**🧩 1. Crear el proyecto base**

Ejecuta en tu terminal:

npx create-expo-app miproyectorn --template

Cuando te pregunte por la plantilla, selecciona **“Blank (TypeScript)”**  
(o ejecuta directamente el siguiente comando para evitar la selección manual):

npx create-expo-app test1 --template expo-template-blank-typescript

Esto creará la estructura básica del proyecto.

Instalar las dependencias compatibles  
npm install expo@~54.0.13 expo-status-bar@~3.0.8 react@19.1.0 react-dom@^19.1.0 react-native@0.81.4 [react-native-web@^0.21.2](mailto:react-native-web@%5e0.21.2)

**🧰 3. Instalar las dependencias de desarrollo**

npm install -D @testing-library/jest-native@^5.4.3 @testing-library/react-native@^12.9.0 @types/jest@^30.0.0 @types/mocha@^10.0.10 @types/react@~19.1.0 babel-preset-expo@^54.0.4 jest@^29.7.0 jest-expo@^54.0.12 [react-test-renderer@^19.1.0](mailto:react-test-renderer@%5e19.1.0)

4. modificar package.json

{

"name": "miproyectorn",

"version": "1.0.0",

"main": "index.ts",

"scripts": {

"start": "expo start",

"android": "expo start --android",

"ios": "expo start --ios",

"web": "expo start --web",

"test": "jest" //añadir

},

**🧠 5. Configurar Jest**

Crea un archivo jest.config.js en la raíz del proyecto con este contenido:

module.exports = {

preset: 'jest-expo',

transformIgnorePatterns: [

'node\_modules/(?!((jest-)?react-native|@react-native(-community)?|expo(-.\*)?))',

],

setupFilesAfterEnv: ['@testing-library/jest-native/extend-expect'],

};

Crear carpeta \_\_test\_\_

Crear archivo test App.test.tsx

import React from 'react';

import { render } from '@testing-library/react-native';

import App from '../App'; // ajusta la ruta si el test está dentro de src/

describe('App Component', () => {

it('renders a welcome message', () => {

const { getByText } = render(<App />);

expect(getByText('Open up App.tsx to start working on your app!')).toBeTruthy();

});

});

Ejecutar test

Npm test

Texto

El contenido generado por IA puede ser incorrecto.