# **Resumen Conceptual sobre Reforzamiento de JavaScript**

 Configuración de la Terminal

Es recomendable no utilizar caracteres especiales ni espacios en los nombres de carpetas y proyectos. En lugar de espacios, se deben usar guiones para evitar problemas con herramientas como Live Server.  
➡️ Ejemplo: En lugar de 'Proyecto Demo', usar 'proyecto-demo'.

 Creación de Proyectos con Create React App

Usando el comando `npx create-react-app`, se puede generar rápidamente una nueva aplicación React. Este comando descarga e instala las dependencias necesarias para trabajar con React y facilita la creación de proyectos desde cero.  
➡️ Ejemplo: Ejecutar `npx create-react-app intro-javascript` para crear un nuevo proyecto de introducción a JavaScript.

 Estructura del Proyecto y Archivos Innecesarios

Tras la creación de un proyecto, es común eliminar archivos innecesarios en la carpeta 'src' para mantener el proyecto limpio. Por ejemplo, se puede eliminar todo el contenido del directorio 'src' y crear un archivo básico como `index.js` con un `console.log('Hola Mundo');`.  
➡️ Ejemplo: Eliminar todos los archivos dentro de 'src' y crear un `index.js` que imprima 'Hola Mundo' en consola.

 Ejecución del Proyecto

Una vez configurado el proyecto, se puede ejecutar con `npm start` o `yarn start`, dependiendo del gestor de paquetes que se esté utilizando. Esto iniciará un servidor local y mostrará la aplicación en el navegador.  
➡️ Ejemplo: Ejecutar `npm start` para iniciar el proyecto y ver los resultados en el navegador.

 Manejo de Versiones de React

Con las versiones recientes de React, es importante tener en cuenta que `ReactDOM.render` ha sido reemplazado por `ReactDOM.createRoot` en React 18. Si no se realiza este cambio, la aplicación puede seguir utilizando la versión anterior de React (v17).  
➡️ Ejemplo: Para evitar problemas, actualizar el código a `ReactDOM.createRoot` cuando se utilice React 18 o superior.