

Proyecto Final Laboratorio III

Laboratorio III

Tecnicatura Universitaria en Programación.

ALUMNOS:

Fenoy Luciana Ariel

Franco Vertiz Lucia

Neira María Eugenia

Laguna Oscar Gabriel

Objetivo:

El objetivo de este informe técnico es proporcionar una visión detallada del desarrollo e implementación de una aplicación ficticia que contiene una lista de supermercados con sus respectivos productos y el precio de cada uno, destinada a gestionar un carrito de compras donde el cliente puede colocar productos de uno o varios supermercados que integran la aplicación.

Resumen Ejecutivo:

Para el cliente:

El sistema desarrollado tiene como propósito principal ofrecer a los clientes una plataforma intuitiva desde la cual puedan realizar compras en línea, consultar promociones en productos específicos o en categorías de productos y gestionar sus listas de compras. Estos pueden elegir los productos de uno o varios supermercados seleccionando por cercanía, por precio o calidad.

En primer lugar, el cliente deberá elegir si la lista de compras será realizada a partir de un supermercado en específico o si la creará independientemente de la compañía que posea esos productos, para luego pasar a buscar los mismos por nombre, por oferta o por categoría.

Realizada la búsqueda, se podrá seleccionar los productos por id (será mostrado por pantalla) para que el sistema los agregue al carrito.

Una vez confeccionada la lista de compras, los consumidores podrán modificar la misma ya sea eliminando productos, adhiriendo nuevos o modificando sus cantidades. Finalmente, el sistema arrojará el total a ser abonado por el consumidor junto con los productos seleccionados.

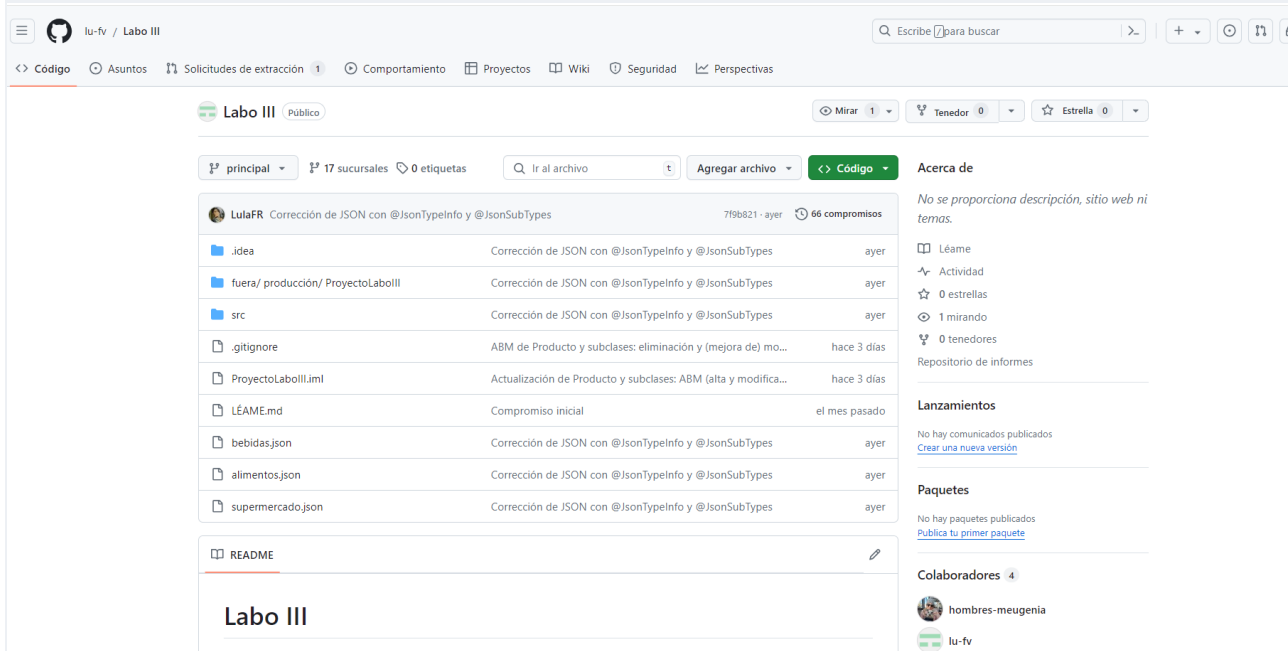
Para el administrador del sistema:

Funciones de ABML de supermercados y productos junto con las funciones de administración de ofertas (aplicación y eliminación).

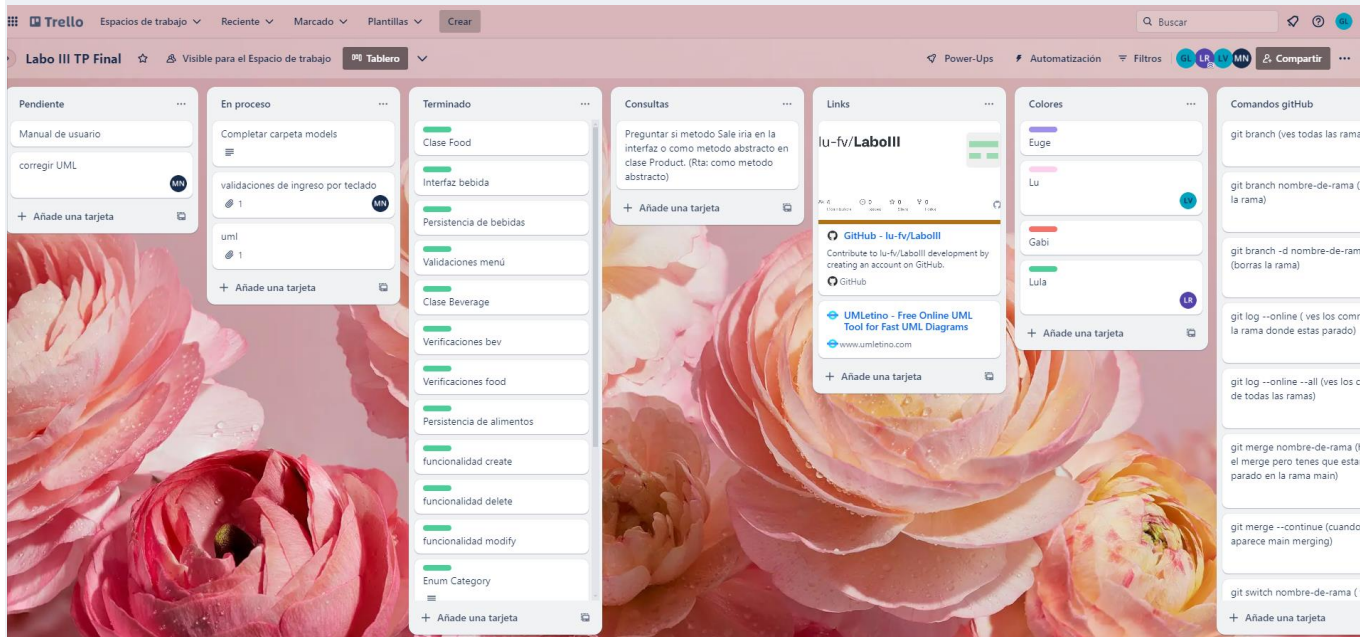
Herramientas utilizadas:

GitHub: se creó un nuevo repositorio denominado “Labo III” donde se fueron subiendo los cambios realizados a lo largo de la confección del proyecto. Los distintos integrantes del grupo crearon su propia

rama para trabajar dentro de una funcionalidad específica para que cuando estuviese lista, se realice el merge de su rama para fusionarla con el Main.



Trello: herramienta utilizada como calendario de tareas y reuniones. También se subió allí el UML inicial y final del proyecto.



Desarrollo y Funcionalidades:

1. Interfaz de Usuario (UI):

- La interfaz ha sido diseñada para ser amigable y accesible, con secciones claras para diferentes categorías de productos, ofertas y búsqueda rápida.
- Se ha implementado un proceso de registro y login seguro solo para el acceso privado, con opciones de autenticación mediante una contraseña creada y guardada como una variable constante.
- No se ha implementado Interfaz gráfica en el proyecto, sino que todas las funcionalidades se utilizan desde la consola.

2. Funcionalidades para Clientes:

- **Crear una lista por Supermercado:** Los clientes pueden explorar el catálogo completo de productos siempre vinculado a un supermercado en particular elegido por nombre, donde éste desplegará una lista de productos. Una vez elegido, el cliente podrá buscar por nombre de producto, por categoría o por productos en oferta (aquellos que estén con un precio reducido). De esta manera irá agregando a su carrito los productos deseados mediante su Id que será informado por pantalla. También contará con las opciones de ver su lista o de modificarla.
- **Crear lista en general:** realiza exactamente las mismas funciones del menú anterior pero aplicada a todos los supermercados que componen la App, imprimiendo por pantalla todos los productos (no repetidos) para que el cliente pueda elegir entre ellos sin distinción del supermercado que los contiene. De esta manera se puede buscar los productos más baratos independientemente de quien lo contenga.
- **Listas de Compras:** Funcionalidad para crear y gestionar listas de compras, se encuentra en las funciones mencionadas precedentemente y se pueden gestionar una vez creadas con métodos de consulta y de modificación de los productos que contiene ya sea con su eliminación o con la modificación de las cantidades.



- **Promociones y descuentos:** Los usuarios pueden ver mediante el menú todos aquellos productos que se encuentren con ofertas especiales o promociones. Éstos serán mostrados con el precio original y el precio aplicado el descuento.

3. Funcionalidades para Administradores:

- **Productos:** funciones de ABML de productos los supermercados contenidos en la App. Permite la creación, modificación de cualquiera de los campos, eliminación o mostrar los listados de productos por cada supermercado. Una vez realizada la opción deseada, estas funciones guardan el mismo en el supermercado seleccionado.
- **Supermercados:** Funciones de ABML de supermercados. Al igual que los productos, en este caso se pueden dar de alta nuevos supermercados e incluirlos a la app, darlos de baja, modificar cualquiera de sus campos (a excepción de la lista de productos ya que esta se realiza desde el menú anterior) y también muestra el listado de los que se encuentran vigentes. También desde este menú, se aplican las ofertas a los productos elegidos por nombre, o también se puede aplicar a grupo de productos por su categoría (almacén, lácteos, bebidas etc.).

Decisiones de diseño tomadas para la confección del sistema:

Se crearon las siguientes clases en la carpeta “Models” con sus correspondientes constructores, getters and setters:

- *Product*: clase abstracta de la que heredan las clases “Beverage” y “Food”. Posee los atributos *Id*, *productName*, *brand* y *category*.
- *ProductForSale*: clase que será utilizada como plantilla para los productos que contenga cada supermercado. Esta contendrá los atributos producto de tipo *Product* (para agregar a la lista tanto *foods* como *beverages*), *price*, *on sail* (dato booleano que indica si está en oferta) y porcentaje de descuento para aplicar una reducción en el precio.
- *Beverage*: clase que hereda de *Product* y a su vez posee un atributo “litres”.
- *Food*: clase que hereda de *Product* y posee otro atributo booleano “perishable” que indica si el producto es perecedero o no.
- *Supermarket*: clase que contiene atributos propios del supermercado como *name*, *cuit*, *address*, *phone* y una *List de Products*.
- *Cart*: clase de tipo genérica (T) que funciona en memoria ram tomando datos del json de *Supermarket* adhiriendo a un *HashMap* (para que no repita productos) los productos deseados, al finalizar la compra muestra el precio total. Sus atributos son “*totalPrice*” y “*cart*”. Una vez cerrado el



programa se perderan los datos cargados.

De todas las clases anteriormente mencionadas se crea una interfaz en la carpeta "Service" y a su vez, otra con la implementacion de esas interfaces con las principales funciones de cada una de manera tal que son llamadas desde la implementacion del service "Menu" que será lo que vemos en la consola de Java.

Resultados y Conclusiones: se ha evaluado exhaustivamente la implementación y funcionalidad del sistema en diferentes áreas clave. Los principales hallazgos indican que la aplicación actual presenta fortalezas significativas en términos de accesibilidad y usabilidad para los consumidores, especialmente en la navegación de los listados de productos y selección de los mismos. Sin embargo, también se identificaron áreas de mejora crítica, como la optimización del sistema de inventario de los supermercados. No se detalla la cantidad de productos que poseen por lo que de ser una aplicación real esto podría ser un gran problema al no poder cumplir con los pedidos de los usuarios. Pese a ello y teniendo en cuenta que el mismo se reduce a la entrega de un trabajo final para aprobar la materia Laboratorio III, lo anteriormente mencionado no se constituye como un factor de riesgo.

Material consultado:

www.baeldung.com

www.github.com

<https://www.youtube.com/watch?v=9ZJ-K-zk> Go
(utilizacion de GitHub).

<https://es.stackoverflow.com/>

diapositivas con el material teórico de la cursada.