3. Identifica y aplica procedimientos y técnicas específicas de una función de acuedo a parámetros establecidos.

4. Comunicación
4. Comunica y recibe información relacionada a su actividad o función, a través de medios y soportes adecuados en contextos conocidos.

APLICACIÓN EN CONTEXTO

5. Trabajo con otros
1. Trabaja colaborativamente en actividades y funciones coordinándose con otros en diversos contextos.

6. Autonomía
1. Se desempeña con autonomía en actividades y funciones especializadas en diversos contextos con supervisión directa.
2. Toma decisiones en actividades propias y en aquellas que inciden en el quehacer de otros en contextos conocidos.
3. Evalua el proceso y el resultado de sus actividades y funciones de acuerdo a parámetros establecidos para mejorar sus prácticas.
4. Busca oportunidades y redes para el desarrollo de sus capacidades

7. Ética y responsabilidad
1. Actúa de acuerdo a las normas y protocolos que guían su desempeño y reconoce el impacto que la calidad de su trabajo tiene sobre el proceso productivo o la entrega de servicios.
2. Responde por cumplimiento de los procedimientos y resultados de sus actividades.
3. Comprende y valora los efectos de sus acciones sobre la salud y la vida, la organización, la sociedad y el medio ambiente.
4. Actúa acorde al marco de sus conocimientos. experiencias y alcance de sus actividades y funciones

CONOCIMIENTO

8. Conocimientos
1. Demuestra conocimientos especificos de su área y de las tendencias de desarrollo para el desempeño de sus actividades y funciones.

Metodología seleccionada

Demostración Guiada

- Esta presentación les ayudará a poder comprender los conceptos necesarios para el desarrollo de su actividad.

Aprendizaje Esperado

- **AE 4.** Diseñar programas de mediana complejidad, que involucren sentencias, estructuras y programación modular en Python para la solución de problemas, de acuerdo a los requerimientos de su especialidad y contexto laboral.

¿Qué vamos a lograr con esta actividad para llegar al Aprendizaje Esperado (AE)?

Diseñar aplicaciones en Python, utilizando sentencias de python y comandos de redes en el desarrollo de aplicaciones de su contexto.

Contenidos

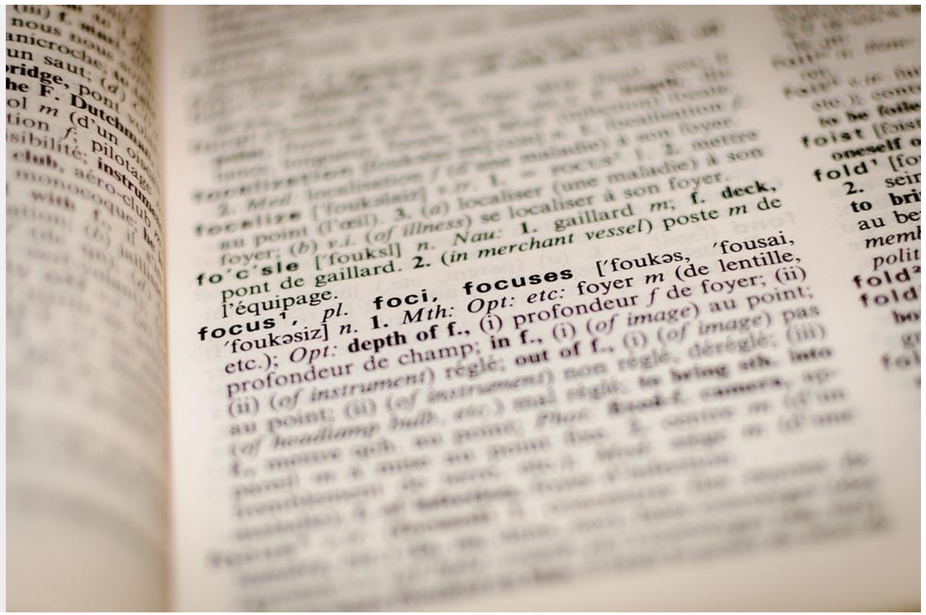
01 DICCIONARIOS

- Definición.
- Características de las claves.
- Casos de uso.
- Creación de un diccionario.
- Mostrar contenido de un diccionario.
- Métodos de diccionarios.

02 PROGRAMABILIDAD DE LAS REDES

- Automatización de las redes.
- Beneficios de la automatización de las redes.
- El script.
- Ejemplos de uso.

Diccionarios



https://pixabay.com/es/photos/diccionario-enfoque-libro-1149723/

Antes de comenzar, reflexionemos...

1. ¿Qué es un diccionario?
2. ¿Para qué sirve?
3. ¿Qué elementos contiene?
4. ¿Cómo se utiliza?

Definición

- Un **Diccionario en Python** es una estructura de datos modificable, con características especiales, que nos permite almacenar cualquier tipo de valor como: enteros, cadenas, listas, etc.

- Estos diccionarios nos permiten además identificar cada elemento por una clave (Key).

- **D = {CLAVE1:VALOR, CLAVE2:VALOR, … , CLAVEN:VALOR}**

Definición

- Para definir un diccionario, se encierra el listado de valores entre llaves.

- Las parejas de clave y valor se separan con comas, mientras que la clave y el valor se separan con dos puntos.

- **D = {NOMBRE:”JUAN PEREZ”, CURSO:”1A”, EDAD: 14}**

Caracteristicas de las claves

01

- Como pueden observar, todos los elementos de un diccionario son pares de valores que tienen la siguiente estructura:

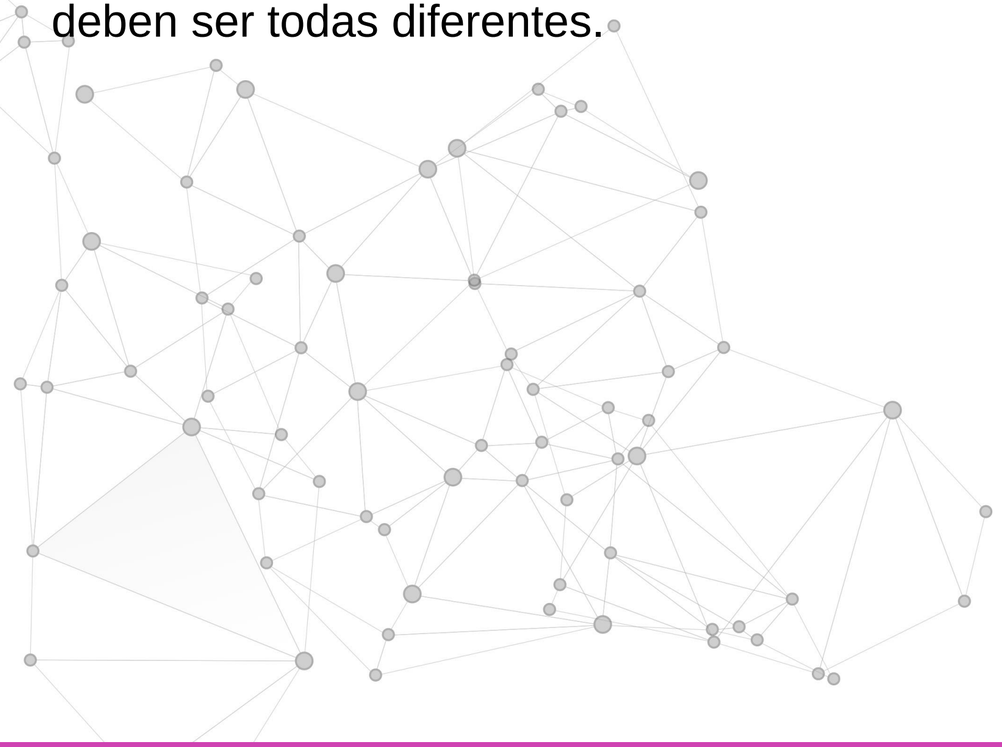
CLAVE : VALOR

02

- La clave, corresponde al nombre con el cual se accederá al valor y es sensible al tamaño.

03

- Las claves no se pueden repetir, ya que identifican el valor con su nombre, por lo cual deben ser todas diferentes.



Casos de usos

01

Una agenda de contactos, con el nombre del contacto y los datos asociados a cada uno.

02

Una lista de alumnos asociando cada alumno con su correspondiente nota, e incluso con más de una nota cada uno.

03

Un diccionario de términos, que asocie cada termino con su significado correspondiente, o con una lista de sinónimos o antónimos.

Casos de usos

04

Un traductor de palabras, que asocie cada palabra con su correspondiente traducción a ingles, francés, etc.

05

Una lista de nombres de usuarios, cada uno asociado a su clave correspondiente.

Creación de un diccionario

01

Para crear un objeto de tipo dict (diccionario) primero se debe definir de la siguiente manera:

D = { }

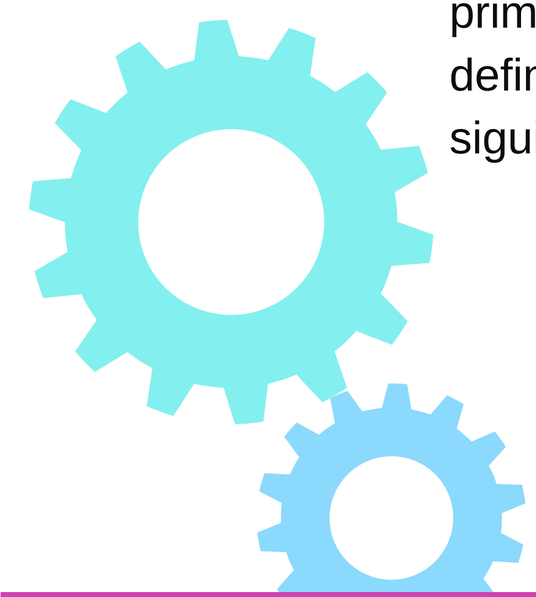
02

Para ingresar valores al diccionario se debe hacer de la siguiente manera:

D[clave] = valor

03

Veamos el ejemplo para ingresar 5 elementos a un diccionario.



Creación de un diccionario

```python
D={}
for i in range(5):
    clave=input("Ingrese Clave: ")
    valor=input("Ingrese Valor: ")
    D[clave]=valor
print(D)
```

Fuente propia

CODIGO PYTHON

SALIDA POR PANTALLA

```
Ingrese Clave: A
Ingrese Valor: 111
Ingrese Clave: B
Ingrese Valor: 222
Ingrese Clave: C
Ingrese Valor: 333
Ingrese Clave: D
Ingrese Valor: 444
Ingrese Clave: E
Ingrese Valor: 555
{'A': '111', 'B': '222', 'C': '333', 'D': '444', 'E': '555'}
```

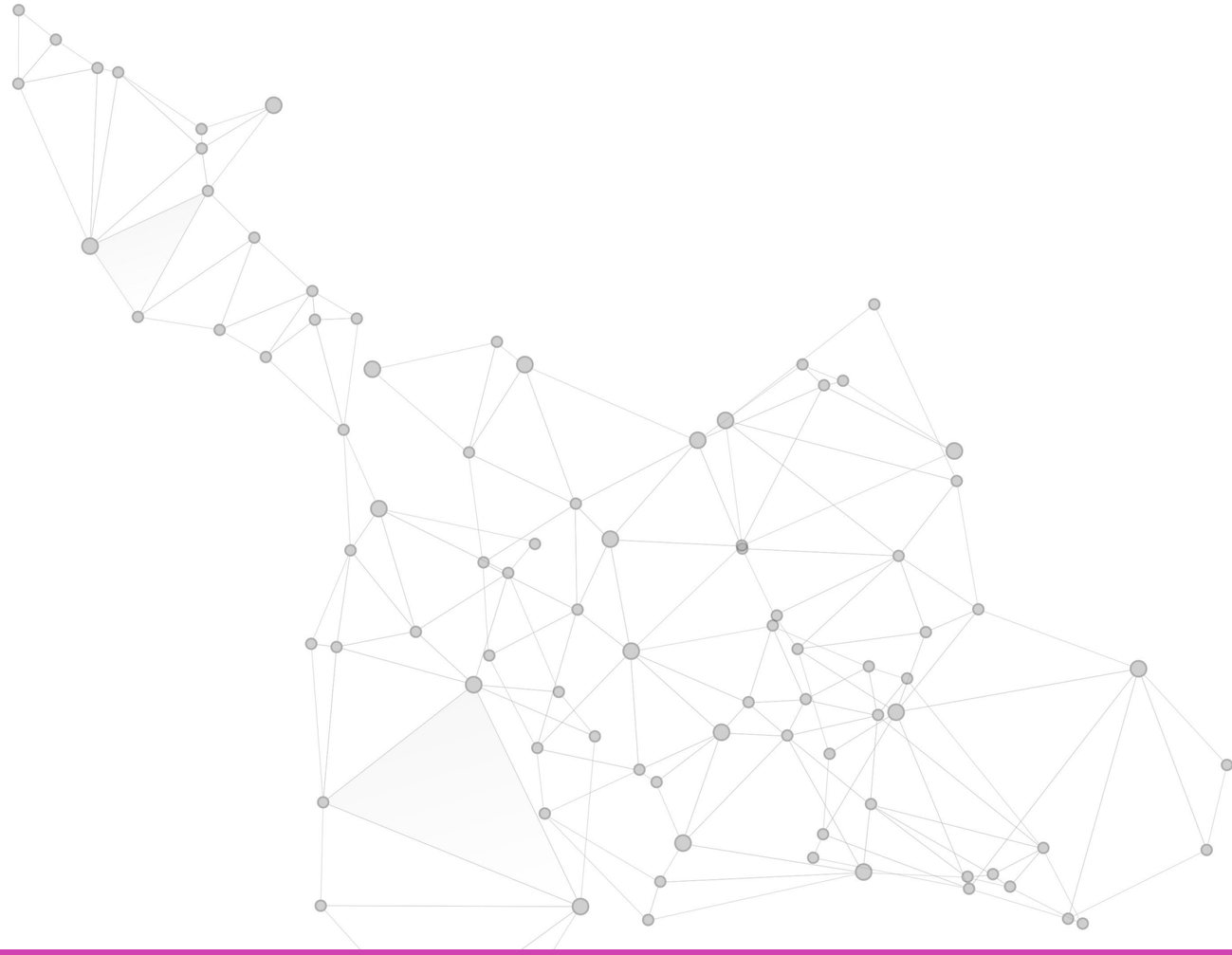
Fuente propia

Mostrar contenido de un diccionario

- Para mostrar el contenido de un diccionario se puede hacer de muchas maneras:

    1. **Mostrar todo el diccionario.**
    2. **Mostrar todo el diccionario por clave y valor (2 formas).**
    3. **Mostrar solo claves.**
    4. **Mostrar solo valores.**

- A continuación veremos cada una de ellas…



Mostrar contenido de un diccionario

```
for key,value in D.items():
    print(key,":",value)
```

```
A : 111
B : 222
C : 333
D : 444
```

```
for key in D.keys():
    print("> ",key)
```

```
>  A
>  B
>  C
>  D
```

Fuente propia

Mostrar contenido de un diccionario

```
print("D = ",D)
```

```
D =  {'A': '111', 'B': '222', 'C': '333', 'D': '444'}
```

```
for clave in D:
    print(clave,":",D[clave])
```

```
A : 111
B : 222
C : 333
D : 444
```

```
for value in D.values():
    print("> ",value)
```

```
>  111
>  222
>  333
>  444
```

Fuente propia

Agregar/actualizer elementos

- **UPDATE:** Permite insertar o actualizar un elemento del diccionario.

- En este caso se agrega la clave ‘sexo’, ya que no existía en el diccionario

```
D = {'nombre' : 'Juan','apellido':'Perez','edad':22}

D.update({'sexo':'masculino'})
print(D)
```

```
{'nombre': 'Juan', 'apellido': 'Perez', 'edad': 22, 'sexo': 'masculino'}
```

Fuente propia

- En este caso se modifica el valor de la clave ‘nombre’, porque ya existía en el diccionario

```
D = {'nombre' : 'Juan','apellido':'Perez','edad':22}

D.update({'nombre':'Maria'})
print(D)
```

```
{'nombre': 'Maria', 'apellido': 'Perez', 'edad': 22}
```

Fuente propia

Eliminar elementos

- **DEL:** Elimina la clave y por ende el valor asociado.
- **POP:** Elimina un elemento por nombre de clave.
- **POPITEM:** Elimina el ultimo elemento.
- **CLEAR:** Vacía el contenido de un diccionario

```
D={'A':1,'B':2,'C':3,'D':4}
del D['B']
print('D = ',D)
```

```
D =  {'A': 1, 'C': 3, 'D': 4}
```

DEL

Fuente propia

```
D={'A':1,'B':2,'C':3,'D':4}
D.pop('B')
print('D = ',D)
```

```
D =  {'A': 1, 'C': 3, 'D': 4}
```

POP

Fuente propia

```
D={'A':1,'B':2,'C':3,'D':4}
D.popitem()
print('D = ',D)
```

```
D =  {'A': 1, 'B': 2, 'C': 3}
```

POPITEM

Fuente propia

```
D={'A':1,'B':2,'C':3,'D':4}
D.clear()
print('D = ',D)
```

```
D =  {}
```

CLEAR

Fuente propia

Copiar diccionarios

- **COPY:** Permite hacer una copia de un diccionario.

```
D1={'nombre':'Juan', 'edad':22,'sexo':'M'}

D2=D1.copy()

print("D1 = ",D1)
print("D2 = ",D2)
```

```
D1 =  {'nombre': 'Juan', 'edad': 22, 'sexo': 'M'}
D2 =  {'nombre': 'Juan', 'edad': 22, 'sexo': 'M'}
```

Fuente propia

Largo de un diccionario

- **LEN:** Permite determinar el largo de un diccionario (cuántos elementos tiene).

```
D={'nombre':'Juan', 'edad':22,'sexo':'M'}

LARGO=len(D)

print("el largo del diccionario es: ",LARGO)
```

```
el largo del diccionario es:  3
```

Fuente propia

¿Cuáles son los pasos para crear un diccionario?

Programabilidad de las redes con Python

Reflexionemos

Si quisiéramos crear 100 usuarios en la red, ¿Cómo lo haríamos de manera rápida y efectiva?

Automatización de las redes

01 Hoy en día todos los procesos que se realizan en las empresas están automatizados a través de aplicaciones que los hacen más precisos y eficientes.

02 ¿Cuántas automatizaciones que has observado podrías mencionar?

03 Por ejemplo el pago de cuentas online, o un autoservicio en una bencinera, las cajas rápidas de autoservicio en una tienda, las compras online, el acceso a portales de ISAPRES, AFP, registro civil o SII para descarga de documentos, etc.

Beneficios de la automatización de las redes

01
- Producción continua por uso de máquinas 24/7.

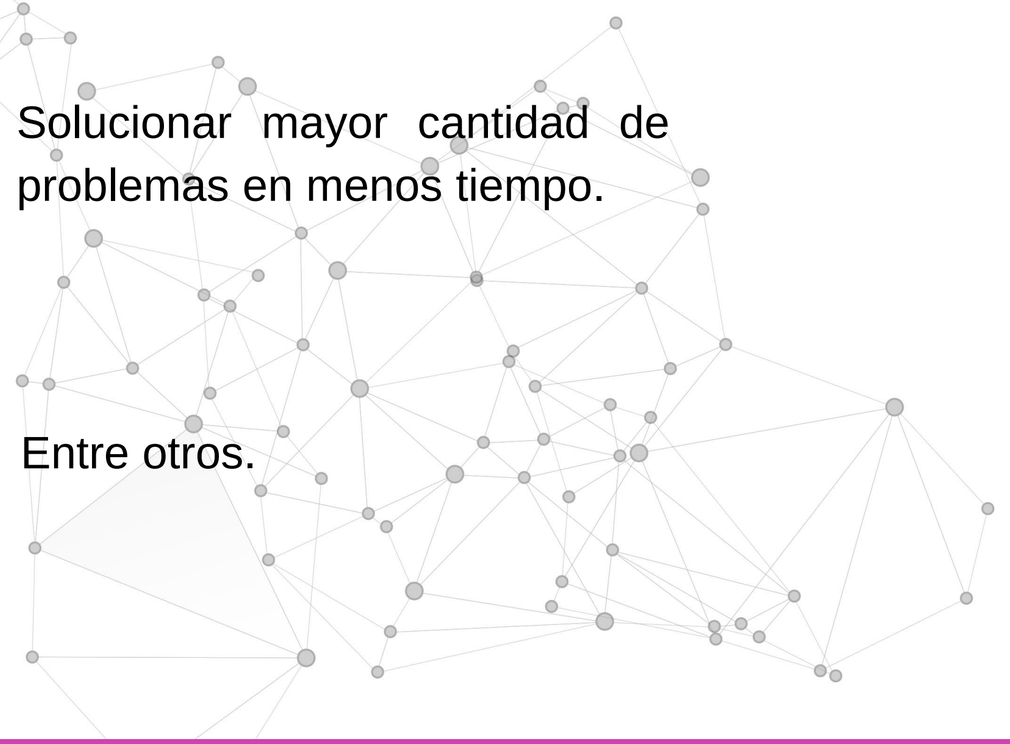
02
- Manejo de grandes cantidades de información de manera rápida.

03
- Resultados más precisos.

04
- Minimización de tiempos de respuesta.

05
- Solucionar mayor cantidad de problemas en menos tiempo.

06
- Entre otros.



Pensemos…

¿Has escuchado la palabra “script”?

¿Sabes qué es un script?

El script

01 Para diseñar aplicaciones que nos permitan automatizar procesos de redes a través de Python utilizaremos ‘scripts’.

02 Un script es un archivo, con extensión .py, escrito en Python, que contiene un código que podrá ser ejecutado completo línea por línea.

03 A continuación veremos algunos ejemplos de la web, sobre scripts sencillos que se pueden aplicar a las redes, y explicaremos para qué sirve cada uno de ellos.

Ejemplo 1 – Chequeo de PING

01

- Este script permite hacer ping a una dirección web y saber si está disponible o no. En este caso, se hace un llamado a Google, y dependiendo de la respuesta del sitio, se muestra un mensaje en consola.

- Se observan elementos conocidos como ‘import’ para utilizar módulos predefinidos de Python, y la sentencia ‘def’ que define la función de chequeo de la conexión.

- La sentencia ‘get’ permite obtener la conexión y la sentencia ‘try/except’ permite chequear el tipo de error recibido.

```
from requests import get, exceptions

def check_internet_connection():
    try:
        get('http://google.com', timeout = 3)
        print('connected')
    except exceptions.ConnectionError:
        print('not connected')

check_internet_connection()
```

Ejemplo tomado del autor Eduardo Saavedra en la pagina Medium.com

Ejemplo 2 – Genera claves

02

```python
import random
import string

def crear_pass(n):
    allChars = list(string.ascii_letters) + list(string.digits) + list(string.punctuation)
    passphrase = []

    for i in range(n):
        tmp = random.choice(allChars)
        passphrase.append(tmp)

    res = "".join(passphrase)
    return res

n = int(input("Ingresa ancho de Password: "))
test = crear_pass(n)
try:
    print(test)
except:
    print("Debe ingresar el ancho del Password")
```

Ejemplo tomado del autor Eduardo Saavedra en la pagina Medium.com

Ejemplo 2 – Genera claves

01 Este script permite generar contraseñas. Este script se puede implementar en caso de tener un sitio web donde se quiera generar contraseñas a los usuarios de manera automática y segura.

02 Se observan elementos conocidos como **‘import’** para utilizar módulos predefinidos de Python, y la sentencia **‘def’** que define la función de que genera y devuelve la contraseña.

03 La sentencia **‘string’** que permite obtener caracteres de diferentes tipos según ascii, y la sentencia **‘try/except’** permite validar que la cantidad de caracteres solicitados para crear la contraseña sea un número válido.

Ejemplo 3 – conexión por TELNET

01

- En el siguiente ejemplo se dejará en claro que el script no funcionará directamente en packet tracer debido a que necesita conexión a equipo físico para ser ejecutado, o un programa (posiblemente con licencia) para simular su ejecución en ambiente real.

02

- De igual manera se explicará el ejemplo en cuanto a lo que realiza y las instrucciones que contiene.

03

- En este ejemplo en particular se utilizan 3 módulos o bibliotecas de Python:

  a. **sys:** este módulo proporciona acceso a funciones que interactúan con el intérprete de comandos.

  b. **getpass:** es un módulo que contiene las funciones getpass y getuser, las cuales solicitan password y usuario respectivamente.

  c. **telnetlib:** este módulo proporciona funciones para implementar el protocolo telnet.

Ejemplo 3 – conexión por TELNET

01

Las funciones utilizadas para establecer conexión son las siguientes:

- **getpass.getpass():** solicita ingreso de contraseña.
- **telnetlib.Telnet():** realiza una conexión a un servidor Telnet.
- **tn.read_until():** lee hasta que se encuentre una cadena determinada o hasta que hayan pasado los segundos del tiempo de espera.
- **tn.write():** escribe una cadena en el socket, puede aparecer un error si la conexión está cerrada.
- **tn.read_all():** lee todos los datos hasta fin de archivo, hasta que la conexión se cierre.
- **tn.close( ):** cierra la conexión.

Ejemplo 3 – conexión por TELNET

02

El siguiente script permite establecer una conexión con usuario y contraseña a través del protocolo telnet, y luego de establecida la conexión permite programar la consola y configurar la interfaz a través de comandos.

- *De la misma manera y bajo las mismas condiciones se podría realizar un script para trabajar con SSH.*

```
import getpass
import sys
import telnetlib

HOST='192.168.111.1'
user = input("Ingrese su Ususario: ")
password = getpass.getpass()

tn = telnetlib.Telnet(HOST)

tn.read_until("Usuario: ")
tn.write(user + "\n")
tn.read_until("Password: ")
if password:
    tn.write(password + "\n")
    tn.write("enable\n")
    tn.write("cisco\n")
    tn.write("configure t\n")
    tn.write("int loop 0\n")
    tn.write("ip address 111.111.111.111 255.255.255.255\n")
    tn.write("end\n")
    tn.write("exit\n")
    print(tn.read_all())
```

Fuente propia

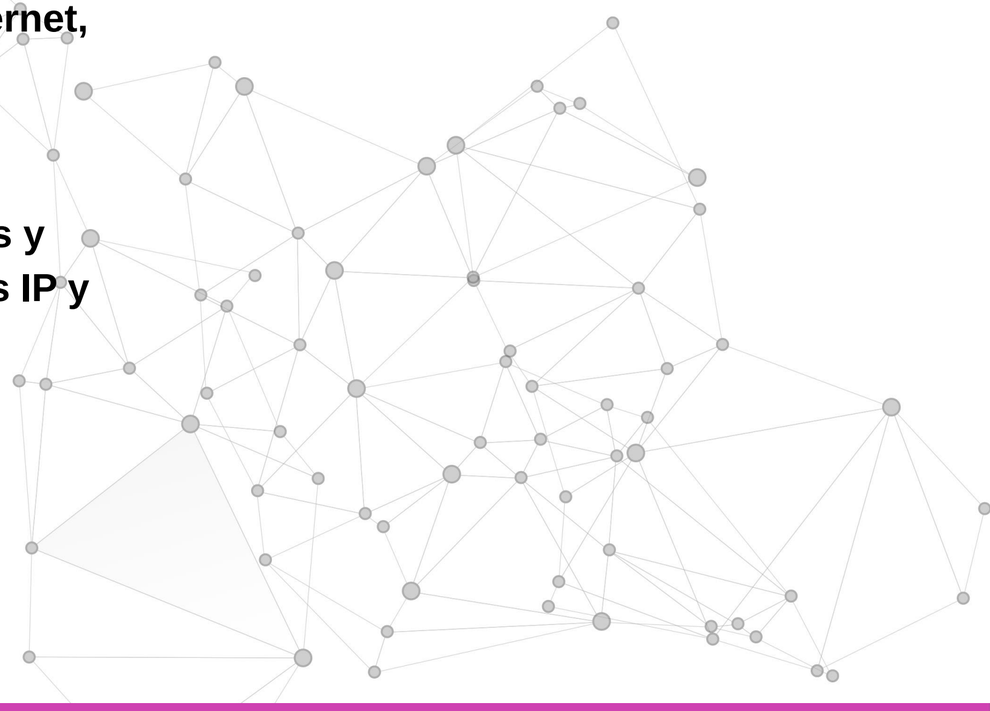
Ejemplo 3 – conexión por TELNET

03

Observación: Para ejecutar este tipo de programas que incluyen conexión telnet o ssh es necesario una de dos opciones:

1. **Tener un PC y un router conectados, habiendo configurado una IP para el router y los servicios básicos correspondientes al caso (ssh/telnet).**

2. **Utilizar un PC con internet, registrarse en developer.cisco.com (devnet) y reservar un laboratorio de pruebas y configurar los equipos IP y ssh.**



Repasemos…

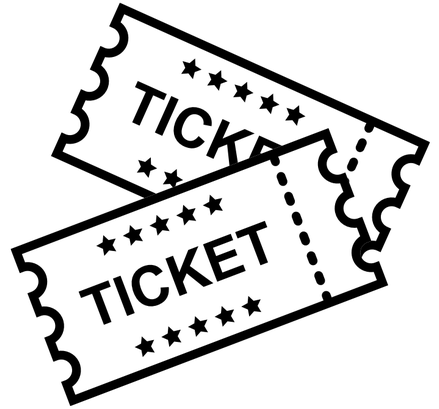
Nombra los pasos básicos para automatizar las redes

¿Tienes preguntas de lo trabajado hasta aquí?

Referencias de imágenes

- **Imágenes de autoría propia,** excepto las que tienen señalado su origen en las mismas, las cuales son gratis para usos comerciales y no es necesario reconocimiento.

Ticket de salida



01

Menciona 2 características de un diccionario en Python.

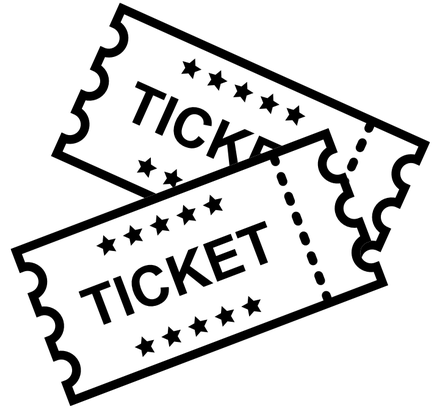
02

¿En qué se parecen los diccionarios de Python a los diccionarios que ya conocías en la vida real?

03

Menciona 2 situaciones en las que podrías utilizar diccionarios en programación.

Ticket de salida



04

¿Cómo le explicarías a una persona, que no tiene conocimientos técnicos, cuál es la utilidad de la programabilidad de las redes?

05

¿Qué procesos frecuentes de red se podrían programar?

06

¿Qué fue lo que más se te dificultó de esta temática?

¿Qué acciones realizarías para comprender mayormente estos contenidos?