

Reto 2: SkyNow 2.0 🌤️

Te han contratado para desarrollar una **actualización** del **sistema inteligente para el análisis de datos meteorológicos** utilizando Python. El director del proyecto, conocedor de las ventajas derivadas del uso de la **Programación Orientada a Objetos (POO)**, ha solicitado construir una nueva versión bajo ese paradigma y al unísono, ha solicitado extender las funcionalidades.

El sistema consta de cinco módulos fundamentales. A continuación, se describen las funcionalidades específicas de cada módulo:

Obtención y Registro de Datos

En este módulo, se utilizará una **API de países** para obtener información de los países del sistema, además se deberá consultar una **API de estados** para obtener la información de todos los estados de cada país. Por otro lado, se contará con una **API meteorológica** para obtener datos sobre el clima de cada país. Las funcionalidades principales incluirán:

1. **Obtención de países:** Se consumirán los datos de la API para obtener información como nombre, capital, población, superficie, moneda e idioma principal.
2. **Obtención de estados:** Se consumirán los datos de la API para obtener información como nombre, capital, población, y superficie de cada estado de un país.
3. **Obtención de datos climáticos:** Se consumirán los datos de la API para obtener información como temperatura, humedad y velocidad del viento de cada uno de los países obtenidos en la API de Países.
4. **Almacenamiento de datos:** A diferencia de la versión anterior, en este caso **no se permitirá la utilización de diccionarios para almacenar los datos**, se determinó

que es mejor almacenar los datos en **listas de objetos**, por lo que se deberán diseñar las clases necesarias para almacenar la información. Para ello, debe utilizar **Programación Orientada a Objetos (POO)**.

Reportes

En este módulo, se deberán proporcionar funcionalidades que permitan al usuario visualizar los datos extraídos de cada API. Las funcionalidades principales incluirán:

1. **Mostrar países:** Al seleccionar esta opción, se deberán mostrar por consola todas las ubicaciones disponibles en el sistema.
2. **Mostrar información de un país:** Al seleccionar esta opción, se le deberá pedir al usuario por consola la ubicación de la cual desea conocer los datos. Posteriormente, se mostrará por consola la información del país seleccionado (*sin mostrar los estados o los datos meteorológicos*).
3. **Mostrar estados de un país:** Al seleccionar esta opción, se le deberá pedir al usuario por consola la ubicación de la cual desea conocer los estados, posteriormente se mostrarán por consola todos los estados del país y la información de cada uno de los estados.
4. **Mostrar datos meteorológicos de un país:** Al seleccionar esta opción, se le deberá pedir al usuario por consola la ubicación de la que desea conocer los datos, posteriormente se mostrarán por consola los datos meteorológicos de dicho país.

Análisis de Datos

En este módulo, se utilizarán técnicas de análisis de datos para analizar la información registrada. Las funcionalidades principales incluirán:

1. **Análisis de superficie y población de los países:** Al seleccionar esta opción, se deberán mostrar por consola cuáles son los países con la población y superficie más alta y más baja. Además, se deberá mostrar el promedio de población y superficie de los países registrados en el sistema.
2. **Análisis de superficie y población de los estados:** Al seleccionar esta opción, se deberá seleccionar el país del cual se quiere realizar el análisis. Posteriormente, se deben mostrar los estados con la población y superficie más alta y más baja de ese país. Además, se deberá mostrar el promedio de población y superficie de los estados del país seleccionado.

3. **Análisis de idiomas:** Al seleccionar esta opción, el sistema deberá mostrar por consola una lista de los idiomas principales, indicando los países que los hablan.
4. **Análisis estadístico de datos meteorológicos:** Se realizarán cálculos estadísticos básicos, como el promedio, la moda y el máximo/mínimo de las variables climáticas. Al seleccionar esta opción, se deberá introducir por consola la localización de la cual se desea hacer el análisis y posteriormente mostrar todos los cálculos previamente mencionados para cada una de las variables climáticas.
5. **Graficar datos (Bono):** Al seleccionar esta opción, se deberá seleccionar el país del cual se quiere realizar el gráfico y posteriormente seleccionar alguna de las siguientes opciones para realizar un gráfico utilizando **Matplotlib**:
 - a. **Superficie de los estados:** Se debe realizar un **gráfico de barras**, en donde se visualice el nombre de cada estado junto con su valor de superficie correspondiente.
 - b. **Población de los estados:** Se debe realizar un **gráfico de barras**, en donde se visualice el nombre de cada estado junto con su valor de población correspondiente.

Filtrado de Datos

En este módulo, se implementarán funcionalidades de filtrado y consulta para facilitar el acceso a los datos almacenados. Las funcionalidades incluirán:

1. **Filtrado de estados por población:** Se le deberá pedir al usuario un país y un valor que corresponderá al número mínimo de habitantes. Posteriormente, se deberán mostrar únicamente los estados con una población superior al número indicado por el usuario, en caso de existir.
2. **Filtrado de estados por superficie:** Se le deberá pedir al usuario un país, un valor mínimo y un valor máximo de superficie. Posteriormente, se deberán mostrar únicamente los estados que se encuentren dentro del rango de valores indicado por el usuario, se debe validar el rango de valores.

Búsqueda y Consulta de Datos

En este módulo, se implementarán funcionalidades de búsqueda y consulta para facilitar el acceso a los datos almacenados. Las funcionalidades incluirán:

1. **Búsqueda por fecha:** Al seleccionar esta opción, el programa deberá pedirle al usuario una ubicación y una fecha para realizar la búsqueda, posteriormente se deberán mostrar todos los datos meteorológicos de la ubicación seleccionada que correspondan a la fecha especificada, en caso de existir.
2. **Búsqueda por variables climáticas:** Al seleccionar esta opción, el programa deberá pedirle al usuario una ubicación, la variable climática que desea consultar (temperatura, humedad, velocidad del viento) y el valor de la misma. Posteriormente, se deberán mostrar todos los datos meteorológicos de la ubicación seleccionada que contengan el valor de la variable especificada, en caso de existir.
3. **Búsqueda de información de un estado:** Al seleccionar esta opción, el programa deberá pedirle al usuario un país, y el estado que desea buscar en dicho país. Luego, se debe utilizar **búsqueda binaria** para encontrar los datos del estado solicitado por el usuario (La lista de estados estará ordenada alfabéticamente en la API para poder utilizar la búsqueda binaria).

Formatos

- **SkyNow** posee una API en donde podrás obtener toda su información:
 - **Formato de los países:**

```
[{
  "location_name": "str",
  "location_capital": "str",
  "population": "int", #Residents
  "area": "int", #km^2
  "currency": "str",
  "main_language": "str"
}]
```

- **Formato de los estados:**

```
[{
  "location_name": "str",
  "location_states": [{
    "state_name": "str",
    "state_capital": "str",
    "population": "int", #Residents
    "area": "int", #km^2
  }]
}]
```

- **Formato de los datos meteorológicos:**

```
[{
  "location_name": "str",
  "location_measurements": [{
    "temperature": "float" #Celsius
    "humidity": "float" #Percentage
    "wind_speed": "int" #Kilometers per Hour
    "date": "string:yyyy/mm/dd"
  }]
}]
```

- **API países:**
 - <https://raw.githubusercontent.com/Andresar16/API-Retos/main/locations-data-api.json>
- **API estados:**
 - <https://raw.githubusercontent.com/Andresar16/API-Retos/main/locations-states-api.json>
- **API datos meteorológicos:**
 - <https://raw.githubusercontent.com/Andresar16/API-Retos/main/locations-api.json>

Criterios de evaluación

- **Modularidad:** El programa debe estar estructurado por Módulos (funciones) que se centren y resuelvan un aspecto del problema, evitando el diseño monolítico.

(4 puntos.)

- **Correctitud funcional:** El sistema construido debe responder de manera correcta a todos los requerimientos expresados en este documento. **(10 puntos.)**
- **Robustez:** El sistema deberá manejar de manera adecuada las entradas erróneas que intente introducir el usuario al momento de escribir datos o hacer peticiones al sistema. **(3 puntos)**
- **Exactitud:** Los resultados arrojados por el sistema como respuesta a las peticiones del usuario deben ser exactos y correctos. **(3 puntos)**

Condiciones de entrega

- El reto deberá ser entregado en **Google Classroom** a más tardar el **02 de agosto de 2023 a las 11:59 PM.**
- El reto es **en pareja**. El uso de **GitHub** es **opcional** para el trabajo colaborativo.
- Debe realizar un **Diagrama de Clases** del proyecto.
- Proyectos entregados que sean muy parecidos serán rechazados y evaluados con **0 (cero) puntos.**
- Proyectos que no “corran” serán evaluados con **0 (cero) puntos.**
- Solo se permite el uso de la librería **requests** y cualquier librería que les pueda ser de utilidad al momento de graficar (**Matplotlib** u otras librerías que sean compatibles con esta).
- La información obtenida de las APIs estará conformada por listas y diccionarios, esta información se debe transformar y guardarse utilizando **listas y objetos**, está prohibido el uso de diccionarios para guardar la información de las APIs. En caso de no utilizar **Programación Orientada a Objetos (POO)** los proyectos serán evaluados con **0 (cero) puntos.**
- Todas las funciones y métodos del proyecto deben estar documentadas usando **Docstring.**
- Los preparadores podrán convocar a los autores de los proyectos en el caso de que lo consideren pertinente para la realización de una evaluación justa y cuidadosa.