**1. ¿Qué clases principales crees que son necesarias para modelar este sistema? Justifica tus elecciones.**

1. **Empleado**:
   * Representa a cada trabajador de la empresa.
   * Justificación: Es fundamental para gestionar la información personal y profesional de los empleados como número de carnet, nombre, categoría, salario, etc.
2. **Proyecto**:
   * Representa los proyectos asignados a los empleados.
   * Justificación: Se necesita una estructura para almacenar detalles como el código único del proyecto, nombre, fecha de inicio y fecha de finalización.
3. **GestorAsignaciones**:
   * Se encarga de la asignación de empleados a proyectos y de gestionar todas las operaciones relacionadas con estas asignaciones.
   * Justificación: Facilita la administración de asignaciones y garantiza que se cumplan las reglas de negocio (como empleados asignados a múltiples proyectos).

**Fase 1: Análisis y Diseño del Sistema**

**1. ¿Qué clases principales crees que son necesarias para modelar este sistema? Justifica tus elecciones.**

1. **Empleado**:
   * Representa a cada trabajador de la empresa.
   * Justificación: Es fundamental para gestionar la información personal y profesional de los empleados como número de carnet, nombre, categoría, salario, etc.
2. **Proyecto**:
   * Representa los proyectos asignados a los empleados.
   * Justificación: Se necesita una estructura para almacenar detalles como el código único del proyecto, nombre, fecha de inicio y fecha de finalización.
3. **GestorAsignaciones**:
   * Se encarga de la asignación de empleados a proyectos y de gestionar todas las operaciones relacionadas con estas asignaciones.
   * Justificación: Facilita la administración de asignaciones y garantiza que se cumplan las reglas de negocio (como empleados asignados a múltiples proyectos).

**2. ¿Qué atributos y métodos deben incluirse en cada clase para cumplir con los requisitos planteados?**

1. **Clase Empleado**
   * **Atributos**:
     + int numero\_carnet
     + std::string nombre
     + std::string fecha\_nacimiento
     + std::string categoria (solo "Administrador", "Operario", "Peón")
     + double salario (entre 250,000 y 500,000)
     + std::string direccion (por defecto "San José")
     + std::string telefono
     + std::string correo (único)
   * **Métodos**:
     + Constructor para inicializar los atributos con validaciones.
     + Métodos para consultar (getters) y modificar (setters) los datos del empleado.
     + Método para mostrar la información completa del empleado.
2. **Clase Proyecto**
   * **Atributos**:
     + int codigo\_proyecto (único)
     + std::string nombre (único)
     + std::string fecha\_inicio
     + std::string fecha\_finalizacion
   * **Métodos**:
     + Constructor para inicializar los atributos con validaciones.
     + Métodos para consultar (getters) y modificar (setters) los datos del proyecto.
     + Método para mostrar la información completa del proyecto.
3. **Clase GestorAsignaciones**
   * **Atributos**:
     + Contenedor (por ejemplo, std::map) para registrar las asignaciones de empleados a proyectos.
   * **Métodos**:
     + Método para asignar un empleado a un proyecto.
     + Método para consultar las asignaciones.
     + Método para mostrar todas las asignaciones con detalles de empleados y proyectos.

Pseudocodigo

+------------------+ asigna a +------------------+

| Empleado |--------------------------->| Proyecto |

+------------------+ +------------------+

| - numero\_carnet | | - codigo\_proyecto|

| - nombre | | - nombre |

| - categoria | | - fecha\_inicio |

| - salario | | - fecha\_fin |

| - direccion | +------------------+

| - telefono |

| - correo |

+------------------+

|

|

v

+------------------------+

| GestorAsignaciones |

+------------------------+

| - asignaciones | (por ejemplo, std::map<Empleado, Proyecto>)

| |

| + asignarEmpleado() |

| + mostrarAsignaciones()|

+------------------------+