

Decisiones de diseño tomadas durante el desarrollo:

- **Uso de clases y estructuras:** Decidí organizar los datos de los empleados y proyectos utilizando clases y estructuras, ya que es una forma clara y ordenada de manejar la información. Esto hace que el código sea más fácil de leer y mantener.
- **Validaciones de datos:** Implementé validaciones dentro de los constructores de las clases para asegurarme de que los datos ingresados fueran correctos. Por ejemplo, me aseguré de que los empleados fueran mayores de edad y que el salario estuviera dentro de un rango válido.
- **Métodos específicos:** Creé métodos dentro de las clases para mostrar la información de los empleados y proyectos, lo que facilita la consulta y actualización de los datos.
- **Uso de vector:** Opté por usar vector para almacenar a los empleados y proyectos, ya que es una estructura dinámica que crece y se ajusta según se agregan más elementos, lo que facilita la gestión de los datos.
- **Transformación de texto:** Implementé un método para convertir las entradas de texto a minúsculas y eliminar las tildes, lo que ayuda a evitar errores al comparar cadenas y garantiza que las entradas sean consistentes.

Aspectos más difíciles:

- **Validación de datos de entrada:** Fue complicado asegurarme de que el usuario ingresara datos correctos. Tuve que implementar múltiples validaciones, como asegurarse de que el salario estuviera dentro de un rango válido o que el carnet del empleado fuera único.
- **Manejo de errores:** Aunque se implementaron varias validaciones, a veces era complicado manejar los errores de

entrada y mantener el flujo del programa sin que se detuviera por un simple error de usuario.

Mejoras o adiciones si tuviera más tiempo:

- Base de datos: Si tuviera más tiempo, implementaría una base de datos para almacenar la información de los empleados y proyectos, lo que permitiría que los datos se mantuvieran de manera más segura y eficiente.
- Interfaz gráfica: Agregar una interfaz gráfica de usuario (GUI) sería una mejora significativa, ya que haría que la interacción con el sistema fuera más amigable y visual.
- Manejo de fechas: Mejoraría la forma en que se gestionan las fechas, ya que en este caso se están utilizando como cadenas de texto, lo que no es lo más eficiente. Usar una librería especializada en fechas sería una opción más adecuada.
- Roles de usuario: Implementaría un sistema de roles para los usuarios, de manera que solo los administradores puedan modificar ciertos datos o realizar ciertas acciones.

Clases necesarias para modelar este sistema:

- Empleado: Para representar a cada trabajador, almacenando toda la información relevante (nombre, carnet, salario, etc.).
- Proyecto: Para representar los proyectos, con sus detalles como código, nombre y fechas.
- SistemaGestion: Para gestionar tanto a los empleados como a los proyectos, y permitir el registro, modificación y consulta de ambos.
- Validaciones: Aunque no es una clase por sí misma, sería útil tener un conjunto de funciones o una clase dedicada a realizar todas las validaciones necesarias para garantizar la calidad de los datos.

Atributos que deben incluirse en cada clase:

- Empleado:
 - carnet: Identificador único del empleado.
 - nombre: Nombre completo del empleado.
 - fechaNacimiento: Fecha de nacimiento para calcular la edad.
 - categoria: El tipo de trabajo del empleado (administrador, operario, peón).
 - salario: Salario mensual del empleado.
 - direccion: Dirección donde vive el empleado.
 - telefono: Número de teléfono para contacto.
 - correo: Correo electrónico.
 - edad: Edad calculada a partir de la fecha de nacimiento.
- Proyecto:
 - codigo: Identificador único del proyecto.
 - nombre: Nombre del proyecto.
 - fechaInicio: Fecha en la que comenzó el proyecto.
 - fechaFin: Fecha de finalización del proyecto.

Diagrama escrito:

1. Clase Empleado:

- Esta clase representa a un empleado en el sistema.

- Contiene varios atributos para almacenar información sobre el empleado, como su carnet (identificador único), nombre, fechaNacimiento (para calcular su edad), categoria (que puede ser "administrador", "operario" o "peón"), salario, direccion, telefono, correo y edad.
- Además, tiene un constructor que inicializa estos atributos y realiza validaciones para asegurarse de que los datos sean correctos (por ejemplo, que el salario esté dentro de un rango válido o que la categoría sea una de las tres permitidas).
- La clase también incluye un método mostrarInformacion() para imprimir los datos del empleado en la consola.

2. Clase Proyecto:

- Esta clase representa un proyecto en el sistema.
- Tiene los atributos codigo (identificador único del proyecto), nombre (nombre del proyecto), fechaInicio (fecha en la que el proyecto comienza) y fechaFin (fecha de finalización del proyecto).
- Al igual que la clase Empleado, la clase Proyecto tiene un constructor que inicializa estos atributos y un método mostrarInformacion() para imprimir los detalles del proyecto en la consola.

3. Clase SistemaGestion:

- Esta clase es el núcleo del sistema y gestiona tanto a los empleados como a los proyectos.
- Contiene dos atributos de tipo vector:
 - empleados: un vector que almacena todos los objetos de tipo Empleado.

- proyectos: un vector que almacena todos los objetos de tipo Proyecto.
- La clase incluye varios métodos para interactuar con los empleados y proyectos:
 - Gestión de empleados: Métodos para registrar, consultar y modificar empleados. Cada vez que se registra un nuevo empleado, se verifica que no haya otro con el mismo carnet y se validan los datos introducidos.
 - Gestión de proyectos: Métodos para agregar, consultar y modificar proyectos. Los proyectos se identifican por su código único, y se pueden modificar detalles como su nombre y fecha de finalización.
- Además, incluye validaciones dentro de cada método para asegurar que los datos introducidos sean correctos (por ejemplo, verificar que el salario de un empleado esté dentro de un rango específico o que la categoría del empleado sea válida).

Relación entre las clases:

- La clase SistemaGestion tiene una relación de "gestión" con las clases Empleado y Proyecto.
- Es decir, SistemaGestion actúa como el contenedor principal que organiza y administra los objetos de tipo Empleado y Proyecto.
- Los métodos de la clase SistemaGestion permiten agregar, modificar y consultar tanto empleados como proyectos, interactuando con las instancias de estas dos clases para realizar las acciones solicitadas.