



dev  
Senior

# Reto #1

# Python

Simulación de un Proyecto de Investigación Científica en Python



# RETO 1 - Proyecto de Investigación Científica en Python

## Introducción

En el curso "Python de Cero a Senior: La Ruta Maestra del Código", los estudiantes aprenderán a utilizar Python para abordar problemas reales de una manera estructurada y eficiente. El primer reto, titulado "**Proyecto de Investigación Científica**", introduce a los estudiantes en el desarrollo de aplicaciones basadas en consola, enfocadas en la recopilación, análisis y manipulación de datos científicos. Este desafío permitirá aplicar conceptos fundamentales de programación, como estructuras de control de flujo, funciones, manipulación de datos y listas, mientras se exploran temas científicos de una manera práctica.

En este reto, los estudiantes simularán un proyecto de investigación que involucra la recopilación de datos experimentales de un laboratorio ficticio, su análisis y la generación de un informe final. Los estudiantes deberán implementar un sistema que permita gestionar la información de los experimentos y los resultados obtenidos, incorporando funcionalidades para procesar los datos y obtener conclusiones. **Objetivos**

Los objetivos principales de este reto son:

### 1. Recopilación de datos experimentales:

- o Los estudiantes deberán desarrollar un sistema que permita agregar, visualizar y gestionar datos de experimentos realizados en un laboratorio. Cada experimento tendrá atributos como el nombre del experimento, fecha de realización, tipo de experimento y resultados obtenidos.

### 2. Análisis de resultados experimentales:

- o La aplicación permitirá a los estudiantes realizar operaciones básicas con los datos recopilados, como calcular promedios, máximos y mínimos de los resultados obtenidos.

### 3. Generación de un informe final:

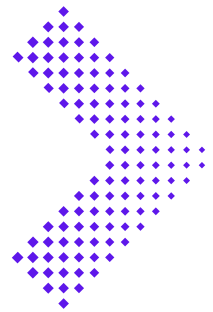
- o La aplicación deberá generar un informe resumen de los experimentos, incluyendo la descripción de los mismos y un análisis de los datos recopilados.

### 4. Validaciones y manejo de errores:

- o Los estudiantes deberán implementar validaciones para garantizar la correcta entrada de datos y manejar errores en el análisis de los experimentos.

### 5. Interacción mediante un menú de opciones:

- o La aplicación será interactiva, con un menú que permita al usuario elegir entre diferentes opciones, como agregar experimentos, analizar resultados y generar informes.



## Beneficios de este Reto

- ❑ **Consolidación de conceptos básicos de Python:**  
Este reto refuerza el uso de **variables**, **estructuras de control**, **funciones** y **listas**, aplicándolos en un contexto de simulación científica.
- ❑ **Desarrollo de habilidades de diseño de software:**  
Los estudiantes aprenderán a descomponer un problema en tareas más pequeñas y manejables, promoviendo la reutilización de código y la organización en módulos.
- ❑ **Aplicación de técnicas de validación y manejo de errores:**  
Se exigirá que los estudiantes validen la entrada de datos y gestionen errores, mejorando sus habilidades para escribir código robusto.

## Generalidades de Requerimientos Funcionales para Proyecto de Investigación Científica

La aplicación de simulación para un proyecto de investigación científica incluye los siguientes requerimientos funcionales:

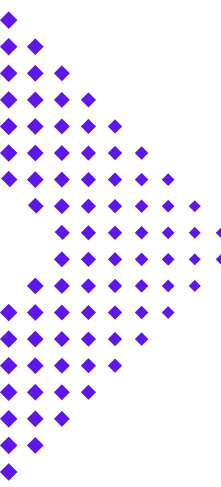
1. **Gestión de Experimentos:** Creación y visualización de experimentos con atributos específicos, como nombre, fecha, tipo y resultados numéricos.
2. **Análisis de Resultados:** Cálculo de estadísticas básicas (promedio, valor máximo y mínimo) y comparación entre experimentos para identificar el desempeño relativo.
3. **Generación de Informe:** Creación de un informe final que resuma los experimentos, estadísticas y conclusiones, con opción de exportar a archivo de texto.
4. **Interacción con el Usuario:** Menú interactivo para acceder a todas las funcionalidades y validación de entradas para asegurar la correcta introducción de datos.
5. **Flujo General del Sistema:** El programa se inicia con un menú de opciones, permite la gestión de experimentos, el análisis de datos, la generación de informes y finaliza con la opción de cerrar el programa.

## Metodología para el Reto 1: Proyecto de Investigación Científica en Python

Para el desarrollo de este proyecto en el curso "Python de Cero a Senior: La Ruta Maestra del Código", se seguirá una metodología colaborativa, que permite a los estudiantes trabajar en equipos, fortaleciendo tanto sus habilidades técnicas como de colaboración. Los estudiantes utilizarán herramientas profesionales, como Git y GitHub, para gestionar el proyecto de manera estructurada y profesional. A continuación, se presenta la metodología recomendada:

### 1. Trabajo en Equipos

- ❑ **Formación de Equipos:** Los estudiantes se organizarán en equipos de dos, lo cual será fundamental para intercambiar ideas, distribuir tareas, y resolver problemas en conjunto.
- ❑ **Distribución de Tareas:** Los equipos deben dividir las tareas de manera equitativa, según las habilidades y fortalezas de cada miembro. Por ejemplo, un estudiante puede enfocarse



en la lógica de recolección y análisis de datos, mientras el otro se encarga de la interfaz y validación de datos. Se sugiere la rotación de roles para que ambos integrantes puedan desarrollar habilidades en diferentes áreas de la aplicación.



## 2. Uso de Git y GitHub

- ❑ **Creación del Repositorio:** Cada equipo creará un repositorio en GitHub para subir el código fuente. Este repositorio debe estar accesible para que los docentes puedan revisarlo y proporcionar feedback.
- ❑ **Manejo de Ramas y Control de Versiones:**

Cada miembro trabajará en su propia rama para desarrollar funcionalidades específicas, evitando así conflictos en el código y facilitando el trabajo en paralelo. Una vez que las ramas estén completas, se realizará una integración en la rama principal (main), precedida por una revisión mutua del código.
- ❑ **Commits Regulares:** Se recomienda realizar commits frecuentes con mensajes claros y descriptivos para reflejar el progreso del proyecto. Es útil usar convenciones de mensajes, como "feat:" para nuevas funcionalidades y "fix:" para correcciones de errores.

## 3. Video de Presentación

- ❑ **Grabación del Video:** Una vez completado el proyecto, los estudiantes deberán grabar un video en el que expliquen las características de la aplicación. Se debe asegurar una buena calidad de audio y video para facilitar la evaluación.
- ❑ **Contenido del Video:**

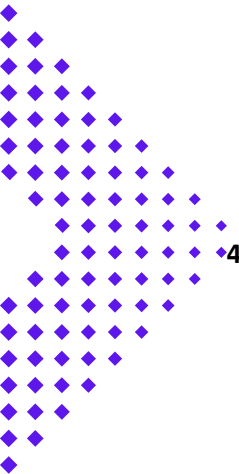
**Introducción:** Presentación breve del proyecto y los objetivos, mencionando desafíos y cómo fueron abordados.

**Demostración del Programa:** Mostrar la ejecución de la aplicación, detallando funciones como la gestión de experimentos, el análisis de resultados, y la generación del informe. También deben ilustrar cómo se implementaron las validaciones y los mensajes de error.

**Reparto Equitativo:** Ambos estudiantes deben participar activamente en la presentación, alternando la explicación de las funcionalidades.
- ❑ **Duración:** El video no debe exceder los 10 minutos. Es recomendable ensayar previamente para cubrir todos los puntos de forma clara y concisa.

## 4. Revisión y Feedback

- ❑ **Entrega del Proyecto:** Al finalizar el reto, los estudiantes deberán compartir los enlaces del repositorio de GitHub y del video de presentación en YouTube. Es importante que el repositorio esté bien documentado, con un archivo README que explique cómo ejecutar el proyecto.



- **Feedback:** Los docentes revisarán el código y el video para proporcionar retroalimentación. La evaluación tomará en cuenta la implementación de los requerimientos funcionales, el uso de Git y GitHub, y la claridad en la presentación. Se alentará a los estudiantes a recibir y reflexionar sobre la retroalimentación y podrán solicitar una reunión para discutirla si lo consideran necesario.



## Requerimientos Funcionales para la Aplicación de Proyecto de Investigación Científica

### 1. Gestión de Experimentos

#### o **Agregar un experimento:**

La aplicación debe permitir crear nuevos experimentos con los siguientes atributos:

- Nombre del experimento: Cadena de texto. Fecha de realización: Fecha en formato "DD/MM/AAAA". Tipo de experimento: Categoría predefinida, como "Química", "Biología" o "Física".
- Resultados: Lista de resultados numéricos obtenidos.

#### o **Visualizar experimentos:**

La aplicación debe permitir visualizar la lista de experimentos, mostrando sus detalles y resultados.

### 2. Análisis de Resultados

#### o **Cálculo de estadísticas básicas:**

Permitir al usuario calcular el promedio, el valor máximo y el mínimo de los resultados de un experimento.

#### o **Comparación entre experimentos:**

Comparar los resultados de dos o más experimentos para identificar el experimento con los mejores o peores resultados.

### 3. Generación de un Informe Resumen

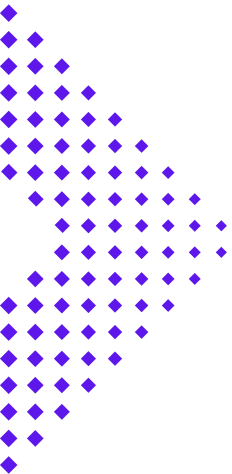
#### o **Crear un informe final:**

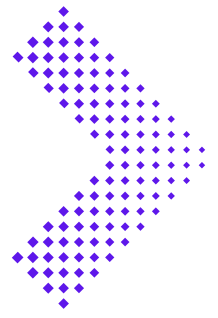
El sistema generará un informe que incluya la descripción de todos los experimentos, los análisis estadísticos realizados y una conclusión basada en los resultados.

#### o **Guardar el informe en un archivo de texto:**

La aplicación debe permitir exportar el informe a un archivo de texto (.txt).

### 4. Interacción con el Usuario





o **Menú de opciones:**

El sistema presentará un menú interactivo con opciones como agregar un experimento, ver experimentos, realizar análisis, generar informe y salir del programa.

o **Validación de entrada de datos:**

El sistema debe validar la entrada del usuario y manejar errores, como fechas mal formateadas o tipos de experimentos no reconocidos.

### Descripción General del Flujo del Sistema

**1. Inicio del Programa:**

El programa comienza presentando un menú con las opciones disponibles.

**2. Gestión de Experimentos:**

El usuario puede agregar, ver y eliminar experimentos según sea necesario.

**3. Análisis de Datos:**

El usuario puede seleccionar un experimento para calcular estadísticas o comparar varios experimentos.

**4. Generación del Informe:**

Se genera un informe final que resume todos los experimentos y los análisis realizados.

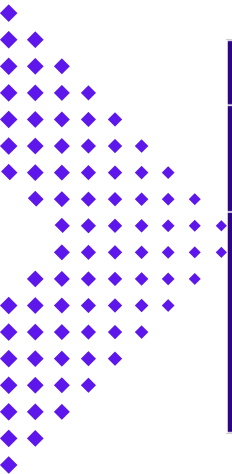
**5. Salir del Programa:**

El usuario puede cerrar el programa seleccionando la opción correspondiente en el menú.

### Rúbrica de Evaluación para el Reto Científico: "Simulación de un Proyecto Científico con Python"

Esta rúbrica especifica los criterios para evaluar el ejercicio práctico sobre la simulación de un proyecto científico con Python. El objetivo es medir la implementación técnica, el uso adecuado de herramientas de desarrollo colaborativo, la presentación del trabajo, y la documentación entregada. La calificación total será de 100 puntos.

Criterio	Descripción	Puntos
<b>1. Implementación Técnica</b>		<b>50</b>
<b>Simulación de datos y experimentación</b>	El sistema permite generar datos científicos simulados con diferentes configuraciones de parámetros, y simula correctamente los resultados del experimento.	15



<b>Análisis y procesamiento de datos</b>	La aplicación incluye funcionalidades para el análisis de datos, como cálculos estadísticos, filtrado de datos y generación de gráficos relevantes.	15
<b>Modelado y ajuste de parámetros</b>	La simulación permite ajustar parámetros del modelo científico y observar el impacto en los resultados de manera clara y comprensible.	10
<b>Validaciones y manejo de errores</b>	El sistema maneja adecuadamente entradas inválidas y asegura la estabilidad del programa durante la ejecución de la simulación y el análisis.	10
<b>2. Uso de Git y GitHub</b>		<b>20</b>
<b>Uso de ramas y trabajo colaborativo</b>	El desarrollo del proyecto fue gestionado con ramas separadas para evitar conflictos de código. El trabajo se distribuyó equitativamente entre los miembros.	10
<b>Commits regulares con mensajes descriptivos</b>	Los commits se hicieron de manera frecuente, y los mensajes son claros, indicando los cambios realizados en cada etapa del desarrollo del proyecto.	10
<b>3. Presentación del Video</b>		<b>20</b>
<b>Explicación del proyecto y objetivos</b>	Los estudiantes presentan adecuadamente el proyecto, explicando sus objetivos, el contexto científico, y la funcionalidad principal de la simulación	5
<b>Demostración de la funcionalidad</b>	El video muestra una ejecución clara del programa, destacando las características clave, como la generación de datos, el análisis y el modelado.	10
<b>Claridad y equidad en la explicación</b>	Ambos estudiantes participan en la presentación, alternando la explicación de diferentes partes del proyecto, y la comunicación es clara.	5
<b>4. Documentación</b>		<b>10</b>
<b>Comentarios y claridad del código</b>	El código está debidamente comentado, explicando funciones y secciones clave, y sigue una estructura clara y comprensible.	5

## README en GitHub

El repositorio contiene un archivo README bien redactado, que describe el proyecto, las instrucciones de instalación y uso, y las funcionalidades principales.

5

### Desglose de Puntos

- ☐ **Implementación Técnica:** 50 puntos
- ☐ **Uso de Git y GitHub:** 20 puntos
- ☐ **Presentación del Video:** 20 puntos
- ☐ **Documentación:** 10 puntos

**Total Máximo: 100 puntos**

### Puntuación Final:

- ☐ **Excelente (90-100 puntos):** El proyecto es completamente funcional, el uso de herramientas colaborativas es adecuado, y la presentación es clara y completa.
- ☐ **Bueno (80-89 puntos):** El proyecto es funcional, con algunas áreas de mejora. La presentación y la documentación son adecuadas, pero pueden ser más detalladas.
- ☐ **Satisfactorio (70-79 puntos):** El proyecto cumple con la mayoría de los requerimientos, pero presenta deficiencias en la implementación o la colaboración no es equitativa.
- ☐ **Insuficiente (menos de 70 puntos):** El proyecto tiene deficiencias significativas en la funcionalidad, la colaboración o la claridad de la presentación en video.