



UNIVERSIDAD RICARDO PALMA
FACULTAD DE INGENIERÍA
ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA

eBAY SIMULATOR

Gómez López, César
cesargomez1503@hotmail.com

García Garay, Diego
diego_jg_g95@outlook.com

Villar Gavidia, Angello David
angelvg11@hotmail.com

ABSTRACT

The present project aims to develop the sales system in which users rectify the new technologies in the means of payment and design of web pages for the various cross-platform web that exists today. It also has a payment integration with credit card, which is associated with the project. The project developed in the course of Data structures and algorithms of the semester 2016-II seeks to satisfy the demands of the real web user, in the simplest and easiest way possible.

RESUMEN

El proyecto desarrollado en el curso de Estructuras de Datos y Algoritmos del semestre 2016-II busca satisfacer las exigencias del usuario web actual, de la forma más simple y fácil posible.

El presente proyecto tiene como finalidad utilizar los conocimientos de estructura de datos para desarrollar un sistema de ventas en donde los usuarios podrán utilizar las nuevas tecnologías en medios de pago y diseño de páginas web para las diversas multiplataforma webs que existen en la actualidad. Además, se tiene una integración de pago con tarjeta de crédito, la cual está asociada al proyecto.

1 INTRODUCCIÓN

EBay es un sitio web en el que distintas personas y empresas compran y venden en línea una amplia variedad de productos, los usuarios deben sentirse cómodos al usar la página web, por ello los algoritmos que hay detrás se crearon de manera efectiva para generar facilidad en todas y cada una de las secciones que posee el proyecto. Los métodos y clases se hicieron para ayudar a que la interfaz gráfica pueda ser muy intuitiva y útil para cualquier tipo de usuario.

2 PRESENTACION DEL PROBLEMA

Un vendedor debe:

- 1) Tener la posibilidad de especificar los datos para realizar completamente la venta de sus artículos que registre para la venta al primer postor. Los datos deben especificarse jerárquicamente mediante una estructura de árbol.
- 2) Procesar la venta sin errores para mantener la confianza de sus usuarios.
- 3) Manejar el envío de los artículos, lo cual siempre será vía aérea.
- 4) Generar reportes de los datos de todos sus artículos. Cada artículo debe tener todos los datos necesarios, además de estados: vendido, en venta, enviándose al comprador.

Un comprador debe:

- 1) Encontrar fácilmente los artículos que busca para la compra inmediata.

- 2) Tener la posibilidad de incluir todos los artículos que quiera en cada compra.
- 3) Tener registrado en la tienda su tarjeta de crédito o débito para la transferencia de dinero hacia el comprador.
- 4) Tener la posibilidad de crear una lista de artículos de preferencia.
- 5) Mantener entre sesiones todos los datos personales o aquellos que se hayan creado por la interacción con la tienda.

El programa:

- 1) Al iniciar el programa, todos los datos van a la memoria principal.
- 2) Al cerrar el programa, todos los datos se guardan en archivos.
- 3) Al reiniciar el programa, todos los datos van otra vez a la memoria y quedan como si no se hubiera cerrado el programa con anterioridad.

3) DESCRIPCIÓN DE LA SOLUCIÓN

Para solventar las necesidades del proyecto se utilizaron diversas estructuras de datos aprendidas a lo largo del curso.

La solución del problema se divide en diferentes puntos:

- Debido a que solo se utilizó Java para realizar cualquier función, la base de datos fue hecha a base de archivos planos guardados dentro del proyecto, los cuales se recuperan al momento de iniciar el programa para mantener la persistencia de datos.
- Los datos de inicio de sesión se guardaron en objetos de una clase llamada 'Usuario' que contiene todos los datos necesarios del usuario y posteriormente estos objetos de dicha clase se guardaron en una tabla de dispersión que es identificada con una llave con el correo de cada persona.
- La relación de productos vendidos, comprados, favoritos y del carrito de compra

se guardaron en una lista de tipo secuencial que se insertó dentro de una tabla de dispersión referenciada a cada usuario con su correo correspondiente.

- Los productos en venta y las categorías se ordenaron jerárquicamente en una estructura de datos de tipo árbol N-rio. Esto permitió que pudiéramos añadir productos y/o categorías sin ningún problema al modificar el árbol.
- Los algoritmos usados fueron creados de manera autentica usando los conocimientos aprendidos en el curso.

4) IMPORTANCIA DEL PROYECTO

La página eBay, puede ser de gran ayuda para la aplicación de las estructura de datos, debido a la complejidad del sistema. Es muy importante para darnos cuenta sobre los conocimientos adquiridos, pues el proyecto nos permite implementar todo lo aprendido a lo largo de este ciclo, y a investigar datos extras para lograr nuestro objetivo; el funcionamiento del programa.

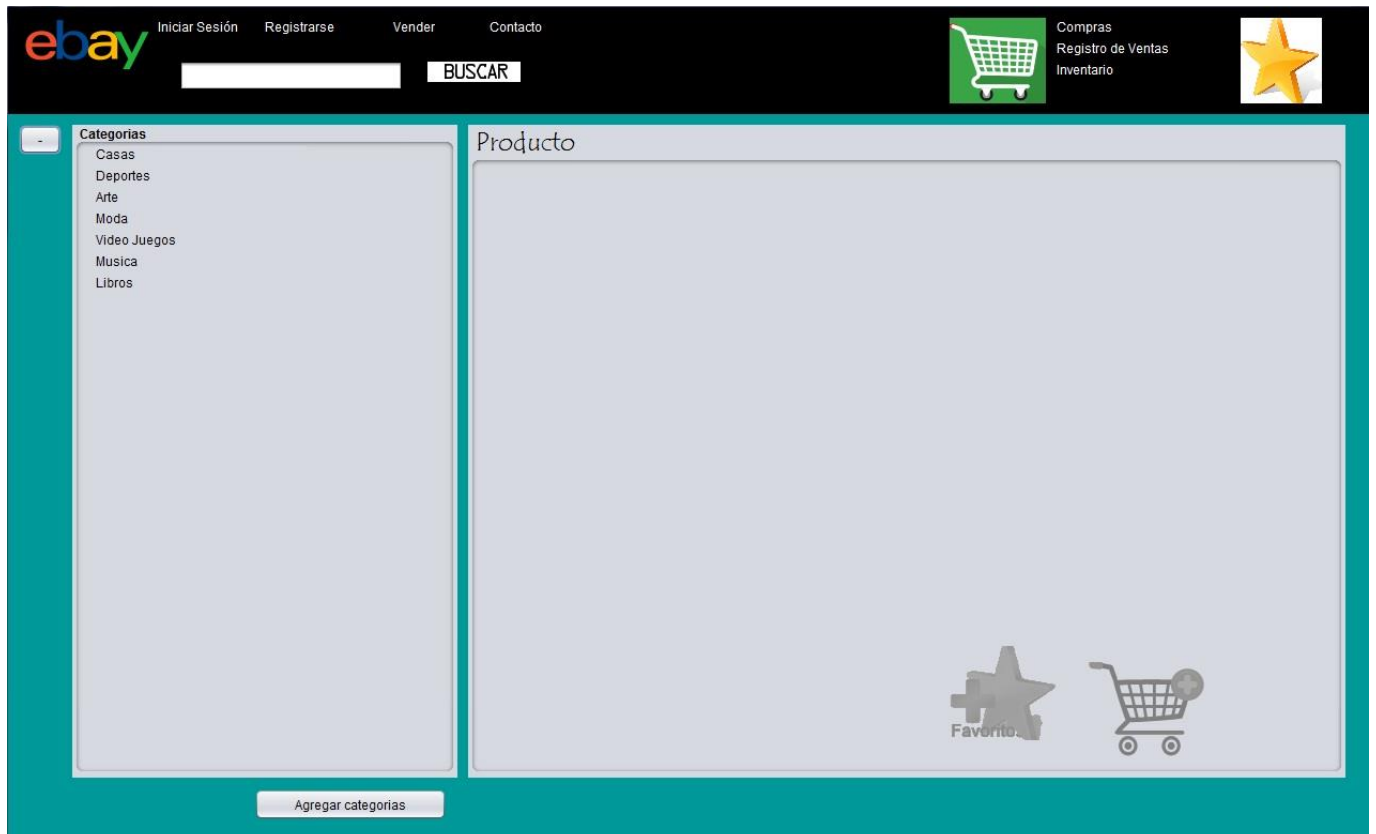
Las temáticas usadas son:

- Construcción de base de datos con archivos planos.
- Conocimiento para diseñar, crear, controlar el acceso y operar bases de datos relacionales.
- Uso herramientas básicas de diseño, implementación, control de acceso y de manipulación de datos con archivos.
- Dominar el modelo racional de base de datos y los procedimientos de modelos e implementación.
- Reconocimiento de la importancia de estos mecanismos de persistencia de datos en los sistemas de información.
- Conocer las estructuras de datos lo mejor posible para poder modificarlas fácilmente.
- Conocer diferentes algoritmos que sirven para utilizar las estructuras de datos a gusto del administrador del software.

1) PROTOTIPO

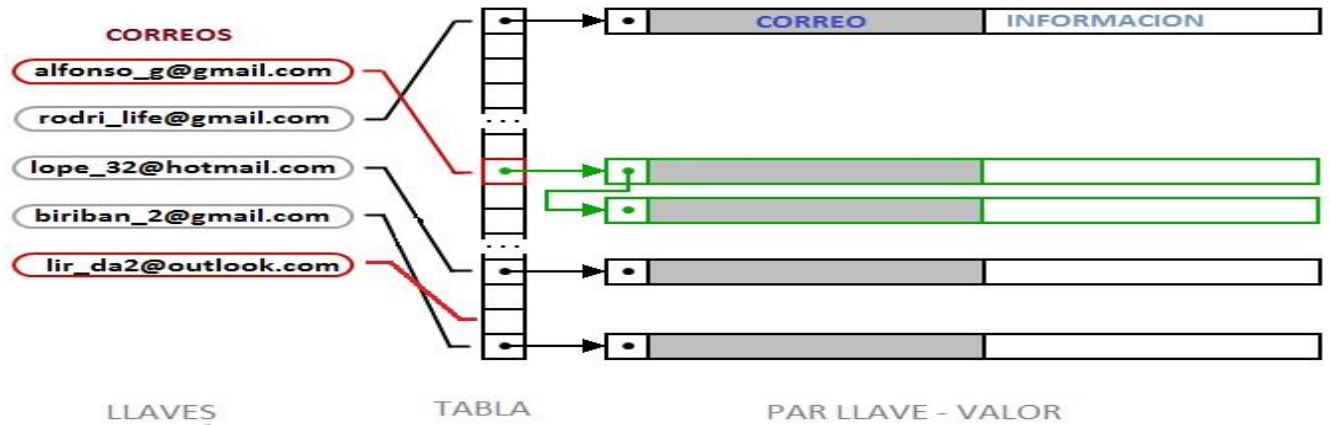
- **Menú Principal:** El menú principal de eBay contiene la conexión de las diferentes secciones. En él también se encuentran la vista de productos mediante categorías en la

cual se usó una estructura de árbol N-ario, mostrada gráficamente a continuación:



- **Registrar usuario:** El registro de usuarios tiene como elemento principal al correo del usuario; este se usó como llave para guardar toda la información del mismo (información

que se almacenó en una clase llamada 'Usuario') en una tabla de dispersión mostrada gráficamente a continuación:



Registro de Usuarios

Correo

Reingrese Correo

Contraseña

Nombre

Apellido

Pregunta Secreta

Respuesta Secreta

Tipo de Tarjeta

Nº de Tarjeta

Codigo de Seguridad

Fecha de vencimiento

- **Inicio de Sesión:** Verifica los usuarios registrados mediante el correo electrónico que se guardaron en una tabla de dispersión

graficada a anteriormente, se usó un algoritmo para revisar todas las llaves de la tabla y así encontrar la indicada:

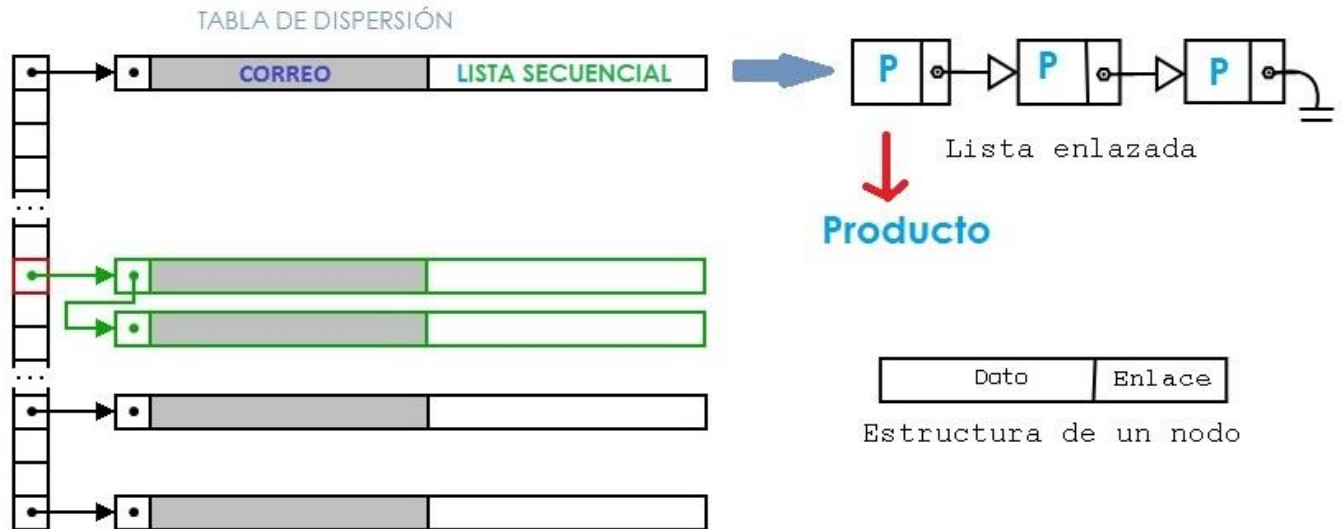
```
private void btnInicioSesionActionPerformed(java.awt.event.ActionEvent evt) {
    //Valores ingresados del usuario
    String clave = txtClave.getText();
    String correo = txtCorreo.getText();
    //Restriccion de campo vacio
    if(correo.length()>0){
        //hUsuarios: Tabla de dispersion que contiene la informacion de usuario.
        //Verifiac si la tabla contiene el correo ingresado
        if(hUsuarios.containsKey(correo)){
            //Se guarda la informacion del usuario en una instancia de la clase 'Usuario'
            //para poder usarla y modificarla.
            Usuario u1 = (Usuario)hUsuarios.get(correo);
            //Verifica si la clave ingresada pertenece a la informacion del usuario obtenida.
            if(clave.equals(u1.getClave())){
                JOptionPane.showMessageDialog(this, "Sesion iniciada como: "+correo);
                MenuPrincipal mp = new MenuPrincipal(u1);
                mp.setVisible(true);
                this.dispose();
                this.setVisible(false);
            }else{
                JOptionPane.showMessageDialog(this, "Contraseña incorrecta ");
            }
        }
        else{
            JOptionPane.showMessageDialog(this, "No se encuentra el usuario "+correo);
        }
    }else{
        JOptionPane.showMessageDialog(this, "Escriba un correo "+correo);
    }
}
}
```



- **Vender producto:** Se guardaron los productos en una lista de tipo secuencial la cual se guardó en una tabla de dispersión con referencia a cada usuario. Esto permitió

que cada usuario tenga su propia lista de productos vendidos. Esto se muestra gráficamente de la siguiente manera:

VENTAS



VENDER PRODUCTO

Título:

Categoría (-)

Casas Deportes Arte Moda Video Juegos Musica

Libros

Condición: Categoría Elegida:

Descripción de la condición:

Características del artículo y detalles:

Precio: Cantidad:

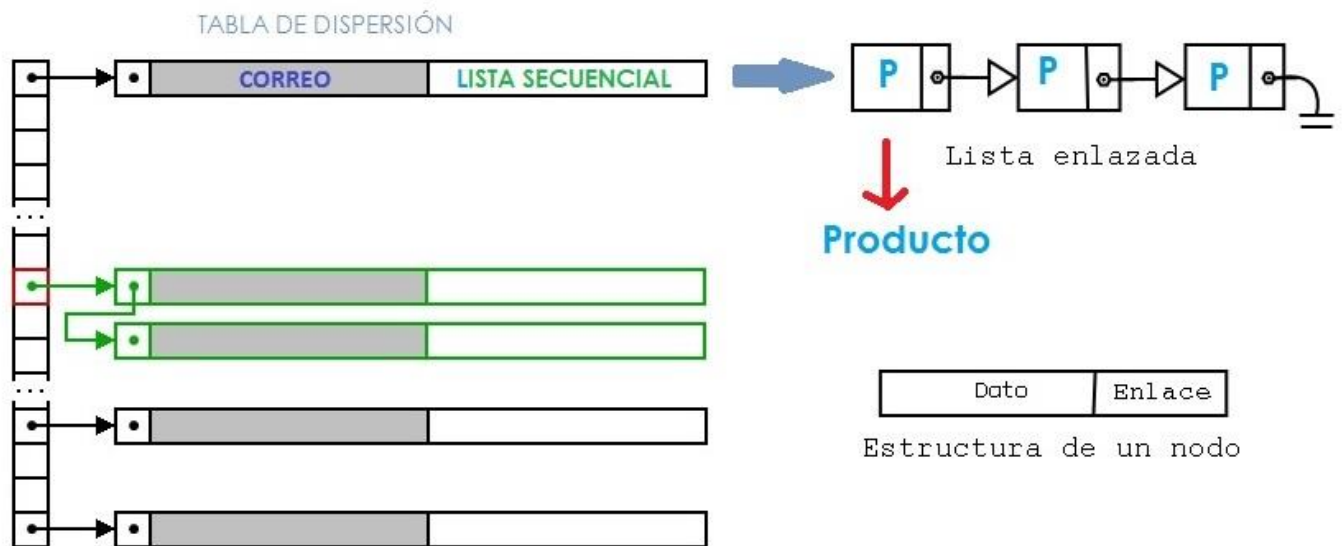
S/.

CANCELAR
FINALIZAR

- **Comprar producto:** Al igual que la venta los productos se guardaron en una lista secuencial que posteriormente se guardó en una tabla de dispersión. El sistema de compras se realizó de un modo parecido al de la página eBay con algunas variaciones,

para adaptarlas a un programa ejecutable, una de estas es que el detalle de cada producto aparecen en una ventana emergente al presionar cada producto y la cantidad se puede ingresar en el carrito de compras, que se verificó a través de algoritmos para que no supere el Stock.

COMPRAS



CARRITO DE COMPRAS DE E-BAY

Comienza a buscar y encuentra ofertas excelentes
Ve a Mi eBay para ver los artículos que marcaste como favoritos

ELIMINAR

**CARRITO
LISTA**

No hay ningun articulo en el carrito.



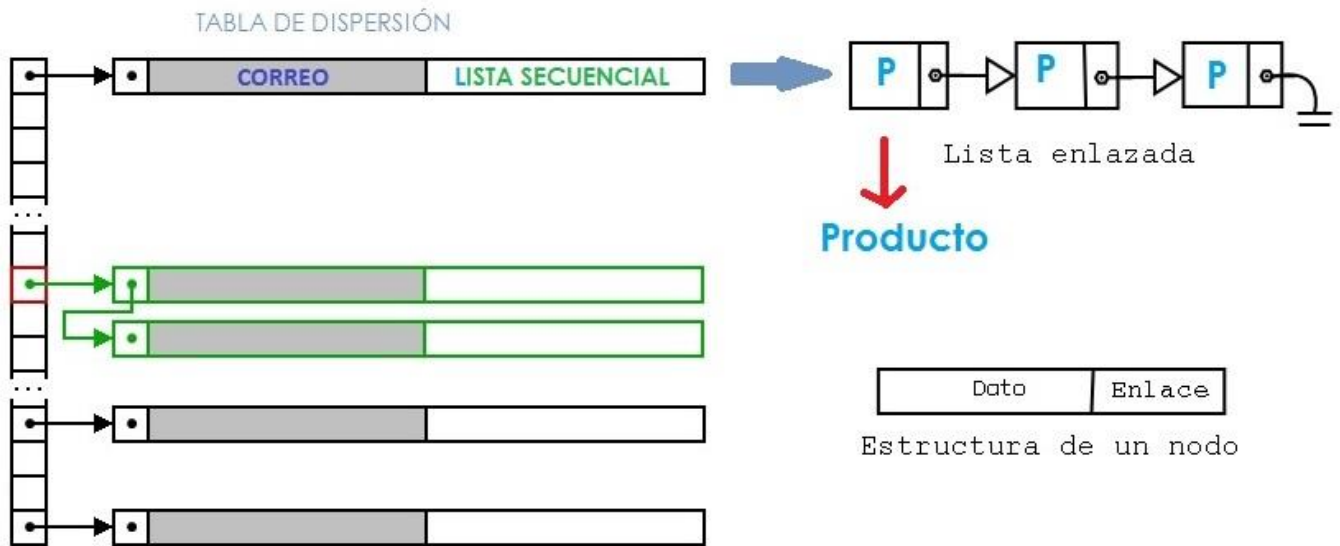
COMPRAR

Cantidad:

- Favoritos:** Al igual que la venta los productos se guardaron en una lista secuencial que posteriormente se guardó en una tabla de dispersión. El sistema de favoritos también se realizó de un modo parecido al de la página eBay con algunas

variaciones, para adaptarlas a un programa ejecutable, una de las opciones es poder ver la información de la lista de productos que el usuario tiene en favoritos y poder eliminarlas o agregarlas al carrito de comprar para su posterior compra.

FAVORITOS



6) Requerimientos del sistema

Método de pago seguro

A fin de que los Clientes se sientan cómodos en ingresar información de su tarjeta de crédito en la aplicación, el simulador de eBay debe proveer un método de pago altamente seguro.

Fácil búsqueda de títulos disponibles

Teniendo una amplia variedad de productos, los clientes deben ser provistos de una rápida forma de búsqueda por categoría.

Habilidad para chequear el estado de una orden

Después que los clientes hacen una compra, ellos querrán tener acceso al estado de sus órdenes.

Habilidad para adicionar y remover productos disponibles para ventas

El inventario va a ser muy dinámico; los nuevos productos son añadidos cuando un usuario registre un producto.

Habilidad para chequear las órdenes de los Clientes

Necesitamos tener rápido acceso a las órdenes para que podamos responder las inquietudes de los Clientes sobre algún retraso en el envío.

Mantenimiento de la información del Cliente

La información del Cliente incluye información personal (nombre, dirección, etc.) así como preferencias productos, necesitan ser guardadas y recuperadas fácilmente.

Agregar productos al carrito de compras

El sistema deberá permitir al usuario, previamente registrado, tener la opción de poder agregar productos seleccionados a su carrito de compra.

Generación de usuarios

El sistema permitirá el ingreso de nuevos usuarios.

Añadir a favoritos

El sistema deberá permitir al usuario tener la opción de poder agregar productos seleccionados a su lista de favoritos.

Acceso al carrito de compras

El sistema deberá permitir al usuario poder ver los productos que agrego en su carrito de compras y/o poder retirarlos.

Cambio de contraseña

El sistema debe permitir al usuario poder modificar la contraseña de su cuenta para tener una mayor seguridad.

Historial de compras

Descripción: El sistema debe permitir ver todas las compras de productos de la tienda realizadas por el usuario.

Historial de ventas

Descripción: El sistema debe permitir ver todas las ventas de productos realizadas por el usuario.

Requerimientos No funcionales

Seguridad

Registro de datos en archivos de texto no encriptados. El ingreso al sistema estará restringido bajo contraseñas cifradas y usuario definidos.

Rendimiento

El sistema debe tener un tiempo de respuesta menor a 3 segundos. El sistema debe soportar el manejo de gran cantidad de información durante su proceso.

Mantenibilidad

Solo administrador se encarga del mantenimiento del sistema solo el administrador programa fechas de mantenimiento o actualizaciones

Portabilidad

El sistema debe brindar comodidad al usuario y a otras áreas que trabajan o que necesitan del programa. Por ejemplo, en el sistema de pago no debe tener problemas en acceder al sistema de personal.

Uso de Java NetBeans para el desarrollo del sistema.

Uso del lenguaje Java para la codificación del sistema.

Desempeño

El sistema no presentara problemas para su manejo e implementación.

Usabilidad

Debe ser fácil de usar. Con ayudas e interfaces intuitivas.

Multiplataforma

El sistema deberá funcionar en distintos tipos de sistemas operativos y plataformas en hardware.