

Challenge: Full Stack Game

Juego de Super Héroes



Descripción:

Este juego consiste en tener un mazo de tarjetas de superhéroes de Marvel, vas a encontrar 108 personajes con diferentes estilos y formas, algunas de estas tarjetas son personajes individuales y otras son tarjetas con un grupo de personajes. Las tarjetas **no tienen características y/o poderes**, para esto debes crear un sistema que permita almacenar estas tarjetas y asignarles una descripción, poderes (XP) y características.

Luego de tener todas las tarjetas sistematizadas, con sus características y/o poderes, se debe diseñar un juego de máximo 6 jugadores y mínimo 2 jugadores. El juego consiste en repartir aleatoriamente 5 tarjetas a los 6 jugadores o a cada jugador creado previamente. El juego consiste en que cada jugador apuesta sus tarjetas y gana el juego el que tenga todas las tarjetas. Cada jugador apuesta basado en su barajade tarjetas y gana la partida la tarjeta que tenga mayor XP.

Instrucciones:

- El sistema es fácil, simplemente se reparte las tarjetas a cada jugador de forma aleatoria y se tiene diferentes rondas, cada ronda los jugadores colocan una tarjeta de forma tal que los demás jugadores no sepan cuál es.
- Luego que todos los jugadores pongan su tarjeta apostada, la tarjeta que sume más puntos gana, significa que el jugador que tenga la mejor tarjeta se lleva las demás tarjetas que están en el tablero (es como un sistema de apuestas convencional).
- El jugador tiene 20 segundos para poner una tarjeta en el tablero, de lo contrario el sistema de forma automática selecciona una tarjeta del mazo de tarjetas del jugador que tiene el turno.
- Después de que el último jugador apueste la tarjeta, se inicia el proceso de validación y asignación de ganador de ronda, quitando y poniendo tarjetas en los mazos de los jugadores.
- El juego finaliza cuando los jugadores quedan sin tarjetas y solo uno de ellos queda vivo en el juego.
- El jugador que adquiere las tarjetas de los demás jugadores puede usar esas mismas tarjetas durante el juego (es decir la puede apostar).
- El juego se reinicia en cualquier momento o finalizando el juego, encontrando el ganador, si un jugador decide retirarse, las tarjetas entran al mazo principal.
- Los puntos se acumulan por jugador, teniendo un puntaje gracias a una cuenta que tenga el usuario al inicio de sesión.

Funcionalidades

- Inicio de sesión con firebase y google (o con cualquier sistema de autenticación)
- Administrador de tarjetas de personajes: CRUD (solo el administrador del juego)
- Inicio de juego: creación de jugadores y repartición de tarjetas de forma aleatoria
- Tablero para poner tarjetas de apuesta (tapa cerrada)
- Validación de apuestas de las tarjetas del tablero (tapa abierta)
- Sistema de rondas: juego del tablero, apuesta de tarjeta y toma de decisiones lógicas
- Puntaje y acumulado: historico de puntos
- Vista del perfil

Características técnicas

- Se debe trabajar con Angular y Spring Boot
- El diseño de arquitectura debe ser orientada a capas con arquitectura limpia
- Se debe desarrollar pruebas unitarias a las funcionalidades más críticas en el backend
- Se debe desplegar usando Netlify y firebase (o alguna plataforma de servicios cloud)

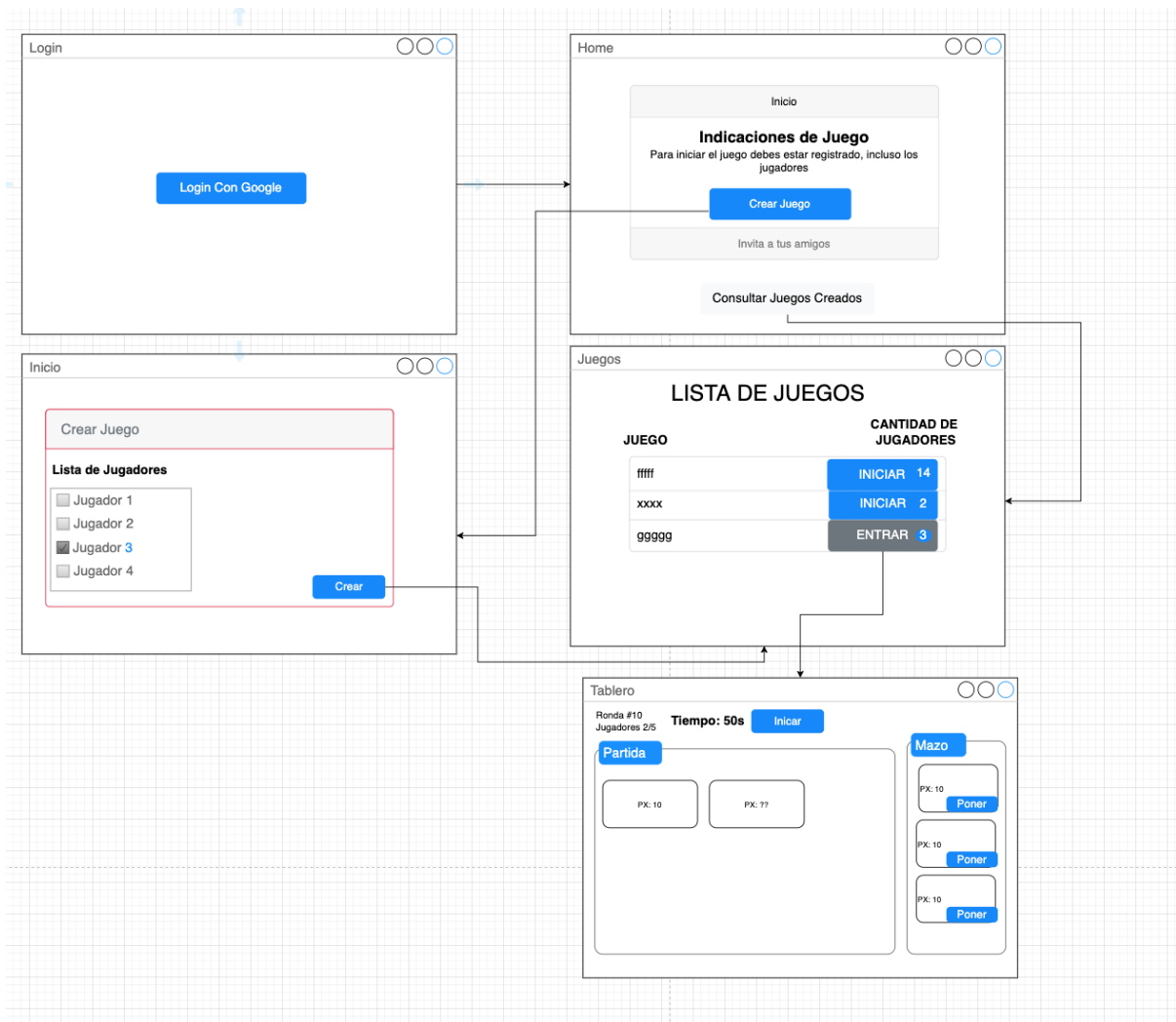
- Se debe tener un sistema de control de versiones en GitHub (mínimos 50 commits)
- Se debe implementar Rabbit como sistema de mensajería.
- Se debe implementar Docker.

Técnica y funcional

- Se debe tener un tablero de apuestas y se debe tener un tiempo para poner la tarjeta.
- Se debe tener un sistema drop/down donde se ponga las tarjetas en el tablero de una forma amigable.
- El manejo de turno y validaciones relacionados al ganador debe ser asignación del backend.
- Se debe diseñar un modelamiento del mismo y crear las API de tal manera este bien contextualizada a nivel funcional
- Se debe trabajar con buenas prácticas de programación y el código debe ser auto descriptivo.
- Si existe un empate el sistema debe permitir a los jugadores disponibles apostar una nueva carga para así sumar puntos y desempatar la partida.

Recursos

- [Pack 108 Pepsicards marvel.zip](#)
- Ejemplo del tablero: Mockup



El proyecto se debe entregar en mínimo 4 - 6 días desde la fecha, se debe sustentar y entregar un resumen de su funcionalidad, la sustentación estará programada para máximo 10 a 15 minutos por medio de una reunión con el coach.

Importante: Tener clara las características y la funcionalidades, dado que con base a esas se realizaría la evaluación que equivale el 75% de la nota del curso. Muchas suerte.