



El futuro digital
es de todos

MinTIC



Vigilada Mineducación



CICLO I:

Fundamentos de Programación en Python



Hechos
QUE CONECTAN

Sesión 17:

Manejo de Archivos

DATOS DE ARCHIVOS

Objetivos de la sesión

Al finalizar esta sesión estarás en capacidad de:

1. Construir un programa con una interfaz de consola o gráfica simple
2. Construir un programa que maneje archivos de texto para almacenar información persistente (formato TXT, JSON, CSV)

Archivos: Acerca del formato CSV

- Los archivos CSV (Valores Separados por Comas) son usados generalmente para intercambiar datos tabulares entre sistemas en texto sin formato.
- Normalmente contienen una fila de encabezado que proporciona los nombres de columna para los datos, sin embargo, estos son considerados semiestructurados. Esto se da y que los archivos CSV no pueden representar de forma natural datos jerárquicos o relacionales.
- Las relaciones de datos normalmente se controlan con varios archivos CSV, donde las claves externas se almacenan en columnas de uno o más archivos, pero las relaciones entre esos archivos no se expresan con el propio formato. Los archivos en formato CSV pueden usar otros delimitadores además de las comas, por ejemplo, tabulaciones o espacios.

Archivos: Acerca del formato CSV

- Los archivos CSV son una opción muy utilizada para el intercambio de datos, a pesar de sus limitaciones, esto se debe a lo compatibles que son con una amplia gama de aplicaciones de negocio, de consumo y científicas.
- Los programas de base de datos y hojas de cálculo pueden importar y exportar archivos CSV. De igual forma, la mayoría de los motores de procesamiento por lotes y de procesamiento de secuencias de datos, como Spark y Hadoop, admiten de forma nativa serializar y deserializar archivos con formato CSV y ofrecen formas de aplicar un esquema en la lectura.

Archivos: Acerca del formato CSV

CodigoDpto	NombreDpto
1	Gerencia
2	Ventas
3	Compras
4	Recursos Humanos
5	Cartera
6	Contabilidad
7	Mercadeo

CodigoEmpleado	Nombre	Apellido	CodigoDpto
10001	Juan	Perez	1
10002	Ana	Marin	1
20001	Julio	Diaz	1
20002	Daniel	Soto	2
30001	Martha	Quiroga	3

Departamentos.csv

CodigoDpto,NombreDpto
 1, Gerencia
 2, Ventas
 3, Compras
 4, Recursos Humanos
 5, Cartera
 6, Contabilidad
 7, Mercadeo

Empleados.csv

CodigoEmpleado,Nombre,Apellido,CodigoDpto
 10001,Juan,Perez,1
 10002,Ana,Marin,1
 20001,Julio,Diaz,1
 20002,Daniel,Soto,2
 30001,Martha,Quiroga,3

Archivos: Acerca del formato JSON

El auge de la web ha llevado **JavaScript** a lugares en los que nunca se imaginó. Es uno de los lenguajes de programación más importantes del mundo tecnológico actual que ha ayudado a expandir el uso de aplicaciones **Web**.

Con JavaScript, entraron en la imagen diferentes extensiones llamadas **JSON**.



El formato de archivo JSON fue especificado por **Douglas Crockford** a principios de la década de 2000. Hoy en día, se ha convertido en el formato informático por defecto para almacenar datos estructurados.

Archivos: Acerca del formato JSON

- Los archivos JSON (Notación de Objetos JavaScript) se representan como pares de clave y valor en un formato semiestructurado.
- JSON es comparado con XML, ya que ambos son capaces de almacenar los datos en un formato jerárquico con los datos secundarios alineados con su elemento primario.
- Ambos son autodescriptivos y legibles, sin embargo, los documentos JSON tienden a ser mucho más pequeños, lo que ha llevado a que se use mucho más en el intercambio de datos en línea, sobre todo en los servicios web basados en REST.

Archivos: Acerca del formato JSON

Los archivos con formato JSON presentan algunas ventajas sobre CSV:

- JSON mantiene las estructuras jerárquicas, lo que facilita la tarea de almacenar datos relacionados en un único documento y representar relaciones complejas.
- Mucho de los lenguajes de programación permiten deserializar JSON en objetos o proporcionan bibliotecas sencilla de serialización de JSON.
- JSON admite listas de objetos, lo que permite evitar traducciones desordenadas de listas en un modelo de datos relacional.

Archivos: Ejemplo del formato JSON

```
{  
  "name": "Miller John",  
  "mobile": "897654321",  
  "age": 45,  
  "address": {  
    "city": "New York",  
    "country": "USA"  
  }  
}
```

```
{  
  "nombre": "Jonh Doe",  
  "profesion": "Programador",  
  "edad": 25,  
  "lenguajes": ["PHP", "Javascript", "Dart"],  
  "disponibilidadParaViajar": true,  
  "rangoProfesional": {  
    "aniosDeExperiencia": 12,  
    "nivel": "Senior"  
  }  
}
```

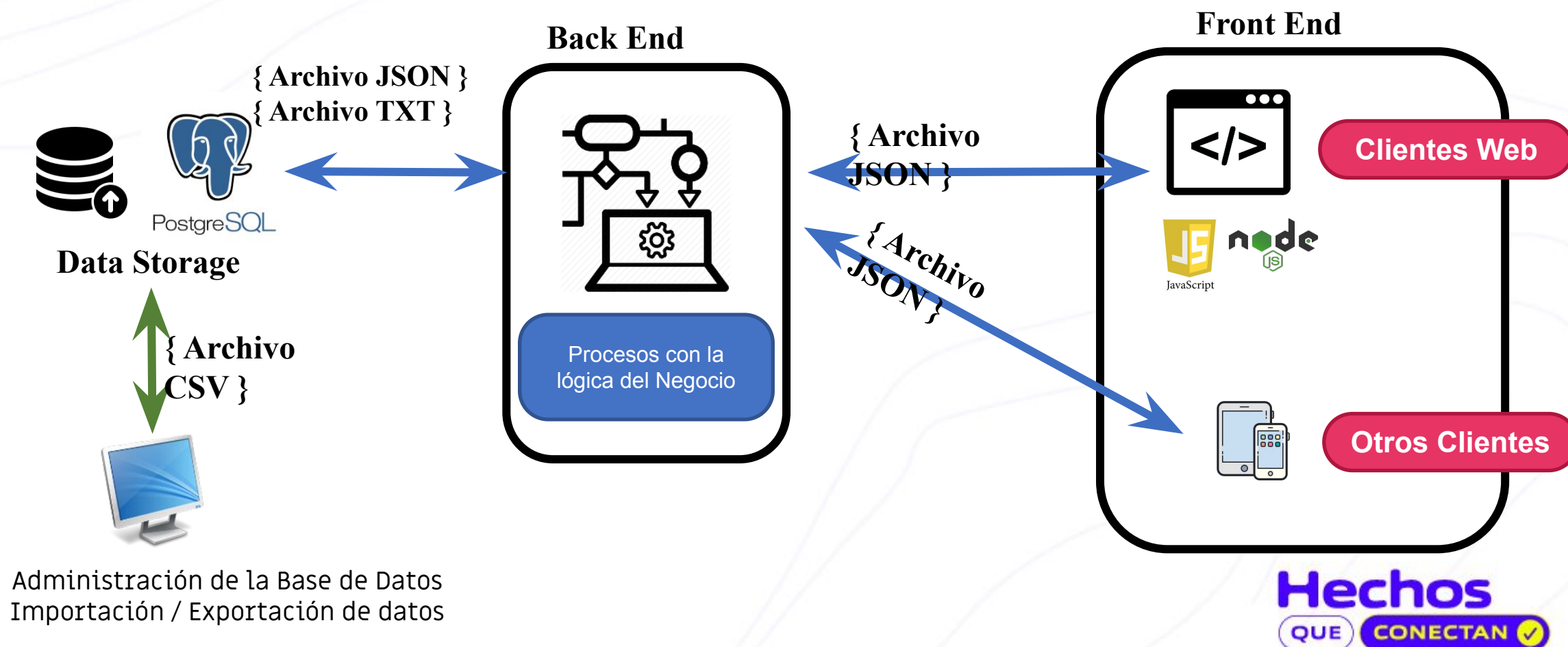
Archivos: Cuándo utilizar los formatos CSV o JSON?

- Los archivos CSV son utilizados generalmente para la exportación y la importación de datos o para su procesamiento para el análisis o el aprendizaje automático.
- Los archivos con formato JSON tienen las mismas ventajas, solo que su uso es más común en soluciones de intercambio de datos activos.
- Los documentos JSON son enviados normalmente por web y por dispositivos móviles para realizar transacciones en línea o por los dispositivos de IoT (internet de las cosas) para comunicaciones unidireccionales.

Archivos: Cuándo utilizar los formatos CSV o JSON?

- Los formatos de archivo CSV y JSON facilitan el intercambio de datos entre diferentes sistemas o dispositivos.
- El hecho de que sus formatos sean semiestructurados, permiten flexibilidad para transferir casi cualquier tipo de datos y la compatibilidad universal con estos formatos hacen que trabajar con ellos sea muy simple.
- Tanto CSV como JSON se pueden utilizar como única fuente sin tratamiento en aquellos casos en los que los datos procesados se almacenan en formatos binarios para una consulta más eficaz.

Archivos: Cuándo utilizar los formatos CSV o JSON?



EJERCICIOS PARA PRACTICAR

Ejercicio 1: Actividad de refuerzo y práctica de conceptos

En este ejercicio deberás crear un script llamado `personas.py` que lea los datos de un archivo de texto, que transforme cada fila en un diccionario y lo añada a una lista llamada `personas`. Luego recorra las personas de la lista y para cada una muestre de forma amigable todos sus campos.

El archivo de texto se denominará `personas.txt` y tendrá el siguiente contenido en texto plano (créalo previamente):

1;Carlos;Pérez;05/01/1989

2;Manuel;Heredia;26/12/1973

3;Rosa;Campos;12/06/1961

4;David;García;25/07/2006

Los campos del diccionario serán por orden:

id, nombre, apellido y nacimiento.

Ejercicio 1: Solución propuesta

```
file = open("./Personas.txt", "r")
```

```
diccionario = []
```

```
for linea in file:
```

```
    pal = linea.split(";")
```

```
    elemento = {
```

```
        "id": pal[0],
```

```
        "nombre": pal[1],
```

```
        "apellido": pal[2],
```

```
        "nacimiento": pal[3]
```

```
    }
```

```
    diccionario.append(elemento)
```

```
file.close()
```

```
print(diccionario)
```

Ejercicio 2: Actividad de refuerzo y práctica de conceptos

En este ejercicio deberás crear un script llamado `contador.py` que realice varias tareas sobre un archivo llamado `contador.txt` que almacenará un contador de visitas (será un número):

El script trabajará sobre el archivo `contador.txt`. Si el archivo no existe o se encuentra vacío lo crearemos con el número 0. Si existe simplemente leeremos el valor del contador.

Luego a partir de un argumento:

Si se envía el argumento `inc`, se incrementará el contador en uno y se mostrará por pantalla.

Si se envía el argumento `dec`, se decrementará el contador en uno y se mostrará por pantalla.

Si no se envía ningún argumento (o algo que no sea `inc` o `dec`), se mostrará el valor del contador por pantalla.

Finalmente guardará de nuevo el valor del contador de nuevo en el archivo.

Utiliza excepciones si crees que es necesario, puedes mostrar el mensaje:

Error: Archivo corrupto.

Ejercicio 2: Solución propuesta

```
file = open("contador.txt", "w")
file.write('0')
fileReader = open("contador.txt", "r")
op=""

while op!='sal':
    op = input("Desea incrementar(inc), decrementar (dec), o salir (sal)")
    dato = fileReader.readline()
```

```
if op == 'inc':
    inc=int(dato)+1
    print(inc)
    file.write(inc)
elif op == 'dec':
    dec=int(dato)-1
    print(dec)
    file.write(dec)
elif op=='sal':
    print("Muchas gracias")
else:
    print(dato)
file.close()
fileReader.close()
```


ENCUESTA SEMANAL

Seguimiento a Habilidades Digitales en Programación

* De modo general, ¿Cuál es grado de satisfacción con los siguientes aspectos?

	Nada Satisfecho	Un poco satisfecho	Neutra	Muy satisfecho	Totalmente satisfecho
Sesiones técnicas sincrónicas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sesiones técnicas asincrónicas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sesiones de inglés	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Apoyo recibido	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Material de apoyo: diapositivas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Material de apoyo: ejercicios prácticos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Completa la siguiente encuesta para darnos retroalimentación sobre tu proceso de aprendizaje en esta semana:

<https://uninorte.questionpro.com/t/ALw8TZshc2>



COMPONENTE PRÁCTICO



El futuro digital
es de todos

MinTIC

Misión
TIC 2022

UN UNIVERSIDAD
DEL NORTE

Vigilada Mineducación

¡GRACIAS

**POR SER PARTE DE
ESTA EXPERIENCIA
DE APRENDIZAJE!**

Hechos

QUE

CONECTAN ✓

