Decisiones tomadas:

Se optó por un desarrollo de manera modular, con tres módulos bien definidos

- AppModule, como base de la aplicación
- CoreModule, que concentra los aspectos básicos del sitio y que se comparten en común entre posibles microservicios que tenga el sitio
- ProductsModule que tiene el desarrollo propiamente pedido para esta oportunidad

La decisión de desarrollarlo así es pensando en que el sitio escale y se pueda trabajar por módulos separados, compartiendo componentes y características de otros módulos cuando se deseen integrar y simultáneamente tenga la flexibilidad para poder desarrollarse y testear independientemente.Por otro lado tiene un gran beneficio en cuanto a la performance de la aplicación ya que se carga de manera lazy. Solo se cargará cuando se acceda al módulo por medio de las rutas que lleven a él y no al cargar e iniciar la aplicación en todo su conjunto.
Dentro del desarrollo se crearon 4 carpetas

- Components
- Models
- Services
- Store

Componentes se tendrán todos los componentes con sus controladores y vistas, manteniendo la característica que provee angular de MVC.
Models: Agrupara todas las clases e interfaces del dominio de negocio de esta aplicación
Services: Agrupara los servicios que utiliza y puede exportar el modulo. Siendo el nexo de comunicación entre los api service y manejo de data de una sola fuente para este modulo.
Store: Se encarga de administrar el estado de la aplicación de manera única y aislando los componentes para enfocarnos únicamente en su funcionalidad, gracias al uso de observables.

La aplicación se basa en un componente selector con 3 hijos que heredan / comparten su vista. Y los controladores se sobreescribe para realizar las llamadas pertinentes para obtener la data o modificarla del store. Y un formulario que tiene el estado de cada opción seleccionada de los selectores y invoca los 3 selectores.
Por otro lado tenemos la lista de productos (products-list) que es filtrada del store, por medio de los actions y effects que se ejecutan.
La complejidad se encuentra en reutilizar la llamada de los effects para diferentes tipos de actions, como ser filtrar la lista de productos al seleccionar o cargar las opciones de cada selector/dropdown, como también realizar el efecto en cascada en cada uno de ellos. Por lo que en los actions de select y dropdown se utilizó una clase y se heredó por tipo de selector.
Y como decisión de diseño se decidió generar en dropdownEffect un método para filtrar los selectores en cascada y no en el padre de dropdownAction, para mantener su poca complejidad y menos código.

Tecnologías utilizadas:

- Angular v8.2.8
- Angular Cli v8.3.6
- ES2019
- Rxjs
- Ngrx
- Boostrap - librería css y scss
- flex-box
- npm

Instructivo de instalación:

Se debe ejecutar el siguiente procedimiento para poder descargar, instalar y ejecutar la aplicación

DESCARGA

1. Utilizando una herramienta como gitbash,en el caso de utilizar windows, dirigirse a la carpeta donde desea descargar e instalar la aplicación. Ej: “C:\Documents\projects”
2. Para descargar el proyecto ingresar en consola:
   git clone https://github.com/cjuhal/iantech
3. Ingresar a la carpeta del proyecto
   cd iantech

INSTALACIÓN

1. Importante: asegúrese de tener node.js y npm instalados a la última versión.
   Verificando por consola npm -v y node -v
   https://nodejs.org/es/
2. Ejecutar por consola:
   npm install
   para descargar e instalar todas las librerías y dependencias de este proyecto, entre los que se encuentra angular, angular cli, etc..

EJECUTAR

1. Para ejecutar la aplicación como desarrolladores debemos utilizar el comando que provee angular cli
   ng serve --o
   (esto generará un dist temporal con la aplicación compilada y abrirá el explorador predeterminado con el sitio http://localhost:4200/)
2. abrir otra consola en paralelo para ejecutar un servicio con la data incluida para este desafío, ejecutar:
   json-server db.json
   (debe estar ubicado en la carpeta raíz del proyecto, ya que el archivo db.json se encuentra ahí)

   INFORMACIÓN ADICIONAL:
   - Si la aplicación será instalada en un servidor web como por ejemplo apache, y las rutas utilizadas para web service es distinta, puede configurarse en la carpeta /enviroments/enviroments.prod.ts o enviroments.ts dependiendo del ambiente y la compilación (atributo urlApi).
   - Si queremos compilar la aplicación y correrla en un servidor de producción el comando a utilizar es
     ng build --env=prod
     --env=prod configura la aplicación indicando que variables de entorno se utilizarán en él.
     Adicionalmente en la carpeta generada /dist, se deberá configurar un archivo .htaccess en el caso de utilizarse apache por ejemplo.