



**UNIVERSIDAD AUTÓNOMA DE QUERÉTARO**

**FACULTAD DE INFORMÁTICA**

**Desarrollo de Aplicaciones Moviles**

**Mtra. María Fernanda Juárez Tirado**

**Fecha de entrega: 20 de Diciembre del 2024**

# Examen Final

Gael Alejandro García Lugo – 278440

**Resumen**

Esta aplicación utilizando TypeScript, Angular, e Ionic , está diseñada para gestionar el inventario de productos de una tienda, usando la API pública https://dummyjson.com/products. La aplicación es una plataforma para visualizar, filtrar, eliminar y exportar información de los productos disponibles que haya en la API.

Es para optimizar la experiencia del usuario mediante una navegación ágil y sencilla. El usuario se beneficia de una búsqueda mejorada gracias a la integración de motores de busqueda, que reduce el tiempo necesario para localizar productos específicos. Al seleccionar un producto, se presenta de manera clara toda la información relevante, incluyendo precio, categoría y disponibilidad, lo que facilita la toma de decisiones informadas.

**Propósito Principal**

El propósito principal de la aplicación es optimizar la gestión de inventarios de productos en una tienda, realizando las tareas de manera rápida, ordenada y sencilla. Mediante una interfaz interactiva, la aplicación permite a los usuarios tener un control total sobre su inventario.

**Objetivos**

- Cargar y visualizar productos: Dar una lista que incluya nombre, descripción, precio, categoría y stock de cada producto.
- Facilitar la búsqueda y el filtrado: Incorporar un buscador interactivo y un ion-select para filtrar por categoría, mejorando así la localización de productos en específico.
- Visualizar detalles de productos: Ofrecer al usuario una vista de cada producto al hacer clic sobre él en la lista principal, mostrando todas las propiedades importantes del producto según los datos que da la API.
- Eliminar productos: Tener la opción de eliminar productos por individual de la lista, permitiendo mantener el inventario actualizado según las necesidades de la tienda.
- Exportar productos: Integrar la funcionalidad de exportación para descargar la lista visible de productos en un archivo JSON, facilitando así su registro y análisis externo. Este es un botón que se encuentra hasta debajo de la app

**Características**

- Cargar productos desde una API: Los datos de los productos se obtienen de la API https://dummyjson.com/products y se cargan en la aplicación.

- Visualización en lista: Se visualiza una lista que muestra las principales propiedades de los productos: nombre, descripción, precio, categoría y stock.
- Búsqueda interactiva: Un componente de búsqueda permite encontrar productos en base a lo que ingreses en el buscador.
- Filtro por categoría: El usuario puede seleccionar una categoría en la botonera mediante un ion-select para filtrar los productos que desee.
- Visualización de detalles: Al seleccionar un producto en la lista, se despliega una vista que muestra todas las propiedades relevantes del producto.
- Eliminación de productos: Los productos pueden ser eliminados individualmente desde la lista, ayudando a mantener actualizado el inventario, según el criterio del usuario.
- Exportación a JSON: Los productos visibles en la lista pueden ser exportados a un archivo JSON descargable para su almacenamiento o análisis de manera externa.

1. **Componentes principales:**
    - app.component.html: Estructura principal de la aplicación.
    - busqueda-productos.component.html: Presenta la barra de búsqueda y los resultados de búsqueda.
    - producto.component.html: Muestra información detallada del producto seleccionado.
    - Exportación de Productos: Permite descargar un archivo JSON con la información del inventario filtrado.
2. **Servicios:**
    - Servicio de Productos: Maneja las operaciones con la API para la obtención, eliminación y gestión de productos.
    - Servicio de Filtrado: Aplica los criterios de tu agrado para categorizar productos de manera dinámica.
3. Modelos:
    - **Type de Producto:** Representa los datos de un producto excluyendo las propiedades "meta", "dimensions" y "reviews" de los datos JSON de la API. Este modelo ayuda a la estructura de los datos a nivel de aplicación, facilitando su uso en diferentes componentes.

**Plugins Utilizados**

- Ionic Searchbar: Barra de búsqueda que permite filtrar productos de manera interactiva. (Facilita la búsqueda mediante un diseño responsive y accesible.)
- Ionic List y Avatar: Componentes utilizados para estructurar y mostrar los resultados de búsqueda. (Presentar la información de los productos de forma ordenada.)
- Ionic Card: Utilizado en producto.component.html para mostrar los detalles del producto. (Estructurar la información del producto con un diseño atractivo.)

**Exportación JSON**

Estructura:

- Los datos exportados estarán en formato JSON.
- El archivo es legible gracias a JSON.stringify con el parámetro de la sangría (null, 2), que añade espacios para que los datos sean amigables al usuario

Flujo de exportación:

- Generar los Datos para Exportar:
  - Se convierte la variable this.productosFiltrados (un array de objetos) en una cadena JSON mediante JSON.stringify. Esto hará que los datos estén en un formato exportable.
- Crear un Objeto Blob:
  - Un objeto Blob contiene los datos JSON en binario. Este objeto permite que los datos puedan ser recibidos por el navegador.
- Crear un Elemento <a>:
  - El código crea dinámicamente un elemento de enlace HTML <a>.
  - El href de este enlace se configura con la URL de los datos contenidos en el objeto Blob usando URL.createObjectURL.
- Configurar la Descarga del Archivo:
  - Se establece el atributo download del elemento <a> con el nombre deseado del archivo: productos.json.
- Iniciar la Descarga:
  - El enlace simulado se activa llamando a link.click(), lo que genera la descarga del archivo JSON para el usuario.
- Liberar Recursos:
  - Se elimina la referencia a la URL del objeto Blob con URL.revokeObjectURL para liberar memoria.

## Conclusión

En base a lo realizado, los objetivos se cumplieron en la app de ionic, puesto que al realizar una consulta ya sea por método de búsqueda o por categorización por medio de ion-select, este nos arroja un resultado o en caso de hacer una búsqueda errónea no nos arroja nada, así mismo dentro de los criterios establecidos en el trabajo se puede observar que el botón de eliminar es un botón independiente por lo que trabaja de manera efectiva, dentro de la misma aplicación podemos visualizar los objetos que tengamos del inventariado de la API mediante una imagen de dicho producto y también las características que nos proporciona la misma. Dentro del ámbito del desarrollo de la aplicación pude notar los diferentes tipos de componentes que en este caso proporciona Ionic para poder realizar aplicaciones móviles, ya que nos permite poder movernos con facilidad dentro del código, claro especificando rutas cada que creemos un nuevo ts o apartado en el cual vayamos a modificar para crear acciones dentro de la app.

La app se desenvuelve de manera esperada por lo que cuando inicio el host no se encuentra ningún problema de start, eso nos quiere decir que se encuentra bien el código, así mismo el uso de los componentes siempre va facilitar la respuesta que podremos recibir mediante ionic ya que ya hay una estructura compuesta dentro del código, únicamente hay que deformarla para la creación de nuevos componentes o así mismo darle movilidad a los componentes ya existentes. Entendí que el desarrollo y exportación mediante JSON es muy fluido y este nos permite tener una vista mas detallada de los datos, se puede realizar de manera en que los datos filtrados aparezcan dentro de la descarga, esto ayuda a temas de personalización con el usuario. Muy bueno Ionic a la hora de la crear o editar componentes.

## Referencias


- Ionicframework. (n.d.). UI Components: Ionic Documentation. Retrieved from https://ionicframework.com/docs/components
- (N.d.). Retrieved from https://www.youtube.com/watch?v=gQq8KvO2WsY
- (N.d.). Retrieved from https://www.youtube.com/watch?v=WO2TkyItEkA