

Facultad de  
Ingeniería, Arquitectura y Diseño



# Taller de programación II

Sesión: Introducción al uso de Streamlit para aplicaciones interactivas





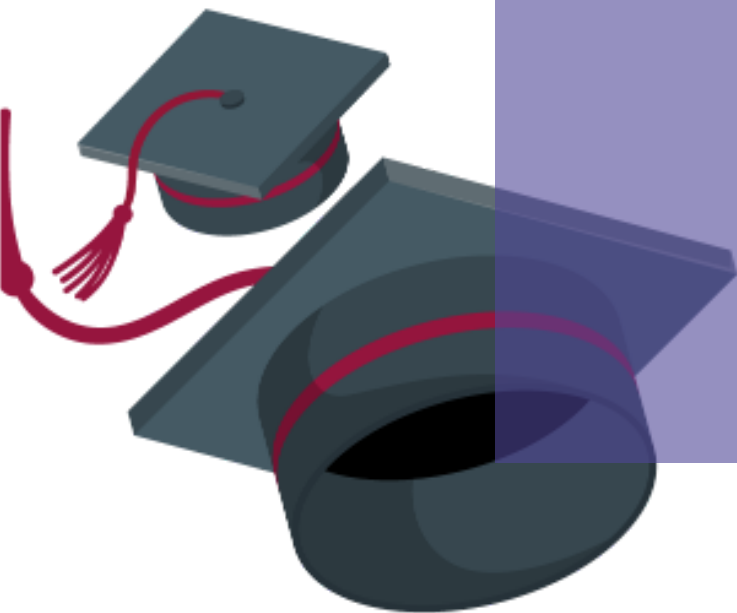
# Contenido

---

1. Introducción a Streamlit.
2. Estructura básica de una aplicación Streamlit.
3. Código proporcionado: explicación paso a paso.
4. Ejemplo de flujo de ejecución.
5. Actividades sugeridas.

# Resultado de aprendizaje

Comprender las funcionalidades básicas de Streamlit, una herramienta poderosa para crear aplicaciones web interactivas en Python. Se usará como base el código proporcionado, que muestra cómo consultar datos desde una API, filtrarlos y visualizarlos.



Antes de empezar, te invitamos a tomar en cuenta los siguientes íconos que aparecen a lo largo del interactivo:



Navega



*Check list*



Descargar  
y revisar  
documento



Observa

Streamlit es un *framework* de Python diseñado para construir aplicaciones web de forma rápida y sencilla. Algunas características clave:

Característica 1

Característica 2

Característica 3



Haz clic en cada característica para conocer su información



## Característica 1

Permite crear interfaces gráficas sin necesidad de conocimientos avanzados de desarrollo web.



Fuente:Storyset.com



## Característica 2

Soporta visualizaciones interactivas.



Fuente:Storyset.com



Fuente:Storyset.com

## Característica 3

Integra bibliotecas como **Pandas**, **Matplotlib**, **Altair** y otras.



**Según lo estudiado, recuerda:**

Streamlit debe estar previamente instalado con pip de la siguiente forma:

```
pip install streamlit
```

Te invitamos a realizar la instalación y seguir los pasos que se proponen a continuación.

## Ejercicio 1: *Check list*

### Estructura básica de una aplicación Streamlit

#### La estructura básica incluye:

You can select more than one answer



Una vez que revises que cuentas con la estructura básica, marca los cuadros haciendo clic para avanzar.

Visualizaciones: usar bibliotecas de gráficos como Matplotlib o Plotly.

Añadir Componentes interactivos: como barras laterales (st.sidebar) y botones (st.button).

Títulos y textos: usar funciones como st.title(), st.write(), etc.

Send

Ejercicio 2: Ejecuta la sentencia

**Código Python con caso de uso:**



**Descarga el código aquí**

Se debe considerar que este código fue guardado en un archivo **Python** de nombre “**ApiRadio2.py**”, por lo tanto, para la visualización de los resultados, y a diferencia de otras oportunidades, su ejecución se realiza mediante la siguiente sentencia:

```
streamlit run ApiRadio2.py
```

## Ejercicio 2: Ejecuta la sentencia

Lo anterior responderá con un mensaje en consola donde nos indica en qué **ip** y puerto está corriendo la aplicación. Además, abrirá de forma automática el navegador por defecto para mostrar la aplicación web funcionando:

```
PS C:\Users\Ricardo\OneDrive\USS\Make it\Python\Unidad 3> streamlit run ApiRadio2.py
```

You can now view your Streamlit app in your browser.

Local URL: <http://localhost:8501>  
Network URL: <http://192.168.0.18:8501>



Navega: al hacer clic en uno de los URL verás lo siguiente.



Opciones de Filtro

Configuración de Visualización

Cargar Datos

Selecciona una Región

Actualizar Datos

Esta aplicación muestra información en tiempo real de los radioaficionados en Chile registrados para comunicación digital.

	callsign	city	country	fname	id	remarks	state	surname
0	CD6CNV	Rio Bueno	Chile	Claudio	7,300,001	DMR		Vallejos
1	CE1WOY	Calama	Chile	Mauricio	7,300,002	DMR	Antofagasta	Romero
2	CD6CNV	Rio Bueno	Chile	Claudio	7,300,003	DMR		Vallejos
3	CE3LWU	Chillan	Chile	Juan Enrique	7,300,004	DMR	Bio Bio	San Martin
4	CE3LWU	Chillan	Chile	Juan Enrique	7,300,005	DMR	Bio Bio	San Martin
5	CA6AMV	Valdivia	Chile	Roberto	7,300,006	DMR		Aburto MartiNez
6	CE2SLD	COQUIMBO	Chile	Jorge	7,300,007	DMR	Coquimbo	ARAYA
7	CD1HCO	Arica	Chile	Ricardo Andres	7,300,008	DMR	Tarapac	Hurtado Cortes
8	CA1MTV	Iquique	Chile	Victor Rodrigo	7,300,009	DMR	Tarapac	Mera Torres
9	CE3CLF	Santiago	Chile	Cristian	7,300,010	DMR	Reg.Metr.de l	Letelier

Datos de Radioaficionados en Chile Según Región

	callsign	city	country	fname	id	remarks	state	surname
0	CD6CNV	Rio Bueno	Chile	Claudio	7,300,001	DMR		Vallejos
2	CD6CNV	Rio Bueno	Chile	Claudio	7,300,003	DMR		Vallejos



Navega en el ApiRadio2 y descubre toda la información.



Opciones de Filtro

Configuración de Visualización

Cargar Datos

Selecciona una Región

Actualizar Datos

Esta aplicación muestra información en tiempo real de los radioaficionados en Chile registrados para comunicación digital.

	callsign	city	country	fname	id	remarks	state	surname
0	CD6CNV	Rio Bueno	Chile	Claudio	7,300,001	DMR		Vallejos
1	CE1WOY	Calama	Chile	Mauricio	7,300,002	DMR	Antofagasta	Romero
2	CD6CNV	Rio Bueno	Chile	Claudio	7,300,003	DMR		Vallejos
3	CE3LWU	Chillan	Chile	Juan Enrique	7,300,004	DMR	Bio Bio	San Martin
4	CE3LWU	Chillan	Chile	Juan Enrique	7,300,005	DMR	Bio Bio	San Martin
5	CA6AMV	Valdivia	Chile	Roberto	7,300,006	DMR		Aburto MartiNez
6	CE2SLD	COQUIMBO	Chile	Jorge	7,300,007	DMR	Coquimbo	ARAYA
7	CD1HCO	Arica	Chile	Ricardo Andres	7,300,008	DMR	Tarapac	Hurtado Cortes
8	CA1MTV	Iquique	Chile	Victor Rodrigo	7,300,009	DMR	Tarapac	Mera Torres
9	CE3CLF	Santiago	Chile	Cristian	7,300,010	DMR	Reg.Metr.de l	Letelier

Datos de Radioaficionados en Chile Según Región

	callsign	city	country	fname	id	remarks	state	surname
0	CD6CNV	Rio Bueno	Chile	Claudio	7,300,001	DMR		Vallejos
2	CD6CNV	Rio Bueno	Chile	Claudio	7,300,003	DMR		Vallejos



Navega en el ApiRadio2 y descubre toda la información.

## Ejercicio 3: Importar

### Importación de bibliotecas:

- `import streamlit as st`
  - `import pandas as pd`
  - `import requests`
  - `import matplotlib.pyplot as plt`
- 
- `streamlit`: para la interfaz gráfica.
  - `pandas`: para procesar los datos obtenidos de la API.
  - `requests`: para realizar solicitudes HTTP.
  - `matplotlib.pyplot`: para generar gráficos.



Observa cómo puedes importar.

## Ejercicio 4: Consulta

### Consulta de datos desde la API

```
url = "https://radioid.net/api/dmr/user/?country=Chile"  
response = requests.get(url)
```



Navega: copia el URL y consulta los datos.

- **requests.get(url):** realiza una solicitud GET a la API para obtener datos de radioaficionados en Chile.
- Si la solicitud es exitosa, se transforma la respuesta JSON en un *dataframe* de Pandas:

```
data = response.json()  
df = pd.DataFrame(data)
```



## Ejercicio 5: Maneja los errores

### Consulta de datos desde la API Manejo de errores

```
if response.status_code != 200:  
    st.error("Error en la consulta de datos")
```

Muestra un mensaje de error en la interfaz si la consulta falla.



¿Pudiste manejar el error?

Send

## Ejercicio 6: Filtros y opciones

### Consulta de datos desde la API

Vamos a aplicar algunos **filtros** y revisar qué **opciones** tenemos al realizar la consulta de datos desde la API.

Descarga el archivo y sigue el paso a paso.



**Descarga**

## Ejercicio 7: Visualiza datos

### Código proporcionado: explicación paso a paso

Para realizar la visualización de datos, descarga el paso a paso de este ejercicio aquí y navega.

You can select more than one answer

Muestra el dataframe completo o filtrado en la interfaz.

Muestra un gráfico de barras con la cantidad de usuarios por región.

Muestra la distribución de categorías de usuarios en un gráfico circular.



*Una vez que completes la visualización de datos, marca los cuadros para avanzar.*

Send

## Ejercicio 8: Flujo de ejecución

### 1. Carga de la aplicación.

- Ejecutar el archivo con el comando:

```
streamlit run nombre_del_archivo.py
```



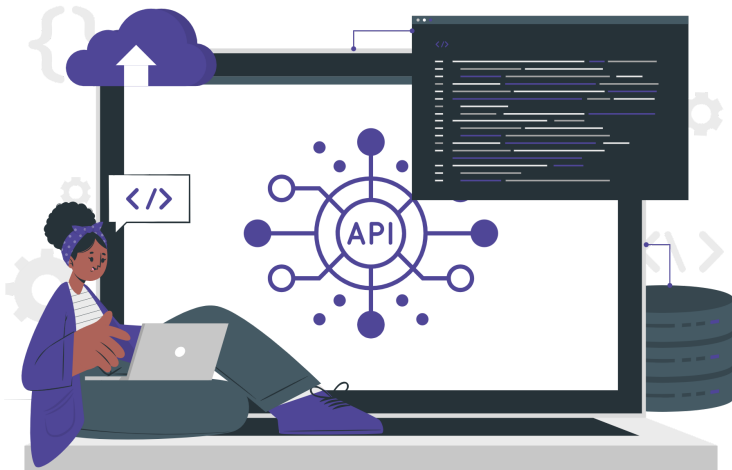
Observa cómo puedes importar.

### 2. Interacción con la interfaz.

- Seleccionar una categoría en la barra lateral.
- Visualizar los datos filtrados en la tabla principal.
- Observar los gráficos generados dinámicamente.

## Actividades extras sugeridas

**Descarga el archivo y sigue practicando**



Fuente:Storyset.com



**Descarga**



## Referencias bibliográficas

---

Dangeti, P., Yu, A., Chung, C., Yim, A., & Petrou, T. (2018). *Numerical Computing with Python (1st ed.)*. Packt Publishing, Limited. Digitalizado.

Vo. T. H. P., Czygan, M., Kumar, A., & Raman, K. (2017). *Python: data analytics and visualization*. Packt Publishing. Digitalizado.



ADVANCE | USS

