

Escuela Superior Politécnica del Litoral

Materia:
Programación Orientada a Objetos

Nombre Proyecto:
Proyecto1-Primer Parcial

Grupo: #2

- Integrantes:**
- **Nelson Alvarado**
 - **Diego Arteaga**
 - **Steeven Fierro**

Periodo:
2020 1T

Tabla de Contenidos

1. Antecedentes.....	3
2. Descripción.....	3
3. Diseño del Sistema.....	3
4. Discusión de la Solución.....	3
5. Evidencias de Participación.....	3
6. Evidencias de ejecución del programa.....	5
7. Referencias Bibliográficas.....	5

1. Antecedentes

- La venta de los productos de los agricultores evitando así intermediarios para su venta.
- Los proveedores que puedan operar con el sistema y realizar cambios.
- Los compradores o clientes que realizan consultas y efectúan la compra del producto con éxito.

2. Descripción

Crear un sistema que permita a los pequeños agricultores vender sus productos evitando cualquier tipo de inconveniente, en donde los proveedores que deseen vender sus productos a través de la plataforma deben primero registrarse en la misma. Una vez registrados pueden ingresar información de sus productos, realizar consultas de los pedidos realizados y gestionar el estado de los pedidos.

Los clientes que ingresan en la plataforma pueden consultar información de los productos que vivan en un radio de 50 km de distancia, realizar pedidos usando distintos medios de pago y consultar el estado de sus pedidos.

3. Diseño del Sistema

Casos de Uso de los usuarios del sistema y diagrama de casos de usos

Casos de uso:

- Proveedor se registra
- Proveedor registra su(s) productos
- Proveedor hace consulta y gestión de los pedidos
- Proveedor edita sus productos.
- Cliente se registra
- Cliente solo consulta información de productos
- Cliente agrega ítem al carrito
- Cliente elimina ítem de carrito
- Cliente realiza compra
- Cliente consulta el estado del pedido

Caso 1: Proveedor se registra

Actor: Proveedor

Precondiciones:

- Haber ingresado a la plataforma.
- Haber pasado por el menú.

Descripción:

- Ingresa su usuario, contraseña y otros datos.
- Se crea un objeto de la clase Proveedor con sus atributos.

Escenario 1.-

Proveedor ingresa a la plataforma y registra sus datos (nombre, número identificación, dirección, ubicación (coordenadas lat. y longitud), número contacto, correo electrónico), además de estos, se registra con un nombre de usuario único y la clave de acceso de su cuenta.

Escenario 2.-

Proveedor ingresa a la plataforma y registra sus datos, pero al momento de ingresar cierto dato, ingresa algo diferente a lo que se pide, por lo que la plataforma debe validar que se ingrese correctamente.

Caso 2: Proveedor registra sus productos

Actor: Proveedor

Precondiciones:

- El proveedor inicio sesión previamente en la plataforma
- Ventana activa del menú numérico.

Descripción:

- Se registra y agrega todos los productos solicitados por el proveedor.

➤ **Escenario 1.-**

Proveedor va a la opción de registrar productos, donde ingresa información de su producto (código, nombre, breve descripción, categoría(s) y precio).

➤ **Escenario 2.-**

Se muestra el menú numérico al proveedor y va a la opción de registrar productos, donde al momento de ingresar algo diferente de lo que se pide, por lo que la plataforma debe validar que se ingrese correctamente.

➤ **Escenario 3.-**

Proveedor no inicia sesión correctamente, por lo que no se le muestra el menú y se le pide que vuelva a ingresar las credenciales.

Caso 3: Proveedor consulta y gestiona los pedidos

Actores: Proveedor, base de datos de pedidos

Precondiciones:

- El proveedor inicio sesión previamente en la plataforma.
- Ventana de opciones del menú activa.
- Tener pedidos.

Descripción:

- Se muestra la información del registro sobre la gestión y disponibilidad del pedido.
- Cambiar el estado de disponibilidad (armando, despachado) del pedido.

➤ **Escenario 1.-**

Se muestra el menú numérico al proveedor y va a la opción de consultar y gestionar pedido. Se muestra el listado de todos los pedidos. Se ingresa el código del pedido del que se quiere validar el estado. Proveedor ingresa nuevo estado y se cambia el estado del pedido.

➤ **Escenario 2.-**

Proveedor va a la opción consultar y gestionar pedido. Se muestra la información (código, fecha, productos, datos del cliente, métodos de pago y el total del pedido).

Caso 4: Proveedor edita sus productos

Actores: Proveedor- Base de datos de productos

Precondiciones:

- El proveedor inicio sesión previamente en la plataforma.
- Ventana de opciones del menú numérico activa.
- Se muestra información de los datos del producto.

Descripción:

- Se pide ingresar el código del producto a editar.
- Se guarda el registro en el cual se editó la gestión y disponibilidad del producto.

➤ **Escenario 1.-**

Se muestra el menú numérico al proveedor y va a la opción de editar producto además se pide los campos por los que se desea filtrar y se muestra el listado de todos los productos que cumplen con el filtro.

➤ **Escenario 2.-**

Proveedor inicia sesión correctamente, se muestra el menú numérico y va a la opción de editar producto. Luego de haber filtrado sus productos, ingresa un código que no se encuentra por lo que se pide que se ingrese un código de producto válido.

- **Escenario 3.-**
Proveedor no inicia sesión correctamente, por lo que no se le muestra el menú y se le pide que vuelva a ingresar las credenciales.
- **Escenario 4.-**
Proveedor inicia sesión correctamente, pero después de mostrarse el menú, ingresa otra opción así que debe poder retroceder hacia el menú.

Caso 5: Cliente se registra

Actor: Comprador o cliente

Precondiciones:

- Haber ingresado a la plataforma.
- Haber pasado por el menú.

Descripción:

- Ingresa su usuario, contraseña y otros datos.
- Se crea un objeto de la clase Cliente con sus atributos.

- **Escenario 1.-**
Cliente ingresa a la plataforma y registra sus datos (nombre, número de identificación, dirección o (coordenadas lat. y longitud), correo electrónico y los datos de su forma de pago preferida. además de esto, se registra con un nombre de usuario único y la clave de acceso de su cuenta.
- **Escenario 2.-**
Cliente ingresa a la plataforma y registra sus datos, pero al momento de ingresar cierto dato, ingresa algo diferente a lo que se pide, por lo que la plataforma debe validar que se ingrese correctamente.

Caso 6: Cliente consulta productos del carrito

Actores: Cliente – Base de Datos de carrito - Sistema de pago

Precondiciones:

- Inicia sesión previamente.
- Ventana del menú numérico activa.
- Tener productos en el carrito.

Descripción:

- Se mostrarán la lista de los productos que se encuentren en el carrito.
- Se solicitará su método de pago (Tarjeta de crédito o PayPal).
- Realiza el pago y finaliza la compra.

- **Escenario 1.-**
El cliente inicia sesión correctamente, se muestra el menú numérico y va a la opción de consultar información de productos. Luego de haber filtrado sus productos, se muestran los artículos que coincidan con el criterio de búsqueda y se muestra su información. Luego se le solicita al usuario ingresar código de un producto o realizar otra búsqueda distinta.
- **Escenario 2.-**
El cliente inicia sesión correctamente, se muestra el menú numérico y va a la opción de consultar información de productos. Luego de haber filtrado sus productos, se muestran todos los artículos de los proveedores que se encuentran hasta 50 km de distancia. Luego se le solicita al usuario ingresar código de un producto o realizar otra búsqueda distinta. En caso de que se ingrese el código del producto, se solicitará agregarlo al carrito y se guardará la información del producto en el carrito de compra y regresará al usuario al menú principal.

Caso 7: Cliente agrega a carrito.

Actores: Cliente- Base de datos de productos- Base de datos de carritos

Precondiciones:

- Inicia sesión previamente.
- Ventana del menú activa.

Descripción:

- Al pedir la información del producto se le mostrará detalladamente todo acerca del mismo y además se agregará ese producto en el carrito de compras.
- Se guarda la información del producto y regresa al menú principal.

➤ **Escenario 1.-**

El cliente inicia sesión correctamente, se muestra el menú numérico y va a la opción de consultar información de productos. Luego de haber filtrado sus productos, se muestran los artículos que coincidan con el criterio de búsqueda y se muestra su información. Luego se le solicita al usuario ingresar código de un producto o realizar otra búsqueda distinta. En caso de que realice otra búsqueda, pues tendrá que volver a filtrar y posteriormente realizar el proceso para agregar el producto que termine escogiendo al carrito.

➤ **Escenario 2.-**

Al cliente se muestran todos los artículos de los proveedores que se encuentran hasta 50 km de distancia. Luego se le solicita al usuario ingresar código de un producto o realizar otra búsqueda distinta. En caso de que realice otra búsqueda, pues tendrá que volver a filtrar y posteriormente realizar el proceso para agregar el producto que termine escogiendo al carrito.

Caso 8: Cliente elimina ítem de carrito

Actores: Cliente – Base de Datos de carrito

Precondiciones:

- Inicia sesión previamente.
- Ventana del menú numérico activa.
- Tener productos en el carrito.

Descripción:

- Se mostrarán la lista de los productos que se encuentren en el carrito.
- El usuario tendrá la opción de eliminar o efectuar su compra del producto.
- Se actualizará la lista de los productos y se volverá a mostrar (en caso de que decide remover un producto).

➤ **Escenario 1.-**

El cliente inicia sesión correctamente, se muestra el menú numérico y va a la opción de consultar información de productos de carrito. Se muestra la lista de productos del carrito con los siguientes datos: código por producto, nombre de productos, cantidad, precio unitario, subtotal y total a pagar.

➤ **Escenario 2.-**

El cliente se le mostrará la opción eliminar producto, por lo que se deberá ingresar el código del producto en cuestión o efectuar la compra de los productos. En caso de que elimine un producto, se remueve el producto del carrito y se muestra los restantes.

➤ **Escenario 3.-**

El cliente inicia sesión correctamente, se muestra el menú numérico y va a la opción de consultar información de productos de carrito. Se muestra la lista de productos del carrito con los siguientes datos: código por producto, nombre de productos, cantidad, precio unitario, subtotal y total a pagar. Luego de eso, se da la opción de si se desea eliminar un producto para lo que se debe ingresar el código del producto en cuestión o efectuar la compra de los productos.

➤ **Escenario 4.-**

El cliente no inicia sesión correctamente, por lo que no se le muestra el menú y se le pide que vuelva a ingresar las credenciales.

Caso 9: Cliente consulta el estado del pedido

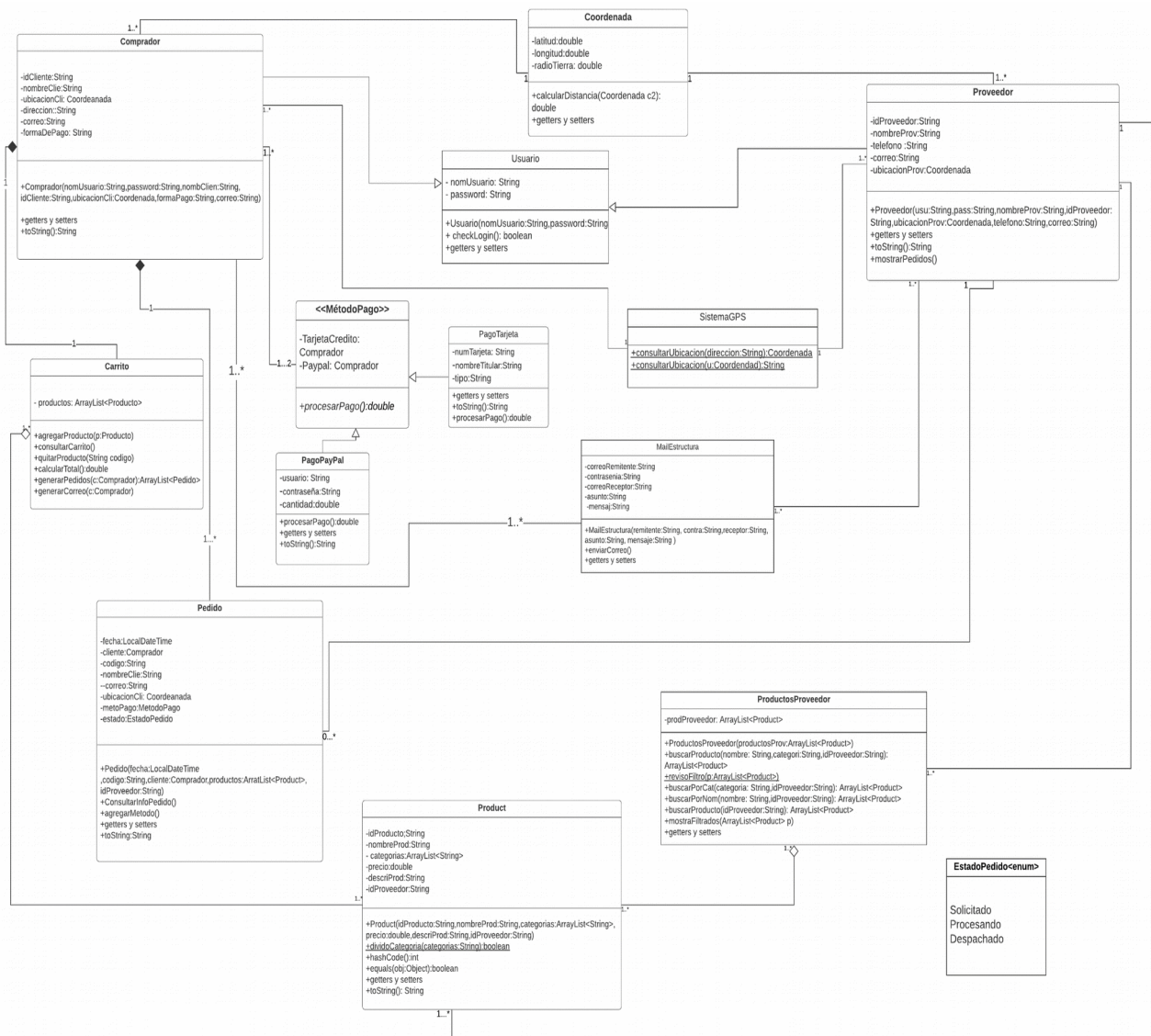
Precondiciones:

- Inicia sesión previamente.
- Ventana del menú numérico activa.
- Haber hecho un pedido.

Descripción:

- El cliente elige la opción que corresponde al estado del pedido.
- El cliente ve el estado del mismo.

- **Escenario 1.-**
El cliente no inicia sesión correctamente, por lo que no se le muestra el menú y se le pide que vuelva a ingresar las credenciales
 - **Escenario 2.-**
En caso de que el cliente realice la compra, se pedirá la forma de pago: tarjeta de crédito o PayPal. En caso de que escoja pagar con tarjeta de crédito, se pedirá que registre el tipo de tarjeta (visa, diners, etc.), número de la tarjeta y nombre del titular de la misma. Si el pago se realizó con éxito, la compra finaliza y se genera un pedido con un código único para cada proveedor y el sistema envía un correo al cliente con la información de su compra.
 - **Escenario 3.-**
En caso de que escoja PayPal, se pide registrar el nombre de usuario y contraseña de la cuenta PayPal. Si el pago se realizó con éxito, la compra finaliza y se genera un pedido con un código único para cada proveedor, también cualquier pedido que se realice tendrá el estado inicial "Procesando". Además de eso, el sistema envía un correo al cliente con la información de su compra.
 - **Escenario 4.-**
Cliente inicia sesión correctamente, pero después de mostrarse el menú, ingresa otra opción así que debe poder retroceder hacia el menú principal.
- Diagrama de clases



- Breve descripción de la responsabilidad de las principales clases.

Paquete	Clases	Responsabilidad
AppTienda	Carrito, Coordenada, EstadoPedido Generador, Pedido, Product, ProductosProveedor SistemGPS	Registra la información para cada objeto de la clase respectiva para poder hacer uso de los datos en el sistema e interactuar unos con otros para resolver el problema.
AppTiendaComp	Comprador, DataCharger, Proveedor, Usuario	Registrar el inicio de sesión de los compradores y proveedores con sus respectivos datos y se procederá a almacenar la información solicitada por el programa la cual esos datos servirán para el proceso de compra del pedido.
Email	MailEstructura	La clase mailEstrucuta nos ayuda a crear una estructura de código en la cual podremos enviar correos a los

		proveedores y compradores de una manera eficaz.
MetodosPago	MetodoPago, PagoPaypal, PagoTarjeta	Registra los métodos de pagos que se vayan a seleccionar en el transcurso del programa.
Testeos	SistemaTiendapp	Cumple con la funcionalidad de interactuar con el usuario y llamar los métodos de las clases anteriores para proceder a pedir la información solicitada del programa para mostrar la solución de este problema.

4. Discusión de la Solución

Realice un análisis de calidad de su software con respecto a los siguientes criterios:

- Funcionalidad
- Flexibilidad y extensibilidad,
- Mantenibilidad
- Usabilidad
- Portabilidad

1.-Funcionalidad

El programa que se realizó en este parcial es sobre la manera de realizar las ventas al cliente e interactuar de forma funcional y eficaz. En tanto a lo solicitado en el proyecto cumple con cada uno de la información propuestas ya que aplicamos los conceptos de herencia, polimorfismo, encapsulamiento, métodos y clases respectivas para cada actor del problema y gracias a esos conceptos podemos hacer uso de pequeños bloques de código para que así el problema sea aún más sencillo de resolver ya que cada clase y métodos tienen sus respectivas funciones en el programa la cual al final se llaman dichos métodos a la clase main o test la cual mostrara la interacción que tiene el usuario con la plataforma y además se muestra una gama de opciones para que la interacción sea de forma agradable.

2.- Flexibilidad y extensibilidad

El programa es flexible porque pueden agregarse una gama alta de datos la cual el software lo trabajará sin problema sin necesidad de dañar o hacer que no se ejecute el programa en la plataforma, los métodos o clases dispuestas para poder realizar este tipo de casos serán los más relevantes del programa que son ProductProveedor, Comprador, Proveedor, Carrito, etc. Su extensibilidad se basará en las acciones e interacciones que tengan estos métodos para desernir el problema y que los requisitos cambiantes como ejemplo la manera de agregar el Geocoding para saber la ubicaciones o direcciones respectivas del comprador o proveedor desee ingresar respecto al satélite del mundo.

El email que es fundamental para poner en práctica los conceptos un poco más avanzados de cómo es la funcionalidad de enviar y notificar correos sobre las compras de los pedidos que se hizo en el programa.

3.- Mantenibilidad

En este caso mantener el problema sin errores es un tanto difícil realizar modificaciones pequeñas en el sistema para corregir algunos errores que pueden ser de sintaxis o al momento de posicionar o no saber la referencia de dicho objeto. Traerá problemas en la codificación del programa la cual habría que ir codificando y justificando por las partes donde seán Trabajano los métodos, una de las herramientas más conocidas es el uso del javadoc del programa, además que es una buena practica poder revisar el javadoc.

4.- Usabilidad

A pesar de cumple todas las funcionalidades y puede manipularse las veces que sea necesaria, el software se puede simplificar aun mas teniendo en cuenta la practica y la experiencia de los desarrolladores que podría llevar a cabo un código de mejor excelencia y mas satisfactoria lanzado como una app.

5.- Portabilidad

El software puede usarse libremente en cualquier plataforma como sea posible, quizás no en diferente contexto ya que habría que editar el Geoocoding para que este analizará varias ubicaciones en distintos países cosa que para este proyecto solo se dejo para el país natal para que así nos de unas coordenadas mas exactas y el email que tiene una pequeña restricción en tanto a la privacidad de la cuenta de Google.

5. Evidencias de Participación

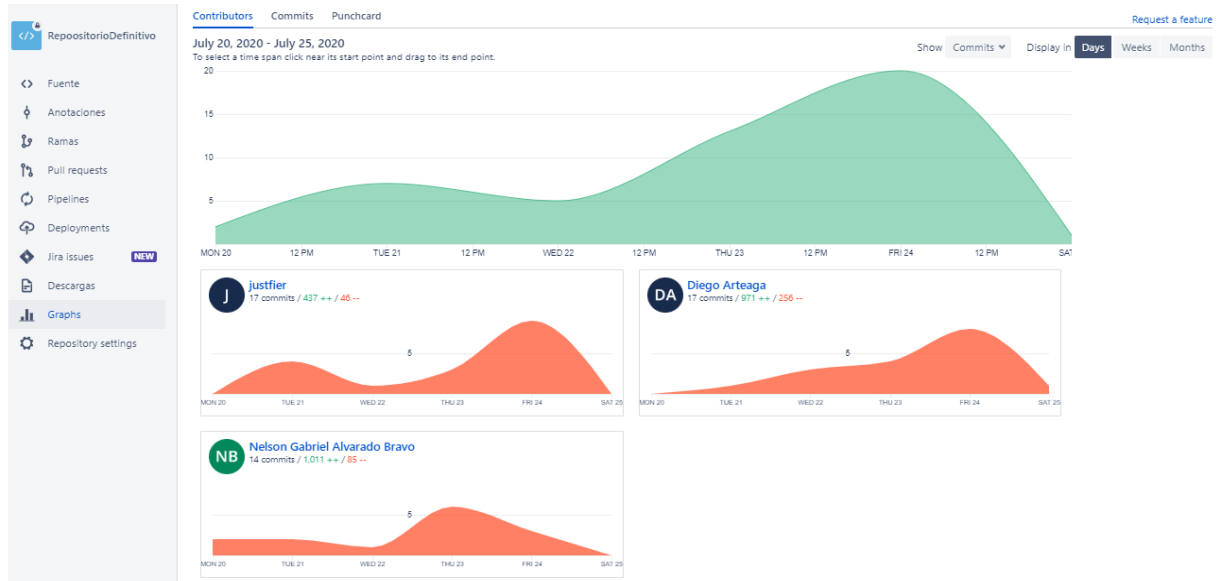
Distribución de Tareas

Nombre	Tarea
Diego Arteaga	SistemaTiendaApp, Usuario, Proveedor, DataCharger, Carrito, ProductosProveedor, Coordenada, MetodosPago
Nelson Alvarado	GeoCooding, SistemaTiendaApp, Generador, Comprador, Pedido, PagoPaypal
Steeven Fierro	Main, Email, ProductosProveedor, PagoTarjeta, SistemaTiendaApp

Evidencias de contribución en el repositorio

- Link del repositorio
<https://bitbucket.org/Koalaspero/repoositoriodefinitivo/src/master/>

Gráfico de Contribución



Commits de los integrantes

	Autor	Registro	Mensaje	Fecha
	Nelson Gabriel Alvarado B	6f4dfc7	SistemaTiendapp.java editado a través de la web en Bitbucket	hace 27 segundos
	Diego Arteaga	c032191	README.txt created online with Bitbucket	hace 7 horas
	Nelson Gabriel Alvarado B	f3ad52d	PagoPaypal.java editado a través de la web en Bitbucket	hace 11 horas
	justfier	2d6e3cb	PagoTarjeta.java editada la clase	hace 11 horas
	justfier	140988b	SistemaTiendapp.java editado el menu de comprador y el de iniciar sesion	hace 12 horas
	justfier	7245823	DataCharger.java agregado el metodo buscarComprador	hace 12 horas
	justfier	f0b09dc	SistemaTiendapp.java agregados metodos para el menu comprador	hace 12 horas
	justfier	81a07c7	PagoPaypal.java modificado el metodo de void a double	hace 12 horas
	Diego Arteaga	d375437	DataCharger.java edited online with Bitbucket	hace 12 horas
	Nelson Gabriel Alvarado B	5d1a316	SistemaTiendapp.java editado a través de la web en Bitbucket	hace 14 horas
	justfier	304854e	DataCharger.java corregido la firma del metodo buscarProdCatNombCil	hace 16 horas
	justfier	b30f059	SistemaTiendapp.java Modificado el metodo opcionesProducto	hace 16 horas
	justfier	a03002b	DataCharger.java modificado el método ubiCincuenta	hace 16 horas
	justfier	375f6dd	ProductosProveedor.java agregado metodos a la clase	hace 16 horas
	Diego Arteaga	cf336f0	DataCharger.java edited online with Bitbucket	hace 18 horas
	Diego Arteaga	f40a843	SistemaTiendapp.java edited online with Bitbucket	hace 18 horas
	Diego Arteaga	15eba5f	SistemaTiendapp.java edited online with Bitbucket	hace 23 horas
	Diego Arteaga	752bc46	Carrito.java edited online with Bitbucket	ayer
	Diego Arteaga	2fb02db	Pedido.java edited online with Bitbucket	ayer
	Diego Arteaga	bf44acd	Pedido.java edited online with Bitbucket	ayer

	Autor	Registro	Mensaje	Fecha
	justfier	b94b04a	PagoTarjeta.java editado el método procesarPago	anteayer
	justfier	70cc65f	MetodoPago.java editado el método de "void" a "double"	anteayer
	justfier	8c1b0e8	MetodoPago.java editado el método abstracto de "void" a "double"	anteayer
	Diego Arteaga	b25140c	DataCharger.java edited online with Bitbucket	anteayer
	Diego Arteaga	c718357	SistemaTiendapp.java edited online with Bitbucket	anteayer
	Nelson Gabriel Alvarado B	f984adb	.gitignore edited online with Bitbucket	anteayer
	Nelson Gabriel Alvarado B	84d92f6	javax.mail-api-1.6.2.jar edited online with Bitbucket	anteayer
	Nelson Gabriel Alvarado B	aF20cfd		anteayer
	Nelson Gabriel Alvarado B	8b9835e	src/main/java/Testeos/SistemaTiendapp edited online with Bitbucket	anteayer
	Nelson Gabriel Alvarado B	4f92039	src/main/java/AppTiendaComp/DataCharger edited online with Bitbucket	anteayer
	Diego Arteaga	af026d9	SistemaTiendapp edited online with Bitbucket	hace 3 días
	Diego Arteaga	99cec00	SistemaTiendapp created online with Bitbucket	hace 3 días
	Diego Arteaga	71096ff	DataCharger created online with Bitbucket	hace 3 días
	Nelson Gabriel Alvarado B	4888aec	Carrito.java editado a través de la web en Bitbucket	hace 3 días
	justfier	1220484	Carrito.java edited Creado el método generarCorreo el cual envía un correo al Comprador con los datos de su compra	hace 3 días
	justfier	5a50008	Pedido.java anaidido el parametro proveedor al constructor	hace 4 días
	Diego Arteaga	d86b58d	Carrito.java edited online with Bitbucket	hace 4 días
	justfier	3842620	EstadoPedido.java editada la variable "procesado" a "procesando"	hace 4 días
	justfier	3329c11	TestCorreo.java creada esta clase para probar que el envío de correo funcione perfectamente	hace 4 días
	justfier	e68f167	MailEstructura.java creada la clase para poder enviar correos	hace 4 días

Diego Arteaga / ProyectoTienda / RepositorioDefinitivo / Commits

Commit

Nelson Gabriel Alvarado Bravo committed 4888aec
3 days ago

Carrito.java editado a través de la web en Bitbucket

[View source](#) [Approve](#)



1220484

master

No tags

[Go to pull request](#)

[View raw commit](#)

[Watch this commit](#)

[Reports](#)

[Run pipeline](#)

Commit

 **Nelson Gabriel Alvarado Bravo** committed 251337b yesterday

SistemaTienapp.java editado a través de la web en Bitbucket

[View source](#) [Approve](#)



a509437

master

No tags 

 Go to pull request


 View raw commit

 Watch this commit

 Reports

 Run pipeline

Commit

 **justfier** committed 140988b 11 hours ago

SistemaTienapp.java editado el menu de comprador y el de iniciar sesion

[View source](#) [Approve](#)



7245823

master

No tags 

 Go to pull request

 View raw commit

 Stop watching

 Reports

 Run pipeline

Commit

 **justfier** committed e68f167 4 days ago

MailEstructura.java creada la clase para poder enviar correos

[View source](#) [Approve](#)



51e1927

master

No tags 

 Go to pull request

 View raw commit

 Stop watching

 Reports

 Run pipeline

Commit

 **Diego Arteaga** committed d375437 11 hours ago

DataCharger.java edited online with Bitbucket

[View source](#) [Approve](#)



5d1a316

master

No tags 

 Go to pull request

 View raw commit

 Watch this commit

 Reports

 Run pipeline

Commit

DA Diego Arteaga committed c832191
7 hours ago

README.txt created online with Bitbucket

[View source](#) [Approve](#)

DA

f3ad52d

master

No tags

[Go to pull request](#)

[View raw commit](#)

[Watch this commit](#)

[Reports](#)

[Run pipeline](#)

6. Evidencias de ejecución del programa

Capturas de pantalla del programa en ejecución de las principales funciones

- Registro de Compradores y Vendedores

```
ProyectoTiendaDef - Apache NetBeans IDE 11.3
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
SistemaTiendaApp.java Output - Run (SistemaTiendaApp) X
***** Bienvenido al menu principal *****
1. Registro
2. Iniciar Sesión
3. Salir
Ingrese una opcion: 1
----- Bienvenido al menu de registro -----
1. Cliente
2. Proveedor
3. Regresar
Ingrese una opcion de Registro : 1
Ingrese sus datos
Nombre (1er nombre y apellido):
Nelson A
ID:
12345
-----Ingrese su direccion-----
¿Desea Ingresar Coordenada o una direccion?:
direccion
Ingresa direccion separada por coma (Lugar,ciudad) :Lago espol,Guayaquil
log4j:WARN No appenders could be found for logger (com.byteowl.s.jopencage.JOpenCageGeocoder).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Correo: nelson@gmail.com
Ingrese su forma de pago preferida: paypal
Username: nel
Password: nel
cliente { idComprador:053-58-5053, nombComprador:Rocio Mera, ubicacion:( -2.050644 , -79.917811 ), correo:remera@espol.edu.ec, formaPago:Tarjeta}
cliente { idComprador:12345, nombComprador:Nelson A, ubicacion:( -2.1471432 , -79.9610621 ), correo:nelson@gmail.com, formaPago:paypal}
```

```
SistemaTiendapp.java  Output - Run (SistemaTiendapp) X
1. Registrar
2. Iniciar Sesión
3. Salir

Ingrese una opcion: 1

----- Bienvenido al menu de registro -----
1. Cliente
2. Proveedor
3. Regresar
Ingrese una opcion de Registro : 2

Ingrese sus datos
Nombre (ler nombre y apellido):
Maria Suarez
ID:
12345
----- Ingrese su direccion -----
¿Desea Ingresar Coordenada o una direccion?:
direccion
Ingrese direccion separada por coma (Lugar,ciudad) :Lago Espol,Guayaquil
log4j:WARN No appenders could be found for logger (com.byteowl.jopenCageGeocoder).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Correo: maria@gmail.com
Numero de contacto: 0982421232
Username: maria
Password: maria
Proveedores(idProveedor=412-61-7508, nombreProv=Roberto Carlos, ubicacionProv=( -2.047585 , -79.913632 ), telefono=0987654321, correoProv=Thuel942@jourrapide.c
Proveedores(idProveedor=226-07-0765, nombreProv=Ronaldo Cristiano, ubicacionProv=( -2.0388 , -79.891752 ), telefono=0987654329, correoProv=Messil939@cuvovx.de)
Proveedores(idProveedor=12345, nombreProv=Maria Suarez, ubicacionProv=( -2.1471432 , -79.9610621 ), telefono=0982421232, correoProv=maria@gmail.com)
```

- Registro de un producto

```
ProyectoTiendDef - Apache NetBeans IDE 11.3
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config> Search (Ctrl F)

Output - Run (SistemaTiendapp)
Building ProyectoTiendDef 1.0-SNAPSHOT
----- exec-maven-plugin:1.5.0:exec (default-cli) @ ProyectoTiendDef -----
Bienvenido
Ingrese una opcion
1. Registrar
2. Iniciar Sesión
3. Salir
2
Escoger:
1. Iniciar Sesión como Proveedor
2. Iniciar Sesión como Comprador
3. Regresar
1
Iniciar Sesión
Usuario:
Roberto234
Contraseña:
12345
Inicio Sesión Corecamonta
Bienvenido al menu de proveedor
1. Registro de Productos
2. Consultar informacion de pedido
3. Gestionar estado de pedido
4. Consulta y edición de productos
5. Salir
1
Registre su producto
Ingrese el id del producto:
940294892
Ingrese el nombre:
yogurt
Descripción del producto:
murtivo
Categoría(s):
Lacteos
Precio:
1.00
Bienvenido al menu de proveedor
```

- Agregar productos al carrito

```
ProyectoTiendaDef - Apache NetBeans IDE 11.3
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Output - Run (SistemaTiendaapp)
4. Consulta y edición de productos
5. Salir
2
No tienes pedidosBienvenido al menu de proveedor
1. Registro de Productos
2. Consultar informacion de pedido
3. Gestionar estado de pedido
4. Consulta y edición de productos
5. Salir
4
¿Desea consultar o editar sus productos?
1. Consultar
2. Editar
3.Regresar
1
¿Desea filtrar? (S/N)
S
Codigo: Uoa94oHG ,Precio: 5.0 ,Nombre: CostillaSimple ,Categoria(s): [Carnicos]Codigo: qWUbgCKx ,Precio: 3.0 ,Nombre: Brocoli ,Categoria(s): [Vegetales]Codigo: ann3mWw ,Precio: 2.5 ,Nombre: Lechuga ,Categoria(s): [Vegetales]C
Nombre: CostillaSimple
Categorias: [Carnicos]
Precio: 5.0
Descripcion: Esta costilla proviene de la parte de la res llamada costillar-falda
Proveedor: 412-61-7508
, Id: qWUbgCKx
Nombre: Brocoli
Categorias: [Vegetales]
Precio: 3.0
Descripcion: El brócoli posee un alto contenido de vitamina C y es rico en fibras que ayudan a mejorar significativamente el tracto digestivo.
Proveedor: 412-61-7508
, Id: ann3mWw
Nombre: Lechuga
Categorias: [Vegetales]
Precio: 2.5
Descripcion: La lechuga es la protagonista del plato más ligero y sencillo del secretario: la ensalada.
Proveedor: 412-61-7508
, Id: 9AUZ0p92S
Nombre: yogurt
Categorias: [Lacteos]
Precio: 3.0
Descripcion: murciago
Proveedor: 412-61-7508
}Bienvenido al menu de proveedor
<
```

```
ProyectoTiendaDef - Apache NetBeans IDE 11.3
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Output - Run (SistemaTiendaapp)
---Bienvenido a Tiendaapp---
¿Desea consultar sus productos? (S/N)
S
¿Desea filtrar sus productos? (S/N)
S
Codigo: Uoa94oHG ,Precio: 5.0 ,Nombre: CostillaSimple ,Categoria(s): [Carnicos]Codigo: qWUbgCKx ,Precio: 3.0 ,Nombre: Brocoli ,Categoria(s): [Vegetales]Codigo: ann3mWw ,Precio: 2.5 ,Nombre: Lechuga ,Categoria(s): [Vegetales]C
opcion 1
opcion 2
opcion 3: Salir
1
Ingrese el codigo del producto.
Uoa94oHG
Id: Uoa94oHG
Nombre: CostillaSimple
Categorias: [Carnicos]
Precio: 5.0
Descripcion: Esta costilla proviene de la parte de la res llamada costillar-falda
Proveedor: 412-61-7508
¿Desea agregar al carrito? (S/N)
S
Ingrese la cantidad de productos para agregar0
Bienvenido al menu de comprador
1. Consulta y venta de productos
2. Ir al Carrito
3. Consultar Carrito
4. Salir
3
Estos son los productos que tiene actualmente en su carrito
-----
Uoa94oHG CostillaSimple 1 5.0 5.0
Total a pagar: 5.0¿Que desea hacer?
1. Quitar Producto
2. Proceder a Compra
Ingrese una opcion: 2
¿Como desea realizar el pago?
1. Tarjeta de Credito
2. PayPal
Ingrese una opcion: 2
Ingrese su usuario de PayPal: chichico
Ingrese su contraseña:
enohufe
<
```

- Efectuar una compra


```

ProyectoTiendaDef - Apache NetBeans IDE 11.3
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Output - Run (SistemaTiendaApp)
¿Desea filtrar sus productos? (S/N)
S
Codigo: Uoa94oNG, Precio: 5.0, Nombre: CostillaSimple, Categoria(s): [Carnicos]Codigo: qWUqCXs, Precio: 3.0, Nombre: Broccoli, Categoria(s): [Vegetales]Codigo: amh3mKw, Precio: 2.5, Nombre: Lechuga, Categoria(s): [Vegetales]
opcion 1
opcion 2
opcion 3: Salir
1
Ingrese el código del producto.
Codigo:
Id: Uoa94oNG
Nombre: CostillaSimple
Categorias: [Carnicos]
Precio: 5.0
Descripcion: Esta costilla proviene de la parte de la res llamada costillar-falda
Proveedor: 412-61-7508
¿Desea agregar al carrito? (S/N)
S
Ingrese la cantidad de productos para agregar
Bienvenido al menu de comprador
1. Consulta y venta de productos
2. Ir al Carrito
3. Consultar Carrito
4. Salir
2
Estos son los productos que tiene actualmente en su carrito
-----
Uoa94oNG CostillaSimple 1 5.0 5.0
Total a pagar: 5.0¿Que desea hacer?
1. Quitar Producto
2. Procceder a Compra
Ingrese una opcion: 2
¿Como desea realizar el Pago?
1. Tarjeta de Credito
2. PayPal
Ingrese una opcion: 2
Ingrese su usuario de PayPal: chichico
Ingrese su contraseña:
enchufe
El mensaje con destino al correo remera@espol.edu.ec se ha enviado con exito.
Su compra se ha generado exitosamente.
<
Run (SistemaTiendaApp) 459:31 INS

```

- Consulta de pedido por parte del comprador

```

Bienvenido al menu de comprador
1. Consulta y venta de productos
2. Ir al Carrito
3. Consultar Carrito
4. Salir
3
Estos son sus pedidos actualmente
vs18DQj2C 2020-07-25T10:26:43.719765400 [Uoa94oNG CostillaSimple 1] [Forkoala (-2.050644 , -79.917811 ) remera@espol.edu.ec ]Tarjeta 5.0Bienvenido al menu de comprador
1. Consulta y venta de productos
2. Ir al Carrito
3. Consultar Carrito
4. Salir

```

- Cambio de estado

```

ProyectoTiendaDef - Apache NetBeans IDE 11.3
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Output - Run (SistemaTiendaApp)
3. Consultar Carrito
4. Salir
4
Gracias por visitarnos
Bienvenido
Ingrese una opcion
1. Registro
2. Iniciar Sesión
3. Salir
2
Escooper
1. Iniciar Sesión como Proveedor
2. Iniciar Sesión como Comprador
3. Registrar
1
Iniciar Sesión
Usuario:
Roberto1234
Contraseña:
12345
Inicio Sesión Correctamente
Bienvenido al menu de proveedor
1. Registro de Productos
2. Consultar informacion de pedido
3. Gestionar estado de pedido
4. Consulta y edicion de productos
5. Salir
5
Bienvenido a la gestion de tus pedidos
vs18DQj2C 2020-07-25T10:26:43.719765400 [Uoa94oNG CostillaSimple 1] [Forkoala (-2.050644 , -79.917811 ) remera@espol.edu.ec ]Tarjeta 5.0¿Desea editar el estado de algun pedido? (S/N)
S
Ingrese el código del pedido a editar: vs18DQj2C
¿A que estado desea cambiarlo?
Procesando
El estado del pedido fue cambiado a Procesando: 2020-07-25T10:31:43.719765400
Bienvenido al menu de proveedor
1. Registro de Productos
2. Consultar informacion de pedido
3. Gestionar estado de pedido
4. Consulta y edicion de productos
5. Salir
5
<
Run (SistemaTiendaApp) 459:31 INS

```

ANEXO

Característica de calidad de software

Funcionalidad

Cumple con todos los requerimientos

Flexibilidad y Extensibilidad

La flexibilidad es la capacidad del software para agregar / modificar / quitar funcionalidad sin dañar el sistema actual. La extensibilidad es la capacidad del software para agregar funcionalidad sin dañar el sistema, por lo que puede ser pensado como un subconjunto de flexibilidad. Esos cambios de funcionalidad pueden ocurrir según los requisitos cambiantes, o una obligación si el proceso de desarrollo es uno de los métodos iterativos. El cambio es inevitable en el desarrollo de software y por lo tanto, esta es una de las propiedades más importantes de software de calidad.

Mantenibilidad y legibilidad

La capacidad de mantenimiento es similar a la flexibilidad, pero se centra en las modificaciones sobre las correcciones de errores y modificaciones de funciones menores, y no de extensiones funcionales importantes. Puede ser soportado con una definición de interfaz útil, documentación y también documentación de código y / o código autodocumentado. Cuanto más adecuada y útil sea la documentación, más fácil se podrá realizar el mantenimiento.

Rendimiento y Eficiencia

El rendimiento es principalmente el tiempo de respuesta del software. Este tiempo de respuesta debe estar en intervalos aceptables (por ejemplo, unos pocos segundos), y no debería aumentar si aumenta el recuento de transacciones. Y también se debe considerar que los recursos son costes. Por lo que, la eficiencia debe ser apoyada con la utilización apropiada de los recursos. Como ejemplo exagerado, la capacidad de realizar una función simple sólo utilizando una máquina de procesador de 32 o 1 TB de espacio en disco no es aceptable. La relación fuente / rendimiento óptimo debe ser controlada.

Usabilidad y accesibilidad

Las interfaces de usuario son las únicas partes visibles del software según el punto de vista del usuario. Por lo tanto, la simplicidad, tomar menos tiempo para completar un trabajo, aprendizaje rápido para el usuario, etc son aspectos muy importantes a considerar. El principio más conocido de esta propiedad es KISS (Keep It Simple Stupid). Simple es siempre lo mejor. Un software utilizable también debe soportar diferentes tipos de control de accesibilidad para personas con discapacidades.

Compatibilidad y portabilidad de la plataforma

Un software de calidad debe funcionar en tantas plataformas como sea posible. Por lo tanto, más personas pueden hacer uso de ella. En

diferentes contextos podemos mencionar diferentes plataformas, esto puede ser plataformas de SO, tipos de navegador, etc. Y la portabilidad trata sobre la adaptación de software que puede ejecutarse en diferentes plataformas, por ser más compatible con la plataforma.

Fuente: <http://inlogiq.com/pautas-considerar-desarrollo-del-software/>