

¿Qué decisiones de diseño tomaste durante el desarrollo y por qué?

Clases y Encapsulamiento:

- Decisión: Utilizar clases para representar empleados, proyectos y asignaciones.
- Razón: El uso de clases permite encapsular los datos y las funciones relacionadas en una sola entidad, lo que facilita la organización del código y la reutilización de componentes

Validaciones en el Constructor:

- Decisión: Incluir validaciones en el constructor de las clases Empleado y Proyecto.
- Razón: Realizar validaciones en el constructor asegura que los objetos se creen con datos válidos desde el principio, evitando errores y problemas más adelante en el programa. Esto también mejora la robustez del sistema.

Uso de Vectores Estáticos para Datos Únicos:

- Decisión: Utilizar vectores estáticos para almacenar correos y nombres de proyectos únicos.
- Razón: Los vectores estáticos permiten mantener un registro de los correos y nombres de proyectos que ya han sido utilizados

Funciones de Consulta y Modificación:

- Decisión: Implementar funciones para consultar y modificar empleados.
- Razón: Estas funciones permiten acceder y actualizar la información de los empleados de manera controlada.

Interfaz de Usuario Interactiva:

- Decisión: Incluir un menú interactivo para que el usuario pueda seleccionar diferentes opciones.
- Razón: Un menú interactivo mejora la experiencia del usuario al proporcionar una forma clara y sencilla de interactuar

Función de Continuación:

- Decisión: Implementar una función para preguntar al usuario si desea continuar después de cada operación.
- Razón: Esta función permite al usuario realizar múltiples operaciones en una sola ejecución

¿Qué aspectos del problema fueron más difíciles de modelar y cómo los resolviste?

Validación de Datos:

Desafío: Asegurar que los datos ingresados por el usuario fueran válidos y consistentes, especialmente para campos como edad, salario, categoría y correos electrónicos únicos.

Solución: Implementé validaciones en los constructores de las clases Empleado y Proyecto. Esto incluyó verificar que la edad fuera mayor o igual a 18

Generación de Reportes:

Desafío: Proporcionar una manera de generar reportes detallados de los empleados registrados en el sistema.

Solución: Implementé una función para generar reportes que recorre el vector de empleados y muestra los datos de cada uno de ellos.

Si tuvieras más tiempo, ¿qué mejorarías o agregarías al sistema?

Poder terminarlo y corregir varios errores que tuve