# Capítulo 03: Nuestro Primer Proyecto

## 1. Organización de carpetas

Para mantener un orden claro, creamos una carpeta base donde estarán todos los proyectos del curso.  
  
Ejemplo:  
00\_Proyectos/  
 ├─ proyecto01/  
 ├─ proyecto02/  
 ├─ proyecto03/  
  
👉 Cada nuevo trabajo práctico será un proyecto independiente dentro de 00\_Proyectos.

## 2. Crear el primer proyecto

1. Abrí la terminal y parate dentro de la carpeta 00\_Proyectos:  
 cd E:\00\_ENGITHUB\Modulo02\_JavaScript\Capitulo2000\_NextJS\00\_Proyectos  
  
2. Ejecutá el comando para crear un nuevo proyecto:  
 pnpm create next-app proyecto01  
  
3. El asistente de Next.js hará una serie de preguntas.

## 3. Preguntas del asistente (respuestas recomendadas)

|  |  |  |
| --- | --- | --- |
| Pregunta | Respuesta recomendada | Explicación |
| Would you like to use TypeScript? | No | Útil, pero al inicio suma complejidad. Lo veremos más adelante. |
| Which linter would you like to use? | ESLint | Ayuda a escribir código más limpio y detectar errores. |
| Would you like to use Tailwind CSS? | Yes | Facilita los estilos con clases utilitarias. |
| Would you like your code inside a `src/` directory? | No | Mejor mantener la estructura más simple (app/). |
| Would you like to use App Router? (recommended) | Yes | Es el sistema moderno de rutas en Next.js. |
| Would you like to use Turbopack? (recommended) | No | Muy rápido, pero aún en desarrollo. Mejor dejarlo para más adelante. |
| Would you like to customize the import alias (`@/\*` by default)? | No | El alias por defecto @/\* funciona bien. |

## 4. Instalación de dependencias

Next.js descargará automáticamente las dependencias necesarias:  
- next (framework)  
- react y react-dom (librería base)  
- tailwindcss (estilos)  
- eslint (linter de código)  
  
⚠️ Nota: Puede aparecer un aviso como este (es normal):  
Ignored build scripts: @tailwindcss/oxide, sharp, unrs-resolver.  
Run "pnpm approve-builds" to pick which dependencies should be allowed to run scripts.  
  
Podés ignorarlo o ejecutar 'pnpm approve-builds' si querés aprobar esas dependencias manualmente.

## 5. Ejecutar el servidor de desarrollo

1. Entrá en la carpeta del proyecto:  
 cd proyecto01  
  
2. Levantá el servidor:  
 pnpm dev  
  
3. Abrí el navegador en:  
- Local: http://localhost:3000  
- Red local (ejemplo en tu WiFi): http://192.168.x.x:3000

## 6. Abrir el proyecto en Visual Studio Code

Es importante abrir siempre el proyecto completo en VS Code, no archivos individuales.  
  
1. Abrí Visual Studio Code.  
2. Menú File → Open Folder...  
3. Seleccioná la carpeta del proyecto (proyecto01).  
  
👉 Así tendrás acceso a todos los archivos, configuraciones y carpetas desde un único entorno de trabajo.

## 7. Primer vistazo al proyecto

Al abrir el navegador, verás la página de inicio que Next.js genera automáticamente.  
- Esa página está definida en app/page.js.  
- Podés modificar el contenido para mostrar un Hello World y ver cómo el cambio se refleja al instante gracias al Hot Reload.

## 8. Resumen

- Organizamos la carpeta base 00\_Proyectos.  
- Creamos el proyecto01 con Next.js.  
- Contestamos las preguntas del asistente con la configuración recomendada.  
- Vimos cómo Next.js instala dependencias automáticamente.  
- Levantamos el servidor y accedimos a la primera página en http://localhost:3000.  
- Abrimos el proyecto en Visual Studio Code, listo para empezar a programar.  
  
Con esto, tu primer proyecto Next.js ya está funcionando y preparado para seguir avanzando 🚀.