

#Synergy-Match-App Monorepo

¡Bienvenido al monorepo de synergy-match-app! Este repositorio contiene la aplicación web (Next.js) y la aplicación móvil (React Native/Expo) para el proyecto Synergy Match.

Este README.md te guiará a través de la configuración del entorno de desarrollo y el flujo de trabajo de Git para contribuir al proyecto.

Inicio Rápido

Sigue estos pasos para poner en marcha tu entorno de desarrollo local.

1.- Requisitos Previos

Asegúrate de tener instalado lo siguiente en tu sistema:

1.1 Git: Descargar Git

1.2 Node.js (versión 18.x LTS): Recomendamos encarecidamente usar nvm (Node Version Manager) para gestionar tus versiones de Node.js.

1.3 npm install -g pnpm (IMPORTANTE)

pnpm -v # Debería mostrar v8.x.x o superior

1.4 Expo Go App (para probar la aplicación móvil en tu teléfono): Android / iOS

1.5 Android Studio (Opcional, para emuladores Android):

Descargar Android Studio

Configura al menos un AVD (Android Virtual Device) para emular un dispositivo Android.

1.6 Descargar Visual Code Node.js

2.- Clonar el Repositorio

Abre tu terminal de Git Bash y clona el repositorio:

2.1 Comando: `git clone https://github.com/snrgapp/desarrollomobile.git`

2.2 Comando: `cd synergy-match-app`

3.- Instalar Dependencias

Una vez dentro de la carpeta raíz del monorepo, instala todas las dependencias:

3.1 Comando: pnpm install

Este comando instalará las dependencias para todos los workspaces (web, mobile, db). Si hay algún mensaje sobre "Ignored build scripts" (por ejemplo, para sharp o unrs- resolver), ejecútalo y confírmalo:

3.1.1 Comando: pnpm approve-builds

Responde 'Y' cuando te lo pida

4.- Iniciar las Aplicaciones desde Visual Code

4.1 Abre una nueva Ventana y abre la carpeta descargada con el proyecto

Puedes iniciar las aplicaciones web y móvil en paralelo.

4.2 Iniciar la Aplicación Web (Next.js)

4.2.1 En una nueva terminal, desde la raíz del monorepo:

Comando: pnpm run dev:web

La aplicación web estará disponible en <http://localhost:3000>.

4.3 Iniciar la Aplicación Móvil (React Native/Expo)

4.3.1 En otra nueva terminal, navega a la carpeta de la aplicación móvil:

Comando: cd apps/mobile

Comando pnpm start

Esto iniciará el Metro Bundler de Expo. Podrás:

Escanear el código QR con la aplicación Expo Go en tu teléfono.

Presionar a en la terminal para abrir la aplicación en un emulador de Android.

Acceder a la versión web de la aplicación móvil en <http://localhost:8081> (útil para depuración en el navegador).



Flujo de Trabajo con Git (IMPORTANTE)

1.- Para mantener el código organizado y colaborativo, seguiremos este flujo de trabajo:

1.1 main: Rama principal de producción. Contiene el código estable listo para el despliegue. Nunca se trabaja directamente en main. Los merge a main solo se hacen desde dev después de una revisión exhaustiva y pruebas finales.

1.2 dev: Rama de desarrollo principal. Es la rama donde se integran todos los cambios de las diferentes funcionalidades antes de ser promovidos a main.

2.- Pasos para un Flujo de Trabajo de Desarrollo

Cada vez que vayas a trabajar en una nueva tarea o funcionalidad:

2.1 Asegúrate de que tu rama dev local esté actualizada:

2.1.1 Abre tu terminal en la raíz del monorepo y ejecuta:

Comando: `git checkout dev` / Nos posicionamos en la rama dev

Comando: `git pull origin dev` / Nos trae de vuelta lo actualizado en la rama dev

2.1.2 Crea una nueva rama de trabajo (feature branch): Nombra tu rama de forma descriptiva, por ejemplo: `feature/nombre-de-la-funcionalidad` o `fix/descripcion-del-bug`.

Comando: `git checkout -b feature/mi-nueva-funcionalidad`

(Consejo: Los nombres de las ramas de características suelen empezar con `feature/`, `fix/`, `chore/`, `refactor/`.)

2.2 Instala o actualiza las dependencias (si es necesario):

Siempre que hayas actualizado tu rama dev o si detectas cambios en el `package.json` de alguno de los workspaces, es una buena práctica ejecutar:

Comando: `pnpm install` Esto asegurará que todas las dependencias estén sincronizadas con la última versión de la rama dev.

3.- Puedes realiza tus cambios y trabaja en tu funcionalidad: Edita el código, añade nuevas características, corrige errores, etc.

3.1 Inicia tu aplicación web (Next.js):

3.1.1 Abre una NUEVA terminal a la raíz de tu monorepo:

Ejecuta el comando para iniciar el servidor de desarrollo de la web:

Comando: `pnpm run dev:web`

¡ESTO ES LO IMPORTANTE! Observa cuidadosamente la salida. Si no hay `SyntaxError`, busca mensajes como `ready - started server on http://localhost:3000` o

similares. Una vez que lo veas, abre tu navegador web y ve a `http://localhost:3000`.

3.2 Inicia tu aplicación móvil (React Native con Expo) - ¡en una terminal SEPARADA!:

3.2.1 Abre otra NUEVA terminal Navega a la carpeta de tu aplicación móvil: ejemplo
`C:/Users/Desktop/synergy-match-app/apps/mobile`

Inicia el servidor de desarrollo de Expo:

Comando: `pnpm start`

Esto abrirá una página en tu navegador (el "Metro Bundler", usualmente en `http://localhost:8081`) y también verás opciones en la terminal.

3.2.1.1 Cargar en el Emulador de Android (si ya lo configuraste y está corriendo):

Asegúrate de que tu emulador de Android esté abierto y visible en tu pantalla.

En la terminal donde ejecutaste `pnpm start`, presiona la tecla `a`. Expo debería detectarlo y empezar a cargar la aplicación.

3.2.1.2 Cargar con la aplicación Expo Go en tu teléfono físico (si no tienes emulador):

Asegúrate de que tu teléfono y tu computadora estén conectados a la misma red Wi-Fi.

Abre la aplicación Expo Go en tu teléfono.

En la página del "Metro Bundler" en tu navegador, verás un código QR.

Dentro de la aplicación Expo Go en tu teléfono, usa la opción para escanear el código QR. La aplicación debería empezar a cargarse.

¡Este es el momento decisivo! Por favor, intenta ejecutar ambas aplicaciones y cuéntame si se inician y funcionan sin problemas. ¡Crucemos los dedos!

3.3 Confirma tus cambios (commits): Haz commits pequeños y descriptivos para validar que funciona tu configuración con el proyecto en el github

3.3.1 git add . # Añade todos los cambios, o especifica los archivos: git add apps/web/src/pages/index.js

3.3.2 git commit -m "feat: Añadida nueva página de perfil de usuario" # O "fix: Corregido error en login"

(Consejo: Usa los prefijos feat:, fix:, chore:, docs:, style:, refactor:, test:, revert: para tus mensajes de commit.)

4.- Sube tus cambios a tu rama remota:

Comando: git push origin feature/mi-nueva-funcionalidad

IMPORTANTE REVISAR Antes de Actualizar la rama dev local (antes de un Pull Request):

Una vez que hayas terminado tu trabajo en la rama feature y antes de preparar un Pull Request, es fundamental que tu rama de trabajo esté al día con la última dev. Esto minimiza conflictos.

Comando: git pull origin dev

Si hay conflictos, resuélvelos en este momento.

Crea un Pull Request (PR) a la rama dev:

Ve a GitHub y crea un Pull Request desde tu rama (feature/mi-nueva-funcionalidad) hacia la rama dev.

Asegúrate de que la rama base sea dev.

Describe claramente los cambios realizados.

Asigna revisores (code owners).

Revisión y Fusión (Merge):

Una vez que el PR sea aprobado por los revisores y todas las pruebas automáticas pasen, se fusionará (merge) a la rama dev.

5.- Sincronización de dev a main (Autoriza Synergy a Eric Chourio a ejecutar este paso)

El proceso de fusión de dev a main será gestionado por un líder técnico o por el equipo de DevOps.

Solo cuando la rama dev haya alcanzado un estado estable y se considere una versión candidata para producción, se creará un Pull Request desde dev a main.

Esto normalmente irá acompañado de un proceso de pruebas de integración y QA final.

EXITOS AL DESARROLLO DE SYNERGY-MATCH-APP A por el exito.....!!!!!!