

Seminario de Lenguajes - Python

Cursada 2024

Streamlit y sus componentes

¿Qué vimos en clase?

- Introducción a Streamlit
- Estructuración con:
 - pages
 - containers
- Widgets
- Callbacks

Les proponemos algunos ejercicios para comprender un poco más los componentes de Streamlit y su funcionamiento.

Dado el archivo **start.py** que contiene un diccionario con información sobre áreas protegidas de Argentina con la siguiente organización:

- `reserved_areas`: son áreas protegidas que son reservas
- `national_parks_only`: son áreas protegidas que son solamente parques nacionales

Realice las apps solicitadas con **Streamlit**.



Desafío 1

Con los datos de **protected_areas** queremos armar una serie de preguntas, que permita identificar cuál es Reserva y NO es Parque Nacional.

Se pide que se muestren 5 opciones:

- 4 son Parques Nacionales
- 1 que sea solamente Reserva

les dejamos una imagen de lo solicitado:

Areas Protegidas de Argentina

Las opciones son áreas protegidas

De las 5 opciones mostradas: 4 son Parques Nacionales y Reservas y 1 sola es solamente Reserva

Marca la opción que es solamente Reserva

Repuestas correctas: 1

- ☐ Monte León
- ☐ Los Cardones
- ☐ Patagonia
- ☐ Iberá - Núcleo San Nicolás/San Alonso
- ☐ Colonia Benítez

La página principal deberá tener dos columnas:

- en la columna 1: mostrar las opciones
- en la segunda columna: ir mostrando la cantidad de respuestas correctas

Para obtener las opciones **all_types** se dejó la invocación a la función **process_areas** que deben completar, debe devolver todas las opciones y cuál es la respuesta correcta **correct_r**.

En la **columna 1** luego de las preguntas se agregó código para mostrar si la respuesta es correcta o no, este mensaje se puede ver? ¿por qué?

En el video vamos a orientar cómo acceder al valor de la respuesta correcta.

Uso de keys, callback y sesión

La opción correcta se pierde ya que al detectar un evento se relee la página, para acceder a la opción antes de releer podemos ahcerlo desde el widget.

- Para obtener los datos de un widget podemos acceder definiendo la key, esto genera una variable con el nombre de la key en la sesión, en **st.session_state**.

Opción para actualizar la cantidad de respuestas correctas utilizando las variables de la sesión:

```
selected_type = st.radio('Marca la opción que es solamente Reserva',  
all_types,index=None,  
key='option', on_change=update,  
args=(correct_r,))
```

- En el ejemplo se ve:
 - definición de la key: **key='option'**, en este momento se genera la variable en **st.session.state**
 - se realiza un **callback** a una función **update, on_change=update**
 - se pasa un argumento a la función **args=(correct_r,)**. Noten que la coma al final de la tupla es necesaria porque como es una tupla, debe ser dos o más elementos. Por lo tanto si se quiere pasar un solo argumento, debe agregarse una coma al final
- Dentro de esa función que se invoca desde el widget consultamos si el valor de la opción elegida es igual a la respuesta correcta.

```
if st.session_state.option == answer:
```

Esta forma de preguntar lo podemos realizar porque lo estamos haciendo en una función que se va a invocar luego de que esté definido la variable de estado **option**.



Desafío 2

Con los datos de las áreas protegidas que se encuentran en el archivo **start.py** genere otra entrada en el sidebar, para lo cual deberá crear el directorio **pages** y crear un nuevo archivo python que realice lo siguiente:

- Generar un formulario que consulte el nombre de un usuario en una pestaña (tab) y si este valor está configurado se muestren las preguntas en otra pestaña.

usuario

Información del usuario

Ingrese su nombre

Responder

Nota

- Para verificar si está configurado el nombre podemos usar variables de estado de Streamlit, que se actualiza desde el widget button.
- El procesamiento del diccionario ya lo realizó en el archivo **start.py** evalúe cómo reorganizar esa función para que ambos archivos puedan acceder y no repetir el código.

🚩 Desafío 3

La información que se encuentra detallada en el diccionario se obtuvo de un archivo, **area_protegida.csv** que adjuntamos. Los datos se obtuvieron del [Instituto Geográfico Nacional](#).

Cree un archivo dentro del directorio **pages** que procese **area_protegida.csv** y encuentre los datos con los cuales se creó el diccionario:

- Para encontrar las reservas, se buscó que la palabra **Reserva** se encuentre en la columna **fna**
- Para encontrar los Parques Nacionales, se buscó que las palabras **Parque** y **Nacional** en encuentre en la columna **fna** y no esté **Reserva**

En caso que no les permita abrir el archivo csv por el tamaño, hay una configuración que incrementa el tamaño máximo, deberán agregarlo antes de abrirlo en el código:

```
In [ ]: import sys
        csv.field_size_limit(sys.maxsize)
```

Esta instrucción está configurando el tamaño máximo que puede abrir csv en función del sistema de cada persona. El módulo **sys** consulta al sistema operativo y devuelve ese valor con **sys.maxsize**

FACULTAD DE INFORMÁTICA - UNLP

¡Nos vemos el martes!



Facultad de
INFORMÁTICA
UNIVERSIDAD NACIONAL DE LA PLATA



UNIVERSIDAD
NACIONAL
DE LA PLATA

In []: