

UNIVERSIDAD MAYOR REAL Y PONTIFICIA DE SAN FRANCISCO DE CHUQUISACA

FACULTAD DE TECNOLOGÍA



Juego Dilemo

NOMBRE: María Luciana Salinas Vilar

NOMBRE: Isaac Joseph Vargas Oropeza

CARRERA: Ing. En ciencias de la computación

MATERIA: Desarrollo de aplicaciones móviles

CU: 111-561 / 111-562

Sucre, 1 de Diciembre de 2025

JUEGO DILEMO

1. DESCRIPCIÓN GENERAL DEL PROYECTO

Dilemo es una aplicación de trivia educativa desarrollada como sistema multiplataforma que permite a los usuarios responder preguntas sobre tecnologías de programación. El proyecto implementa una arquitectura cliente-servidor con capacidad de ejecución tanto en navegadores web como en dispositivos móviles Android.

2. ARQUITECTURA TECNOLÓGICA

2.1 Stack Tecnológico

Frontend:

- React: Framework principal para la interfaz de usuario
- Node.js: Entorno de ejecución JavaScript

Backend:

- Node.js: Servidor de aplicaciones
- Express: Framework para manejo de rutas y peticiones HTTP

Base de Datos:

- MySQL: Sistema de gestión de base de datos relacional
- phpMyAdmin: Interfaz de administración de base de datos

Herramienta de Conversión Multiplataforma:

- Capacitor: Framework utilizado para empaquetar la aplicación web y convertirla en aplicación nativa para Android

3. ESTRUCTURA DE MINIJUEGOS

El sistema contiene tres modalidades de juego principales:

3.1 Python Quiz

Preguntas básicas sobre el lenguaje de programación Python con sistema de respuestas verdadero/falso, organizado en tres niveles de dificultad progresivos.

3.2 Git Quiz

Preguntas generales sobre el sistema de control de versiones Git, también estructurado en tres niveles de dificultad creciente con formato verdadero/falso.

3.3 Logo Quiz

Modalidad de reconocimiento visual donde se presenta un logotipo y el usuario debe identificar correctamente si corresponde a la tecnología indicada mediante respuestas verdadero/falso.

4. SISTEMA DE GAMIFICACIÓN

La aplicación incorpora elementos de engagement del usuario:

- Sistema de recompensas por participación
- Registro de rachas de juego consecutivas
- Ranking individual por minijuego
- Ranking global unificado
- Panel de ajustes personalizables de cuenta

5. CONFIGURACIÓN DE BASE DE DATOS

5.1 Nombre de Base de Datos

dilemo_db

5.2 Gestor de Base de Datos

XAMPP proporciona el servidor MySQL y phpMyAdmin para administración.

5.3 Archivo de Configuración

El archivo .env en el directorio raíz del proyecto contiene las credenciales y parámetros de conexión a la base de datos MySQL.

6. PROCEDIMIENTOS DE EJECUCIÓN

6.1 Requisitos Previos

- Node.js instalado en el sistema
- XAMPP instalado y configurado
- Android Studio instalado para desarrollo móvil
- Dispositivo móvil o emulador Android configurado

6.2 Inicialización de la Base de Datos

Paso 1: Iniciar XAMPP Abrir el panel de control de XAMPP y activar los servicios Apache y MySQL.

Paso 2: Verificar acceso a phpMyAdmin Acceder mediante navegador a <http://localhost/phpmyadmin> para confirmar que la base de datos dilemo_db está disponible.

6.3 Ejecución del Backend

Abrir terminal y ejecutar los siguientes comandos en secuencia:

```
cd backend  
npm start
```

Este proceso inicia el servidor Node.js que gestiona las peticiones de la aplicación y la comunicación con la base de datos MySQL.

6.4 Ejecución del Frontend para Desarrollo Web

En una terminal separada, desde el directorio raíz del proyecto:

```
npm run dev -- --host
```

El parámetro --host permite que el servidor de desarrollo sea accesible desde otros dispositivos en la misma red local, no solo desde localhost.

6.5 Acceso desde Navegador Web

Una vez ejecutados ambos servidores, acceder a la aplicación mediante navegador utilizando la dirección IP local proporcionada en la consola.

6.6 Simulación de Vista Móvil en Navegador

Para trabajar con dimensiones de dispositivos móviles durante el desarrollo:

Paso 1: Abrir las herramientas de desarrollador presionando Ctrl+Shift+I

Paso 2: Activar el modo de vista de dispositivo mediante el icono correspondiente o presionando Ctrl+Shift+M

Paso 3: Seleccionar el modelo de dispositivo deseado o establecer dimensiones personalizadas

Esta metodología permite desarrollar y ajustar la interfaz con las proporciones exactas de dispositivos móviles sin necesidad de compilar constantemente la aplicación Android.

7. PROCESO DE MIGRACIÓN A ANDROID

7.1 Compilación del Proyecto

Desde el directorio raíz del proyecto, ejecutar:

```
npm run build
```

Este comando genera los archivos optimizados para producción en el directorio de distribución.

7.2 Sincronización con Capacitor

Inmediatamente después de la compilación, ejecutar:

```
npx cap sync android
```

Este comando realiza tres operaciones fundamentales:

1. Copia los archivos web compilados al proyecto Android
2. Actualiza las dependencias nativas de Capacitor
3. Sincroniza los plugins instalados con el proyecto Android

7.3 Apertura en Android Studio

Después de la sincronización exitosa, abrir Android Studio y cargar el proyecto ubicado en el directorio android generado por Capacitor.

7.4 Configuración de Conexión de Red

Aspecto Crítico: Para que la aplicación Android pueda comunicarse con el backend Node.js, ambos sistemas deben estar conectados a la misma red local.

Configuración Requerida:

1. El dispositivo móvil o emulador debe estar en la misma red WiFi que el equipo donde se ejecuta el backend
2. El archivo de configuración del frontend debe apuntar a la dirección IP local del servidor backend, no a localhost
3. Verificar que el firewall del sistema operativo permita conexiones entrantes en el puerto del backend

7.5 Ejecución en Dispositivo Móvil

Desde Android Studio:

Opción 1 - Dispositivo Físico:

1. Conectar dispositivo Android mediante cable USB
2. Habilitar modo de depuración USB en el dispositivo
3. Seleccionar el dispositivo en la lista de dispositivos disponibles
4. Presionar el botón Run o Shift+F10

Opción 2 - Emulador:

1. Iniciar el emulador Android desde AVD Manager
2. Esperar que el sistema operativo arranque completamente
3. Seleccionar el emulador en la lista de dispositivos disponibles
4. Presionar el botón Run o Shift+F10

8. FLUJO DE TRABAJO COMPLETO

8.1 Para Desarrollo Web Local

1. Iniciar XAMPP y verificar servicios MySQL y Apache
2. Abrir terminal y ejecutar: `cd backend && npm start`
3. Abrir segunda terminal y ejecutar: `npm run dev -- --host`
4. Acceder mediante navegador a la URL proporcionada
5. Utilizar Ctrl+Shift+I para activar vista de dispositivo móvil
6. Desarrollar y probar cambios en tiempo real

8.2 Para Desarrollo y Testing en Android

1. Completar todos los pasos de desarrollo web local
2. Realizar cambios necesarios en el código
3. Ejecutar `npm run build` para compilar
4. Ejecutar `npx cap sync android` para sincronizar
5. Abrir Android Studio

6. Verificar que dispositivo y servidor estén en la misma red
7. Compilar y ejecutar desde Android Studio
8. Probar funcionalidad en dispositivo móvil

9. CONSIDERACIONES TÉCNICAS IMPORTANTES

9.1 Gestión de Direcciones IP

En entorno de desarrollo local, el backend responde en localhost o 127.0.0.1. Sin embargo, cuando la aplicación Android intenta conectarse, localhost hace referencia al propio dispositivo móvil, no al equipo donde se ejecuta el servidor.

Solución: Configurar la URL del backend utilizando la dirección IP local del equipo servidor, por ejemplo 192.168.1.100, de manera que el dispositivo móvil pueda localizar correctamente el backend en la red local.

9.2 Persistencia de Base de Datos

La base de datos MySQL gestionada por XAMPP debe permanecer activa durante todo el tiempo que se utilice la aplicación, tanto en versión web como móvil. Cualquier interrupción del servicio MySQL resultará en pérdida de conectividad.

9.3 Sincronización de Código

Cada vez que se realicen modificaciones significativas en el código del frontend, es necesario repetir el proceso de compilación y sincronización (`npm run build` seguido de `npx cap sync android`) para que los cambios se reflejen en la aplicación Android.