FrontEnd Test:  
Descripción del Ejercicio

1. Vista Principal de Lista de Productos (PLP):
   * Crea una vista similar a la página de Adidas mencionada, enfocándose en listar productos.
   * Utiliza Angular para desarrollar esta vista.
   * Los productos deben cargarse desde un archivo JSON local que simule una llamada a API.
2. Navegación a la Página de Detalle del Producto (PDP):
   * Al hacer clic en un producto en la lista, el usuario debe ser redirigido a una página de detalle de producto (PDP).
   * Esta página debe mostrar información más detallada del producto seleccionado.

### Requisitos Técnicos

* Angular: Utiliza versiones recientes de Angular para aprovechar las características más actuales del framework.
* Routing: Implementa Angular Router para gestionar la navegación entre la vista de lista de productos y la vista de detalle de productos.
* Servicios de Angular: Utiliza un servicio para manejar la carga y el manejo de datos de productos.
* Componentes: Desarrolla componentes reutilizables, uno para la lista de productos y otro para los detalles del producto.
* Responsive Design: Asegúrate de que la vista sea responsive para adaptarse a diferentes tamaños de pantalla.
  + Media Queries de Tailwind: Asegúrate de que cada componente de tu interfaz sea responsive, utilizando las media queries que proporciona Tailwind. Esto garantiza que tu aplicación se vea bien en dispositivos móviles, tablets y desktops.
  + Pruebas de Responsive: Realiza pruebas en varios dispositivos y resoluciones de pantalla para asegurarte de que la experiencia de usuario sea coherente y funcional en todos ellos.
* Despliegue en Vercel
  + Integración con Git: Asegúrate de tener tu proyecto en un repositorio de Git. Esto facilitará la integración continua y el despliegue automático con Vercel.
  + Configuración en Vercel: Conecta tu repositorio de GitHub, GitLab o Bitbucket con Vercel desde su panel de control. Sigue las instrucciones para configurar el despliegue automático cada vez que hagas push a tu rama principal.
* Git y gestión del tiempo
  + Control de versiones con Git: Utiliza ramas para las nuevas características y merges para integrarlas al main. Implementa commits claros y descriptivos para cada cambio significativo.
  + Registro de tiempo: Podrías considerar herramientas como Toggl o Clockify para llevar un registro del tiempo dedicado a cada tarea del proyecto. Esto te ayudará a gestionar mejor tus esfuerzos y evaluar la eficiencia del desarrollo.
* Uso de Tailwind CSS
  + Integración con Angular: Instala Tailwind CSS en tu proyecto Angular. Esto se puede hacer instalando las dependencias necesarias (tailwindcss, postcss, autoprefixer) y configurando Tailwind en tu proyecto según la [documentación oficial](https://tailwindcss.com/docs/guides/angular).
  + Estilos responsivos: Utiliza las clases de utilidad de Tailwind para diseñar la UI. Aprovecha las características de Tailwind para responsive design, como sm:, md:, lg:, etc., para adaptar tu diseño a diferentes tamaños de pantalla.