

**UNIVERSIDAD TÉCNICA DE BABAHOYO**

**FACULTAD DE ADMINISTRACIÓN FINANZAS E INFORMÁTICA**

**ESCUELA DE SISTEMA**

**TEMA:**

**Sistema de venta e inventario para la empresa Calzado Sport Fhasion.**

**ESTUDIANTE:**

**Denisse Thalía Díaz**

**Quinto Sistemas “Nocturno”**

**Docente**

**Ing. Geovanny Vega**

**Periodo**

**Abril – Septiembre 2018**

ÍNDICE DE CONTENIDOS

[INTRODUCCIÓN 3](#_Toc521056324)

[DOMINIO DEL PROBLEMA 4](#_Toc521056325)

[Ámbito del software 4](#_Toc521056326)

[Detalle de información 4](#_Toc521056327)

[OBJETIVOS 5](#_Toc521056328)

[Objetivo General 5](#_Toc521056329)

[Objetivos Específicos 5](#_Toc521056330)

[ORGANIGRAMA 5](#_Toc521056331)

[ALCANCES Y NIVELES DE DECISIÓN 6](#_Toc521056332)

[Limitaciones del Proyecto 6](#_Toc521056333)

[PLANIFICACIÓN TEMPORAL 6](#_Toc521056334)

[ANÁLISIS DE REQUISITOS 6](#_Toc521056335)

[Metas del Sistema 6](#_Toc521056336)

[Requisitos Funcionales 7](#_Toc521056337)

[Requerimiento Técnico 7](#_Toc521056338)

[INGENIERÍA DE ANÁLISIS 8](#_Toc521056339)

[Ciclo de vida 8](#_Toc521056340)

[Usuarios 8](#_Toc521056341)

[Diagrama casos de uso 9](#_Toc521056342)

[MODELAMIENTO DEL SISTEMA 10](#_Toc521056343)

[Diagrama entidad relación 10](#_Toc521056344)

[Diccionario de datos 11](#_Toc521056345)

[HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO 14](#_Toc521056346)

[NETBEANS 14](#_Toc521056347)

[POSTGRESQL 14](#_Toc521056348)

[Herramienta PgAdmin 3 14](#_Toc521056349)

[DISEÑO DE LA BASE DE DATOS 15](#_Toc521056350)

[DISEÑO DE INTERFACES 15](#_Toc521056351)

[CODIFICACIÓN 18](#_Toc521056352)

[CONCLUSIONES 20](#_Toc521056353)

[RECOMENDACIONES 20](#_Toc521056354)

[BIBLIOGRAFÍA 20](#_Toc521056355)

# INTRODUCCIÓN

El presente trabajo explica el desarrollo de un sistema de control de ventas e inventario, puesto que nace de la inquietud de sustentar los factores que determinan la importancia de llevar un control de inventarios dentro de las empresas, sin hacer exclusión por el tipo de negocio o por su tamaño, el llevar un control de inventario recae en la forma de ver los activos de modo tangible, en muchas ocasiones resulta ser una tarea no fácil, pero tampoco imposible y aunque puede resultar ser un ejercicio tedioso, sobre todo para pequeños negocios, es la base de un plan de negocios y rentabilidad del mismo.

Es por ello que Calzado Sport Fhasion ha optado por implementar un sistema de control de ventas e inventario para registrar todas sus operaciones comerciales.

El control del inventario es un elemento muy importante para el desarrollo, tanto en grandes empresas como en pequeñas y medianas; una mala administración, puede ser el culpable de generar clientes descontentos por el no cumplimiento de la demanda, además de ocasionar problemas financieros que pueden llevar a la compañía a la quiebra

Es muy importante que las empresas tengan su inventario atentamente controlado, vigilado y ordenado, dado a que de éste depende el proveer y distribuir adecuadamente lo que se tiene, colocándolo a disposición en el momento indicado. Sin duda alguna, para cualquier tipo de empresa se hacen necesarios los inventarios dado a que la base de todas las organizaciones consiste en la compra y venta de bienes y servicios, haciéndose necesaria la existencia de los inventarios, los cuales le van a permitir tener control de la mercancía y a su vez generar reportes de la situación económica de la empresa

# DOMINIO DEL PROBLEMA

La empresa Calzado Sport Fhasion se dedica a la venta de zapatos al público en general. Actualmente Calzado Sport Fhasion dispone de un sistema manual para el manejo de existencias, el cual consiste en un cuaderno de anotaciones, donde quedan registradas las ventas de productos, anotando el nombre y el precio del producto correspondiente en cada transacción. El administrador verifica una vez terminada la jornada laboral la cantidad de productos vendidos y el monto total por concepto de ventas.

Las compras son realizadas a grandes proveedores existentes en la región, las cuales son registradas en su respectivo libro contable (libro de compras y ventas), labor que lleva a cargo el contador de la empresa, no quedando ningún registro en el local.

## Ámbito del software

La solución consiste en diseñar y construir un sistema de escritorio que permita a Calzado Sport Fhasion llevar un control sobre el stock de los productos disponibles. Este proyecto permitirá automatizar el sistema actual, ya que no solo se mantendrá información de los productos, sino también de los proveedores, clientes y movimientos de stock. Con esto se consigue obtener rápidamente un detalle tanto de las entradas como salidas de productos según periodo de fechas.

Con respecto al control de stock, el sistema mantendrá un registro de las salidas de productos, con lo cual el administrador poseerá una información oportuna para evaluar o tomar decisiones de cuando reabastecerse. Se utilizará a JAVA como lenguaje de programación conjuntamente con PostgreSQL para el almacén de datos.

## Detalle de información

Una vez analizada la situación actual y los problemas que posee la empresa, se puede establecer lo siguiente. La empresa no posee equipo computacional en la actualidad, pero existe la intención por parte de la empresa en hacer la inversión que sea necesaria, de lo anterior se desprende que existe un gran entusiasmo de parte de la empresa Calzado Sport Fhasion , para solucionar los problemas mencionados anteriormente.

# OBJETIVOS

## Objetivo General

Desarrollar un de venta e inventario para la zapatería Calzados Mechita.

## Objetivos Específicos

* Realizar el estudio pertinente de requerimientos para la realización del sistema.
* Mejorar el proceso de venta y registro de inventario de los zapatos de Calzado Sport Fhasion.
* Agilizar el proceso de gestión e impresión de reportes de inventario y de ventas.

# ORGANIGRAMA

# ALCANCES Y NIVELES DE DECISIÓN

La disponibilidad de los recursos necesarios para llevar a cabo los objetivos o metas señalados, sirve para recopilar datos relevantes sobre el desarrollo de un proyecto y en base a ello tomar la mejor decisión, si procede su estudio, desarrollo y/o implementación.

## Limitaciones del Proyecto

* El sistema no realizará ventas vía Internet.
* El sistema no realizará balances, ni otra documentación contable.
* El sistema no registrará los pagos a los proveedores.

# PLANIFICACIÓN TEMPORAL

|  |  |  |
| --- | --- | --- |
| Tarea | Duración (días) | Dependencia |
| Recolección de requerimientos (T1) | 5 |  |
| Análisis (T2) | 4 | T1 |
| Diseño (T3) | 7 | T2 |
| Análisis de Funcionalidad (T4) | 3 |  |
| Desarrollo (T5) | 30 | T3 |
| Pruebas (T6) | 7 | T5 |
| Implementación | 5 | T4, T5, T6 |

# ANÁLISIS DE REQUISITOS

## Metas del Sistema

* Diseño y construcción de un sistema, escritorio, que facilite el acceso de los usuarios.
* Registro de existencias de los productos.
* Registro de Ventas y Compras
* Automatización del sistema de inventario.

## Requisitos Funcionales

“El primer reto del trabajo de los requisitos es encontrar, comunicar y recordar (que normalmente significa registrar) lo que se necesita realmente, de manera que tenga un significado claro para el cliente y los miembros del equipo de desarrollo”.

Los requerimientos se clasifican en las siguientes categorías

Evidente: Debe realizarse, y el usuario debería saber que se ha realizado.

Oculta: Debe realizarse, aunque no es visible para los usuarios. Esto se aplica a muchos servicios técnicos subyacentes, como guardar información de un mecanismo persistente de almacenamiento. Las funciones ocultas a menudo se omiten (erróneamente) durante el proceso de obtención de requerimientos.

Superflua:

Opcionales: su inclusión no repercute significativamente en el costo, ni en otras funciones.

## Requerimiento Técnico

**HARDWARE**

* **Procesador:** Procesador de 86 o 64 bits
* **Memoria RAM:** 2 o 4 GB de RAM
* **Mostrar:** Súper VGA con una resolución de 1024 x 768

**SOFTWARE**

* JRE 8.1 (Java Rutine Enviroment)
* PostgreSQL 9.5

# INGENIERÍA DE ANÁLISIS

## Ciclo de vida

El ciclo de vida de este proyecto será el modelo en cascada, puesto que es un sistema pequeño. El modelo de desarrollo en cascada sigue una serie de etapas de forma sucesiva, la etapa siguiente empieza cuando termina la etapa anterior. El tiempo que se pasa en diseñar el producto en las primeras fases del proceso puede evitar problemas que serían más costosos cuando el proyecto ya estuviese en fase de desarrollo.

## Usuarios

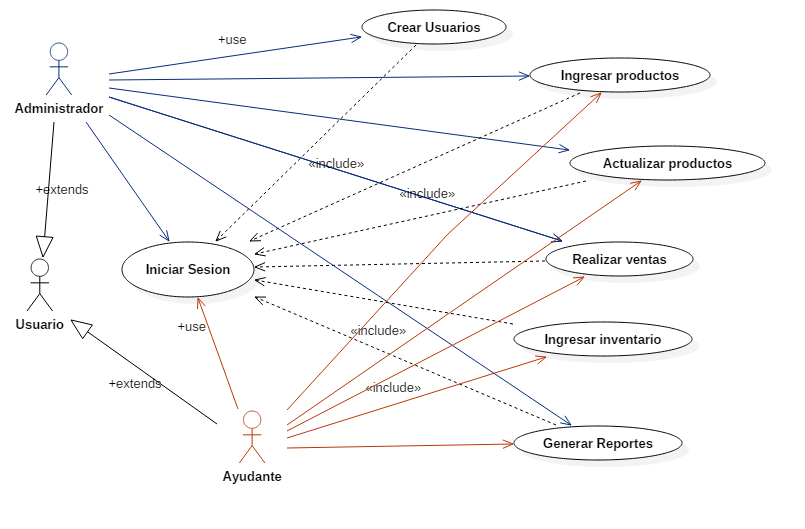
**Administrador**

* Encargado del soporte, mantenimiento y corrección de errores por parte de los usuarios.
* Crear un nuevo administrador y otros usuarios
* Agrega Productos
* Actualiza Productos
* Ingresa clientes y proveedores
* Realiza Ventas
* Genera Reportes

**Ayudante**

* Podrá registrarse escoger horarios, matricularse y realizar pagos.
* Agrega Productos
* Actualiza Productos
* Ingresa clientes y proveedores
* Realiza Ventas
* Genera Reportes

## Diagrama casos de uso



# MODELAMIENTO DEL SISTEMA

## Diagrama entidad relación



## Diccionario de datos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabla** | **Nombre de Campo** | **Tipo de Dato** | **Primary Key** | **Foreign Key** |
| **articulos** | codcategoria | int4 | No | Sí |
| codsubcategoria |  |  |  |
| codarticulo | varchar(6) | Sí | No |
| descripcion | varchar(40) | No |  |
| grabaiva | bpchar(1) |  |  |
| costo | float8 |  |  |
| pvp |  |  |  |
| stock |  |  |  |
| estado | bpchar(1) |  |  |
| **categorias** | codcategoria | int4 | Sí |  |
| descripcion | varchar(40) | No |  |
| **ciudades** | codciudad | varchar(4) | Sí |  |
| codprovincia | varchar(2) | No | Sí |
| nombreciudad | varchar(60) |  | No |
| **costos** | codsaldo | int8 | Sí |  |
| codarticulo | varchar(20) | No | Sí |
| stock\_c | float8 |  | No |
| costo\_c |  |  |  |
| fecha | timestamp |  |  |
| **detallefactura** | codfactura | int8 |  | Sí |
| codarticulo | varchar(20) |  | No |
| cantidad | float8 |  |  |
| precio |  |  |  |
| iva |  |  |  |
| **detallemovimiento** | codmovimiento | int8 |  | Sí |
| codarticulo | varchar(20) |  |  |
| orden | int4 |  | No |
| cantidad | float8 |  |  |
| costo |  |  |  |
| costotal |  |  |  |
| cantidadtotal |  |  |  |
| costounitariototal |  |  |  |
| **estados** | codestado | bpchar(1) | Sí |  |
| descripcion | varchar(30) | No |  |
| **factura** | codfactura | int8 | Sí |  |
| codmovimiento |  | No | Sí |
| coditpofactura | int4 |  |  |
| numerofactura | varchar(9) |  | No |
| concepto | varchar(200) |  |  |
| codtiporesponsable | bpchar(1) |  |  |
| identificacion | varchar(13) |  |  |
| totalfactura | float8 |  |  |
| fechahoraingreso | timestamp |  |  |
| fechafactura | date |  |  |
| idperiodo | int8 |  | Sí |
| **ingresoalsistema** | codiniciosis | int4 |  | No |
| codusuario | varchar(10) |  |  |
| codestado | bpchar(1) |  | Sí |
| fecha | timestamp |  | No |
| isusuario | bool(5) |  |  |
| **movimientos** | codtipomovimiento | varchar(3) |  | Sí |
| naturaleza | bpchar(1) |  | No |
| codusuario | varchar(10) |  |  |
| codtiporesponsable | bpchar(1) |  | Sí |
| identificacion | varchar(13) |  |  |
| codmovimiento | int8 | Sí | No |
| concepto | varchar(200) | No |  |
| fechamovimiento | date |  |  |
| totalmovimiento | float8 |  |  |
| fechahoraingreso | timestamp |  |  |
| fechahoraactualizacion |  |  |  |
| fechahoraanulacion |  |  |  |
| usuarioanulacion | varchar(10) |  |  |
| usuarioaactualizacion |  |  |  |
| estado | bpchar(1) |  |  |
| idperiodo | int8 |  | Sí |
| **perfiles** | codperfil | int4 | Sí | No |
| descripcion | varchar(30) | No |  |
| **periodo** | idperiodo | int8 | Sí |  |
| descripcion | varchar(30) | No |  |
| fechainicio | date |  |  |
| fechafin |  |  |  |
| **provincia** | codprovincia | varchar(2) | Sí |  |
| descripcion | varchar(30) | No |  |
| **responsables** | codtiporesponsable | bpchar(1) | Sí | Sí |
| identificacion | varchar(13) |  | No |
| razonsocial | varchar(80) | No |  |
| codciudad | varchar(6) |  | Sí |
| direccion | varchar(100) |  | No |
| telefono | varchar(10) |  |  |
| contacto | varchar(50) |  |  |
| **subcategorias** | codcategoria | int4 |  | Sí |
| codsubcategoria |  | Sí | No |
| descripcion | varchar(40) | No |  |
| **tipofactura** | coditpofactura | int4 | Sí |  |
| descripcion | varchar(10) | No |  |
| **tipomovimiento** | codtipomovimiento | varchar(3) | Sí |  |
| descripcion | varchar(50) | No |  |
| naturaleza | varchar(2) |  |  |
| **tiporesponsable** | codtiporesponsable | bpchar(1) | Sí |  |
| descripcion | varchar(20) | No |  |
| **usuarios** | codperfil | int4 |  | Sí |
| codusuario | varchar(10) | Sí | No |
| nombre | varchar(40) | No |  |
| clave | varchar(10) |  |  |
| codestado | bpchar(1) |  | Sí |

# HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

## NETBEANS

Es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun MicroSystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos (Actualmente Sun Microsystems es administrado por Oracle Corporation).

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

El NetBeans IDE soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring.

## POSTGRESQL

Es un Sistema de gestión de bases de datos relacional orientado a objetos y libre, publicado bajo la licencia PostgreSQL, similar a la BSD o la MIT. PostgreSQL es un potente sistema de base de datos relacional de objetos de código abierto con más de 30 años de desarrollo activo que le ha valido una sólida reputación de fiabilidad, robustez de las características y rendimiento.

Hay una gran cantidad de información que se encuentra describiendo cómo instalar y usar PostgreSQL a través de la documentación oficial. La comunidad de PostgreSQL brinda muchos lugares útiles para familiarizarse con la tecnología, descubrir cómo funciona y encontrar oportunidades profesionales.

PostgreSQL ha tenido una larga evolución, la cual se inicia en 1982 con el proyecto Ingres en la Universidad de Berkeley. Este proyecto, liderado por Michael Stonebraker, fue uno de los primeros intentos en implementar un motor de base de datos relacional. Después de haber trabajado un largo tiempo en Ingres y de haber tenido una experiencia comercial con el mismo, Michael decidió volver a la Universidad en 1985 para trabajar en un nuevo proyecto sobre la experiencia de Ingres, dicho proyecto fue llamado post-ingres o simplemente POSTGRES.

El proyecto post-ingres pretendía resolver los problemas con el modelo de base de datos relacional que habían sido aclarados a comienzos de los años 1980. El principal de estos problemas era la incapacidad del modelo relacional de comprender "tipos", es decir, combinaciones de datos simples que conforman una única unidad. Actualmente estos son llamados objetos. Se esforzaron en introducir la menor cantidad posible de funcionalidades para completar el soporte de tipos. Estas funcionalidades incluían la habilidad de definir tipos, pero también la habilidad de describir relaciones - las cuales hasta ese momento eran ampliamente utilizadas pero mantenidas completamente por el usuario. En Postgres la base de datos «comprendía» las relaciones y podía obtener información de tablas relacionadas utilizando reglas. Postgres usó muchas ideas de Ingres pero no su código.

## Herramienta PgAdmin 3

PgAdmin es la plataforma de administración y desarrollo de código abierto más popular y rica en características para PostgreSQL, la base de datos de código abierto más avanzada del mundo.

PostgreSQL es un Sistema de Gestión de Bases de Datos Open Source, que tiene las siguientes características:

* Es un gestor de bases de datos orientadas a objetos, cumple los estándares SQL92 y SQL99, y soporta funcionalidades avanzadas.
* Es el sistema de gestión de bases de datos de código abierto más potente del mercado.
* Es de código libre, publicado bajo la licencia BSD.
* No es espacial, no admite datos geográficos. Si queremos trabajar con datos espaciales es necesario trabajar con PostGIS.

PostGIS es una extensión de PostgreSQL, que añade soporte de objetos geográficos a PostgreSQL, convirtiéndola en una base de datos espacial. Tiene las siguientes características:

Permite el uso de objetos GIS incluyendo el soporte para índices GiST y funciones básicas para el análisis de objetos GIS.

Está publicado bajo licencia GNU.

Con PostGIS podemos usar los objetos de especificaciones de OpenGIS: puntos, líneas, polígonos, multilíneas, multipuntos, y colecciones geométricas.

pgAdmin 4 es una aplicación gráfica para gestionar el gestor de BD PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ y se puede usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. El interfaz gráfico soporta todas las características de PostgreSQL y facilita enormemente la administración.

**Java (lenguaje de programación)**

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos diez millones de usuarios reportados.

La primera característica, orientado a objetos (“OO”), se refiere a un método de programación y al diseño del lenguaje. Aunque hay muchas interpretaciones para OO, una primera idea es diseñar el software de forma que los distintos tipos de datos que usen estén unidos a sus operaciones. Así, los datos y el código (funciones o métodos) se combinan en entidades llamadas objetos. Un objeto puede verse como un paquete que contiene el “comportamiento” (el código) y el “estado” (datos). El principio es separar aquello que cambia de las cosas que permanecen inalterables. Frecuentemente, cambiar una estructura de datos implica un cambio en el código que opera sobre los mismos, o viceversa. Esta separación en objetos coherentes e independientes ofrece una base más estable para el diseño de un sistema software. El objetivo es hacer que grandes proyectos sean fáciles de gestionar y manejar, mejorando como consecuencia su calidad y reduciendo el número de proyectos fallidos.

Otra de las grandes promesas de la programación orientada a objetos es la creación de entidades más genéricas (objetos) que permitan la reutilización del software entre proyectos, una de las premisas fundamentales de la Ingeniería del Software. Un objeto genérico “cliente”, por ejemplo, debería en teoría tener el mismo conjunto de comportamiento en diferentes proyectos, sobre todo cuando estos coinciden en cierta medida, algo que suele suceder en las grandes organizaciones. En este sentido, los objetos podrían verse como piezas reutilizables que pueden emplearse en múltiples proyectos distintos, posibilitando así a la industria del software construir proyectos de envergadura empleando componentes ya existentes y de comprobada calidad, conduciendo esto finalmente a una reducción drástica del tiempo de desarrollo. Podemos usar como ejemplo de objeto el aluminio. Una vez definidos datos (densidad, maleabilidad, etc.), y su “comportamiento” (soldar dos piezas, etc.), el objeto “aluminio” puede ser reutilizado en el campo de la construcción, del automóvil, de la aviación, etc.

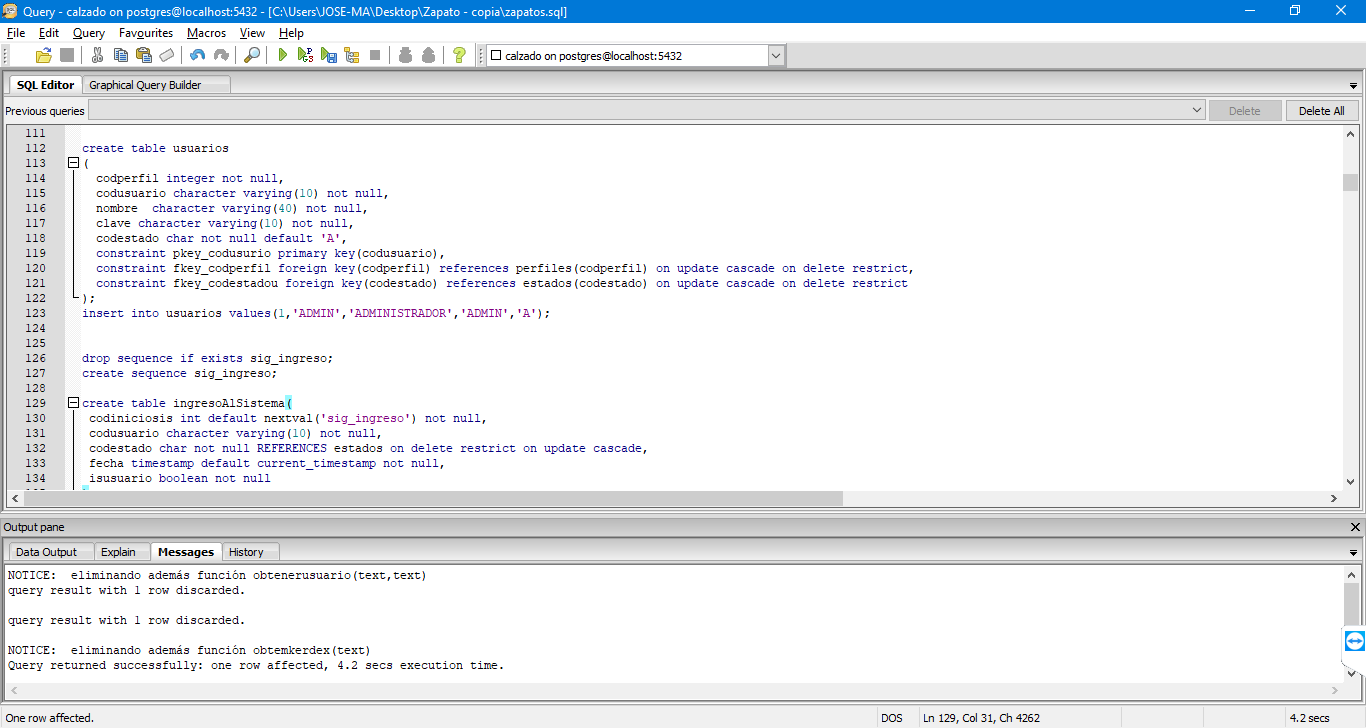
La reutilización del software ha experimentado resultados dispares, encontrando dos dificultades principales: el diseño de objetos realmente genéricos es pobremente comprendido, y falta una metodología para la amplia comunicación de oportunidades de reutilización. Algunas comunidades de “código abierto” (open source) quieren ayudar en este problema dando medios a los desarrolladores para diseminar la información sobre el uso y versatilidad de objetos reutilizables y bibliotecas de objetos.

Java Development Kit

Java Development Kit (JDK) es un software que provee herramientas de desarrollo para la creación de programas en Java. Puede instalarse en una computadora local o en una unidad de red.

En la unidad de red se pueden tener las herramientas distribuidas en varias computadoras y trabajar como una sola aplicación.

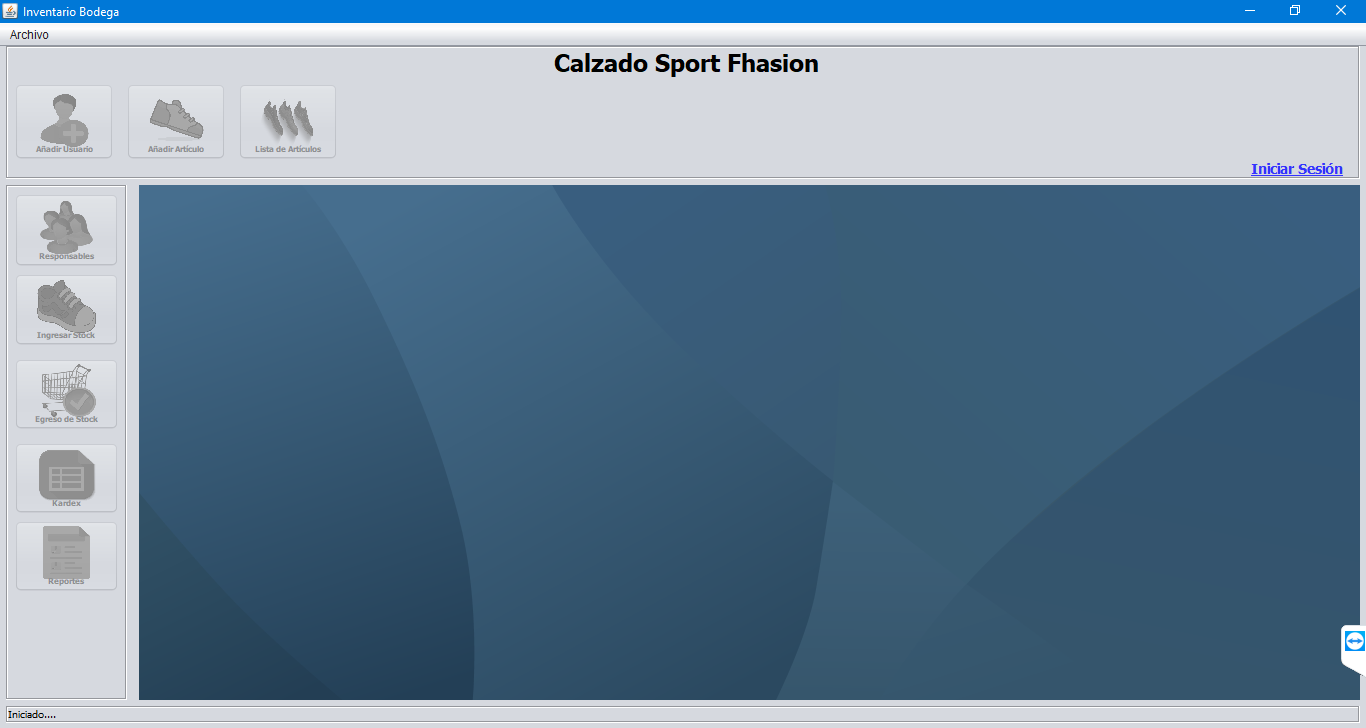
# DISEÑO DE LA BASE DE DATOS



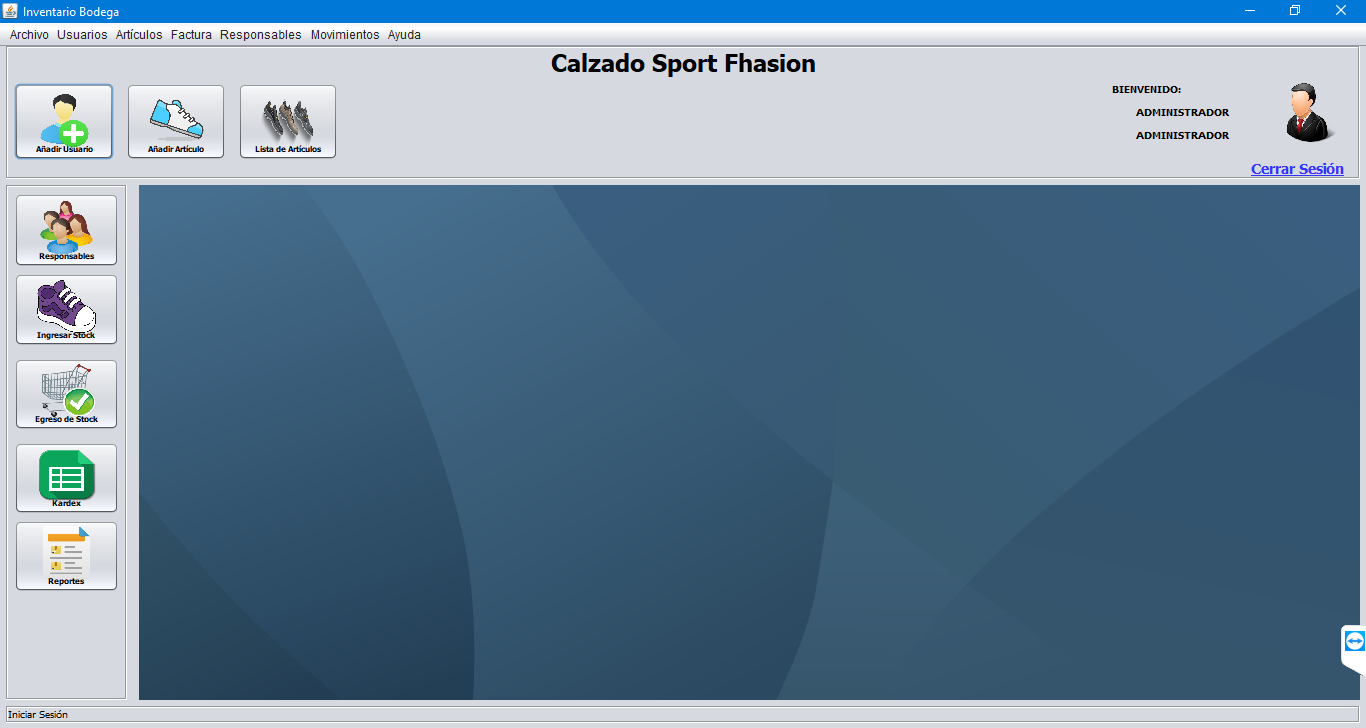
# DISEÑO DE INTERFACES



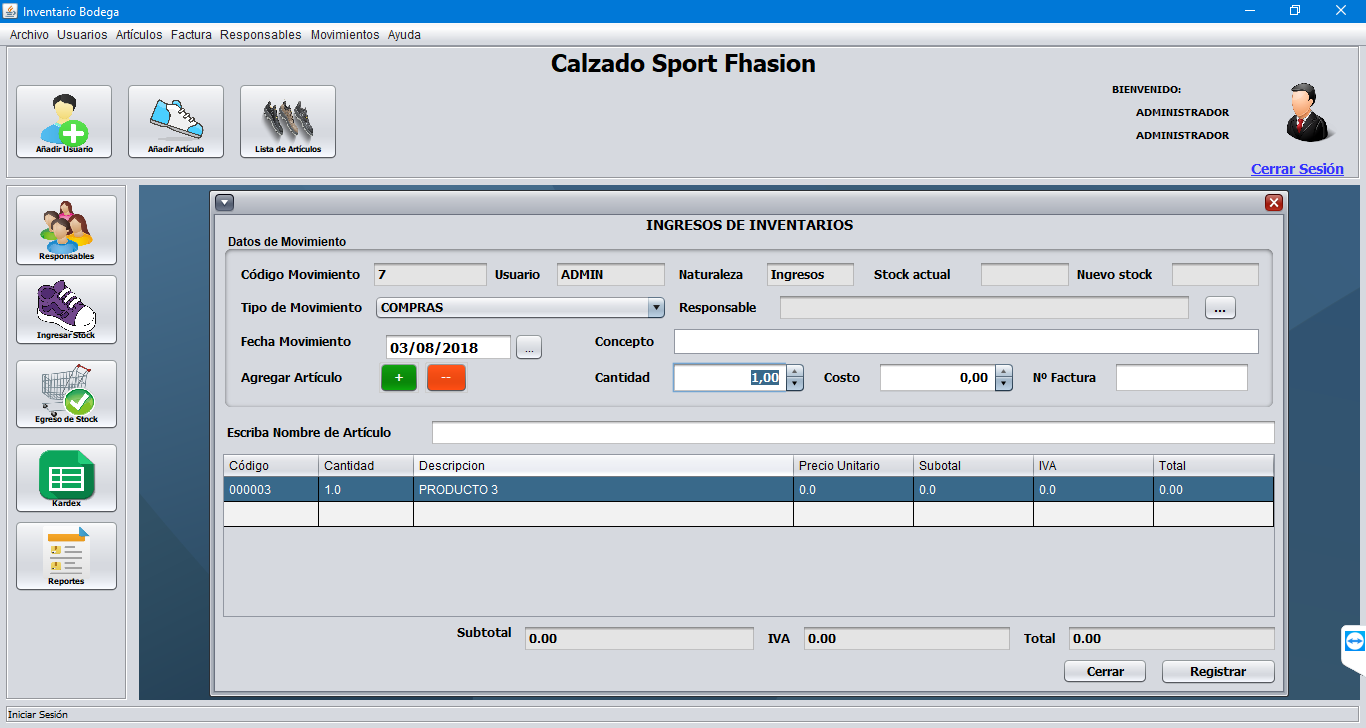
Inicio



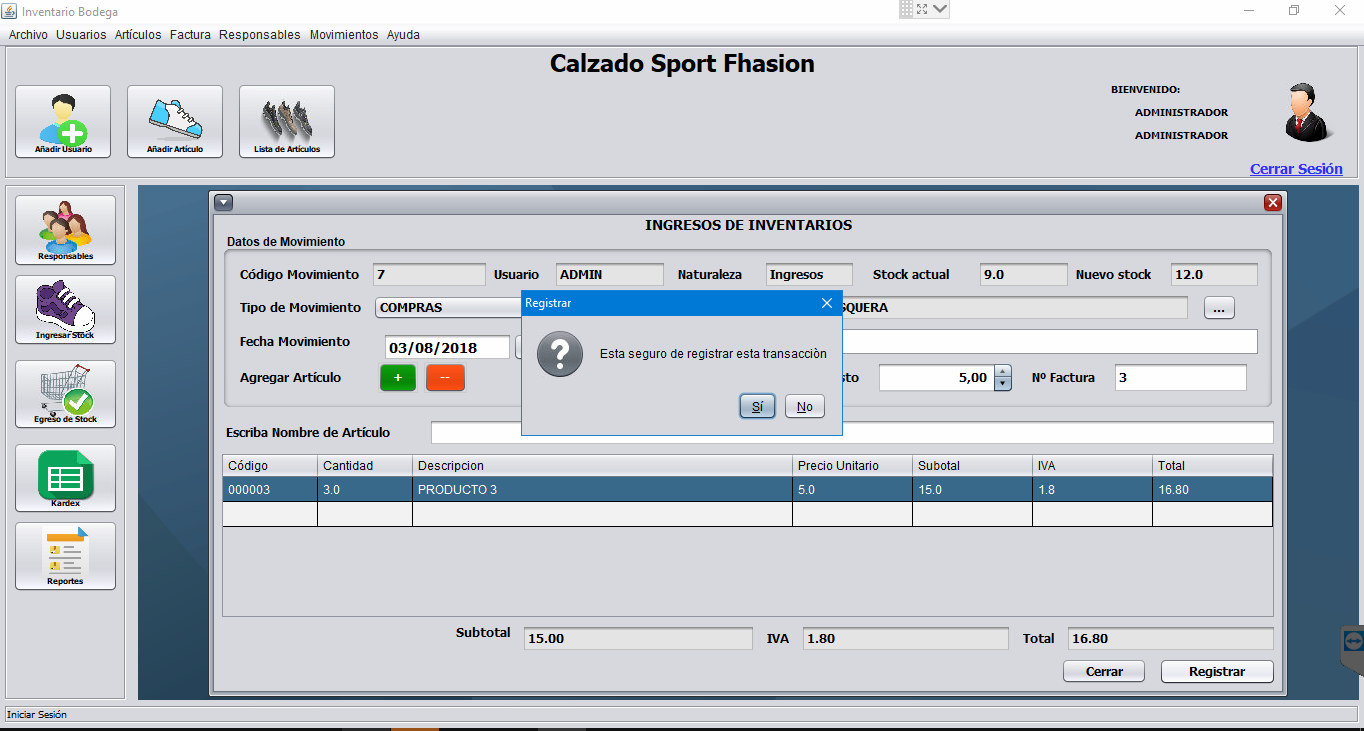
Usuario en el Sistema



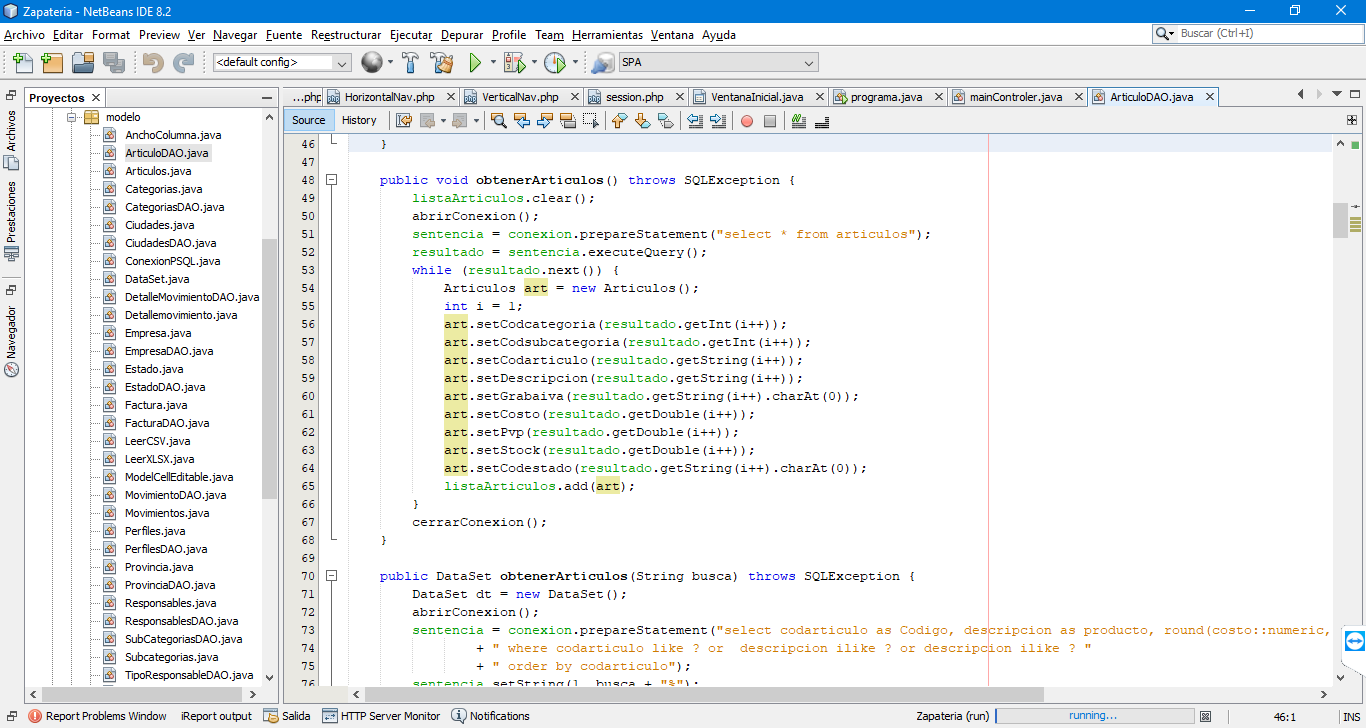
Realizar Movimiento

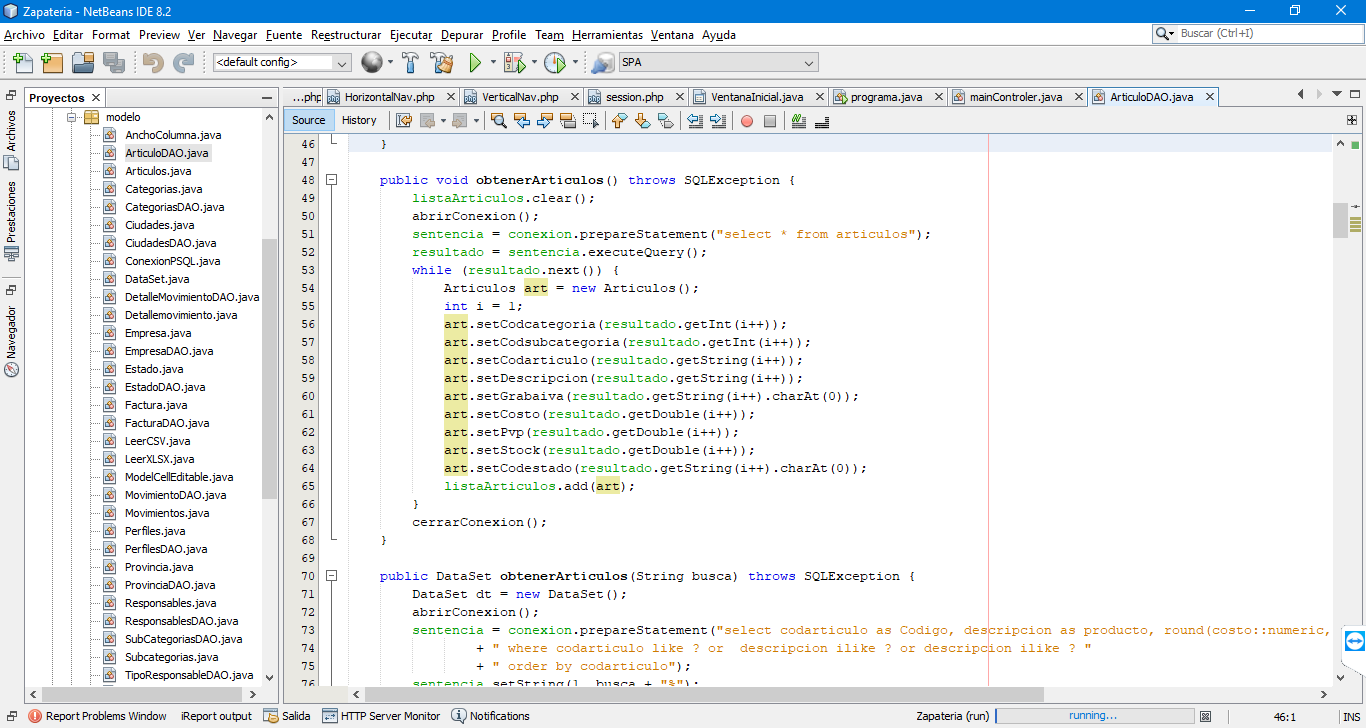


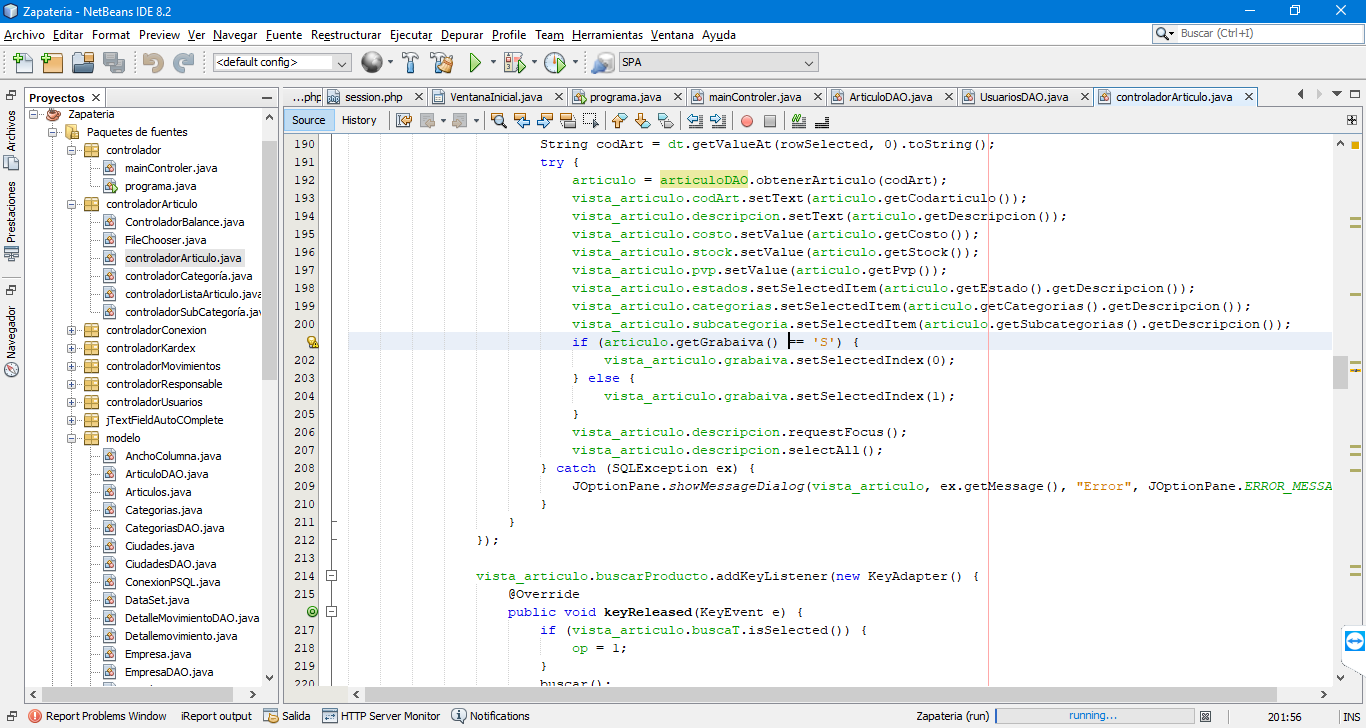
Mensaje de Confirmación



# CODIFICACIÓN







# CONCLUSIONES

Debido al constante crecimiento que han experimentado las empresas y la necesidad de disponer en forma rápida y oportuna de información, se hace indispensable la construcción de sistemas de software para cumplir con estos objetivos.

La empresa Calzado Sport Fhasion en su afán por estar a la vanguardia de la tecnología solicitó la construcción de un sistema el cual le permita facilitar y automatizar el control de su inventario.

Para hacer posible el desarrollo del sistema se debió realizar un análisis de las tecnologías utilizadas por la empresa, a fin de determinar la más conveniente.

Se decidió que las herramientas a utilizar generaran un ahorro para la empresa, es por este motivo que se optó por la tecnología JAVA y un motor de base de datos PostgreSQL los cuales son unas de las herramientas openSource más usadas, debido a la disminución significativa en el tiempo de programación y además al poco rendimiento de memoria. El diseño con la arquitectura MVC, permite a futuro agregar funcionalidades al nuevo sistema, debido al desacoplamiento de sus capas, por lo cual es fácil cambiar alguna de ellas, no afectando al sistema completo.

# RECOMENDACIONES

Con respecto a las pruebas resultaron exitosas, ya que, se encontraron diversos errores, siendo una de las más importantes las pruebas de carga o estrés, esta prueba fue de vital importancia para observar las falencias que poseía el sistema y realizar los cambios pertinentes para su mejora.

Por último es importante mencionar que debido a factores de tiempo, existen algunos requerimientos que no pudieron ser abordados en este proyecto, los cuales han sido dejados para trabajos futuros, tales como, comercio electrónico y módulo de finanzas, libro diario.

# BIBLIOGRAFÍA

* LARMAN, Craig . UML y Patrones. Introducción al Análisis y diseño orientado a objetos. 2da. Edición. Prentice Hall.
* LARMAN, Craig. (2003). UML y Patrones. Una Introducción al Análisis y Diseño
* Orientado a Objetos y al Proceso Unificado. 2da. Edición. Prentice Hall.
* PRESSMAN, R. (2002). “Ingeniería del software, un enfoque práctico”. MacGraw – Hill, 5ta edición.
* PRESSMAN, Roger S. (2005). Ingeniería del Software: un enfoque práctico. McGraw – Hill. 6ta. edición.

Anexos