**Sistema de reserva de canchas de paddle.**

Tecnologías: React, Python con FastAPI, SQLAlchemy y base de datos Postgres. (Tecnologías obligatorias).

Se puede usar cualquier librería de React o Python que necesiten. Bootstrap se dió en clases pero es optativa, si alguien maneja o quiere usar otra librería no hay problema.  
  
**Características:**  
  
**- Cancha:** tiene un nombre. por ejemplo: "cancha 1", si está techada o no.  
**- Reserva**: Se relaciona con una cancha. Tiene día, hora, y duración. También tiene un teléfono y nombre del contacto.  
  
**Requerimientos funcionales:**

**- Reservas:**  
1- Creación de reservas

2- modificación de reserva.

3- Eliminación de reserva.  
Nota: Al crear o modificar una reserva no debe coincidir con una reserva existente. Por ejemplo: Si hay una reserva para el lunes a las 16hs de 2 horas de duración no debe ser posible crear una reserva para ese lunes a las 17hs en la misma cancha.  
4- Consultar las reservas por día y por cancha. Por ejemplo: Listar las reservas de "cancha 1" del "lunes X de diciembre".

Requerimientos no funcionales:

1- Canchas: Las canchas deben estar en la base de datos y ser leídas a través de la API.

Criterios estéticos:  
- No hay criterios estéticos estrictos, no voy a reprobar a nadie por la estética aunque los sistemas en blanco y negro, sin CSS van a tener menos nota que los sistemas con algún estilo.  
  
Notas: - Lo que no está definido lo pueden hacer bajo el criterio de cada uno.

Por ejemplo:

- pueden hacer ABMs de canchas o directamente entrar los datos en la tabla.  
- La duración mínima o máxima de la reserva, queda a criterio de cada alumno también.--

- El diseño del menú y pantallas queda a criterio del alumno. Igual que la página de consulta.  
  
Criterio de evaluación:  
1. El sistema tiene que tener implementadas los requerimientos funcionales y no funcionales.  
2. Manejo de errores. Cualquier error debe ser debidamente notificado al usuario.  
3. La API va a ser probada por fuera del front-end. Cada endpoint debe notificar si funcionó o hubo un error.  
4. Validación de datos. Por ejemplo: no se puede crear una reserva sin elegir una cancha o sin especificar un horario o sin datos de contacto.

Ya saben cualquier consulta por email o discord.

Saludos