

Universidad ORT Uruguay

Facultad de Ingeniería

Ingeniería de Software 1

Letra del Obligatorio 2

Marcos Novelli (201663)

Matias Salles (201701)

Entregado como requisito de la materia Ingeniería de
Software 1

25 de noviembre de 2019

Declaraciones de autoría

Nosotros, AUTORES , declaramos que el trabajo que se presenta en esa obra es de nuestra propia mano. Podemos asegurar que:

- La obra fue producida en su totalidad mientras realizábamos
- Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad;
- Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra;
- En la obra, hemos acusado recibo de las ayudas recibidas;
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuido por otros, y qué fue contribuido por nosotros;
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.

Resumen

Este informe presenta un programa de compra-venta online de productos comestibles, el cual promueve un sistema ecológico.

Se basa en la eliminación, por parte de la empresa, de envases en donde vender sus productos(frutas, verduras o frutos secos). El cliente es el encargado de llevar un envase a la tienda para poder retirar su producto comprado previamente por medio de la aplicación.

Formas de comprar en EcoShop:

- Compra presencial: El cliente entra a la tienda con un envase y compra el producto deseado.
- Compra por medio de la aplicación: El usuario entra a la aplicación EcoShop y elige los productos que desea comprar agregandolos al su "carrito" e ingresando la cantidad(en kg) deseada. Luego, el usuario ingresa el día y el local en donde desea retirar su los productos. Tras concretar la compra en el sistema, solo resta ir a buscar, con los envases necesarios, el producto el día y al local especificado.

Índice general

1. Versionado	2
1.1. Repositorio utilizado	2
1.2. Criterios de versionado	2
1.3. Resumen del log de versiones	3
2. Codificación	4
2.1. Estándar de codificación	4
2.2. Pruebas unitarias	6
2.3. Análisis de código	7
3. Interfaz de usuario y usabilidad	8
3.1. Criterios de interfaz de usuario	8
4. Evaluación de usabilidad	10
5. Pruebas funcionales	12
5.1. Técnicas de prueba aplicadas	12
5.2. Casos de prueba	12
5.3. Sesiones de ejecución de pruebas	12
6. Reporte de defectos	13
6.1. Definición de categorías de defectos	13
6.1.1. Defectos encontrados por iteración	13
6.2. Estado de calidad global	14
7. Reflexión	16

1. Versionado

1.1. Repositorio utilizado

En el proyecto utilizamos la plataforma Github: El repositorio se encuentra en el siguiente link:

<https://github.com/matiassalles99/Obligatorio-2.git>

Identificamos los siguientes elementos de la configuración de software:

- Diagramas de clases/conceptuales para desarrollar la aplicación EcoShop de una forma ordenada.
- Proyecto de NetBeans: Utilizado para desarrollar la aplicación EcoShop.
- Archivos .txt con explicaciones de ciertas decisiones (ej: explicando el diagrama de clases), o con código temporal (ej: inicializar una ventana en JavaFx).
- Imágenes utilizadas para el proyecto de Overleaf.

El proyecto de Overleaf se dejó por fuera de los elementos de la configuración de software, ya que para modificarlo era más cómodo trabajarlo directamente desde la página, que bajarlo y subirlo al repositorio luego de cambios.

1.2. Criterios de versionado

Para el desarrollo de la aplicación se utilizaron dos ramas: master y develop. Utilizamos la master para versiones estables, de la cual se realizaban releases cuando la aplicación se encontraba lo suficientemente sólida para ser utilizada por un usuario. Utilizamos la rama develop para el desarrollo de la aplicación, en la cual se realizó la mayor parte del trabajo. Para la realización del proyecto, se arreglaron reuniones entre ambos integrantes del equipo, para discutir el estado de la aplicación y los pasos a seguir. También hubo una distribución del trabajo orientada hacia la interfaz de usuario por parte de Marcos, y hacia el dominio de parte de Matias. Con cada cambio agregado, se incluyó un mensaje en el commit describiendo los principales cambios subidos al repositorio, sobre estos, se aprendió en el transcurso del proyecto la ventaja de realizar más commits con menos cambios para luego tener una mejor visión de cada cambio realizado.

1.3. Resumen del log de versiones

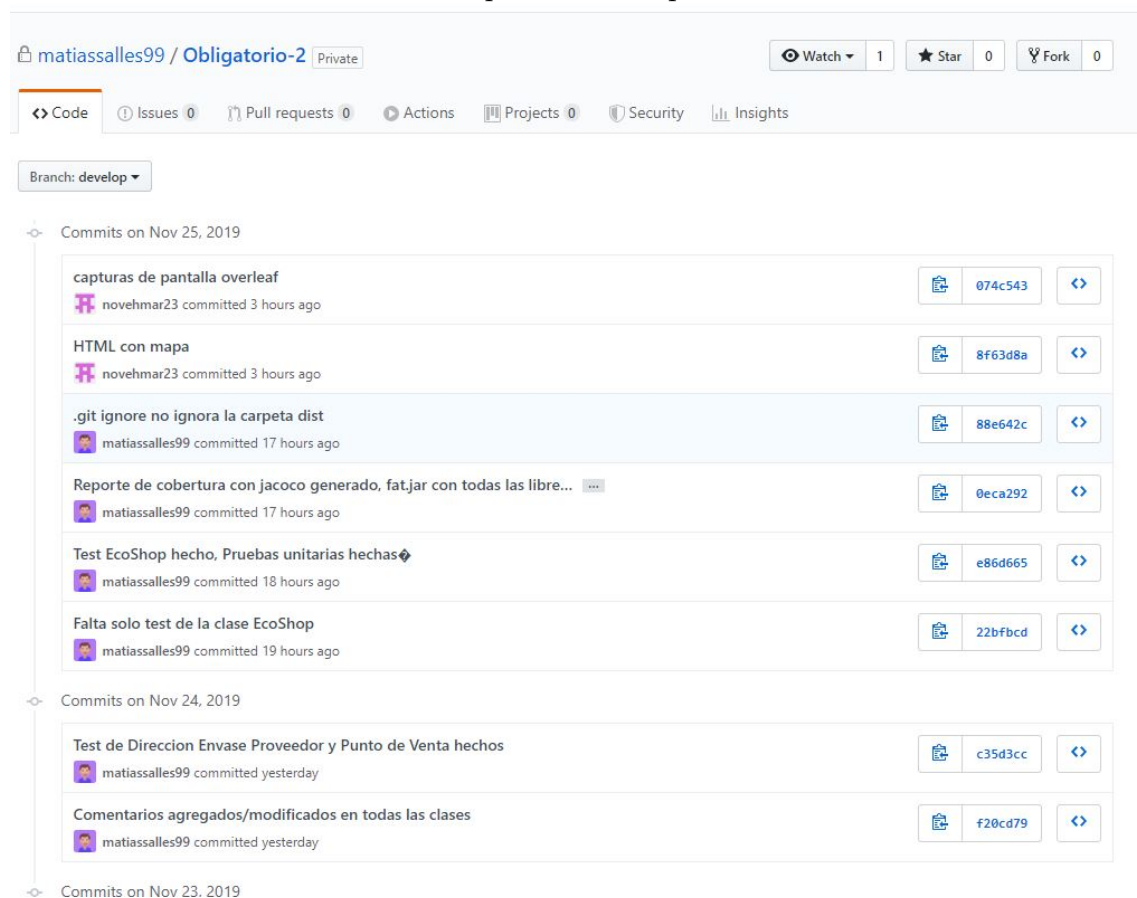
A continuación se dejan imágenes del repositorio en GitHub, dejando evidencia de el progreso del proyecto en el ultimo mes.

barra.JPG barra.JPG

Repositorio Obligatorio 2 Ingenieria de Software - Marcos Novelli - Matias Salles



develop.JPG develop.JPG



2. Codificación

2.1. Estándar de codificación

Este proyecto respeta los estándares de codificación para un código más prolijo, entendible y eficaz que facilite el trabajo en equipo y también al momento de revisar el código. Presentaremos fragmentos del código como modo de ejemplo de varios puntos que conforman una codificación prolija.

ELEMENTOS BÁSICOS:

En la siguiente imagen se pueden ver los siguientes Elementos Básicos de los estándares:



```
106  /**
107   * Carga nuevo artículo: Manzana Verde
108   */
109  private void cargarManzanaVerde() {
110      //Vaiaables
111      double densidadDeManzanaVerde = 947;
112      double precioPorKGManzanaVerde = 59;
113      String rutaDeImagenManzanaVerde
114          = "interfazGrafica/imagenes/manzanaVerde2.jpg";
115      String calificadorManzanaVerde = "Fruta";
116
117      //Creo Artículo
118      IArticulo manzanaVerde
119          = new Articulo("Manzana Verde", rutaDeImagenManzanaVerde,
120                        precioPorKGManzanaVerde, densidadDeManzanaVerde,
121                        calificadorManzanaVerde);
122
123      //
124      Seteo los posibles envases para este artículo
125      manzanaVerde.agregarEnvase(obtenerEnvase("Bolsa Mediana"));
126      manzanaVerde.agregarEnvase(obtenerEnvase("Bolsa Grande"));
127
128      //Ingreso este nuevo artículo al sistema
129      sistemaEcoShop.registrarArticulo(manzanaVerde);
130  }
```

Figura 2.1: Método de ejemplo: "cargarManzanaVerde"

- Comienzo y fin de estructuras: El inicio del método se encuentra alineado con el corchete que le da cierre al mismo.
- Agrupamiento de sentencias: Es evidente los 4 subgrupos de de líneas de código, separados por un espacio, y comentados
- Indentación, anidamiento: Está perfectamente indentado distinguiendo así, las acciones que se encuentras dentro de otras. También se nota un buen uso del anidamiento en el momento de crear el artículo "manzanaVerde".
- Nombres de elementos: Son nemotécnicos logrando una buena comprensión de su funcionalidad.
- Mayúsculas y minúsculas: Se encuentran siempre presentes para hacer más fácil la lectura de nombres con mas de una palabra.
- Paréntesis: Existe un correcto control de los paréntesis no teniendo paréntesis innecesarios.
- Espaciado de parámetros: Se pueden ver en la creación del artículo
- Estructura de directorio y nombres de archivo

ESTÁNDAR DE CODIFICACIÓN JAVA:

En la imagen anterior se puede ver que se cumplen los siguientes puntos:

- Clases comienzan en mayúscula: ej: "Articulo" o "IArticulo"
- Paquetes y métodos comienzan en minúscula: "cargarManzanaVerde"
- Uso de camelCase para diferenciar palabras
- Llaves estilo Kernighan and Ritchie

Estilo Kernighan&Ritchie/Kernel

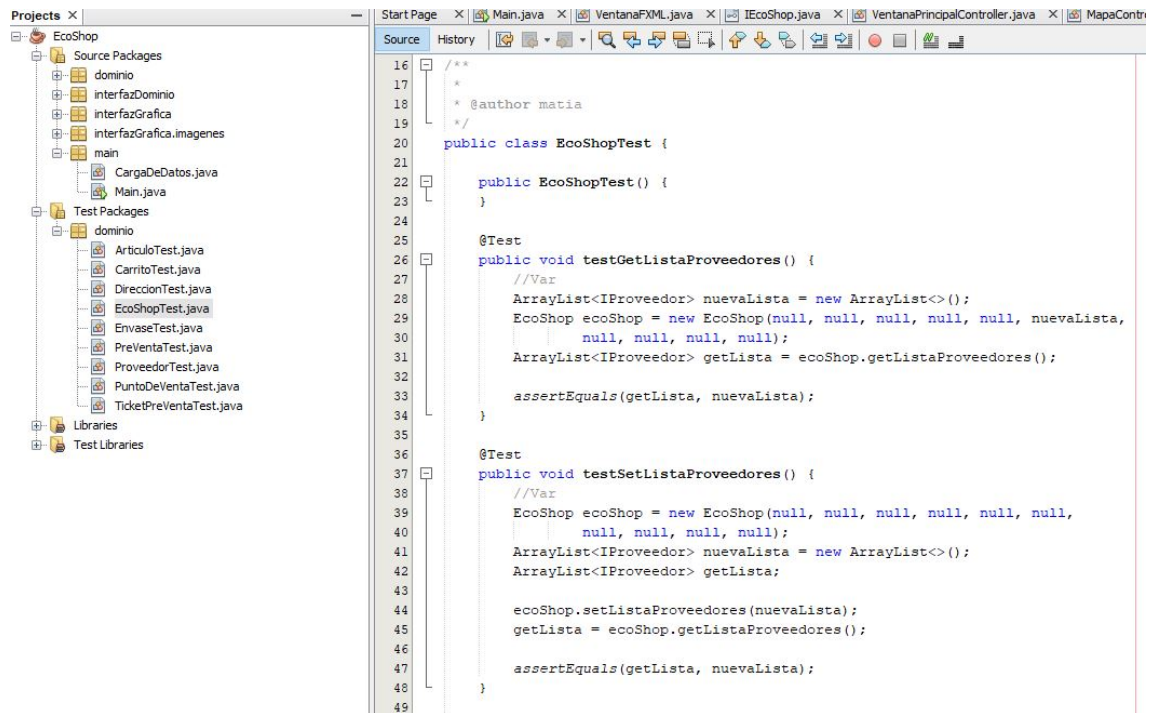
```
if(condicion) {
    sentencia0;
    sentencia1;
}
```

Figura 2.2: Ejemplo estilo Kernighan and Ritchie

También, se tiene en cuenta la portabilidad al solicitar la imagen "manzanaVerde2.jpg" por no tener direcciones de directorios o servidores fijos

2.2. Pruebas unitarias

Se utilizó JUnit y sus funcionalidades para realizar las pruebas unitarias, destacando los `assertArrayEquals`, `assertTrue/False` y `assertEquals`. A continuación se incluye un ejemplo de prueba unitaria en la cual podemos destacar los 3 bloques de: inicialización, manipulación y chequeo, y el análisis de cobertura.



```

16  /**
17   *
18   * @author matia
19   */
20  public class EcoShopTest {
21
22      public EcoShopTest() {
23      }
24
25      @Test
26      public void testGetListaProveedores() {
27          //Var
28          ArrayList<IProveedor> nuevaLista = new ArrayList<>();
29          EcoShop ecoShop = new EcoShop(null, null, null, null, null, nuevaLista,
30              null, null, null, null);
31          ArrayList<IProveedor> getLista = ecoShop.getListaProveedores();
32
33          assertEquals(getLista, nuevaLista);
34      }
35
36      @Test
37      public void testSetListaProveedores() {
38          //Var
39          EcoShop ecoShop = new EcoShop(null, null, null, null, null, null,
40              null, null, null, null);
41          ArrayList<IProveedor> nuevaLista = new ArrayList<>();
42          ArrayList<IProveedor> getLista;
43
44          ecoShop.setListaProveedores(nuevaLista);
45          getLista = ecoShop.getListaProveedores();
46
47          assertEquals(getLista, nuevaLista);
48      }
49  }

```

matiasalles99/Obligatorio-2: Re: x dominio

← → ↺ Archivo | C:/Users/novhm/Documents/_ORT/2019/Ing%20Software%201/Obligatorio%20GIT/NUEVO/Obligatorio-2/Reporte%20de%20c...

Aplicaciones Icons

JaCoCoverage analysis of project "EcoShop" (powered by JaCoCo from EcEmma) > dominio

dominio

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
EcoShop	<div><div></div></div>	87%	<div><div></div></div>	73%	18	88	20	187	3	55	0	1
Articulo	<div><div></div></div>	85%	<div><div></div></div>	100%	2	31	11	71	2	30	0	1
Carrito	<div><div></div></div>	94%	<div><div></div></div>	73%	10	42	4	94	0	20	0	1
TicketPreVenta	<div><div></div></div>	95%	<div><div></div></div>	75%	4	20	4	69	2	14	0	1
PreVenta	<div><div></div></div>	87%	<div><div></div></div>	25%	3	20	3	31	1	18	0	1
Envase	<div><div></div></div>	80%	<div><div></div></div>	n/a	1	9	4	17	1	9	0	1
Direccion	<div><div></div></div>	100%	<div><div></div></div>	n/a	0	21	0	44	0	21	0	1
PuntoDeVenta	<div><div></div></div>	100%	<div><div></div></div>	n/a	0	14	0	30	0	14	0	1
Proveedor	<div><div></div></div>	100%	<div><div></div></div>	n/a	0	10	0	22	0	10	0	1
Total	182 of 1.970	91%	36 of 128	72%	38	255	46	565	9	191	0	9

2.3. Análisis de código

Para mejorar la calidad de código se incluyo el uso de una interfaz para el dominio, la cual fue documentada cumpliendo con los estándares para luego generar un JavaDoc (incluido en el proyecto). Esta decisión nos permitió separar la implementación del dominio de la interfaz gráfica, distinguir funciones elementales las cuales eran necesarias para el funcionamiento de la interfaz gráfica y obtener una forma de trabajo mas ordenada. También el uso de una clase independiente para la carga de datos ayudo a hacer fácil el ingreso, modificación de datos a ingresar al sistema.

3. Interfaz de usuario y usabilidad

3.1. Criterios de interfaz de usuario

Discutir los criterios aplicados para la construcción de la interfaz de usuario. Por ejemplo: legibilidad, navegación, layout, controles, gestión de errores.

- Parte gráfica: El GUI presenta una paleta de colores con relación a los objetivos de la aplicación. Se manejan todos de verdes opacos y no escandalosos(intensos) para transmitir sensaciones de salud, vida, paz, tranquilidad y "el camino del bien". Pequeños tonos de marrón transmiten sensación de calidez, hogar.
- Interfaz por teclado: acciones que generalmente se realizan por medio de determinadas teclas del keyboard, están habilitadas. Como lo está, por ejemplo, la tecla "Enter" al buscar un artículo en la barra de búsqueda; o esta misma tecla presionada como paso siguiente al ingreso del kilaje a comprar.
- Dimensiones: todas las acciones a realizar se pueden lograr en pocos pasos y estos son claros sin tener exceso de uniformación. Por lo tanto podemos decir que es un sistema denso y preciso. Se navega con total naturalidad ya que tiene sus componentes disfrutados tal y como los tienen la mayoría de las páginas de compra online(nos basamos en las páginas web de Tienda Inglesa y de Amazon).
- Restricciones de uso: El sistema apela a restricciones para poder evitar errores al correr el programa, como por ejemplo, la utilización de dos spinners(unidades y decimales) en vez de un textfield para el ingreso de un numero float.
- Legibilidad: El tamaño de la tipología configurada permite la fácil lectura.
- Gestión de errores: Frente al problema del ingreso del kilaje al agregar un artículo al carrito, primero optamos por una ventana emergente comunicando el error. Tras la devolución del profesor, optamos por cambiar a un texto en rojo, pequeño, que no corte con el curso del sistema pero sí que se note. La tercera y última opción fue, la realización de spinners controlando el ingreso de información.

Todo error es informado al usuario, como por ejemplo los siguientes casos



Manzana Verde

Proveedor: La Serenísima
Estados Unidos, Florida

\$59.0 x kg

Realize su compra: Inserte el peso: kg

Agrega al carrito



Manzana Verde

Proveedor: La Serenísima
Estados Unidos, Florida

\$59.0 x kg

Realize su compra: Inserte el peso: kg

Por favor ingrese un valor

Agrega al carrito

Por favor, seleccione número de local donde desea retirar su pedido:

Registre día de retiro:

n° del Local - Direccion

COMPRAR

Por favor, seleccione número de local donde desea retirar su pedido:

Registre día de retiro:

n° del Local - Direccion

COMPRAR

Seleccione un envase para cada artículo



Manzana Verde

Cantidad Seleccionada: 1.1 kg

Precio: \$ 64.9

Seleccione el envase a llevar

Sacar del Carrito

Por favor, seleccione número de local donde desea retirar su pedido:

Registre día de retiro:

n° del Local - Direccion

COMPRAR

La compra se ha realizado exitosamente

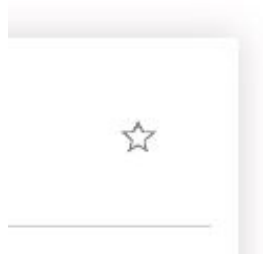
4. Evaluación de usabilidad

El programa está realizado teniendo en cuenta estándares de comunicación con el usuario. Como por ejemplo iconos internacionales de botones con ciertas funcionalidades(ej: la estrella de favorito, la lupa de buscar, el carrito de compras, etc). Orden de distribución de elementos.

Para una correcta evaluación de usabilidad, se utilizó la metodología de pruebas exploratorias. Le presentamos el programa a una mujer de 61 años de edad(Alicia Harispe, madre de Marcos Novelli), le leímos el resumen de este documento y le pedimos que realice diferentes acciones.

- Buscar todas las verduras:
 - Comentarios: "Muy sencillo."
 - Tiempo en realizar la acción: "3 segundos"
- Buscar la dirección del local número 2:
 - Comentarios: "Fácil."
 - Tiempo en realizar la acción: "5 segundos"
- Comprar 0.5kg de pistachos:
 - Comentarios: "Fácil pero si quiero comprar 0.75 kg no podría. Se entiende muy bien que se agregó al carrito de compras porque apareció el número uno en rojo. Muy buenas las imágenes de referencia de los envases. "
 - Tiempo en realizar la acción: "43 segundos"
- Descargar un ticket de compra:
 - Comentarios: "Muy fácil."
 - Tiempo en realizar la acción: "8 segundos"
- Agregar las bananas a favoritos y luego ver mi lista de favoritos:
 - Comentarios: "Muy fácil."
 - Tiempo en realizar la acción: "9 segundos"

Todas las acciones tienen una respuesta en la interfaz que permite confirmar la realización de las mismas. Como lo son, por ejemplo, los siguientes casos:



5. Pruebas funcionales

5.1. Técnicas de prueba aplicadas

5.2. Casos de prueba

5.3. Sesiones de ejecución de pruebas

6. Reporte de defectos

6.1. Definición de categorías de defectos

Definir las categorías de defectos que se van a usar en el reporte. Se debe definir en forma separada el tipo y la severidad del defecto.

- - Tipo: Dominio
 - Severidad: Alta
- - Tipo: Interfaz
 - Severidad: Alta
- - Tipo: Limitaciones de interfaz
 - Severidad: Media

6.1.1. Defectos encontrados por iteración

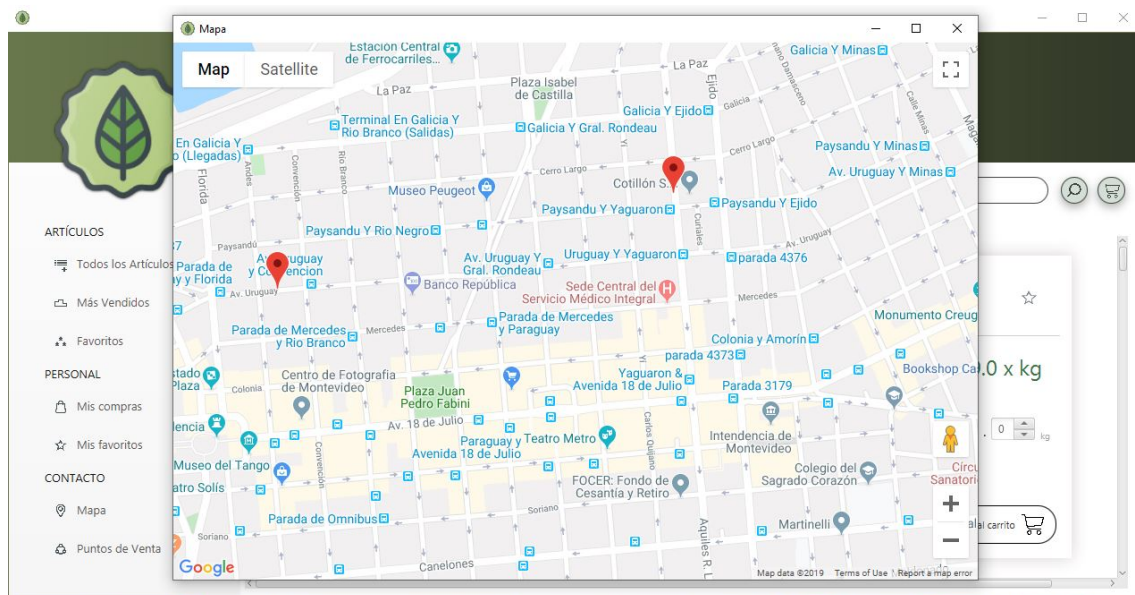
- Dominio:
 - Si estoy viendo los favoritos en el mundo y quiero buscar un producto, no funciona el buscador.
- Interfaz:
 - Si maximizo la pantalla, se notan errores en el GUI porque es un programa diseñado para una específica resolución.
 - Cuando estoy en mis favoritos y elimino un articulo de mis favoritos(des-seleccionola estrellita), no se me borra automaticamente en pantalla. Tengo que actualizar la pantalla para que desaparezca.
 - Si intento buscar un artículo que no existe en la lista de artículos, no me comunica la no existencia, solamente no me aparecen artículos en en pantalla.
- Limitaciones de Interfaz:
 - Los spinners para ingresar el kilaje del producto a comprar, solo permiten ingresar valores en gramos que sean múltiplos de 100.

6.2. Estado de calidad global

El programa actualmente muestra un correcto funcionamiento de todas las funcionalidades implementadas, no se encontraron errores al ser utilizado por periodos extensos por distintas personas (amigos, familia y desarrolladores(nosotros)). En cuanto a las funcionalidades implementadas, el software:

- Gestiona los artículos en venta, con un stock total de 12 artículos disponibles al usuario.
- Gestiona los envases aplicables a cada articulo, en la sección del "Carrito" donde el usuario puede encontrar todos los artículos que fueron seleccionados para comprar y decidir que envase llevara para cada uno de ellos.
- Gestiona los puntos de venta EcoShop, desplegando la información necesaria (departamento, calle, barrio, numero de local) para que el usuario encuentre el local ideal para el retiro de su compra. "online." comprar en el local mismo.
- Gestiona las Pre-Ventas realizadas por el usuario, mediante la sección del "Carrito" seleccionando la fecha de retiro de la compra de un calendario, el local en el cual va a realizar el retiro y los envases respectivos a cada articulo en el "Carrito".
- Ofrece la opción de descargar un ticket electrónico (PDF) con la información de la venta.
- Gestiona los artículos mas vendidos, ofreciendo una sección en la cual el usuario puede navegar entre los artículos ordenados decreciente, de mas vendidos a menos vendidos.
- Ofrece la opción de agregar artículos a favoritos, al igual que una sección donde se muestran los "top 4 items favoritos en el mundo.
- Ofrece la posibilidad de buscar artículos mediante una barra de búsqueda y un menú de opciones que contiene las siguientes categorías en las cuales se encuentran los artículos: Frutas, Verduras, Frutos Secos.
- Motiva al usuario a seguir utilizando la metodología propuesta por EcoShop para evitar el uso de bolsas, brindándole en la sección de puntos de venta un mensaje que contiene la cantidad estimada del uso de bolsas hasta el momento desde el inicio de EcoShop.

También se logro incluir, mediante el uso de JxBrowser y Google Maps Javascript API, un mapa interactivo con los puntos de venta como marcadores en el mapa. Esta funcionalidad se elimino de la entrega ya que con las librerías de JxBrowser el proyecto superaba los 40mb por lo que no se podía subir a gestión. En el .zip que contiene el proyecto de NetBeans, se incluye un index.html con el cual se puede visualizar el mapa mencionado anteriormente (este archivo era el que JxBrowser cargaba). A continuación se deja una imagen de lo mencionado anteriormente:



Comentarios: Cuando probamos en las computadoras del laboratorio el EcoShop-fat.jar no se ejecuta al hacerle doble click, sin embargo, utilizando la consola y colocándonos en el directorio del ejecutable para luego mediante el comando `java -jar EcoShop-fat.jar` ejecutarlo, el programa se ejecuta correctamente.

7. Reflexión

Con respecto al programa, se tomaron las decisiones/riesgos de utilizar JavaFx para realizar la interfaz gráfica del programa y la realización de una interfaz del dominio que luego sería implementada. Al tomar estas decisiones, tuvimos que aprender desde cero estos nuevos conceptos que no habían sido utilizados por ninguno de los integrantes hasta el momento. El proceso de aprendizaje inicial nos llevó entre una y dos semanas para luego estar cómodos utilizando JavaFx y implementando cada clase del dominio con su respectiva interfaz. Creemos que se tuvo un buen alcance, obteniendo una aplicación fácil de mantener gracias a la interfaz del dominio, que también fue documentada con los estándares de JavaDoc en todas las funciones, y una aplicación agradable a la vista, con una interfaz de usuario fácil de usar y por encima de lo que se podría conseguir con JavaSwing. Por otra parte, el uso de Git y GitHub nos ayudó a colaborar en conjunto en el proyecto de forma simultánea, y aprender esta herramienta la cual es usada a nivel mundial por la mayoría de las empresas. También el uso de JUnit enriqueció nuestros conocimientos, acercándonos a los conceptos/herramientas de Testing que no habíamos utilizados hasta el momento. En caso de continuar el desarrollo del programa nos gustaría seguir agregando funcionalidades, por ejemplo:

- Calcular cuantos envases serían necesarios dado el envase y el artículo. Este mismo se lograría a través del registro del volumen del envase y la densidad del artículo, la cual puede ser medida experimentalmente, con la que luego se podría despejar el volumen que ocuparía el artículo sabiendo su densidad y masa (cantidad que el usuario compra en kg). Finalmente utilizando un algoritmo que calcule cuantas veces $\frac{\text{envase}}{\text{artículo}}$ el volumen del artículo en el volumen del envase, se podría informar al usuario no solo que envases podría llevar, sino que también cuantos de cada tipo, así, poder tener la opción de llevar distinta combinación de envases dependiendo lo que el usuario disponga en el momento,
- Un sistema que gestione una recolección de envases por los locales de EcoShop, que luego estén disponibles a los usuarios que no disponen de envases en el momento.

Finalmente el proyecto nos dejó enseñanzas en cuanto al manejo del tiempo, el cual muchos libros de Ingeniería de Software tratan pero que se aprenden mayoritariamente con la experiencia. Con respecto a esto, nos hubiese gustado haber tenido más tiempo para desarrollar la documentación, la cual no se logró completar en su totalidad.

Bibliografía

- [1] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, eight ed. McGraw-Hill, 2014.
- [2] I. Sommerville, *Software Engineering*, 10th ed. Pearson, 2015.
- [3] Universidad ORT Uruguay. (2013) Documento 302 - Facultad de Ingeniería. [Online]. Available: <http://www.ort.edu.uy/fi/pdf/documento302facultaddeingenieria.pdf>