# Guía paso a paso — Proyecto Expo + React Native + TypeScript con Testing

## 1) Crear el proyecto Expo con TypeScript

Ejecuta los siguientes comandos en tu terminal:  
npx create-expo-app@latest MiProyectoRN2 --template expo-template-blank-typescript  
cd MiProyectoRN2

## 2) Instalar dependencias de testing

Instala Jest, Testing Library y React Test Renderer con versiones compatibles:  
npm install --save-dev jest@29.7.0 jest-expo@54.0.12 @testing-library/react-native@12.4.3 @testing-library/jest-native@5.4.3 react-test-renderer@19.1.0 --legacy-peer-deps

## 3) Configurar archivos del proyecto

### babel.config.js

module.exports = {  
 presets: ['babel-preset-expo'],  
};

### jest.config.js

module.exports = {  
 preset: 'jest-expo',  
 testEnvironment: 'jsdom',  
 setupFilesAfterEnv: ['@testing-library/jest-native/extend-expect'],  
 transformIgnorePatterns: [  
 'node\_modules/(?!((jest-)?react-native|@react-native|react-clone-referenced-element|@expo|expo(nent)?|@expo(nent)?/.\*|expo-modules-core|@react-navigation/.\*))',  
 ],  
 modulePathIgnorePatterns: ['<rootDir>/.expo/'],  
};

### tsconfig.json

{  
 "compilerOptions": {  
 "target": "esnext",  
 "module": "esnext",  
 "jsx": "react-native",  
 "strict": true,  
 "esModuleInterop": true,  
 "skipLibCheck": true,  
 "forceConsistentCasingInFileNames": true,  
 "moduleResolution": "node",  
 "resolveJsonModule": true,  
 "types": ["jest"]  
 },  
 "exclude": ["node\_modules"]  
}

## 4) Crear componentes y tests de ejemplo

Archivo App.tsx:

import React from 'react';  
import { Text, View } from 'react-native';  
  
export default function App() {  
 return (  
 <View style={{flex:1, alignItems:'center', justifyContent:'center'}}>  
 <Text>Open up App.tsx to start working on your app!</Text>  
 </View>  
 );  
}

Archivo App.test.tsx:

import React from 'react';  
import { render } from '@testing-library/react-native';  
import App from './App';  
  
describe('App', () => {  
 it('renders correctly', () => {  
 const { getByText } = render(<App />);  
 expect(getByText('Open up App.tsx to start working on your app!')).toBeTruthy();  
 });  
});

## 5) Limpieza y reinstalación

Si surgen errores de dependencias o caché, ejecuta:  
  
Windows (PowerShell):  
rd /s /q node\_modules  
del package-lock.json  
npx jest --clearCache  
npm install --legacy-peer-deps  
  
macOS / Linux:  
rm -rf node\_modules package-lock.json  
npx jest --clearCache  
npm install --legacy-peer-deps

## 6) Ejecutar la app y los tests

Para iniciar Expo:  
npm start  
  
Para ejecutar tests:  
npm test

## 7) Problemas comunes y soluciones

- ERESOLVE: usar --legacy-peer-deps  
- SyntaxError: ajustar jest.config.js y babel.config.js  
- Archivos .ts no transformados: incluirlos en transformIgnorePatterns  
- Error de caché: ejecutar npx jest --clearCache