**Titulo del Proyecto:**

Sistema de Gestión para Club de Voleibol (CVO)

**Integrantes:**

Joseph Caleb Perez Carrascal

Desarrollador y Documentador

192464

**Introducción:**

Por medio de Java, trabajé para ejecutar el código de un proyecto, Implementando el SDLC, que me permitiera satisfacer unas necesidades de quien en este caso seria mi cliente, club de voleibol Ocaña.

**Problemática:**

El club de voleibol Ocaña (cvo) quiso actualizarse en el método de control de pagos de mensualidades, ya que todo hoy en día se esta manejando digitalmente, pusieron en mis manos un programa que les permita llevar el control de la cantidad de jugadores con sus datos, identificarlos a cada uno con un numero de ID distinto, y concorde a ello, que me muestre las estadísticas financieras de cada jugador, sus meses cancelados y los faltantes por pagar, el monto total y demás.

**Desarrollo:**

**Detalle de los módulos desarrollados**

El sistema se estructura en torno a una aplicación Java orientada a objetos para gestionar jugadores del club de voleibol Ocaña. Los principales módulos o clases desarrolladas son:

* **Persona**  
  Clase base que contiene atributos comunes como nombre, apellido. Sirve como superclase para otras entidades del sistema, fomentando la reutilización de código.
* **Jugador** (extiende de Persona)  
  La clase Jugador representa una especialización de la entidad Persona, diseñada para modelar a los integrantes activos del club de voleibol. Su función principal es almacenar y validar los datos específicos de un jugador
* **ClubVoleibol**  
  Clase que gestiona una colección de jugadores mediante una lista interna (ArrayList). Permite agregar, mostrar, buscar y eliminar jugadores.
* **Main**  
  Clase principal con un menú interactivo basado en consola para facilitar la gestión de los jugadores. Utiliza la clase Scanner para la entrada del usuario y delega las acciones a ClubVoleibol.

**Tecnologías empleadas:**

* **Lenguaje:** Java
* **Paradigma:** Programación orientada a objetos.
* **Colecciones:** Uso de ArrayList para la gestión dinámica de jugadores
* **Entrada/Salida:** Interfaz por consola usando Scanner
* **Validaciones:** Control mediante condicionales (if, try-catch) para asegurar la entrada correcta del usuario

**Justificación de decisiones técnicas:**

* **Modularización con Orientación a objetos:** Se optó por el enfoque orientado a objetos para separar responsabilidades: la clase Jugador se enfoca en representar entidades, mientras que ClubVoleibol gestiona la lógica de negocio. Esto facilita la mantenibilidad y escalabilidad.
* **Herencia:** La herencia de Jugador desde Persona reduce redundancia, permitiendo que varias clases compartan atributos comunes.
* **ArrayList:** Se eligió esta estructura por su flexibilidad para agregar y eliminar elementos sin necesidad de gestionar el tamaño.

**Fases SDLC:**

**Planificación:** Principalmente en esta fase, nos hacemos un mapa o modelo mental, planificando mas o menos como se estructuraría el código.

**Análisis:** Empezamos a reconocer requisitos, necesidades, los objetivos a cumplir y demás para poner en marcha la siguiente fase, en este caso, ya sabemos que lo que tenemos que poner en marcha es un programa que nos muestre el reporte de cada jugador en cuanto a financieramente.

**Diseño:** se definió una estructura modular basada en los principios de la programación orientada a objetos, con el objetivo de garantizar claridad, reutilización de código y facilidad de mantenimiento.

**Desarrollo:** implementar de forma progresiva y modular el sistema de gestión para un club de voleibol, siguiendo el diseño previamente establecido. Se empleó el lenguaje Java como tecnología principal, aplicando los principios de la programación orientada a objetos

**Implementación:** La fase de implementación consistió en traducir el diseño modular del sistema a código funcional en Java, asegurando que cada clase cumpliera con sus responsabilidades específicas. La implementación fue guiada por principios de claridad, validación de datos y separación de responsabilidades.

**Pruebas:** Realizamos distintas pruebas del programa antes de entregárselo al club de voleibol, validando que todo funcione correctamente y no tenga ningún percance al momento de la entrega, verificando que sea un trabajo de calidad y funcional.

**Mantenimiento:** Es una fase posterior a la entrega, quedamos atentos y abiertos a la problemática del cliente, con el propósito de corregir, mejorar o implementar nuevas herramientas al programa.

**Diagrama UML:**

