

Sistema control de inventarios para Pymes

Juan Camilo Tabares, Paulino Acuña, Miguel Currea

No. de Equipo Trabajo: {1}

I. INTRODUCCIÓN

Tomando como base los conocimientos y herramientas adquiridas previamente en programación orientada a objetos y las que se están adquiriendo en estructuras de datos se pretende desarrollar un proyecto donde se realice un control de inventarios de cualquier empresa. En el siguiente documento se encontrarán los aspectos básicos sobre el diseño y funcionamiento de esta aplicación.

II. DESCRIPCIÓN DEL PROBLEMA A RESOLVER

Realizar una aplicación de escritorio completamente funcional en la cual se pueda dar solución o mejorar la experiencia de manejo de inventarios que se debe realizar en una empresa pequeña o mediana.

III. USUARIOS DEL PRODUCTO DE SOFTWARE

Dentro del desarrollo e implementación de el proyecto se encontrarán los siguientes perfiles:

1. Perfil de ingeniero: este corresponde a el o los personajes que se ven involucrados con el desarrollo de la aplicación y su mantenimiento a la hora de estar en funcionamiento, tienen acceso a el codigo de el programa ya que será el encargado de realizar cambios para alterar su funcionamiento o mejorarlo.
2. rol de administrador: este corresponde a el o los personajes que pueden manejar todas las funcionalidades que presenta la aplicación final como también administrar a los diferentes usuarios que se verán involucrados en la aplicación, tienen acceso a las bases de datos para consultar, modificar y agregar cualquier tipo de dato.
3. rol de usuario: este corresponde a el o los personajes que pueden realizar funcionalidades a las cuales el administrador les permite acceder.

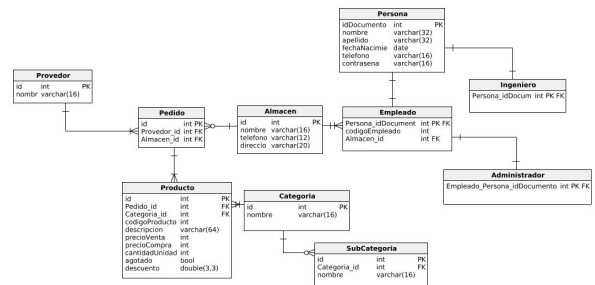
IV. REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

Entre los posibles requerimientos funcionales de un sistema, se incluyen:

- Descripciones de los datos a ser ingresados en el sistema.

Los datos que se ingresaran a la aplicación provendrán de una base de datos la cual se realizará con MySQL y serán los correspondientes a todas las características que tiene un producto las cuales son: nombre, codigo, categoria, precio, cantidad, descripción, proveedor y almacén en el que está disponible.

- Descripción del modelo relacional diseñado para la base de datos.



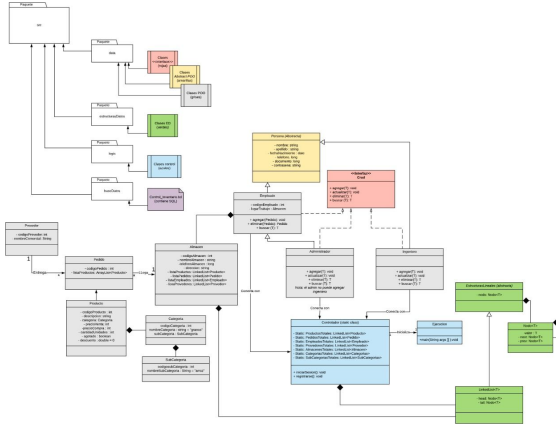
Vertabelo

- Descripciones de las operaciones a ser realizadas en cada pantalla que se presenta.

La pantalla principal que se encontrará al ingresar a la aplicación será una pantalla de login donde se pedirá usuario y contraseña para acceder a el sistema y según el rol que tenga podrá entrar a las siguientes ventanas:

1. Ventana de administrador: dentro de esta se verán las opciones de control de usuarios, control de inventario y configuración. dentro de la opción de control de usuarios puede añadir o eliminar perfiles como también modificarlos y definir a qué funcionalidades se les permitirá acceso.
2. Ventana de Usuario: dentro de esta se encontraran las opciones de inventario donde se podrá visualizar todos los productos que se encuentren o realizar búsquedas parciales, y la opción de pedido donde el usuario ingresara el número del pedido y directamente se descargan todos los datos correspondientes a el pedido que llegó a la empresa.

- Descripción de los flujos de trabajo realizados por el sistema.



- Descripción de los reportes del sistema y otras salidas.

La aplicación tendrá las opciones de imprimir parcial o totalmente los productos y categorías que se encuentren, tendrá una opción de imprimir para el administrador y esta será una diferente la del usuario ya que el administrador tiene acceso a todo tipo de datos mientras que el usuario solo a los designados por el administrador.

- Definición de quiénes pueden ingresar datos en el sistema.

El administrador será el personaje que tendrá acceso total a los datos de usuarios y de inventario como se señaló anteriormente en la ventana de administrador pero el usuario tendrá la opción de agregar pedido donde podrá modificar el inventario de manera controlada por el proveedor de la empresa.

Funcionalidades:

- **Agregar :**
Descripción: Para esta entrega pide por parámetros el código y nombre de el proveedor, realiza la creación de el objeto tipo proveedor y lo agrega delante de la lista enlazada.
Acciones iniciadas y comportamiento esperado: Al realizar la ejecución de la aplicación aparecerá un menú de usuario en el cual se podrá ver esta funcionalidad, al seleccionarla realiza la instanciación del nuevo objeto y se añade en la cabeza de la lista enlazada, no mostrará ningún mensaje.
Requerimientos funcionales: Ingresar datos correctos, el codigo es de tipo int y el nombre de tipo String.

- **Buscar:**
Descripción: Para esta entrega pide por parámetros el código del proveedor el cual lo utiliza para buscar el correspondiente proveedor en la lista enlazada.
Acciones iniciadas y comportamiento esperado: Al realizar la ejecución de la aplicación aparecerá un menú de usuario en el cual se podrá ver esta funcionalidad, al seleccionarla pedirá el código de el proveedor que se desea buscar, este conecta con la lista enlazada y la recorre hasta encontrar el elemento, devuelve el código y el nombre del proveedor si es encontrado, de no ser así devuelve un mensaje donde dice que no se ha encontrado.

Requerimientos funcionales: Ingresar una variable de tipo int la cual va a corresponder a el código que se buscará en la lista enlazada, de no ser así la aplicación de detendra.

- **Consultar todos los Datos:**
Descripción: al seleccionar esta funcionalidad imprimirá en pantalla todos los datos que este almacenados en la lista enlazada que en este caso será la de proveedor.
Acciones iniciadas y comportamiento esperado: al seleccionar esta acción en el menú de usuario que se ejecutara al correr la aplicación procederá a imprimir todos los datos almacenados, si la lista está vacía solo imprimirá “Lista de Proveedores” seguido de una línea en blanco.
Requerimientos funcionales: Digitar el número 4 que retornara la opción “consultar lista de proveedores”
- **Actualizar:**
Descripción: Pide por parámetro primero codigo de el objeto que se desea modificar, después pide los nuevos datos de el objeto y por ultimo guarda el objeto con estos nuevos datos .
Acciones iniciadas y comportamiento esperado: Al realizar la ejecución de la aplicación aparecerá un menú de usuario en el cual se podrá ver esta funcionalidad, al seleccionarla pedirá por entrada el código del objeto que se desea modificar, procede a buscar en la lista enlazada este objeto, después pide los nuevos datos del y por último imprime el nuevo objeto en pantalla
Requerimientos funcionales: los códigos que pide la aplicación son de tipo Int y los nombres de tipo String, si la lista se encuentra vacía retorna un mensaje reportando esto.
- **Eliminar:**
Descripción: Pide por parámetro el codigo de el proveedor a eliminar, después consulta la lista

enlazada y la recorre hasta encontrar el proveedor, posteriormente lo elimina.

Acciones iniciadas y comportamiento esperado: Al realizar la ejecución de la aplicación aparecerá un menú de usuario en el cual se podrá ver esta funcionalidad, al seleccionarla pedirá el código de el objeto a eliminar, después procederá a recorrer la lista enlazada hasta encontrarlo y lo eliminará.

Requerimientos funcionales: el tipo de Dato que pide por entrada será de tipo Int, si se ingresa otro tipo de dato procederá a cerrarse la aplicación, si la lista se encuentra vacía retorna un mensaje reportando esto.

- **Ordenar:**

Descripción: Ordena todos los datos pertenecientes a la lista enlazada en orden alfabético.

Acciones iniciadas y comportamiento esperado: Al realizar la ejecución de la aplicación aparecerá un menú de usuario en el cual se podrá ver esta funcionalidad, al seleccionarla se recorre toda la lista enlazada comparando los datos y ordenándolos en por orden alfabético.

Requerimientos funcionales: Si la lista se encuentra vacía retornada un mensaje reportando esto.

- **Salir:**

Descripción: Finaliza la aplicación

V. DESCRIPCIÓN DE LA INTERFAZ DE USUARIO PRELIMINAR



Luego de tener claro el enfoque de la aplicación, se procede a diseñar un esqueleto gráfico de la aplicación y las principales funciones en un *mock-up* realizado en HTML5.

VI. ENTORNOS DE DESARROLLO Y DE OPERACIÓN

Dentro de los recursos tecnológicos necesarios para la creación del proyecto se encuentran:

1. Administrador de bases de datos, para esto se pretende hacer uso de SQL.
2. NetBeans, IDE usado para la programación del código necesario para el funcionamiento esperado de la aplicación.
3. Interfaz gráfica, simple e intuitiva, que permita al usuario navegar por las distintas opciones que ofrece el sistema de inventarios, esta será programada usando las capacidades POO de Java en NetBeans.

El entorno de operación será Microsoft™ Windows® ya que esta se desarrollara como una aplicación de escritorio

VII. PROTOTIPO DE SOFTWARE INICIAL

Para la realización del software inicial se designó a una persona que realizará el proyecto base donde estarían contenidas todas las clases necesarias para todo el proyecto y se designó a un integrante que creará el repositorio en github y subiera este proyecto base.

Repositorio GitHub:

data	Entrega Final
docs	Entrega Final
nbproject	Creacion funcionalidad LinkedList.agregar()
src	Entrega Final
build.xml	Creacion del repositorio
manifest.mf	Creacion del repositorio

Figura 1. Estructura del repositorio.

Dentro de este prototipo de software se encontrarán todas las clases que se utilizarán a lo largo de el desarrollo de la aplicación aunque a lo largo de este puede surgir la necesidad de generar más clases según los retos que vayan apareciendo en el desarrollo.

Tomando de referencia los conocimientos adquiridos en POO se realizó la abstracción de las características que debía contener cada clase, en algunas de ellas se implementó la herencia y el polimorfismo se ve de manera muy clara al utilizar los métodos principales con diferentes tipos de objetos en el software.

VIII. PRUEBAS DEL PROTOTIPO

I. ANÁLISIS COMPARATIVO

Las siguientes tablas corresponden a un análisis funcional entre la cantidad de datos ingresados a la estructura **lista enlazada** y sus respectivos tiempo de ejecución. Cada tabla contempla dicha relación respecto a una funcionalidad específica.

Agregar Elemento	
Datos [cantidad]	Tiempo [milisegundos]
10.000	0,1
100.000	0,1
500.000	0,1
1.000.000	0,1
5.000.000	0,1
Complejidad: $O(1)$	

Actualizar Elemento	
Datos [cantidad]	Tiempo [milisegundos]
10.000	3,1
100.000	4,7
500.000	31
1.000.000	50
5.000.000	233
Complejidad: $O(n)$	

Buscar Todos Los Elementos	
Datos [cantidad]	Tiempo [milisegundos]
10.000	957

100.000	7.923
500.000	49.643
1.000.000	88.937
5.000.000	441.917
Complejidad: $O(n)$	

Buscar Un Elemento	
Datos [cantidad]	Tiempo [milisegundos]
10.000	2,8
100.000	8,3
500.000	33
1.000.000	52
5.000.000	250
Complejidad: $O(n)$	

Ordenar Elementos	
Datos [cantidad]	Tiempo [milisegundos]
10.000	12.643
100.000	207.172
500.000	N.A
1.000.000	N.A
5.000.000	N.A
Complejidad: $O(n^2)$	

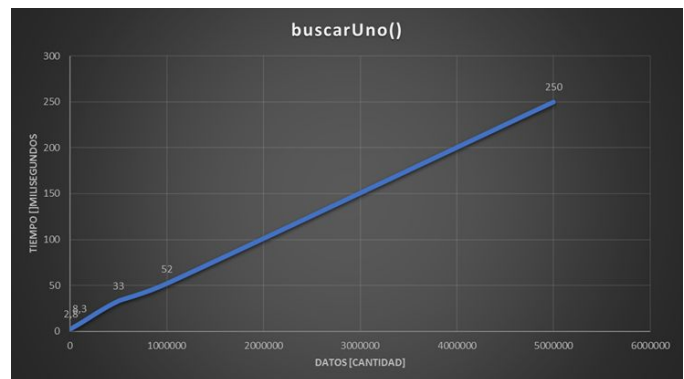
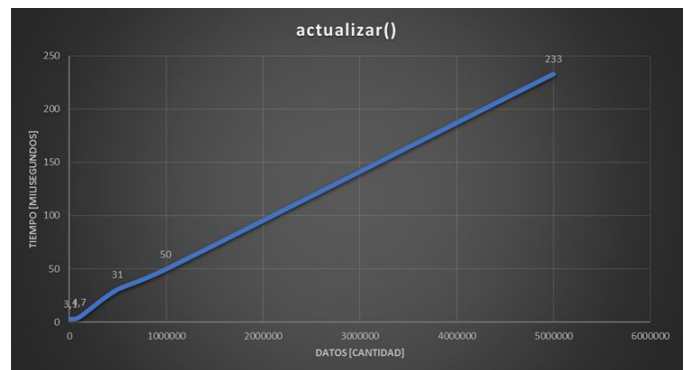
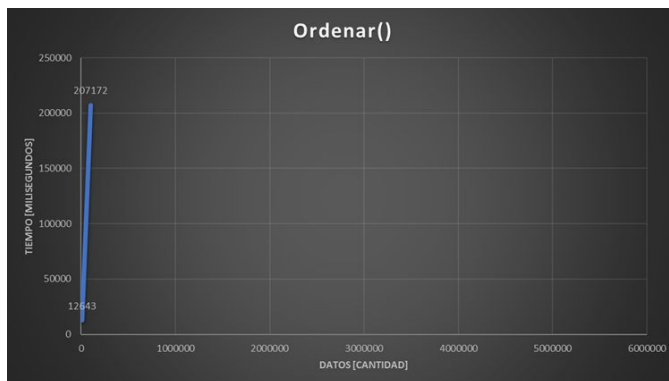
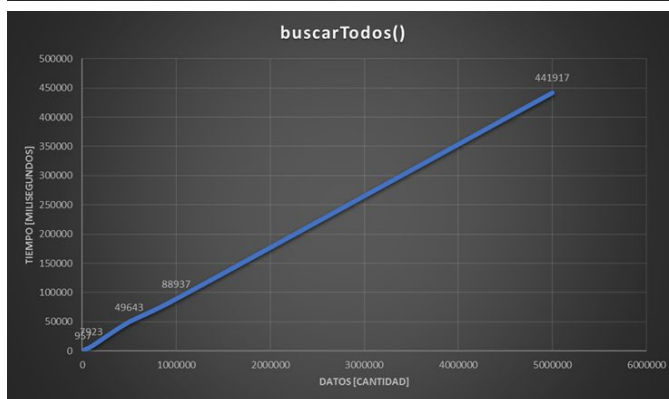
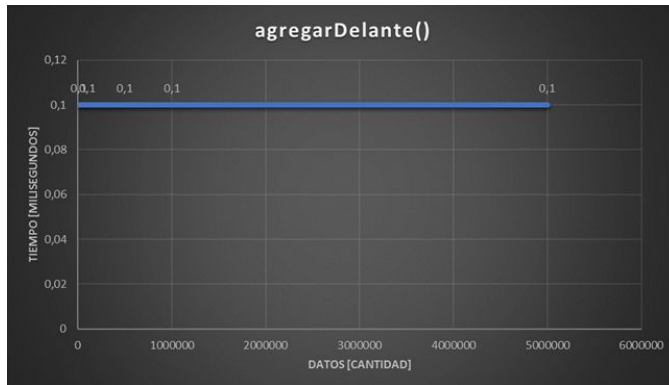
La notación N.A en la tabla “Ordenar Elementos” indica que no fue posible calcular estos valores debido al crecimiento exponencial de realizar esta operación con la cantidad de datos requerida.

Dentro del método “ordenar()” en la clase “LinkedList<T>” se encuentran 2 ciclos while anidados, por lo que esta operación no está disponible por el momento para una entrada de 500.000 datos.

Adicionalmente se puede observar que en la tabla “Agregar Elemento”, el tiempo de ejecución se mantiene sin importar la cantidad de datos de entrada ya que dentro de el método

“agregarDelante(key)” de la clase “LinkedList<T>” no es necesario realizar un recorrido sobre los datos para agregar un nuevo elemento.

Como equipo de prueba se ha utilizado un computador *Hewlett-packard Company* modelo *p6203w* con procesador *AMD Athlon(tm) 2 X2 215 2.70 GHz* y memoria RAM de 4.00 GB sobre un sistema operativo *Windows 7 home premium*.



II. ROLES Y ACTIVIDADES

Juan Camilo Tabares: Desarrollador y secretario.

Paulino Acuña: Desarrollador y especialista.

Miguel Currea: Desarrollador y coordinador.

III. DIFICULTADES Y LECCIONES APRENDIDAS

A continuación se describe una lista con las las mayores dificultades encontradas durante el desarrollo de esta primera entrega:

- Generación de datos.
- Conversión de datos de formato CSV a formato SQL.
- Diseño de la estructura del software y la conexión con la base de datos MySQL.
- Manejo del sistema de control de versiones Git y la plataforma GitHub.
- Casteo y manejo de variables generics.