

Guion Explicativo - Sistema Académico

Escena: Presentación del sistema en un video explicativo

◇ Interlocutores:

- **José** → Explica las funciones generales.
- **Ricardo** → Complementa con detalles técnicos y ejemplos prácticos.

Introducción

José: ¡Hola a todos! Bienvenidos a la presentación del **Sistema Académico**, una plataforma diseñada para mejorar la gestión educativa, facilitar la comunicación y optimizar el aprendizaje.

Ricardo: Así es. En este video, te mostraremos **cómo funciona cada pestaña**, explicando su propósito y cómo interactuar con ella. También analizaremos el código para que los interesados en el desarrollo puedan entender su implementación.

1. Registro de Estudiantes

José: Empezamos con la pestaña de **Registro de Estudiantes**, donde los usuarios pueden ingresar el **nombre, apellido y nota** de un estudiante.

Ricardo: Desde el punto de vista técnico, esta pestaña usa **campos de entrada en Tkinter**, almacenando los datos en una lista llamada estudiantes. Cuando se presiona el botón "Registrar Estudiante", el sistema verifica que los datos sean correctos antes de registrarlos.

Ejemplo en código:

```
python
def
    nombre =
    apellido =
    nota =

    registrar_estudiante():
        nombre_estudiante.get().strip()
        apellido_estudiante.get().strip()
        nota_estudiante.get().strip()

    if nombre and apellido and nota.isdigit():
```

```
estudiantes.append({"nombre": nombre, "apellido": apellido, "nota": int(nota)})
messagebox.showinfo("Registro", f"{nombre} {apellido} registrado con nota {nota}.")
```



José: Básicamente, si un usuario ingresa los datos correctamente y pulsa el botón, el sistema guarda la información del estudiante automáticamente.

2. Mensajes Académicos

José: La siguiente pestaña es **Mensajes Académicos**, donde los usuarios pueden ver **notificaciones sobre exámenes, entregas y otros eventos importantes**.

Ricardo: La pestaña usa una **lista de mensajes preexistentes** almacenados en una Listbox. Esto facilita la navegación dentro del sistema, permitiendo ver anuncios sin intervención del usuario.

Ejemplo en código:

```
python
mensajes_preexistentes = [
    " Recordatorio: La entrega de proyectos finales es el 15 de junio.",
    " Examen de Matemáticas programado para el 20 de junio."
]
```

José: Así, los estudiantes pueden acceder a la pestaña en cualquier momento y revisar la información sin perder fechas clave.

3. Calendario Académico

José: Luego tenemos el **Calendario Académico**, que muestra eventos programados, como exámenes y entregas de trabajos.

Ricardo: A nivel técnico, esta pestaña usa un **Listbox** con eventos predefinidos. En futuras mejoras, podríamos agregar la opción de personalizar fechas o integrar una base de datos para actualizar eventos dinámicamente.

4. Rendimiento Académico

José: La pestaña de **Rendimiento Académico** permite analizar el desempeño de los estudiantes mediante cálculos automáticos.

Ricardo: Esta sección procesa las notas registradas, **calculando el promedio, la nota máxima y la nota mínima**.

Ejemplo en código:

```
python
def calcular_estadisticas():
    if estudiantes:
        notas = [est["nota"] for est in estudiantes]
        promedio = sum(notas) / len(notas)
        nota_maxima = max(notas)
        nota_minima = min(notas)
```

José: Una función bastante útil para los docentes, ya que permite obtener un análisis rápido del rendimiento académico de los alumnos.

5. Material de Apoyo

José: En la pestaña **Material de Apoyo**, los usuarios pueden **subir archivos y agregar enlaces** a contenido educativo.

Ricardo: Esta funcionalidad permite cargar archivos mediante un **file dialog**, y guardar enlaces en una lista interactiva que abre las páginas en el navegador.

Código para abrir un enlace:

```
python
def abrir_enlace(event):
    seleccion = lista_enlaces.curselection()
    if seleccion:
        enlace = lista_enlaces.get(seleccion[0])
```

```
webbrowser.open(enlace)
```

José: Esto facilita el acceso a información complementaria sin necesidad de salir del sistema.

6. Foro Académico

José: Finalmente, tenemos el **Foro Académico**, un espacio donde los estudiantes pueden **intercambiar ideas y resolver dudas**.

Ricardo: Desde la parte técnica, los mensajes se agregan a una Listbox, permitiendo visualizar conversaciones en tiempo real.

Ejemplo en código:

```
python
def                                     publicar_mensaje():
    mensaje                             = entrada_mensaje_foro.get().strip()
    if                                  mensaje:
        mensajes_foro.insert(tk.END, f"Usuario: {mensaje}")
```

José: Es una excelente opción para fomentar la comunicación académica y el aprendizaje colaborativo.

Cierre y Conclusión

José: Como hemos visto, el **Sistema Académico** es una plataforma versátil que permite **registrar estudiantes, compartir recursos y analizar rendimiento académico**.

Ricardo: Además, su implementación en **Python con Tkinter** lo hace accesible y fácil de mejorar con futuras actualizaciones.

José: ¡Gracias por acompañarnos en esta presentación! Esperamos que el sistema les resulte útil.