

Iniciación con Python

Clase 16 - “Proyecto Final”

¡Les damos la bienvenida!



Vamos a comenzar a grabar la clase

Clase **15.**

► SQLite

1. Definimos y creamos la base de datos.
2. Agregamos a las funciones creadas antes de las consultas de SQL.
3. Revisión del enunciado del TFI.

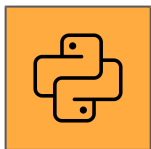
Clase **16.**

► Proyecto Final

1. Módulo Colorama.
2. Retroalimentación general del curso y de los proyectos.
3. Conclusiones y cierre final.



Colorama



Módulo Colorama

En esta última clase, aprenderemos a usar el módulo Colorama, una herramienta que permite añadir color y estilos a los textos en la consola. Esto te permitirá mejorar la presentación visual de tus programas, facilitando la lectura y destacando información importante.

		red	green	yellow	blue		cyan	white
black	X	X X X	X X X	X X X	X X X	X	X X X	X X X
red	X X X	X X X	X X X	X X X	X X X	X X X	X X X	X X X
green	X X X	X X X	X X X	X X X	X X X	X X X	X X X	X X X
	X X X	X X X	X X X	X X X	X X X	X X X	X X X	X X X
blue	X X X	X X X	X X X	X X X	X X X	X X X	X X X	X X X
magenta	X	X X X	X X X	X X X	X X X	X	X X X	X X X
cyan	X X X	X X X	X X X	X X X	X X X	X X X	X X X	X X X
white	X X X	X X X	X X X	X X X	X X X	X X X	X X X	X X X



Instalación de Colorama

Colorama es un módulo externo, lo que significa que necesitás instalarlo antes de usarlo. La instalación se realiza fácilmente mediante el gestor de paquetes **pip** con el siguiente comando:

```
pip install colorama
```

Con este paso, el módulo queda listo para que puedas utilizarlo en tus proyectos.



Importación del módulo

Una vez instalado, el módulo Colorama se importa usando la sintaxis que ves a continuación.

```
from colorama import init, Fore  
  
init() # Inicializa Colorama
```

La función `init()` es fundamental, ya que prepara el entorno para usar colores y estilos en la consola, especialmente en sistemas Windows.

Con este comando de inicialización, queda todo listo para que puedas comenzar a aplicar colores a los textos que tu programa mostrará en la terminal.



Secuencias ANSI

^[0;49;31m	^[1;49;31m	^[2;49;31m	^[4;49;31m	^[5;49;31m	^[7;49;31m
^[0;49;32m	^[1;49;32m	^[2;49;32m	^[4;49;32m	^[5;49;32m	^[7;49;32m
^[0;49;33m	^[1;49;33m	^[2;49;33m	^[4;49;33m	^[5;49;33m	^[7;49;33m
^[0;49;34m	^[1;49;34m	^[2;49;34m	^[4;49;34m	^[5;49;34m	^[7;49;34m
^[0;49;35m	^[1;49;35m	^[2;49;35m	^[4;49;35m	^[5;49;35m	^[7;49;35m
^[0;49;36m	^[1;49;36m	^[2;49;36m	^[4;49;36m	^[5;49;36m	^[7;49;36m
^[0;49;37m	^[1;49;37m	^[2;49;37m	^[4;49;37m	^[5;49;37m	^[7;49;37m
^[0;49;90m	^[1;49;90m	^[2;49;90m	^[4;49;90m	^[5;49;90m	^[7;49;90m
^[0;49;91m	^[1;49;91m	^[2;49;91m	^[4;49;91m	^[5;49;91m	^[7;49;91m
^[0;49;92m	^[1;49;92m	^[2;49;92m	^[4;49;92m	^[5;49;92m	^[7;49;92m
^[0;49;93m	^[1;49;93m	^[2;49;93m	^[4;49;93m	^[5;49;93m	^[7;49;93m
^[0;49;94m	^[1;49;94m	^[2;49;94m	^[4;49;94m	^[5;49;94m	^[7;49;94m
^[0;49;95m	^[1;49;95m	^[2;49;95m	^[4;49;95m	^[5;49;95m	^[7;49;95m
^[0;49;96m	^[1;49;96m	^[2;49;96m	^[4;49;96m	^[5;49;96m	^[7;49;96m
^[0;49;97m	^[1;49;97m	^[2;49;97m	^[4;49;97m	^[5;49;97m	^[7;49;97m
^[0;49;39m	^[1;49;39m	^[2;49;39m	^[4;49;39m	^[5;49;39m	^[7;49;39m

Colorama facilita el uso de secuencias **ANSI**, un conjunto de códigos que permite aplicar colores y estilos al texto en la terminal. Estas secuencias de texto indican a la terminal cómo debe presentarse el texto, pero suelen ser complejas y poco intuitivas. Colorama simplifica este proceso, haciendo que nuestro código sea más fácil de leer y compatible con distintos sistemas operativos.



Función init()

La función `init()` es esencial para que Colorama funcione correctamente en todas las plataformas. Ofrece parámetros como `autoreset`, que permite que el color de texto vuelva automáticamente al valor original después de cada línea.

Esto es útil para evitar que los cambios de color afecten el resto de la salida en consola.

```
from colorama import init, Fore
init(autoreset=True)
print(Fore.RED + "Este texto es rojo.")
print("Este texto vuelve al color original automáticamente.")
```

```
Este texto es rojo.
Este texto vuelve al color original automáticamente.
```



Fore: cambiar el color del texto

```
from colorama import init, Fore
init(autoreset=True)

print(Fore.RED + "Cuidado: estás a punto de eliminar un producto.")
print(Fore.GREEN + "Producto registrado con éxito.")
```

```
Cuidado: estás a punto de eliminar un producto.
Producto registrado con éxito.
```

Fore es el componente que usamos para cambiar el color del texto en pantalla.

Colorama nos permite utilizar colores como `Fore.RED`, `Fore.GREEN`, y `Fore.BLUE`, que son útiles para destacar alertas, confirmaciones y otros mensajes importantes.



Back: cambiar el color de fondo

Back permite cambiar el color de fondo de los mensajes en consola, lo cual es útil para hacer que ciertos mensajes destaquen aún más.

Al igual que Fore, Colorama incluye colores básicos para el fondo, como Back.YELLOW o Back.RED.

```
from colorama import init, Back
init(autoreset=True)

print(Back.YELLOW + "Aviso: el stock de este producto es muy bajo.")
```

```
Aviso: el stock de este producto es muy bajo.
```



Style: cambiar el estilo del texto

```
from colorama import init, Style, Fore
init(autoreset=True)

print(Style.BRIGHT + Fore.BLUE + "Bienvenido/a a la gestión de
inventario.")
print(Style.DIM + "Nota: asegúrate de ingresar un ID válido.")
```

```
Bienvenido/a a la gestión de inventario.
Nota: asegúrate de ingresar un ID válido.
```

Con **Style**, podemos ajustar la intensidad del color del texto. Por ejemplo, `Style.BRIGHT` es ideal para títulos, mientras que `Style.DIM` se usa para mensajes secundarios.

Esto nos ayuda a diferenciar distintos tipos de información en nuestros programas.



Ejemplo práctico

```
from colorama import Style, Back, Fore
print(Back.YELLOW + Fore.RED + Style.BRIGHT + "Alerta: el stock de
este producto es crítico. Se recomienda reponer.")
```

Uniendo **Fore**, **Back** y **Style**, podemos personalizar completamente los mensajes en consola.

```
Alerta: el stock de este producto es crítico. Se recomienda reponer.
```

Esta funcionalidad es importante para destacar advertencias y confirmaciones.



Mejora visual del menú del TFI

```
Menú de Gestión de Personas
```

1. Registrar persona
2. Mostrar personas
3. Actualizar edad de una persona
4. Eliminar persona
5. Buscar persona
6. Reporte de personas menores de edad
7. Salir

```
Seleccione una opción: 7
```

```
Saliendo del programa...
```

En tu proyecto final, podés mejorar el menú de opciones utilizando Colorama para hacerlo más atractivo visualmente, añadiendo color a las opciones de menú y mensajes de error, facilitando la navegación y comprensión de la interfaz.

(Código en la diapositiva siguiente)



Mejora visual del menú del TFI

```
from colorama import init, Fore, Back, Style
init(autoreset=True)

def mostrar_menu():
    while True:
        print(Style.BRIGHT + Fore.BLUE + Back.WHITE + "\nMenú de Gestión de Personas".center(40, " "))
        print(Style.BRIGHT + Fore.CYAN + "1. Registrar persona")
        print(Style.BRIGHT + Fore.CYAN + "2. Mostrar personas")
        print(Style.BRIGHT + Fore.CYAN + "3. Actualizar edad de una persona")
        print(Style.BRIGHT + Fore.CYAN + "4. Eliminar persona")
        print(Style.BRIGHT + Fore.CYAN + "5. Buscar persona")
        print(Style.BRIGHT + Fore.CYAN + "6. Reporte de personas menores de edad")
        print(Style.BRIGHT + Fore.RED + "7. Salir")
        opcion = input(Fore.YELLOW + "\nSeleccione una opción: ")

        if opcion == "7":
            print(Fore.RED + "Saliendo del programa...")
            break
        else:
            print(Back.RED + Fore.WHITE + "Opción inválida. Por favor, intente nuevamente.")

mostrar_menu() # Iniciar el programa llamando al menú principal
```



¡Felicitaciones por llegar hasta aquí!



A lo largo de este curso, juntos recorrimos un camino lleno de aprendizajes: desde los fundamentos de Python, el uso de estructuras como diccionarios y bucles, hasta la implementación de bases de datos para gestionar información con persistencia de datos.

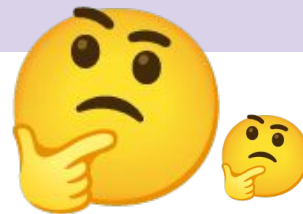
Han dado vida a una aplicación funcional y profesional con su **Proyecto Final Integrador**, demostrando no sólo dominio técnico, sino también creatividad y esfuerzo. Este logro es una muestra de su capacidad para enfrentar desafíos y aprender de cada paso del proceso. Les animamos a que sigan avanzando, explorando y ampliando sus horizontes en la programación, porque este es sólo el comienzo de un camino lleno de posibilidades.



¡El futuro está en sus manos!



¿Cómo puedo seguir formándome?



Data Scientist: Data Analytics

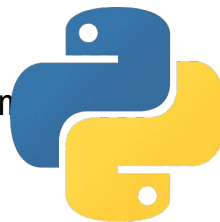
Es un campo que combina programación, análisis y estadísticas para extraer información valiosa de grandes volúmenes de datos.

En el curso de “**Data Analytics**” aprenderás a analizar datos para identificar patrones, responder preguntas específicas y tomar decisiones. Es una especialidad que te permitirá manejar herramientas y lenguajes como Python, SQL y librerías especializadas para el análisis de datos.

Ambos caminos complementan lo que ya han aprendido en este curso y son una excelente manera de ampliar sus habilidades en áreas de alta demanda en el mercado laboral.



Google Sheets



 **pandas**

 **NumPy**

Full Stack Developer: Front-End Js

Si te apasiona crear soluciones tecnológicas completas, el camino de **Full Stack Developer** es una excelente oportunidad para continuar tu formación. Este rol te permite diseñar aplicaciones tanto en su apariencia como en su funcionamiento interno, convirtiéndote en un profesional altamente versátil y demandado.



Markup Language
Content
HTML



Style sheet Language
Presentation
CSS



Programming Language
Behavior
JavaScript

En el **Front-End**, aprenderás a construir interfaces atractivas y funcionales usando **HTML**, **CSS** y **JavaScript**. Estas herramientas son esenciales para desarrollar páginas web dinámicas y amigables, donde cada clic y cada diseño están pensados para la mejor experiencia del usuario.

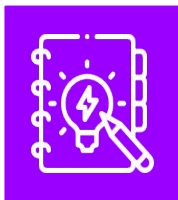
Game Developer: Unity 2D

Si te interesa combinar creatividad, lógica y tecnología para crear mundos interactivos, el camino del **Game Developer** es ideal para vos.



El desarrollo de videojuegos no solo es apasionante, sino que también te permite aplicar todo lo que aprendiste sobre programación.

Podés empezar con **Unity 2D**, donde aprenderás a desarrollar juegos en dos dimensiones, trabajando con mecánicas sencillas y divertidas que te permitirán dominar los fundamentos del diseño y la **programación de videojuegos**.



Entrega de Proyecto



Entrega de Proyecto



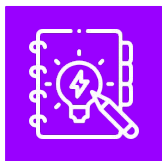
Obligatorio | Entregable

¡Felicidades por haber llegado al final de la cursada! 🎉

Es momento de presentar el proyecto final que has estado desarrollando durante este recorrido de aprendizaje.

Recuerda que las consignas detalladas fueron discutidas durante la clase 15. Te recomendamos revisarlas cuidadosamente antes de la entrega para asegurarte de cumplir con todos los requisitos.

¡Estamos ansiosos por ver tu trabajo y el esfuerzo que has puesto en él! 🚀



Entrega de Proyecto



Obligatorio | Entregable

Formato de entrega: Compartir un link al drive (público) que contenga los archivos y carpetas que conforman tu proyecto. Los links deberán ser entregados en el apartado de “Pre-Entrega de Proyecto” en el Campus Virtual.



Tu aplicación le debe permitir a la persona que lo utiliza efectuar:

- **Registro de productos:** Ingresar nuevos productos al inventario, solicitando nombre, descripción, cantidad disponible, precio y categoría.
- **Consulta de productos:** Consultar el inventario y ver la información detallada de cada producto, como stock disponible y precio.
- **Actualización de productos:** Modificar la cantidad disponible de un producto específico.
- **Eliminación de productos:** Permitir eliminar productos del inventario.
- **Listado Completo:** Generar un listado completo del inventario.
- **Reporte de Bajo Stock:** Mostrar un reporte de productos con bajo stock.