



## Proyecto POO – Git

Vais a realizar el siguiente proyecto por parejas, repartiendo la funcionalidad.

**Objetivo:** desarrollar un sistema de gestión para un centro deportivo

Tendremos las siguientes clases:

### 1. Cancha (cancha.py)

#### Lo implementa el alumno1

Atributos:

- Número de cancha
- Deporte
- Precio
- Habilitada
- Lista de reservas
- Lista de empleados

Acciones:

- Crear una cancha
- Agregar una cancha en la lista del centro. No se podrá agregar si en la misma ya se encuentra registrada.
- Listar las canchas totales según un tipo de deporte.
- Quitar una cancha de la lista del centro. No se podrá quitar la cancha si la misma posee reservas pendientes.

### 2. Clientes (personas.py)

#### Lo implementa el alumno2

Atributos:

- Nombre
- Apellido
- Teléfono
- Identificador



- Activo

Operaciones:

- Crear un cliente
- Agregar un cliente en la lista del centro. No se podrá agregar si en la misma ya se encuentra registrado.
- Quitar un cliente en la lista del centro. No se podrá quitar si el mismo tiene reservas pendientes.
- Listar clientes morosos de una cancha o de la lista de clientes totales.

### 3. Reservas (reserva.py)

**Lo implementa el alumno1**

Atributos:

- Numero de reserva
- Fecha
- Cliente
- Cancha

Operaciones:

- Crear una reserva. Para poder crear una reserva, tanto la cancha como el cliente deben estar habilitados. El horario correspondiente y la cancha debe estar desocupada en ese horario. Al crear la reserva el cliente tendrá un movimiento negativo para liquidar a final de mes.
- Listar las reservas actuales de una cancha.
- Listar las reservas de un cliente.
- Un cliente no puede reservar si tiene un saldo negativo menor a -2000.
- Cada reserva generada el cliente tendrá un nuevo saldo negativo el cual será el costo de esta.
- El centro deberá registrar el pago del cliente aportando un movimiento positivo al mismo.
- Mostrar saldo de un cliente

### 4. Empleados (personas.py)

**Lo implementa el alumno2**

Atributos:

- Nombre
- Apellido



- Desocupado (Si/no)
- Lista de tareas

Operaciones:

- Registrar un empleado a una cancha. Para esto la cancha debe estar habilitada y el empleado no debe estar registrado en ninguna otra cancha.
- Asignar tarea a un empleado. El empleado deberá poder tener una lista de tareas, las mismas se pueden asignar y quitar. El empleado que no posee ninguna tarea estará como desocupado.
- Listar los empleados que se encuentran desocupados.
- Quitar un empleado de la cancha

## 5. Centro (centro\_x.py)

**Lo implementan ambos alumnos.**

**centro\_1.py y centro\_2.py**

Atributos:

- Nombre
- Dirección
- Lista de canchas
- Lista de clientes

Operaciones:

- Implementa libremente las operaciones del centro para hacer uso de las operaciones del resto de clases.

## Acciones en Git – GitHub

1. Los alumnos crearán un repositorio público común en el GitHub de uno de los dos.
2. Realizarán en **su working directory** los módulos que le correspondan e irán confirmando los cambios en el repositorio según consideren.
3. Antes de comenzar con el desarrollo del módulo **centro\_x.py** o en un momento anterior se debe hacer **pull** del repositorio para recuperar todos los cambios del compañero o compañera y poder utilizarlas.



4. Los integrantes del equipo deben acordar cuales van a ser las funcionalidades a desarrollar en el módulo centro\_x.py. Se acordará también dejar una de las funciones pendientes de implementar para que la implemente el compañero o compañera.
5. Al terminar la implementación del módulo centro\_x.py este se debe hacer **pull** y **push** del mismo.
6. Cada miembro del equipo implementará en el módulo del compañero/a la función que acordaron dejar pendiente. Se hará **push** y **pull**
7. Cada miembro del equipo probará la funcionalidad desarrollada por el otro.
8. Se deberá ir acordando cómo solucionar los fallos encontrados y cómo mandarlos al repositorio.

Recordar poner los comentarios suficientes en las acciones realizadas contra el repositorio

#### ENDES:

- Se debe presentar el enlace al repositorio público