Estudiante: Calloni Antonela – legajo: 14532

Profesores: Pioli Pablo, Ferreyra Juan Pablo

trabajo integrador diseño de sistemas

Contenido

[Explicación de la consigna 2](#_Toc57363889)

[Resolución 3](#_Toc57363890)

[Explicación de los archivos en GitHub 6](#_Toc57363891)

[Los archivos JSON 7](#_Toc57363892)

[Tests de aceptación 9](#_Toc57363893)

# Explicación de la consigna

Se comenzó trabajando sobre la idea de un Trabajo Práctico Integrador, a través de la realización de una plataforma e-commerce, a la que llamamos “OpenShop”.

Se subdividieron los estudiantes por grupos para trabajar en las diferentes áreas de la aplicación, siendo la nuestra Catálogo, que era seguida por Carrito, Cobranza y Despacho. Los integrantes del grupo Catálogo somos:

* Calloni, Antonela.
* Escudero, Carlos.
* Massoni, Ignacio.
* Pellarolo, Matías.

Para permitir la relación entre estas diferentes áreas, utilizamos estructuras JSON que son archivos emitidos y recibidos por las áreas con la información que cada una necesita.

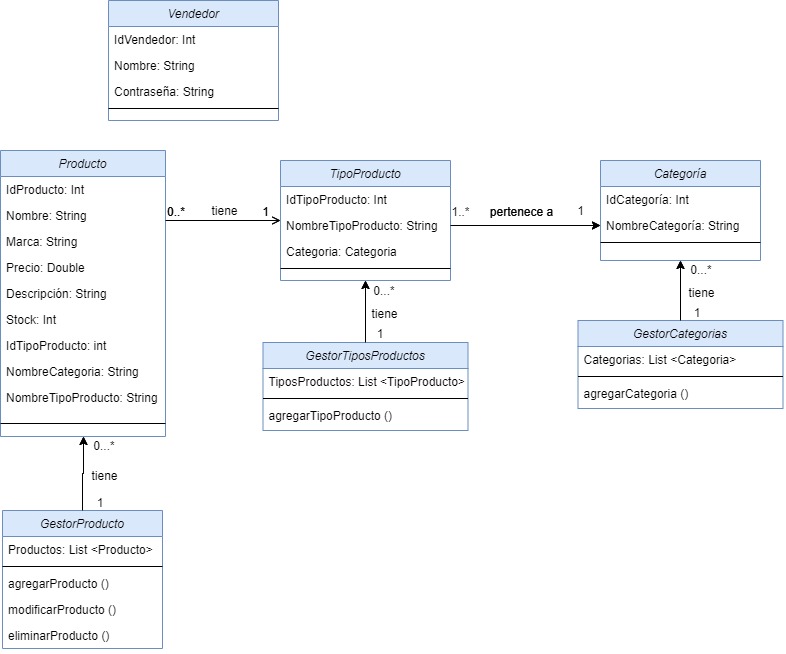
Los principales requerimientos del área Catálogo fueron:

* Permitir al Vendedor iniciar sesión.
* Permitir al Vendedor agregar un artículo para venta en conjunto con precio, características y existencia.
* Permitir al Vendedor modificar diversas características de un artículo para venta.
* Permitir al Vendedor eliminar un artículo para venta en conjunto con precio, características y existencia.
* Permitir transferir listado de productos para venta a grupo Carrito.
* Permitir obtener listado de productos vendidos de Despacho para actualización constante de stock.

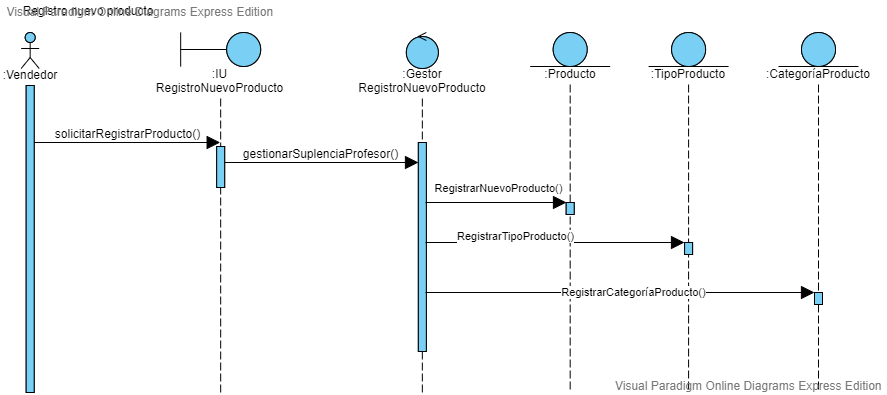
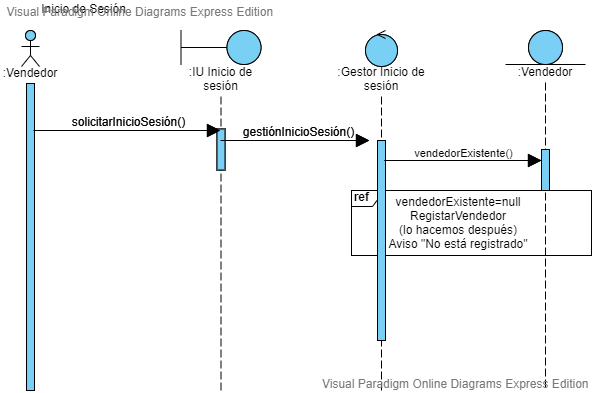
# Resolución

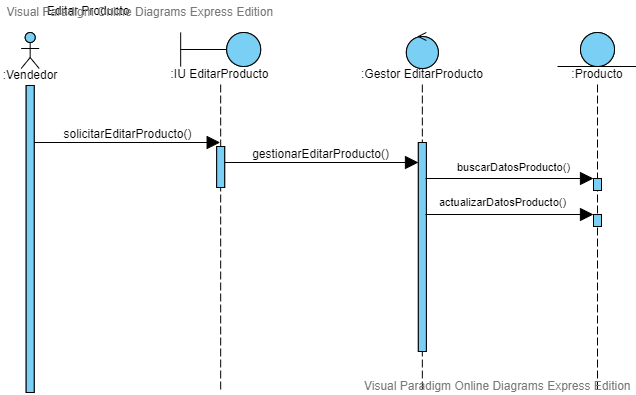
Luego de varias reuniones virtuales, discutimos entre los integrantes del grupo sobre las funcionalidades básicas del programa y realizamos diagramas que reflejan el diseño del mismo, así como dan las pautas principales para la posterior codificación.

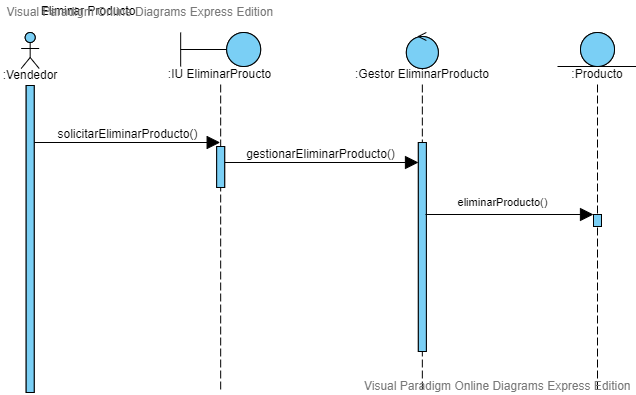
En primer lugar, para modelar la interconexión entre los objetos, realizamos un Diagrama de Clases, el cual nos ayudó a pensar mejor la estructura del programa.



Luego, hicimos diagramas de secuencia rápidos, como para ir adentrándonos en las funciones principales que iba a tener que cumplir el programa según sus requerimientos.







Una vez finalizado lo anterior, se prosiguió con la programación del Catálogo de manera individual. De esta manera, en la práctica, se realizaron las funcionalidades de los requerimientos básicos y se actualizó el Diagrama de Clases según lo cambiado en el código en función del tiempo.

A la hora de programar tuvimos dos lenguajes de programación como opciones a utilizar. Estos eran C# y Java. En este trabajo, trabajé con C# y utilicé Visual Studio 2017.

## Explicación de los archivos en GitHub

Dentro de la carpeta "DSI2020" se encuentran dos sub carpetas, una nombrada "OpenShop-Inicial" donde se encuentra todo lo realizado durante la primera parte del año respecto al proyecto, y la otra nombrada "OpenShopCatalogo" donde se encuentra todo lo realizado para el trabajo final de la materia.

Dentro de la carpeta OpenShopCatalogo se encuentra la documentación del Catálogo en este documento junto con los test de aceptación, una carpeta " XUnitTestProject1" con los tests unitarios, y otra carpeta " WindowsFormsApp1" donde está el archivo programa con todas sus clases y formularios.

Cabe aclarar, que además de lo pedido pedido por los docentes para la nota mínima que era: Tener un repositorio en Github con una implementación propia del proyecto OpenShop que incluya la implementación del área de trabajo a la que pertenezca incluyendo la funcionalidad de interconexión con las otras áreas, y los tests unitarios y de aceptación de la implementación del punto anterior; realicé adicionalmente la implementación de una interfaz gráfica en una aplicación de Windows Forms, además de una función adicional de poder registrar a diferentes vendedores para poder ingresar luego al sistema gestor de productos (que no era parte de nuestros requisitos iniciales).

Como conclusión… ¿quedaron cosas para mejorar? MUCHAS. Si pudiera dedicarle más tiempo, relacionaría a los vendedores con sus productos para que sólo un vendedor pueda editar sus productos y no los de otros, seguiría testeando para probar más posibles errores, agregaría la funcionalidad de poder agregar registros de Tipos de Productos con sus respectivas categorías, etc etc.

Sin embargo, lo poco que pude realizar en este trabajo me ayudó muchísimo para entender cómo programar algo desde cero, cómo usar los diagramas para orientarnos y la cantidad de dudas que surgen con ellos antes de programar, la IMPORTANCIA del testeo de la aplicación y lo importante de programar ya pensando en las posibles restricciones y errores, para no tener que hacer tantas modificaciones luego, la importancia de un buen diseño y de pensar bien los requerimientos del programa y poder discutirlos con nuestros clientes, la importancia también de la buena documentación y cómo nos ayuda cuando dejamos algo por dos semanas y queremos volver a ponernos y no sabemos por dónde empezar; pero por sobre todo me ayudó a sentarme y a amigarme con la programación, que no es de mis hobbies favoritos, pero que entiendo es una herramienta fundamental para nuestra profesión (si ya puedo decirle “nuestra”) y que además es un lenguaje universal, que nos ayuda a ejercitar el pensamiento abstracto y el ingenio.

## Los archivos JSON

El archivo JSON que mandamos desde catálogo:

[

{

"IdProducto": ,

"Nombre": ,

"Marca": ,

"Precio": ,

"Descripcion": ,

"Stock": ,

"Imagen": ,

"NombreCategoría": ,

"NombreTipoProducto": ,

},

...

]

EJEMPLO:

[

{

"IdProducto": "5487",

"Nombre": "Calculadora Casio fx-85MS",

"Marca": "Casio",

"Precio": "19394.99",

"Descripcion": "bla bla bla",

"Stock": "23",

"Imagen": "https://www.casio-intl.com/product/image/1425458815389/",

"NombreCategoría": "Tecnología",

"NombreTipoProducto": "Calculadoras"

},

{

"IdProducto": "3248",

"Nombre": "Celular Xiaomi Note 8",

"Marca": "Xiaomi",

"Precio": "38000.00",

"Descripcion": "bla bla bla",

"Stock": "7",

"Imagen": "https://d2pg6pehp4f12r.cloudfront.net/wp-content/uploads/2019/10/note-8-1.jpg",

"NombreCategoría": "Tecnología",

"NombreTipoProducto": "Celulares"

}

]

El archivo JSON que recibimos de Despacho:

{

"articulosDespachados":

[

{

"id":

"nombre":

"marca":

"precio":

"descripcion":

"cantidad":

}

]

}

Ejemplo:

{

"articulosDespachadosdos":

[

{

"id": "1";

"nombre": "Remera clasica";

"marca": "Nike"

"precio": "$1000";

"descripcion": "Remera de algodón";

"cantidad": "1";

}

]

}

## 

## Tests de aceptación

Para finalizar, realizamos diferentes tests de aceptación a nuestro programa, para intentar encontrar errores de una forma más eficiente y pudiendo dejar registro de los mismos.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nº | Nombre | Valores ingresados | Resultado esperado | Resultado observado | Resultado de prueba | Tipo error | Prioridad | Estado de Error |
| 1 | Registro nombre ingresando números | Nombre: an34 | Notificación de que deben ser solo letras | Se recibe la notificación | Éxito |  |  |  |
| 2 | Registro nombre ingresado correctamente | Nombre: Antonela | Desaparición del símbolo de error | Se recibe la notificación | Éxito |  |  |  |
| 3 | Registro validación de contraseña (ingrese contraseña nuevamente) mal ingresada | Contraseña1: 123456123 Contraseña2: 111 | Notificación de que las contraseñas no son iguales | Se recibe la notificación | Éxito |  |  |  |
| 4 | Registro validez contraseña (ingrese contraseña nuevamente) bien ingresada | Contraseña1: 123456123 Contraseña2: 123456123 | Desaparición del símbolo de error | Se recibe la notificación | Éxito |  |  |  |
| 5 |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |  |