

**Universidad Autónoma del Estado de México**

**Unidad Académica Profesional Tianguistenco**

**Ingeniería en software**

**Unidad de aprendizaje:**

Análisis y Diseño de Software

**Sistema de administración de una tienda super kompras**

Índice

[Introducción 3](#_Toc58873403)

[Planteamiento del problema 4](#_Toc58873404)

[Objetivo específico: 5](#_Toc58873405)

[Objetivos generales: 5](#_Toc58873406)

[Marco teórico 6](#_Toc58873407)

[Diseño de base de datos 9](#_Toc58873408)

[Diccionario de datos. 10](#_Toc58873409)

[Diagrama entidad relación 12](#_Toc58873410)

[Modelo Relacional Normalizado 13](#_Toc58873411)

[Listas de requerimientos 14](#_Toc58873412)

[Requerimientos Funcionales 14](#_Toc58873413)

[Requerimientos no funcionales 16](#_Toc58873414)

[Plantillas caso de uso 18](#_Toc58873415)

[Diagrama de caso de uso 45](#_Toc58873416)

[Diagrama de clases 46](#_Toc58873417)

[Diagrama de objetos 47](#_Toc58873418)

[Diagrama de actividades 48](#_Toc58873419)

[Máquinas de Estado 49](#_Toc58873420)

[Diagramas de secuencia 55](#_Toc58873421)

[Diagramas de comunicación 61](#_Toc58873422)

[Diagrama de paquetes 63](#_Toc58873423)

[Diagrama de componentes 65](#_Toc58873424)

[Diagrama de despliegue 66](#_Toc58873425)

[Conclusiones 67](#_Toc58873426)

[Bibliografía 70](#_Toc58873427)

[Anexos 71](#_Toc58873428)

[Enlace de la carpeta con el video y el programa: 71](#_Toc58873429)

# Introducción

La razón por la cual se elabora es que hay una pérdida exagerada de tiempo, generando un problema dentro de la tienda, pausando actividades a empleados de diferentes sectores; además de generar una modernización en el sistema de la tienda, haciendo que la interacción humano-computadora sea más fácil para cualquier usuario. Se creará un sistema de registro para los productos que se almacenan en el almacén de la tienda. De esta manera, se tendrá un control total del inventario, y así, no tener pérdidas. También ayudará al registro de artículos vendidos, para que cada cajero sea responsable de los artículos que vende y de que dé el monto esperado al final del día acorde a las ventas que tuvo la tienda.

# Planteamiento del problema

El problema que presenta actualmente la tienda SuperKompras #24 es una falla en el control de su inventario interno, es una situación en la cual no hay control sobre los productos que están dentro de la tienda, ya sean productos con caducidad, productos extraviados e incluso aquellos que llegan dañados de alguna manera. La tienda al contar con una bodega de un tamaño muy pequeño no cuenta con el almacenamiento correcto, por lo que muchas veces llegan productos en exceso, y ante esto, la manera de registrar cada producto en el software de inventario de la tienda genera una pérdida de tiempo que es esencial para los trabajadores internos de la tienda. En algunas ocasiones hay productos que no son registrados y que no encuentran dentro del programa, por lo que el personal de la tienda que hacer el registro de manera manual y después traspasar la información de manera digital para ingresarla posteriormente al software correspondiente. La obstrucción en las actividades en los mercaderes (encargados de acomodar los productos dentro de la tienda, atender al cliente dentro de la misma) al tener que hacer el inventario de manera manual, genera conflictos como: mal conteo de productos, falta de reporte de producto dañados, extraviados o que no son aptos para la venta al público, además de que el gerente o subgerente deben de estar presentes para realizar el inventario pertinente, ya que hay productos denominados “productos de alto valor” y al no registrar todos estos productos desde el principio de la llegada de mercancía, puede generar perdidas monetarias a la tienda.

# Objetivo específico:

La finalidad con la que se está realizando este proyecto, es crear un sistema que se encargue de optimizar el registro de inventario, agilizando los procesos al momento de generar y crear el inventario interno de la tienda.

# Objetivos generales:

• General estadísticas de compra venta de productos.

• Jerarquizar los productos más demandados y los menos demandados.

• Tener un control de la entrada y salida de los productos, así como las ganancias de estos.

• Control sobre las personas que puedan acceder al programa.

• Control sobre la cantidad de los productos y el estado en el que llegan.

• Listado de las ventas y productos más solicitados.

• Registro de la cantidad de productos que llegan a la tienda.

• Implementación de tecnología para optimizar los procesos.

• Registro de entradas, salidas y ganancias de los productos periódicamente.

# Marco teórico

Descripción del software

Las acciones que se pueden realizar en cada una de las ventanas dependerán del nivel de usuario, debido a que no todos tienen el mismo acceso en el programa, por motivos de seguridad, para evitar la edición de información fundamental de la tienda.

La primera ventana será el login al programa, el cual tendrá 3 opciones, salir del programa, registrarse o iniciar sesión.

Al momento de registrar un usuario lo primero que se solicitará, será la clave de acceso del gerente, para posteriormente llenar los campos de:

* Nombre y apellidos
* Sexo
* Edad
* Lugar donde vive
* Correo electrónico
* Numero telefónico
* Cargo

Cada uno de estos campos será obligatorio para el registro de todos los usuarios sin importar el cargo.

La segunda ventana para la ventana del gerente podrá observar la lista de proveedores y productos, cada una de ellas tendrá la opción de dar de altas, bajas, consultar o modificar productos o proveedores, ya que es el usuario con mas opciones dentro del programa y en cada acción deberá de colocar su clave de acceso para poder realizar la acción que el desea.

En el caso de iniciar sesión con el subgerente, solamente podrá consultar o dar de alta productos o proveedores, sin poder modificar ningún otro apartado dentro del programa.

Para el usuario mas bajo que en este caso es el mercadeador, solamente podrá hacer consultas de productos o proveedores, ya que no podrá modificar datos de ninguna manera.

Al momento de alta algún producto o proveedor solamente se podrá registrar con la fecha en el momento en el que se realiza esta acción, no se podrá modificar de ninguna manera, aunque el mayor usuario (gerente) quiera modificarlos, el sistema debe de dar la fecha sin modificación alguna.

Al momento de dar de baja algún producto o proveedor solamente será de manera lógica, ya que se necesitará un registro previo de cada uno de estos, para que siga existiendo en la base de datos de manera inactiva.

Al momento en que usuario se encuentra en estado “inactivo” no tendrá acceso de ninguna manera al programa, solamente aparecerá en la base de datos, ya que los encargados de esta área deben de llevar un registro de empelados y productos.

La base de datos se debe de actualizar cada 24 hrs para evitar la perdida de datos importantes del sistema y evitar problemas a mayor escala.

El gerente siempre estará notificado de las acciones que hagan otros usuarios para evitar la alteración de información, en caso de que se realice la modificación de un producto o proveedor, se debe de registrar el usuario que hizo el cambio, además de agregar la fecha y hora de esta acción.

En caso de haber un error con el programa (error de conexión a la base de datos, error de inicio de sesión, etc) se deberá de notificar lo mas pronto posible al equipo encargado que le dará mantenimiento al software.

Diagrama

Descripción generada automáticamente

Uso de java

La razón por la cual todo se desarrollará en Java, será por su flexibilidad con el sistema operativo con el que se esta trabajando (Windows) para evitar la compra de alguna licencia de otro SO, o para evitar problemas con la interacción con el usuario, ya que windows suele ser el sistemas mas usado y sobre todo mas fácil de usar.

Además, el lenguaje, contiene librerías y demás opciones de programación que harán más rápido el desarrollo del software, y otra de las grandes ventajas de programar en este lenguaje, es la programación en 3 capas: interfaz, código y base de datos, agilizando el tiempo de programación de manera considerada y sobre todo sin problemas entre los desarrolladores.

Uso de documentación Dorcur

Documentación de Requerimientos Centrada en el Usuario, es un tipo de documentación lo suficientemente entendible para el desarrollo del sistema, ya que toca puntos importantes en la revisión e implementación de los requerimientos.

Para la elaboración de cada uno de los puntos que toca la documentación DorCu, se realizó anteriormente en el proyecto y actualmente sirvió para la rectificación de los mismos.

Metodología

Todas las interfaces deben de llevar la misma estética, como el uso de los colores de la empresa, tipografía y los iconos deben de estar adecuados para que el usuario pueda entender el programa y que sea mas sencilla la interacción entre usuario-computadora.

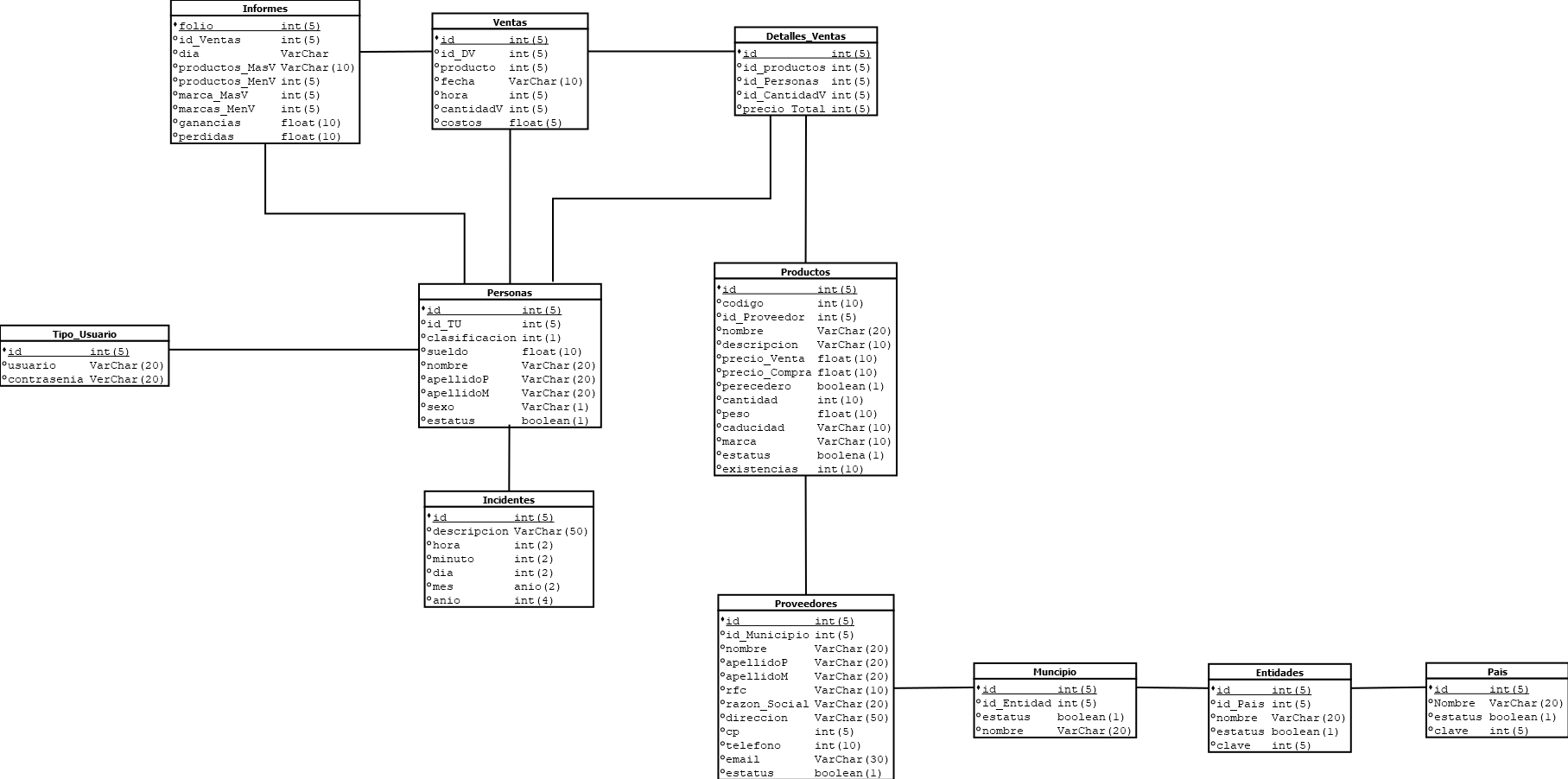
Además, la base de datos debe de estar bien conectada con el almacén central para evitar la fuga de datos al momento de solicitar un producto, o notificar sobre la perdida de estos.

En el caso de los proveedores, solamente se podrán modificar los datos en la tienda, ya que no son los mismos en todas las tiendas, y para evitar conflictos mayores, solamente llegaran los datos a la central de la base de datos, con los datos correspondientes del proveedor. Esto mismo pasara con la acción realizada en los productos.

Al momento de generar el respaldo de la base de datos, predeterminara una hora, por lo que en caso de que algún usuario este realizando una actividad le saldrá una notificación en pantalla, para que pueda guardar las acciones que se encuentra realizando.

El uso del lenguaje de programación en java podrá permitir un mejor proceso de desarrollo para los programadores, ya que permite hacer ciertas tareas con comando facilitando las tareas y reduciendo en tiempos y costos, además de poder verificar errores antes del lanzamiento final del software.

# Diseño de base de datos



# Diccionario de datos.

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tamaño | Type | Descripción |
| Tipos\_Usuarios | | | |
| Id | 5 | Int | Identificador del usuario |
| Usuario | 20 | VarChar | Nombre de usuario |
| Contraseña | 20 | VarChar | Contraseña del respectivo usuario |
| Estatus | 1 | Boolean | Activo o inactivo |
| Personas | | | |
| Id | 5 | Int | Identificador de la persona |
| Clasificación | 1 | Int | 1=Gerente, 2=Subgerente, 3=Proveedores |
| Sueldo | 10 | Float | Lo que gana la persona |
| Nombre | 20 | VarChar | Nombre de la persona |
| ApellidoP | 20 | VarChar | Apellido paterno |
| ApellidoM | 20 | VarChar | Apellido Materno |
| Sexo | 1 | Boolean | Sexo de la persona 1=Mujer, 2=Hombre |
| Estatus | 1 | Boolean | Activo o Inactivo |
| Incidentes1 | | | |
| id | 5 | Int | Identificador del reporte |
| Descripcion | 150 | String | Descripcion del reporte |
| hora | 2 | int | Hora en la que se dio de alta el reporte |
| minuto | 2 | int | Minuto en la que se dio de alta el reporte |
| fecha | 10 | String | Fecha en la que se dio de alta el reporte |
| Ventas | | | |
| Id | 5 | Int | Identificador de las ventas |
| Fecha | 10 | String | Fecha que se realizó las ventas |
| Hora | 5 | String | Hora en que se realizó las ventas |
| Costos | 5 | Float | Costos de las ventas |
| Detalles\_Ventas | | | |
| Id | 5 | Int | Identificador de las ventas |
| Id\_Ventas | 5 | Int | Identificador de productos |
| Id\_Productos | 5 | Int | Identificador de la persona que realizo la venta |
| CantidadV | 5 | Int | Cantidad por pieza vendida del producto |
| pesoV | 5 | Float | Cantidad por kg o gramos vendida del producto |
| Precio\_Tota | 5 | Float | Precio total de la venta |
| Productos | | | |
| Id | 5 | Int | Identificador de los productos |
| Código | 10 | Int | Código del producto en sistema |
| Id\_Proveed | 5 | Int | Identificador del proveedor |
| Nombre | 20 | VarChar | Nombre del producto |
| Descripcion | 10 | VarChar | Descripción del producto |
| Prec\_Venta | 10 | Float | Precio de venta del producto |
| Prec\_Comp | 10 | Float | Precio de compra del producto |
| Perecedero | 1 | Boolean | Perecedero Falso o verdadero |
| Cantidad | 10 | Int | Dato si el producto es por cantidad |
| Peso | 10 | Float | Dato si el producto es por peso |
| Caducidad | 10 | VarChar | Caducidad del producto si es que tiene |
| Marca | 20 | VarChar | Marca del producto |
| Estatus | 1 | Boolean | Activo o Inactivo |
| Existencias | 10 | Int | Existencias del producto |
| Proveedores | | | |
| Id | 5 | Int | Identificador del proveedor |
| Id\_Mun | 5 | Int | Identificador del municipio |
| Nombre | 20 | VarChar | Nombre del proveedor |
| ApellidoP | 20 | VarChar | Apellido paterno |
| ApellidoM | 20 | VarChar | Apellido materno |
| RFC | 10 | VarChar | Rfc del proveedor |
| Razón\_Soc | 20 | VarChar | Razón social del proveedor |
| Direccion | 50 | VarChar | Dirección del proveedor |
| CP | 5 | Int | Código postal |
| Telefono | 10 | Int | Teléfono para comunicarse con el proveedor |
| Email | 30 | VarChar | Email para comunicarse con el proveedor |
| Estatus | 1 | Boolean | Activo o Inactivo |
| Municipio | | | |
| Id | 5 | Int | Identificador del municipio |
| Id\_Entidad | 5 | Int | Identificador de la entidad |
| Estatus | 1 | Boolean | Activo o Inactivo |
| Nombre | 20 | VarChar | Nombre del municipio |
| Entidad | | | |
| Id | 5 | Int | Identificador entidad |
| Id\_Pais | 5 | Int | Identificador del país |
| Nombre | 20 | VarChar | Nombre de la entidad |
| Estatus | 1 | Boolean | Inactivo o Activo |
| País | | | |
| Id | 5 | Int | Identificador del país |
| Nombre | 20 | VarChar | Nombre del país |
| Estatus | 1 | Boolean | Inactivo o Activo |

# Diagrama entidad relación

Imagen que contiene electrónica, oscuro, pantalla, cerca

Descripción generada automáticamente

# Modelo Relacional Normalizado

# Listas de requerimientos

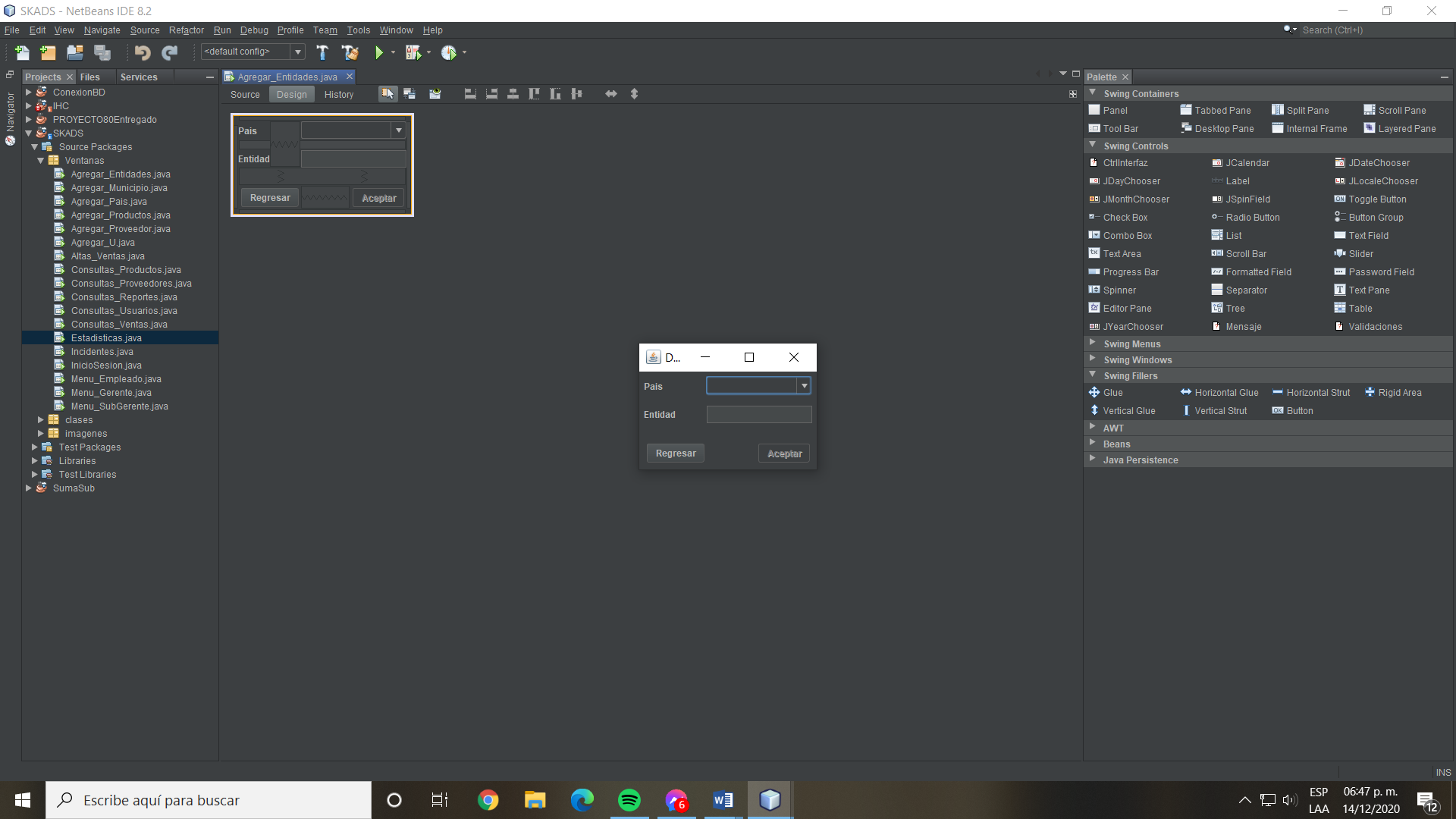
## Requerimientos Funcionales

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requerimiento | Descripción | Estatus | Observaciones | Diccionario de datos | Interfaz | Tabla de base de datos |
| 1 | El sistema controlará el acceso y lo permitirá solamente a usuarios autorizados. Los usuarios deben ingresar al sistema con un nombre de usuario y contraseña. | Satisfecho | La interfaz ya tiene los campos requeridos para diferenciar entre gerente y empleado, aunque falta añadir subgerente. | Gerente  Subgerente Empleado | Iniciar Sesión | Tabla Personas |
| 2 | El sistema enviará una alerta al administrador del sistema cuando ocurra alguno de los siguientes eventos: Registro de nueva cuenta, ingreso al sistema por parte del cliente, 2 o más intentos fallidos en el ingreso de la contraseña de usuario y cambio de contraseña de usuario. | Satisfecho | Esta opción al ser backend todavía no puede ser implementado hasta que todas las interfaces y bases de datos estén bien interconectadas. | -------- | Iniciar Sesión | Tablas incidentes |
| 3 | El apartado de monto acepta únicamente valores numéricos con dos decimales. | Satisfecho | En la interfaz se verifica los datos que son metidos en precio y verifica las reglas léxicas de los precios “0.0”. | Productos | Registro de productos | Tabla de productos |
| 4 | La pantalla de registro de factura puede imprimir los datos en pantalla a la impresora. | Satisfecho | En la interfaz de cobro aún falta generar el pdf de la impresora. | ------- | Satisfecho | Tabla de Informes |
| 5 | El apartado fecha de transacción acepta únicamente fecha de este día. | Satisfecho | Se ha desarrollado la interfaz de ventas | ------- | Satisfecho | Tabla ventas |
| 6 | La pantalla de registro de pago puede imprimir los datos en pantalla a la impresora. | Satisfecho | Se ha desarrollado el apartado de ventas. | Gerente  Subgerente  Empleado | Ventas | Tablas informes |
| 7 | El sistema controlará el acceso y lo permitirá solamente a usuarios autorizados. | Satisfecho | El sistema ya verifica usuarios. Pero usando solo datos guardados en archivo binario, pero la adaptación a base de datos no es problema. | Gerente  Subgerente Empleado | Iniciar sesión | Tabla Personas |
| 8 | El sistema enviará una alerta al administrador del sistema cuando ocurra alguno percance o distintas actividades. | Satisfecho | Se tiene la conexión adecuada para poder realizar esta alerta. | Gerente | Gerente | Tabla de gerente |
| 9 | El sistema hará las estadísticas de los productos más vendidos y menos vendidos. | Satisfecho | Ya tenemos el modelo matemático solo falta implementarlo y hacer los ajustes requeridos | Informe | Estadísticas | Tabla de informe |
| 10 | Los productos serán catalogados según su tipo, marca, precio. | Satisfecho | Al dar de alta un producto se tiene que registrar estos datos. | Productos | Registro de productos | Tabla productos |
| 11 | Habrá una lista de proveedores donde cada uno tendrá su sub lista de productos. | Satisfecho | Ya tenemos el modelo del objeto proveedores, solo falta añadirlo y conectarlo con los otros. | Proveedores | Proveedores | Tabla de proveedores |
| 12 | Todas las clases (empleados, gerentes, subgerentes, productos, proveedores) tendrán estatus de activo o inactivo. | Satisfecho | Todos cuentan ya con su propiedad de activo o inactivo | Gerente  Subgerente Empleado | Registro de usuarios | Tabla Personas |
| 13 | El estatus inactivo tendrá la función de no mostrar esa clase en las consultas a menos que se deba ver. | Satisfecho | La conexión a la base de datos se encuentra conectada, por lo que el usuario en estado “inactivo” no puede ingresar al sistema | Gerente  Subgerente Empleado | Ver usuarios | Tabla Personas |

## Requerimientos no funcionales

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requerimiento | Descripción | Estatus | Observaciones | Diccionario de datos | Interfaz | Tabla de base de datos |
| 1 | El apartado de monto acepta únicamente valores numéricos con dos decimales. | Satisfecho | Se añadió el apartado de precio de venta y precio de compra.  Se agregaron en las ventanas de los productos (altas, modificaciones). | Se encuentra unida al diccionario de datos | Altas y modificaciones de productos | Tabla de productos  Tabla de Ventas |
| 2 | El sistema controlará el acceso y lo permitirá solamente a usuarios autorizados. | Satisfecho | El sistema ya verifica usuarios. Pero usando solo datos guardados en archivo binario, pero la adaptación a base de datos no es problema. | Gerente  Subgerente Empleado | Inicio de sesión | Tabla de incidentes |
| 3 | El sistema enviará una alerta al administrador del sistema cuando ocurra alguno percance o distintas actividades. | Satisfecho | Conexión adecuada a la base de datos. | ------- | Incidentes | Tabla de incidentes |
| 4 | Tendrá un pequeño costo el sistema | Satisfecho | El sistema se encuentra en funcionamiento. | ------- | NO APLICA | ------- |
| 5 | Será fácil el uso del sistema | Satisfecho | El sistema está desarrollado de tal manera que el usuario lo use de manera fácil y concreta. | ------- | Todas las interfaces | ------- |
| 6 | El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 1 hora. | Satisfecho | Aun no podemos medir el tiempo que tarde un usuario en aprender el uso del sistema. | ------- | Todas las interfaces | -------- |
| 7 | El sistema debe contar con manuales de usuario estructurados adecuadamente | Satisfecho | Debemos terminar el sistema al ciento por ciento para poder hacer un manual actualizado de todas las funciones del sistema. | ------- | NO APLICA | ------- |
| 8 | El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final. | Satisfecho | Los avisos se encuentran realizados, y aparecen conforme al error. | ------- | Satisfecho | ------- |
| 11 | El sistema enviara notificaciones cuando el producto este en el almacén o la venta | Satisfecho | La conexión se encuentra lista para mostrar la cantidad de productos | Productos | Satisfecho | Tabla de productos |
| 12 | Los permisos de acceso al sistema podrán ser cambiados solamente por el gerente de acceso a datos. | Satisfecho | Solo el apartado de gerente tiene estas opciones especiales. | Gerente | Ventana gerente | Tabla Personas  Tabla tipo de usuario  Tabla gerente |
| 14 | Todos los sistemas deben respaldarse cada 24 horas | Satisfecho | Implementado el respaldo. | ------- | Todas las interfaces | ------- |
| 15 | El sistema debe poseer interfaces gráficas bien formadas. | Satisfecho | Las interfaces tiene un diseño intuitivo para cada tipo de usuario | ------- | Todas las interfaces | ------- |

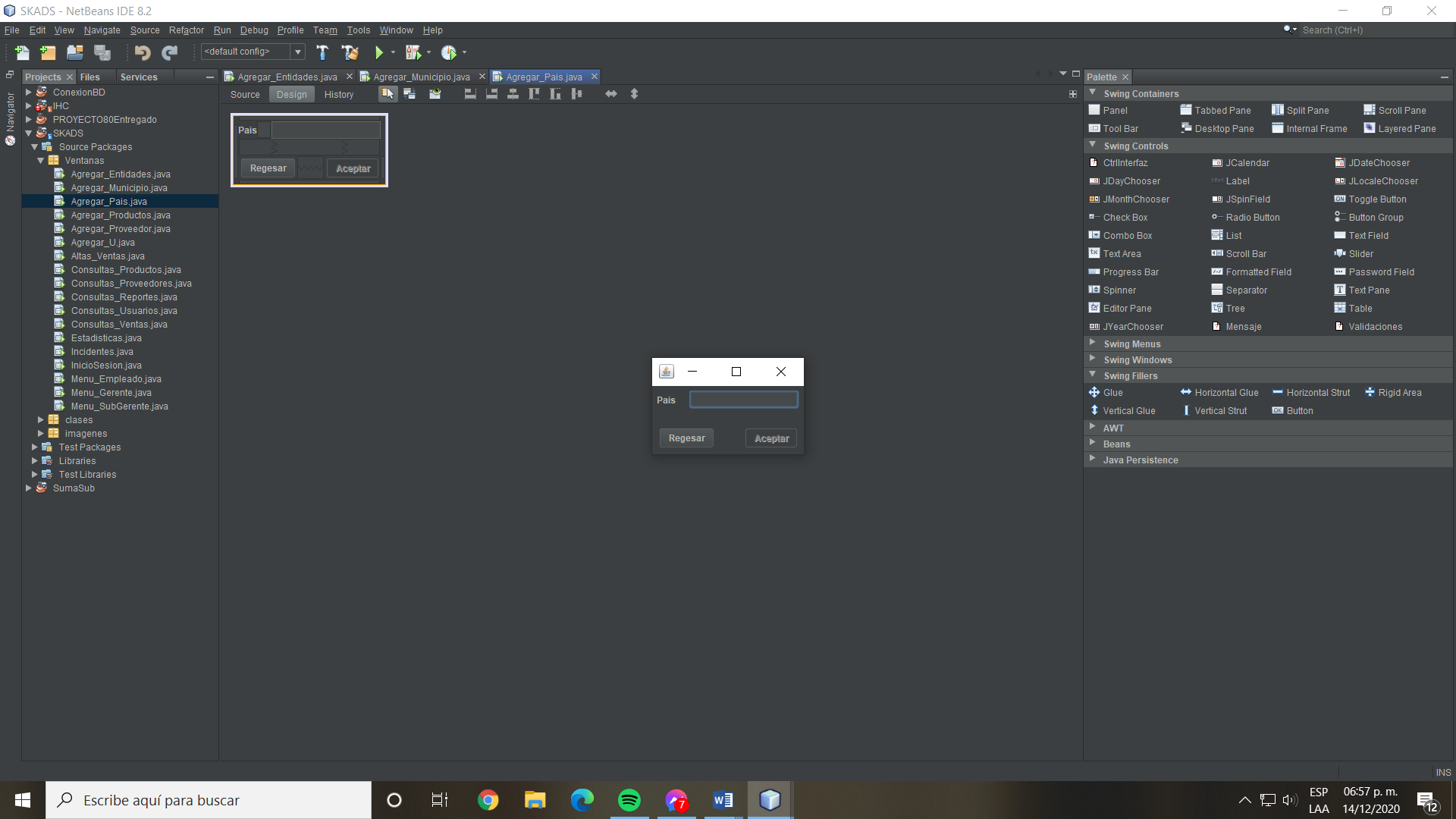
# Plantillas caso de uso



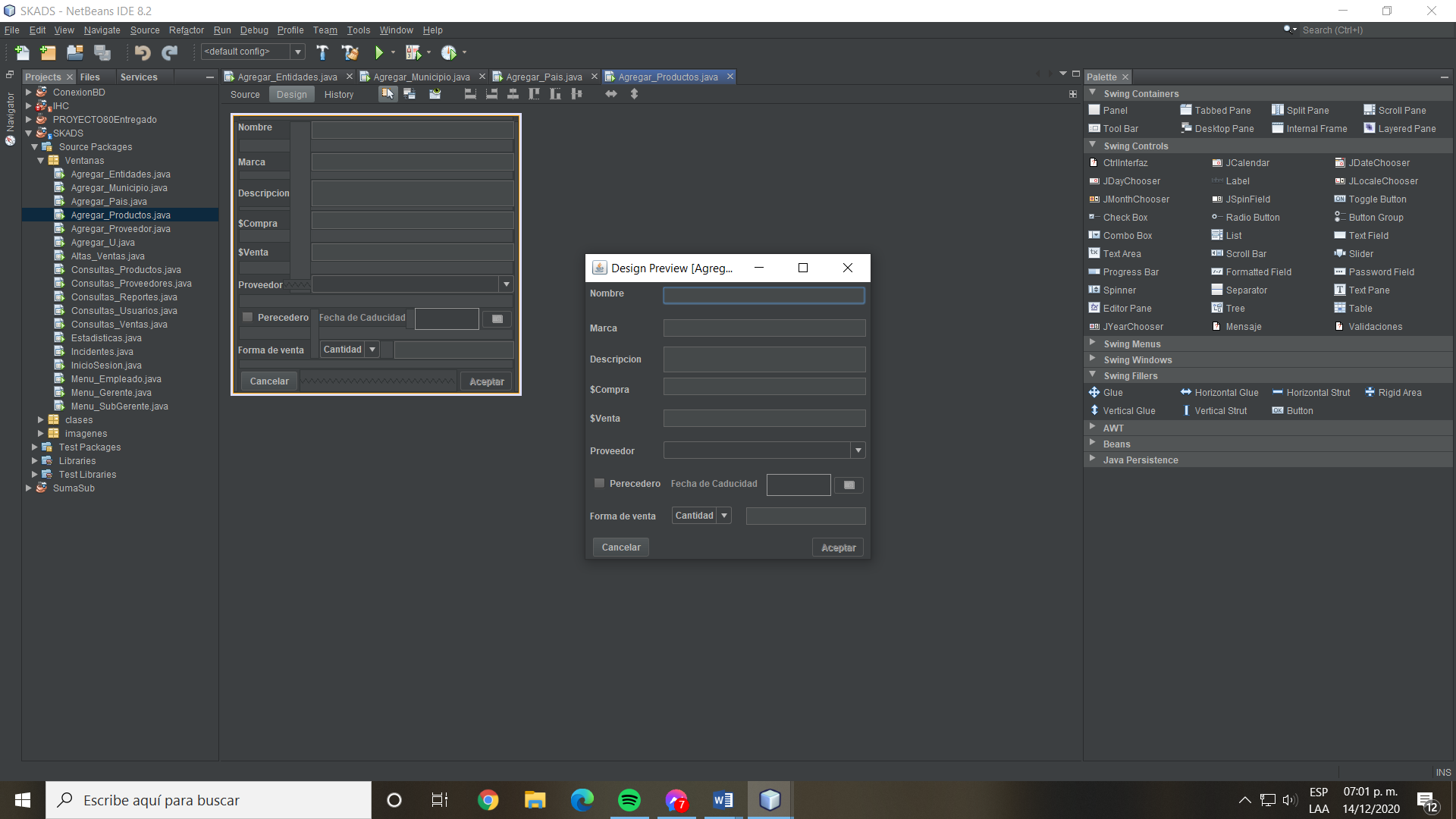
|  |  |  |  |
| --- | --- | --- | --- |
| **RF- <11>** | | <Agregar País-Proveedores> | |
| **Versión** | | <0.2> | |
| **Autores** | | <Kevin Jesús, Gerardo Pastrana, Julio Estrada> | |
| **Fuentes** | | <NetBeans 8.2, GitHub> | |
| **Objetivos asociados** | | <Agrega las entidades> | |
| **Descripción** | | En esta interfaz, se dan de alta la entidad de donde pertenece el proveedor, de acuerdo al país. | |
| **Precondición** | | <Se debe iniciar sesión como administrador y haber registrado un país anteriormente.> | |
| **Secuencia Normal** | |  | |
|  | **Actor** | **Sistema** | |
| **1** | **Gerente** | El gerente selecciona el país al que corresponde la entidad. | |
| **2** |  | El gerente coloca el nombre de la entidad | |
| **3** |  | El gerente da aceptar y se registra la entidad en la Base de datos. | |
| **Excepciones** | |  | |
|  | **Actor** | **Sistema** | |
| **1.1** | **Gerente** | El sistema no permite que se puedan ingresar números. | |
| **2.1** |  | En caso de cancelar, se limpiara la entidad y no se modificara la base de datos. | |
| **Rendimiento** | | **Paso** | **Cota de tiempo** |
|  | | 1 | 1 segundos |
|  | | 2 | 3 segundos |
|  | | 3 | 2 segundos |
| **Frecuencia esperada** | | <1º de veces> veces / <5s> | |
| **Importancia** | | { vital} | |
| **Post-condición** | | <Regresa al menú del gerente.> | |
| **Urgencia** | | {hay presión} | |
| **Comentarios** | | <Se encuentra validado para evitar números y los espacios los elimine en la base de datos para agregarse adecuadamente.> | |



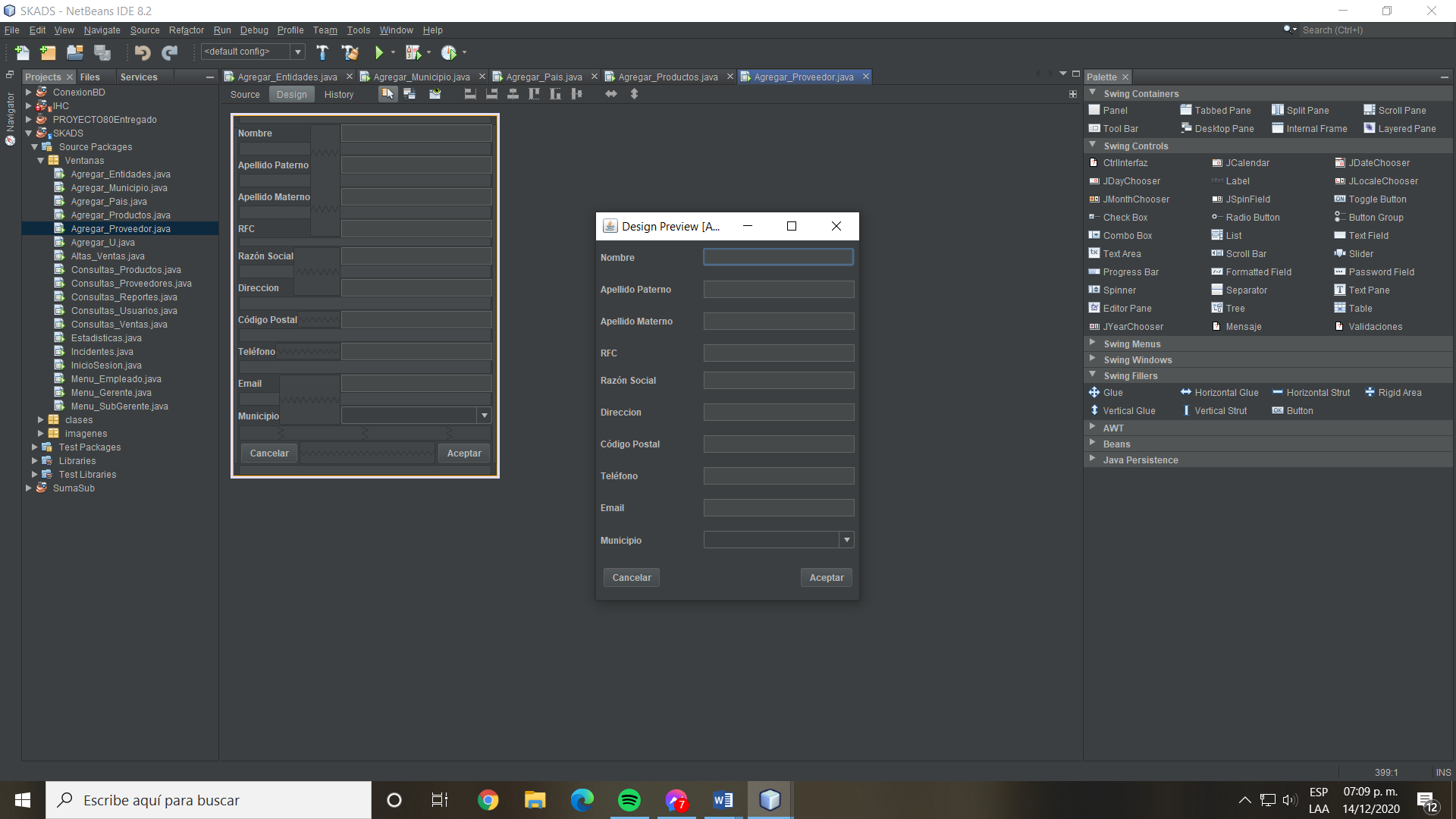
|  |  |  |  |
| --- | --- | --- | --- |
| **RF- <11>** | | <Agregar entidad de proveedor> | |
| **Versión** | | <0.2> | |
| **Autores** | | <Kevin Jesús, Gerardo Pastrana, Julio Estrada> | |
| **Fuentes** | | <NetBeans 8.2, GitHub> | |
| **Objetivos asociados** | | <Agrega los municipios> | |
| **Descripción** | | En esta interfaz, se dan de alta de municipios de donde pertenece el proveedor, de acuerdo a la entidad. | |
| **Precondición** | | <Se debe iniciar sesión como administrador y haber registrado un país y una entidad anteriormente.> | |
| **Secuencia Normal** | |  | |
|  | **Actor** | **Sistema** | |
| **1** | **Gerente** | El gerente selecciona la entidad al que corresponde la entidad. | |
| **2** |  | El gerente coloca el nombre de la entidad | |
| **3** |  | El gerente da aceptar y se registra la entidad en la Base de datos. | |
| **Excepciones** | |  | |
|  | **Actor** | **Sistema** | |
| **1.1** | **Gerente** | El sistema no permite que se puedan ingresar números. | |
| **2.1** |  | En caso de cancelar, se limpiara el municipio y no se modificara la base de datos. | |
| **Rendimiento** | | **Paso** | **Cota de tiempo** |
|  | | 1 | 1 segundos |
|  | | 2 | 3 segundos |
|  | | 3 | 2 segundos |
| **Frecuencia esperada** | | <1º de veces> veces / <5s> | |
| **Importancia** | | { vital} | |
| **Post-condición** | | <Regresa al menú del gerente.> | |
| **Urgencia** | | {hay presión} | |
| **Comentarios** | | <Se encuentra validado para evitar números y los espacios los elimine en la base de datos para agregarse adecuadamente.> | |



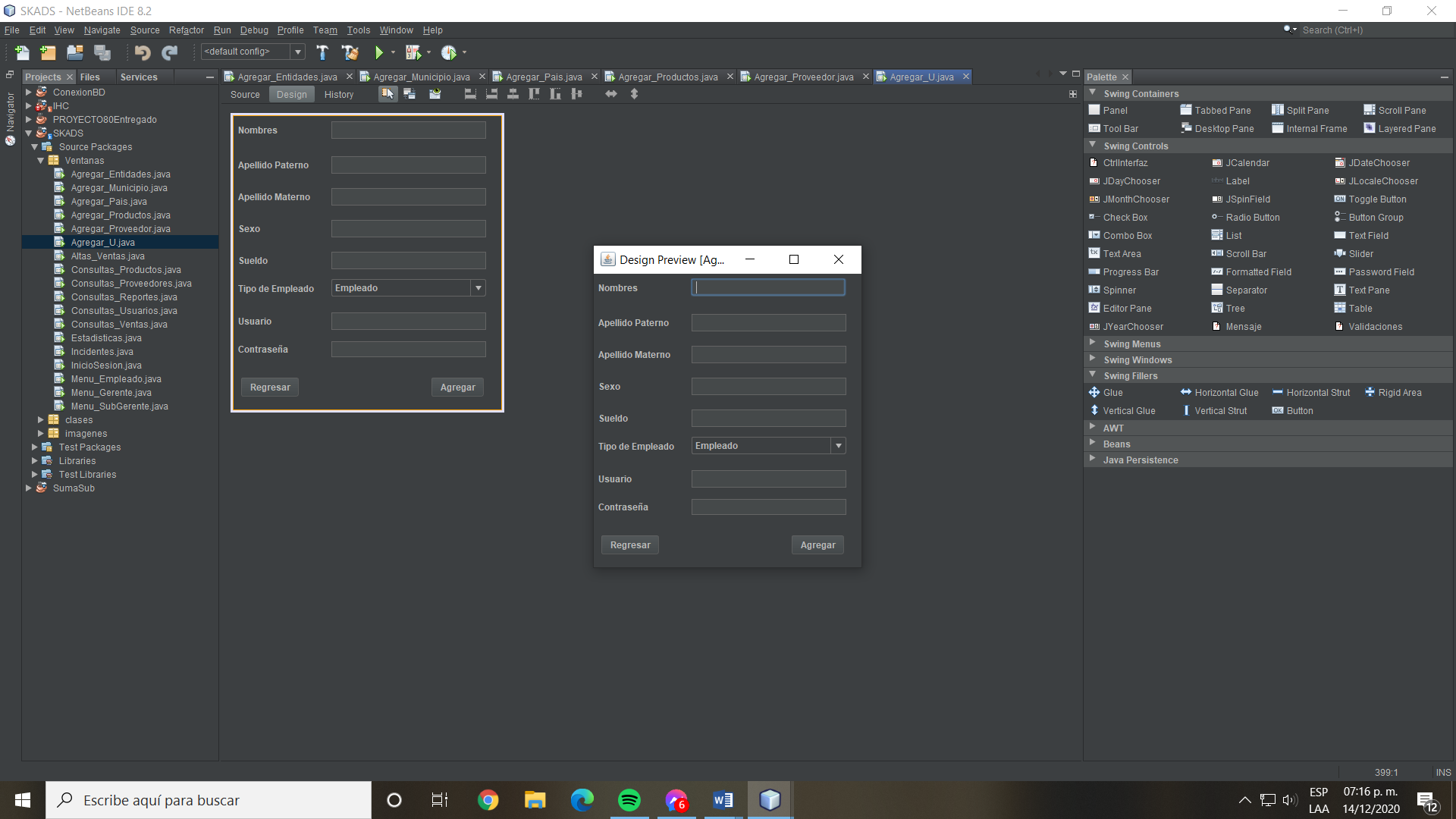
|  |  |  |  |
| --- | --- | --- | --- |
| **RF- <11>** | | <**Lista de proveedores**> | |
| **Versión** | | <0.2> | |
| **Autores** | | <Kevin Jesús, Gerardo Pastrana, Julio Estrada> | |
| **Fuentes** | | <NetBeans 8.2, GitHub> | |
| **Objetivos asociados** | | <Agrega el pais> | |
| **Descripción** | | En esta interfaz, se dan de alta del país de donde pertenece el proveedor, de acuerdo a la entidad. | |
| **Precondición** | | <Se debe iniciar sesión como gerente.> | |
| **Secuencia Normal** | |  | |
|  | **Actor** | **Sistema** | |
| **1** | **Gerente** | El gerente coloca el nombre de la entidad | |
| **2** |  | El gerente da aceptar y se registra la entidad en la Base de datos. | |
| **Excepciones** | |  | |
|  | **Actor** | **Sistema** | |
| **1.1** | **Gerente** | El sistema no permite que se puedan ingresar números. | |
| **2.1** |  | En caso de cancelar, se limpiara el país y no se modificara la base de datos. | |
| **Rendimiento** | | **Paso** | **Cota de tiempo** |
|  | | 1 | 1 segundos |
|  | | 2 | 3 segundos |
|  | | 3 | 2 segundos |
| **Frecuencia esperada** | | <1º de veces> veces / <5s> | |
| **Importancia** | | { vital} | |
| **Post-condición** | | <Regresa al menú del gerente.> | |
| **Urgencia** | | {hay presión} | |
| **Comentarios** | | <Se encuentra validado para evitar números y los espacios los elimine en la base de datos para agregarse adecuadamente.> | |



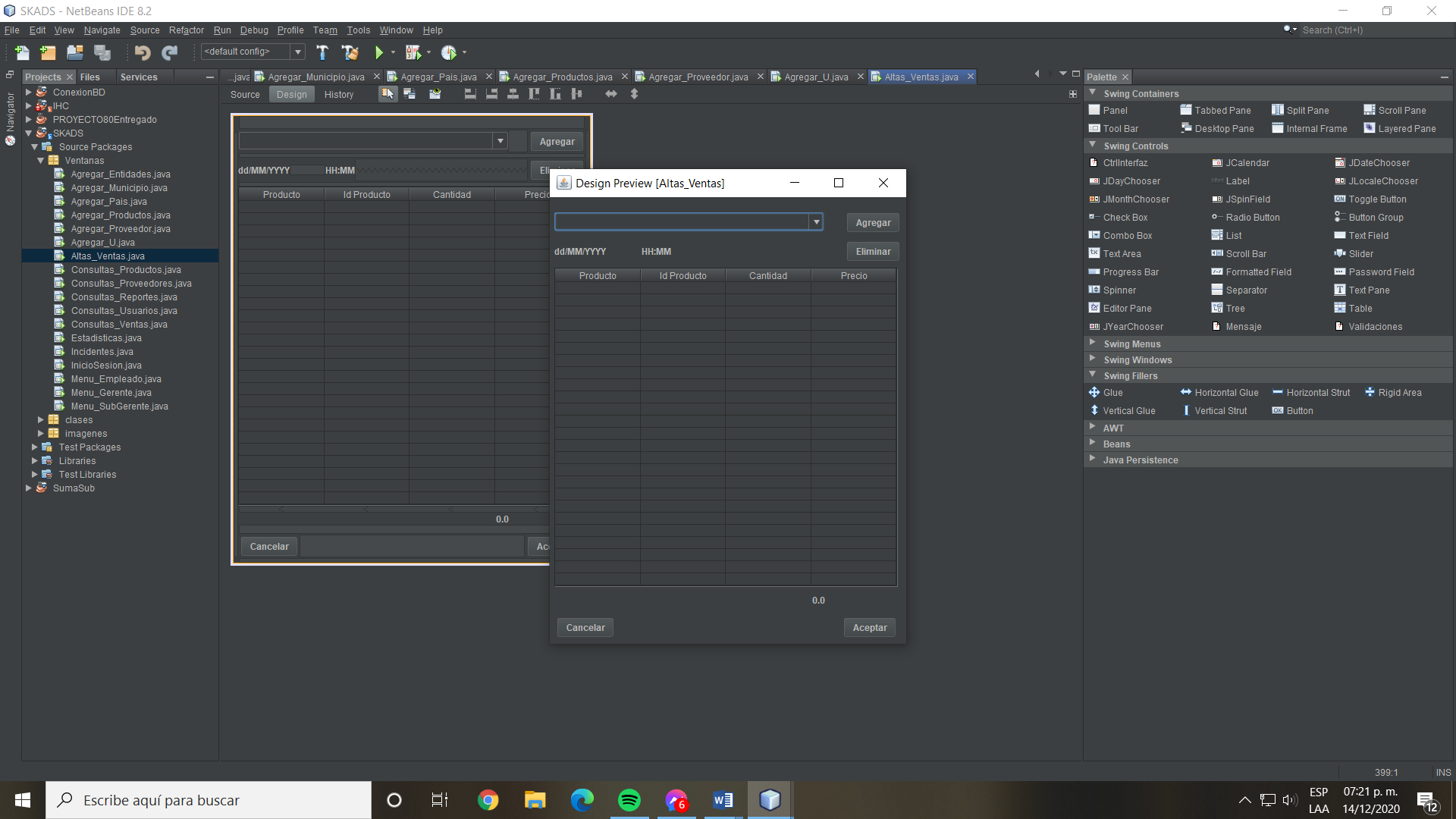
|  |  |  |  |
| --- | --- | --- | --- |
| **RF- <** RF3, RF10**>** | | RF3, RF10 | |
| **Versión** | | Versión 0.2 17/11/2020 | |
| **Autores** | | Julio Piña, Kevin Escutia, Gerardo Pastrana | |
| **Fuentes** | | Java netbeans 8.2, github. | |
| **Objetivos asociados** | | Registrar los productos | |
| **Descripción** | | Este sistema registrara los productos, por tipo de nombre, marca, cantidad proveedor, por cantidad o peso y si es perecedero o no. | |
| **Precondición** | | El gerente o subgerente debe iniciar sesión y elegir dar de alta un nuevo producto. | |
| **Secuencia Normal** | | | |
|  | **Actor** | **Sistema** | |
| **1** | **Gerente** | El gerente accede a la ventada de alta de productos. | |
| **2** | **Gerente** | El gerente rellena los datos necesarios | |
| **3** | **Sistema** | El gerente elige si es perecedero o no. | |
| **3.1** | **Gerente** | Si es perecedero se elige la fecha que caduca. | |
| **4** | **Gerente** | El gerente guarda los datos. | |
| **5** | **Sistema** | Sistema guarda los cambios en la base de datos. | |
| **Excepciones** | | Se debe seleccionar un proveedor, si es perecedero o no y si se vende por cantidad o por peso. | |
|  | **Actor** | **Sistema** | |
| **1.1** |  |  | |
| **2.1** |  | Los datos se deben rellenar de acuerdo al tipo de variable. | |
| **3.1** |  | En caso de que no ser perecedero se mandara automáticamente no perecedero. | |
| **Rendimiento** | | **Paso** | **Cota de tiempo** |
|  | | 1 | 0 |
|  | | 2 | 40 s. |
|  | | 3 | 1 s. |
|  | | 4 | 4 s. |
|  | | 5 | 5 s. |
| **Frecuencia esperada** | | nº de veces, 50 s. | |
| **Importancia** | | Importante | |
| **Post-condición** | | Se regresa al menú de Gerente o de subgerente de acuerdo al inicio de sesión. | |
| **Urgencia** | | Hay presión | |
| **Comentarios** | | Mayúsculas identificadas. | |



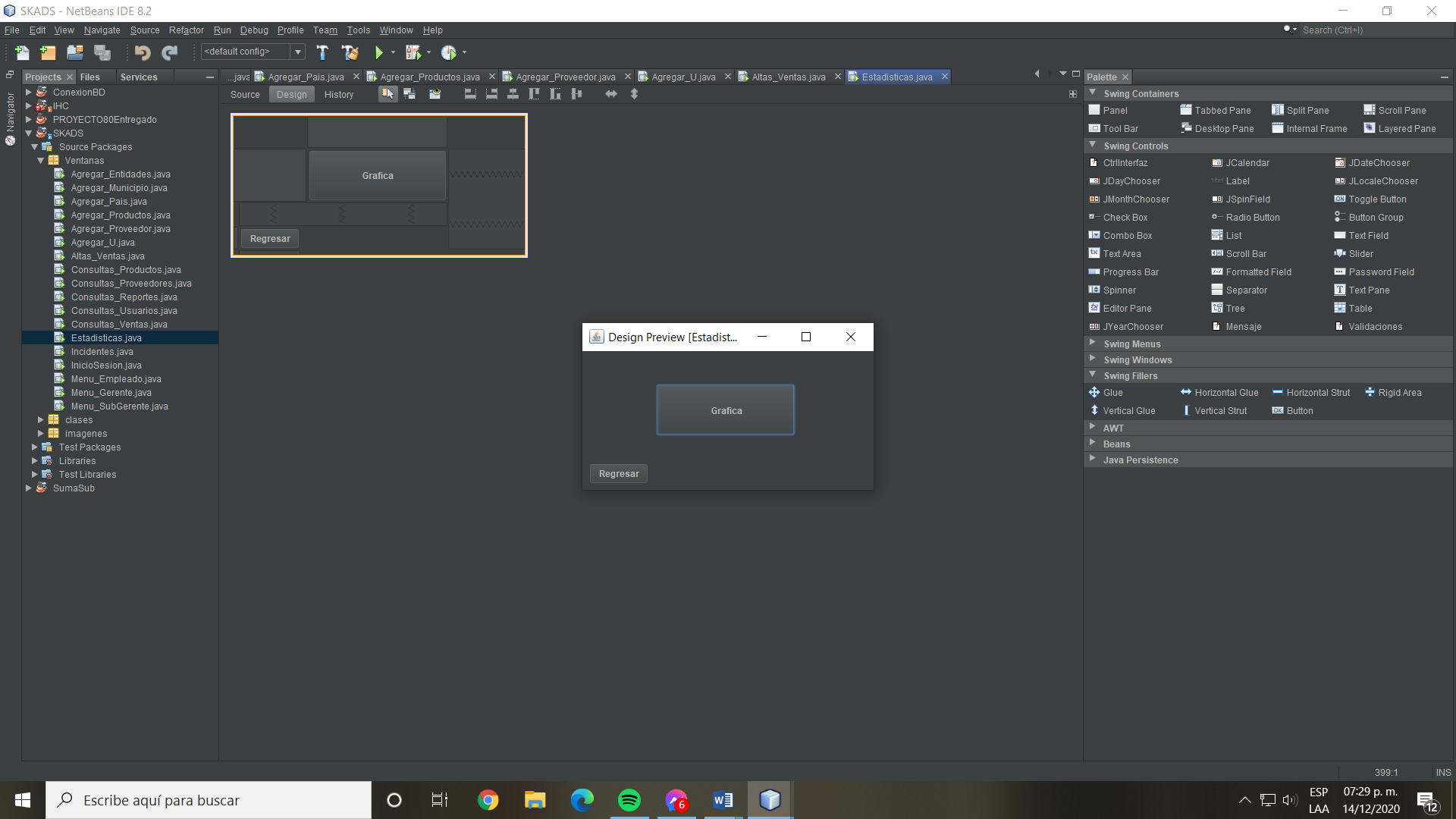
|  |  |  |  |
| --- | --- | --- | --- |
| **RF- <11>** | | <**11 Lista de proveedores**> | |
| **Versión** | | <0.2> | |
| **Autores** | | <Kevin Jesús, Gerardo Pastrana, Julio Estrada> | |
| **Fuentes** | | <NetBeans 8.2, GitHub> | |
| **Objetivos asociados** | | <Agregar un nuevo proveedor> | |
| **Descripción** | | Agregara un nuevo proveedor a la base de datos. | |
| **Precondición** | | <Se debe iniciar sesión como gerente y debe de haber un registro de país, entidad y municipio para agregar.> | |
| **Secuencia Normal** | |  | |
|  | **Actor** | **Sistema** | |
| **1** | **Gerente** | El gerente llena los datos de las cajas | |
| **2** |  | El gerente da clic en botón aceptar | |
| **3** |  | El sistema enviara los datos a la base de datos | |
| **Excepciones** | |  | |
|  | **Actor** | **Sistema** | |
| **1.1** | **Gerente** | El sistema validara que estén correctamente llenos los campos para poder continuar. | |
| **2.1** |  | En caso de cancelar no se modificara la base de datos | |
| **Rendimiento** | | **Paso** | **Cota de tiempo** |
|  | | 1 | 40 segundos |
|  | | 2 | 3 segundos |
|  | | 3 | 5 segundos |
| **Frecuencia esperada** | | <1º de veces> veces / <48 segundos> | |
| **Importancia** | | { vital} | |
| **Post-condición** | | <Regresara al menú de gerente> | |
| **Urgencia** | | {hay presión} | |
| **Comentarios** | | <La casilla de email cuenta con validación de correo real, los datos del municipio serán enviados automáticamente.> | |



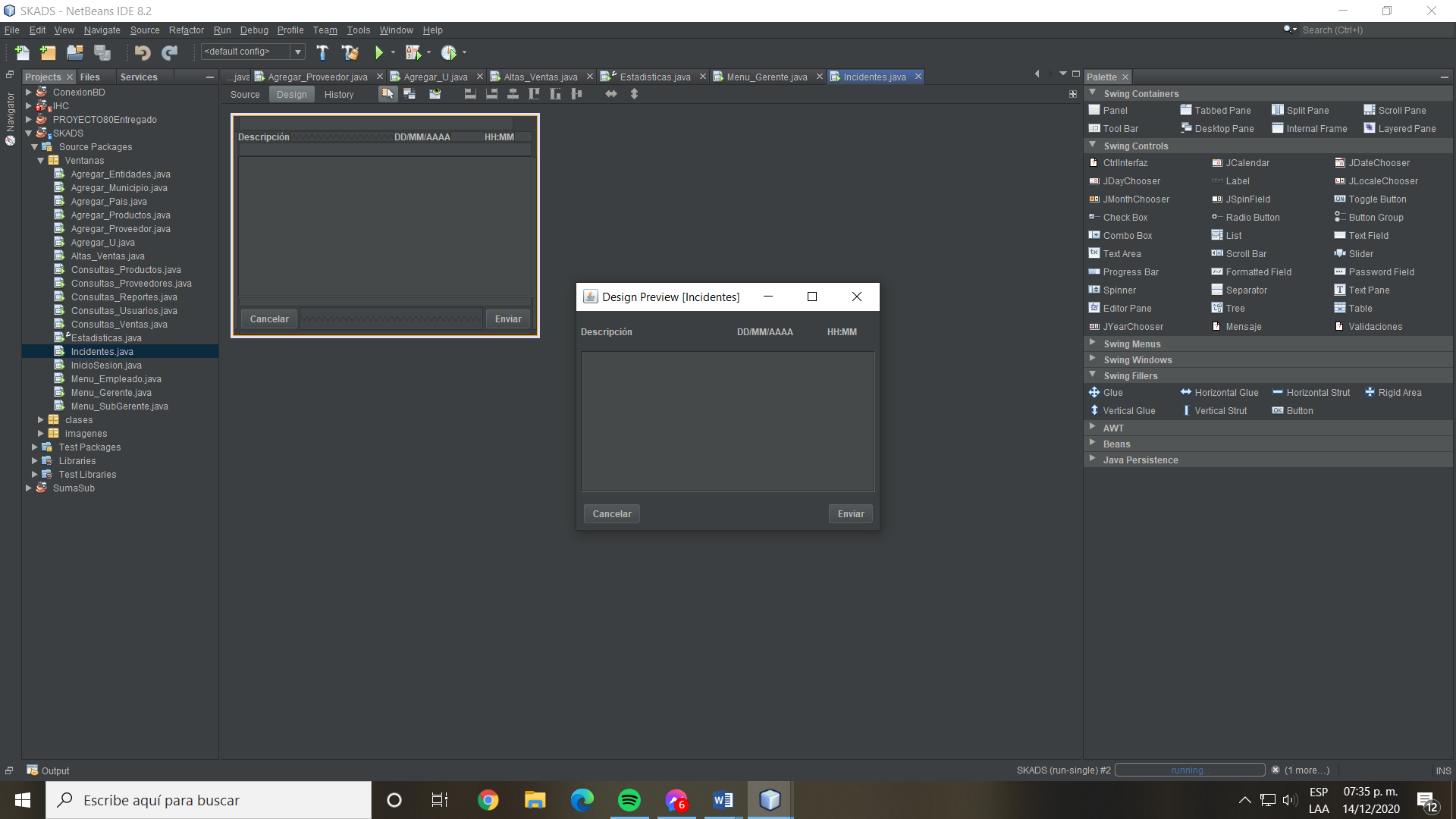
|  |  |  |  |
| --- | --- | --- | --- |
| **RF- <11>** | | <**Lista de proveedores**> | |
| **Versión** | | <0.2> | |
| **Autores** | | <Kevin Jesús, Gerardo Pastrana, Julio Estrada> | |
| **Fuentes** | | <NetBeans 8.2, GitHub> | |
| **Objetivos asociados** | | <Agregar Usuario> | |
| **Descripción** | | Habrá una lista de proveedores donde cada uno tendrá su sub lista de productos. | |
| **Precondición** | | <Se debe iniciar sesión como gerente y seleccionar dar de alta un nuevo empleado.> | |
| **Secuencia Normal** | |  | |
|  | **Actor** | **Sistema** | |
| **1** | **Gerente** | El gerente llena los datos de las cajas | |
| **2** |  | El gerente selecciona el nivel del nuevo usuario (empleado, subgerente o gerente) | |
| **3** |  | El gerente da aceptar | |
| **4** |  | Se guardan los datos en la base de datos. | |
| **Excepciones** | |  | |
|  | **Actor** | **Sistema** | |
| **1.1** | **Gerente** | El sistema validara que estén correctamente llenos los campos para poder continuar. | |
| **2.1** |  | En caso de cancelar no se modificara la base de datos. | |
| **Rendimiento** | | **Paso** | **Cota de tiempo** |
|  | | 1 | 40 segundos |
|  | | 2 | 3 segundos |
|  | | 3 | 5 segundos |
|  | | 4 | 5 Segundos |
| **Frecuencia esperada** | | <1º de veces> veces / <53 segundos> | |
| **Importancia** | | { vital} | |
| **Post-condición** | | <Regresara al menú de gerente> | |
| **Urgencia** | | {hay presión} | |
| **Comentarios** | | <Los datos de tipo de usuario se generan automáticamente.> | |



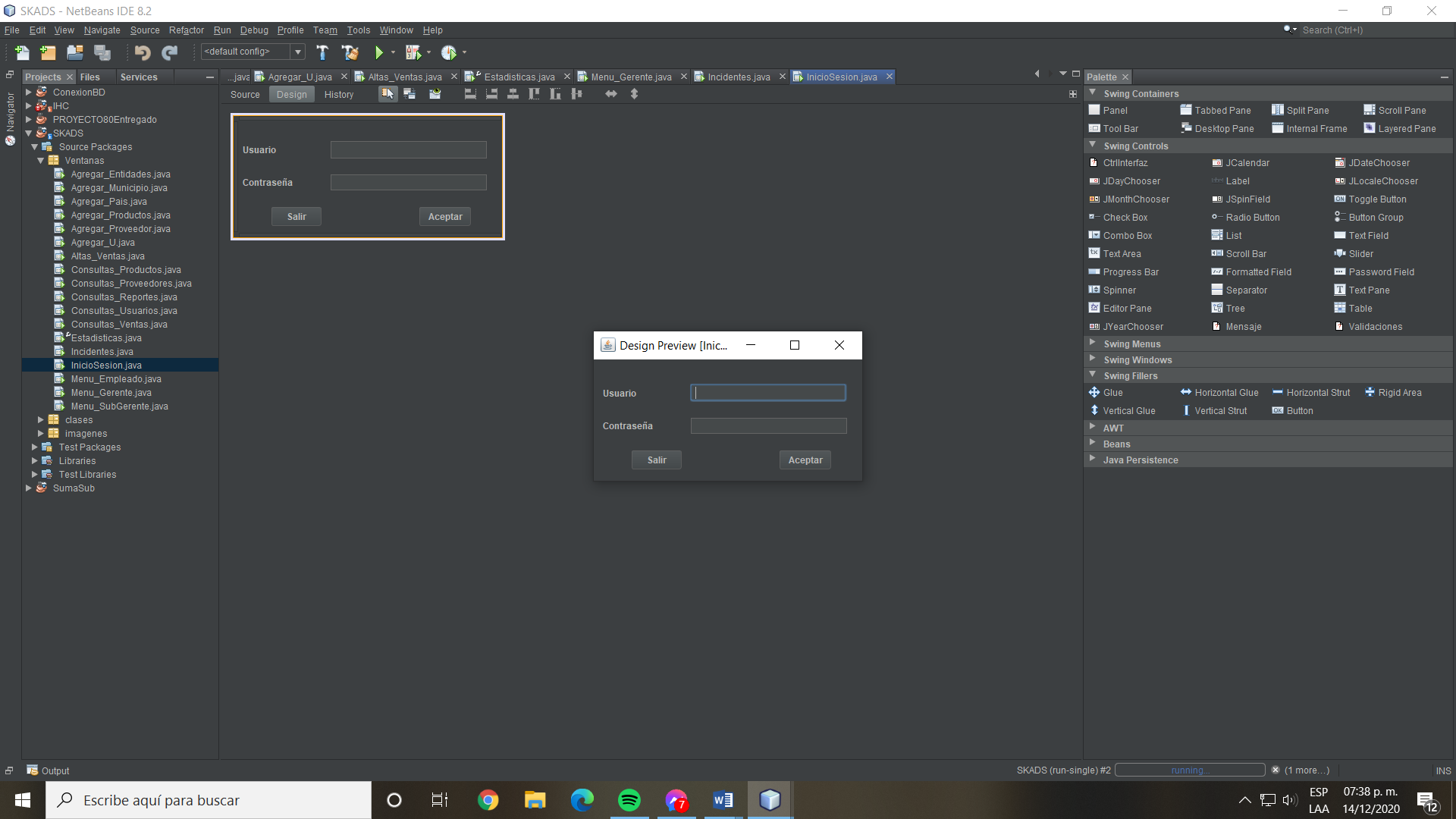
|  |  |  |  |
| --- | --- | --- | --- |
| **RF- <10>** | | <RF10> | |
| **Versión** | | Versión 0.2 17/11/2020 | |
| **Autores** | | Julio Piña, Kevin Escutia, Gerardo Pastrana | |
| **Fuentes** | | Java netbeans 8.2, github. | |
| **Objetivos asociados** | | Ventas | |
| **Descripción** | | Realizar la venta de los productos registrados | |
| **Precondición** | | <Ingresar y seleccionar en realizar una venta, deben de existir los productos que desea vender> | |
| **Secuencia Normal** | |  | |
|  | **Actor** | **Sistema** | |
| **1** |  | Selecciona el producto que desea agregar | |
| **2** |  | Se agrega el producto. | |
| **3** |  | Se modifica la cantidad de venta | |
| **4** |  | Se acepta la venta realizada | |
| **Excepciones** | |  | |
|  | |  | |
|  | **Actor** | **Sistema** | |
| **2.1** |  | En caso de que no se quiera el producto, se dará en eliminar antes de aceptar la venta, eliminando el último producto agregado. | |
| **3.1** |  | En caso de que no exista suficiente producto se le informara al usuario | |
| **Rendimiento** | | **Paso** | **Cota de tiempo** |
|  | | 1 | 1 segundos |
|  | | 2 | 1 segundos |
|  | | 3 | 5 segundos |
|  | | 4 | 5 segundos |
| **Frecuencia esperada** | | 1 vez / 12 s. | |
| **Importancia** | | Sin importancia | |
| **Post-condición** | | Regresar al menú de acuerdo al inicio de sesion. | |
| **Urgencia** | | Puede esperar | |
| **Comentarios** | | <La suma de la venta se muestra en la parte inferior> | |



|  |  |  |  |
| --- | --- | --- | --- |
| **RF- <RF9>** | | <RF**9**> | |
| **Versión** | | <0.2> | |
| **Autores** | | <Kevin Jesús, Pastrana Gómez, Julio Piña> | |
| **Fuentes** | | <NetBeans 8.2, GitHub> | |
| **Objetivos asociados** | | <nombre del objetivo> | |
| **Descripción** | | El sistema hará las estadísticas de los productos más vendidos y menos vendidos. | |
| **Precondición** | | <El usuario debe iniciar como gerente y seleccionar en estadisticas> | |
| **Secuencia Normal** | |  | |
|  | **Actor** | **Sistema** | |
| **1** | **Gerente** | Se selecciona gráfica y se muestra una grafica con la información. | |
| **Excepciones** | |  | |
|  | **Actor** | **Sistema** | |
| **1.1** | **Gerente** | En caso de que no existan ventas se mostrara la grafica vacia. | |
| **Rendimiento** | | **Paso** | **Cota de tiempo** |
|  | | 1 | 1 segundos |
|  | | 2 | 8 segundos |
| **Frecuencia esperada** | | <1> veces / <9s> | |
| **Importancia** | | {importante} | |
| **Postcondición** | | <Se retorna al menú del gerente> | |
| **Urgencia** | | {hay presión} | |
| **Comentarios** | | <SC> | |

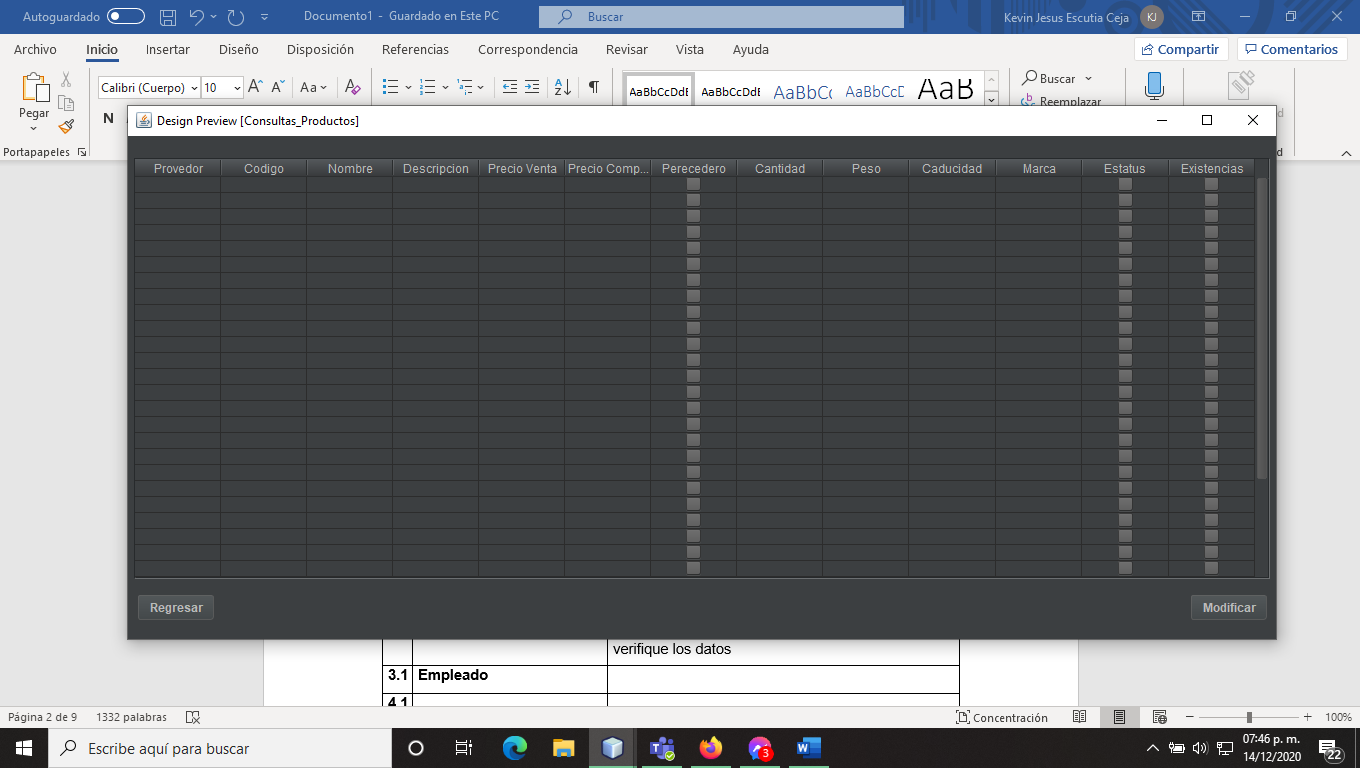


|  |  |  |  |
| --- | --- | --- | --- |
| **RF- <RF2,RF8,RNF3>** | | RF2, RF8, RNF3 | |
| **Versión** | | Versión 0.2 17/11/2020 | |
| **Autores** | | Julio Piña, Kevin Escutia, Gerardo Pastrana | |
| **Fuentes** | | Java netbeans 8.2, github. | |
| **Objetivos asociados** | | Se crean aquí los incidentes. | |
| **Descripción** | | En caso de que ocurra algún incidente cualquier usuario podrá hacer un reporte del incidente. | |
| **Precondición** | | Se deberá seleccionar incidente en el menú | |
| **Secuencia Normal** | |  | |
|  | **Actor** | **Sistema** | |
| **1** | **Llenar** | Se deberá llenar el incidente con la información de lo que sucedió. | |
| **2** | **Enviar** | Se enviara el incidente que ocurrió en la tienda | |
| **Excepciones** | |  | |
|  | | En caso de que sobre pase los 250 caracteres se le informaran de que debe ser más breve el reporte de incidente. | |
|  | **Actor** | **Sistema** | |
| **1.1** |  | nd | |
| **Rendimiento** | | **Paso** | **Cota de tiempo** |
|  | | 1 | 50 segundos |
|  | | 2 | 5 segundos |
| **Frecuencia esperada** | | 1 vez / 55 s. | |
| **Importancia** | | Sin importancia | |
| **Post-condición** | | Se regresara al menú de acuerdo al inicio de sesión. | |
| **Urgencia** | | Puede esperar | |
| **Comentarios** | | SC. | |

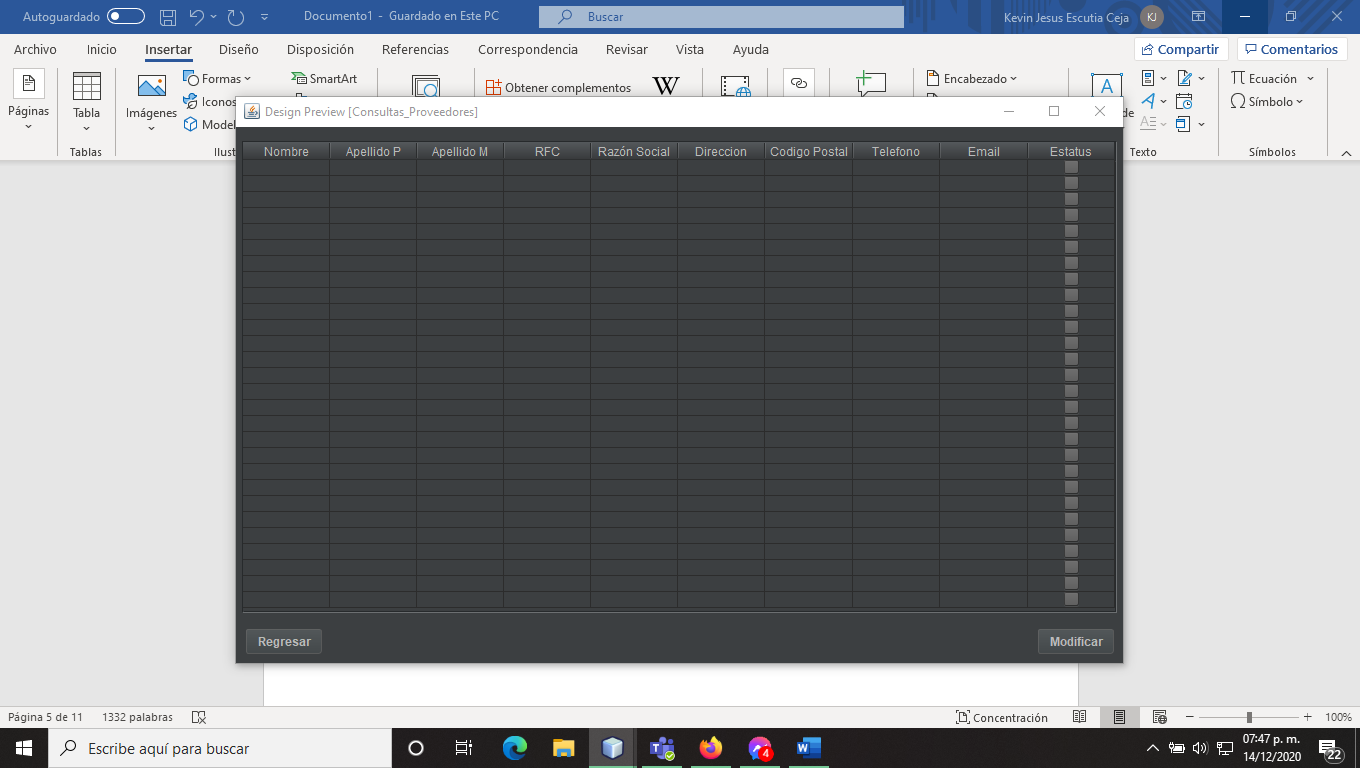


|  |  |  |  |
| --- | --- | --- | --- |
| **RF-** | | RF1, RF2, RF7, RNF2, | |
| **Versión** | | Versión 0.2 17/11/2020 | |
| **Autores** | | Julio Piña, Kevin Escutia, Gerardo Pastrana | |
| **Fuentes** | | Java netbeans 8.2, github. | |
| **Objetivos asociados** | | Inicio de sesión. | |
| **Descripción** | | Permitirá iniciar sesión para los diferentes usuarios. | |
| **Precondición** | | Inicializar el programa | |
| **Secuencia Normal** | |  | |
|  | **Actor** | **Sistema** | |
| **1** | **Usuario** | Rellenar con el código de usuario del empleado | |
| **2** | **Contraseña** | Ingresar la contraseña del usuario. | |
| **3** | **Aceptar** | Se comprobara el tipo de usuario y se le llevara al submenú del tipo de usuario ingresado | |
| **4** | **Salir** | Saldrá del programa | |
| **Excepciones** | |  | |
| **1** | | Se mostrara un error de que el usuario no es el correcto. | |
| **2** | | Se mostrara un erro de que la información es incorrecta. | |
|  | **Actor** | **Sistema** | |
| **1.1** |  | Se deberá llenar con caracteres adecuados | |
| **2.1** |  | La contraseña debe ser precisa | |
| **3.1** |  | Se enviara a la base de datos para comprobar la información. | |
| **Rendimiento** | | **Paso** | **Cota de tiempo** |
|  | | 1 | 5 segundos |
|  | | 2 | 5 segundos |
| **Frecuencia esperada** | | 1 vez / 10 s. | |
| **Importancia** | | Vital | |
| **Post-condición** | | Se enviara al submenú dependiendo del tipo de usuario | |
| **Urgencia** | | Inmediatamente | |
| **Comentarios** | | La contraseña no será visible. | |

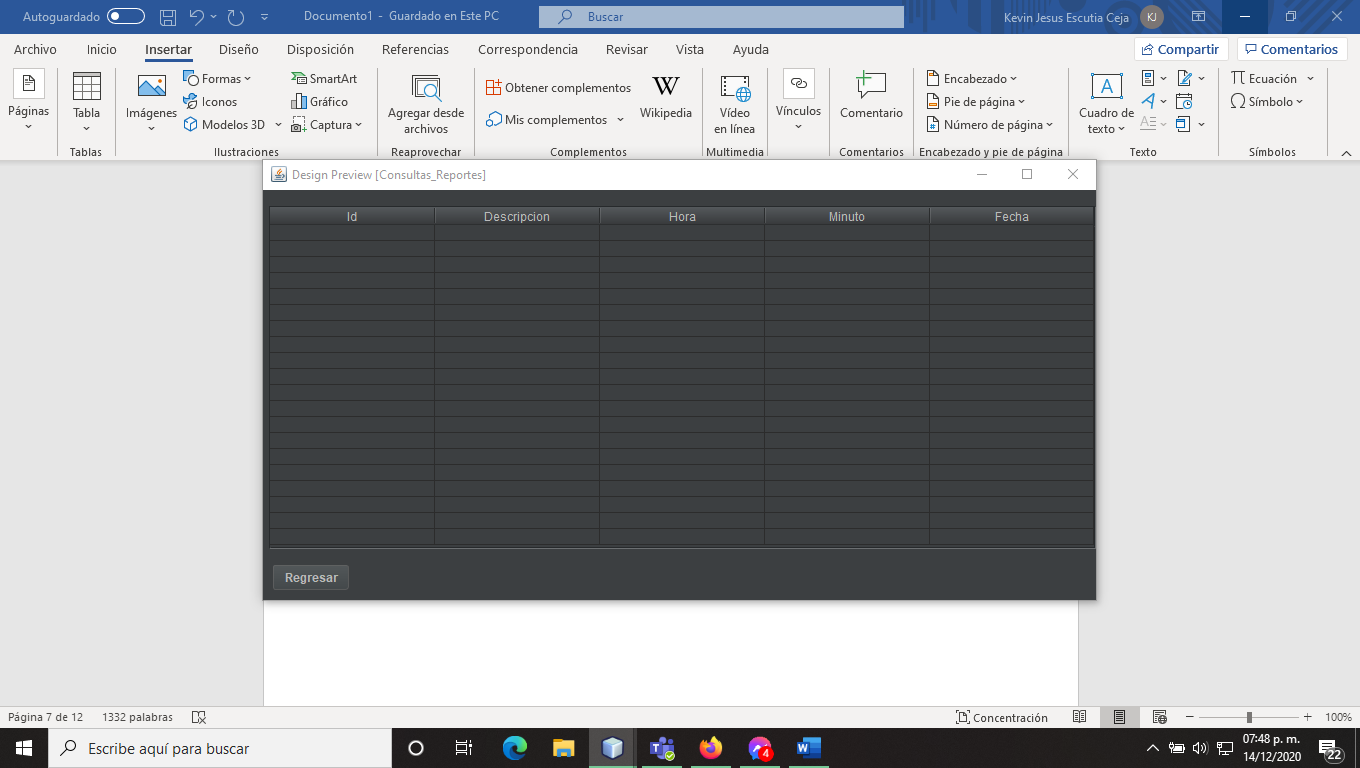
|  |  |  |  |
| --- | --- | --- | --- |
| **RF-** | | <**Consultas\_Productos**> | |
| **Versión** | | <2.0> | |
| **Autores** | | <Kevin Jesús, Gerardo Pastrana, Julio Estrada> | |
| **Fuentes** | | <NetBeans 8.2, GitHub> | |
| **Objetivos asociados** | | <Administrar los productos> | |
| **Descripción** | | Habrá una lista de productos | |
| **Precondición** | | <El usuario debe elegir el boton de productos en el menú\_Gerente o menú\_Subgerentes> | |
| **Secuencia Normal** | |  | |
|  | **Actor** | **Sistema** | |
| **1** | **Gerente** | El actor visualiza los datos de productos | |
| **2** | **SubGerente** | El actor da clic en uno de los espacios | |
| **3** | **Empleados** | El sistema revisa que todas las casillas tengan sus correspondientes datos | |
| **4** |  | El sistema actualizara los datos en la BD | |
| **5** |  |  | |
| **6** |  |  | |
| **Excepciones** | |  | |
|  | **Actor** | **Sistema** | |
| **1.1** | **Gerente** | Si el usuario es Empleados no tiene permitido modificar datos | |
| **2.1** | **Sub Gerente** | El sistema mandara aviso al usuario para que verifique los datos | |
| **3.1** | **Empleado** |  | |
| **4.1** |  |  | |
| **5.1** |  |  | |
| **Rendimiento** | | **Paso** | **Cota de tiempo** |
|  | | 1 | n segundos |
|  | | 2 | n segundos |
| **Frecuencia esperada** | | <1º de veces> veces / <60> | |
| **Importancia** | | { vital} | |
| **Postcondición** | | <Pasa al menú\_Gerentes o menú\_SubGerentes o menú\_Empleados> | |
| **Urgencia** | | {hay presión} | |
| **Comentarios** | | <una vez tengamos las interfaces esqueleto el resto será más sencillo de predecir> | |



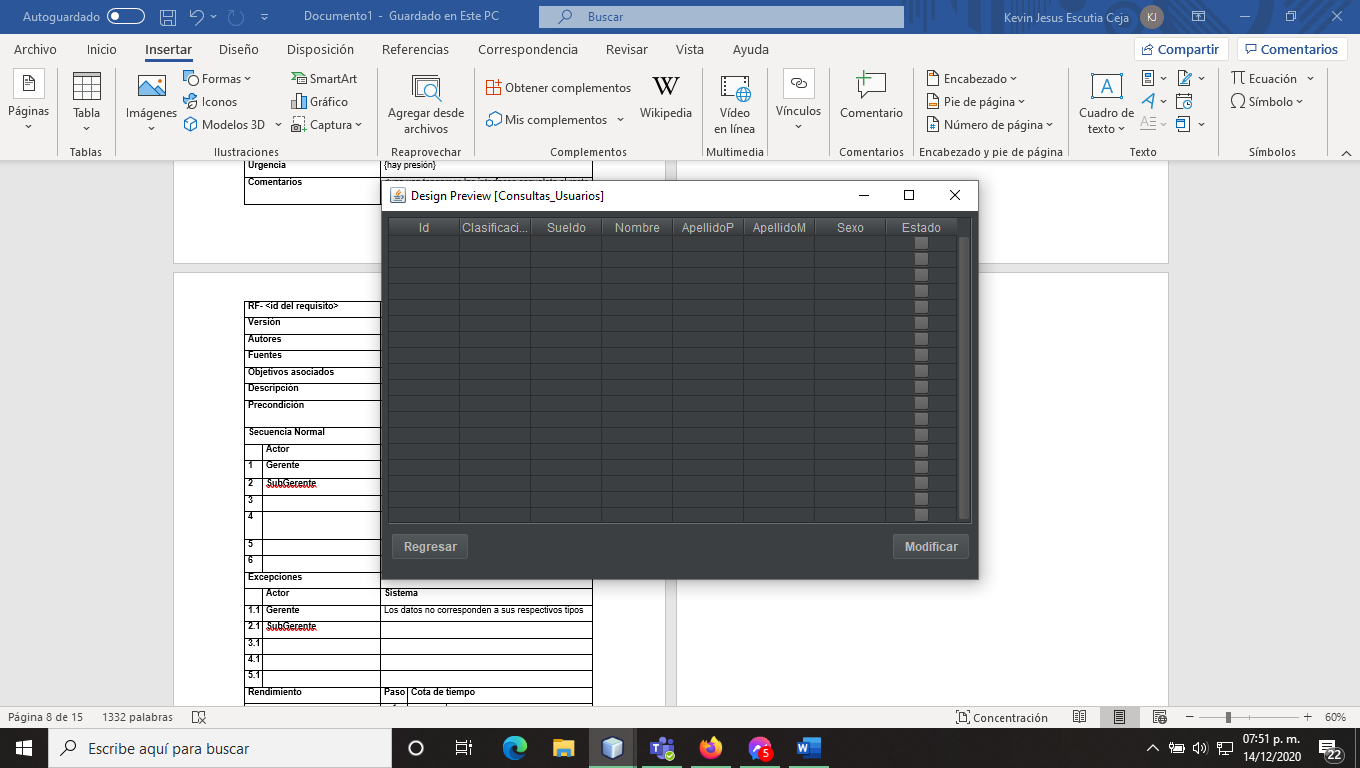
|  |  |  |  |
| --- | --- | --- | --- |
| **RF-** | | <**Consultas\_Proveedores**> | |
| **Versión** | | <2.0> | |
| **Autores** | | <Kevin Jesús, Gerardo Pastrana, Julio Estrada> | |
| **Fuentes** | | <NetBeans 8.2, GitHub> | |
| **Objetivos asociados** | | <Administrar los productos> | |
| **Descripción** | | Habrá una lista de proveedores | |
| **Precondición** | | <El usuario debe elegir el boton de proveedores en el menú\_Gerente o menú\_Subgerentes> | |
| **Secuencia Normal** | |  | |
|  | **Actor** | **Sistema** | |
| **1** | **Gerente** | El actor llena los datos de las cajas | |
| **2** | **SubGerente** | El actor da clic en botón aceptar | |
| **3** |  | El sistema revisa que todas las casillas tengan datos | |
| **4** |  | El sistema enviara los datos a la base de datos | |
| **5** |  |  | |
| **6** |  |  | |
| **Excepciones** | |  | |
|  | **Actor** | **Sistema** | |
| **1.1** | **Gerente** | El sistema identificara que los datos ya existen en bd | |
| **2.1** | **SubGerente** | El sistema mandara aviso al usuario para que verifique los datos | |
| **3.1** | **Empleado** |  | |
| **4.1** |  |  | |
| **5.1** |  |  | |
| **Rendimiento** | | **Paso** | **Cota de tiempo** |
|  | | 1 | n segundos |
|  | | 2 | n segundos |
| **Frecuencia esperada** | | <1º de veces> veces / <60> | |
| **Importancia** | | { vital} | |
| **Postcondición** | | <Pasa al menú\_Gerentes o menú\_SubGerentes o menú\_Empleados> | |
| **Urgencia** | | {hay presión} | |
| **Comentarios** | | <una vez tengamos las interfaces esqueleto el resto será más sencillo de predecir> | |



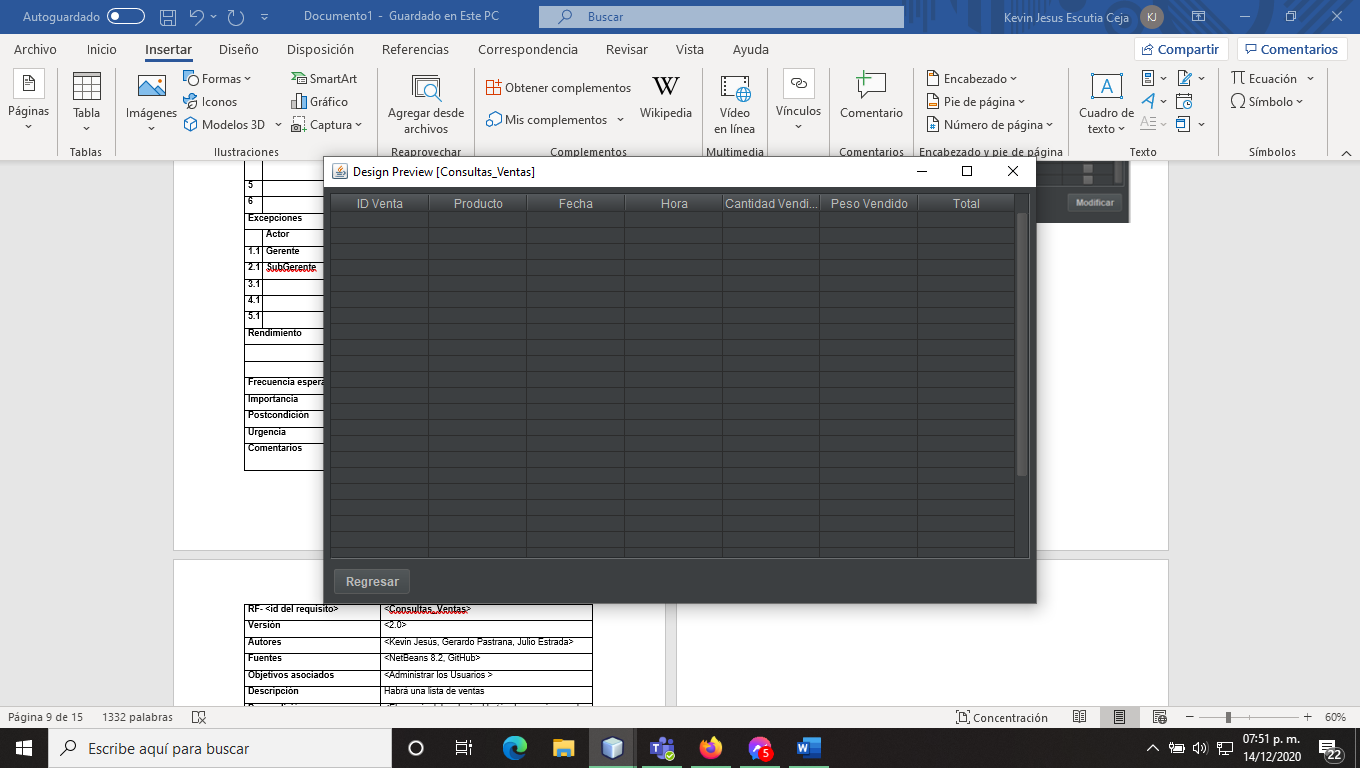
|  |  |  |  |
| --- | --- | --- | --- |
| **RF-** | | <**Consultas\_Reportes**> | |
| **Versión** | | <2.0> | |
| **Autores** | | <Kevin Jesús, Gerardo Pastrana, Julio Estrada> | |
| **Fuentes** | | <NetBeans 8.2, GitHub> | |
| **Objetivos asociados** | | <Administrar los reportes> | |
| **Descripción** | | Habrá una lista de reportes | |
| **Precondición** | | <El usuario debe elegir el botón de reportes en el menú\_Gerente o menú\_Subgerentes o menú\_Empleados> | |
| **Secuencia Normal** | |  | |
|  | **Actor** | **Sistema** | |
| **1** | **Gerente** | El actor visualizara los datos en la tabla no puede hacer nada más | |
| **2** | **SubGerente** | El actor da clic en botón regresar | |
| **3** | **Empleado** |  | |
| **4** |  |  | |
| **5** |  |  | |
| **6** |  |  | |
| **Excepciones** | |  | |
|  | **Actor** | **Sistema** | |
| **1.1** |  |  | |
| **2.1** |  |  | |
| **3.1** |  |  | |
| **4.1** |  |  | |
| **5.1** |  |  | |
| **Rendimiento** | | **Paso** | **Cota de tiempo** |
|  | | 1 | n segundos |
|  | | 2 | n segundos |
| **Frecuencia esperada** | | <1º de veces> veces / <60> | |
| **Importancia** | | { vital} | |
| **Postcondición** | | <Pasa al menú\_Gerentes o menú\_SubGerentes o menú\_Empleados> | |
| **Urgencia** | | {hay presión} | |
| **Comentarios** | | <una vez tengamos las interfaces esqueleto el resto será más sencillo de predecir> | |



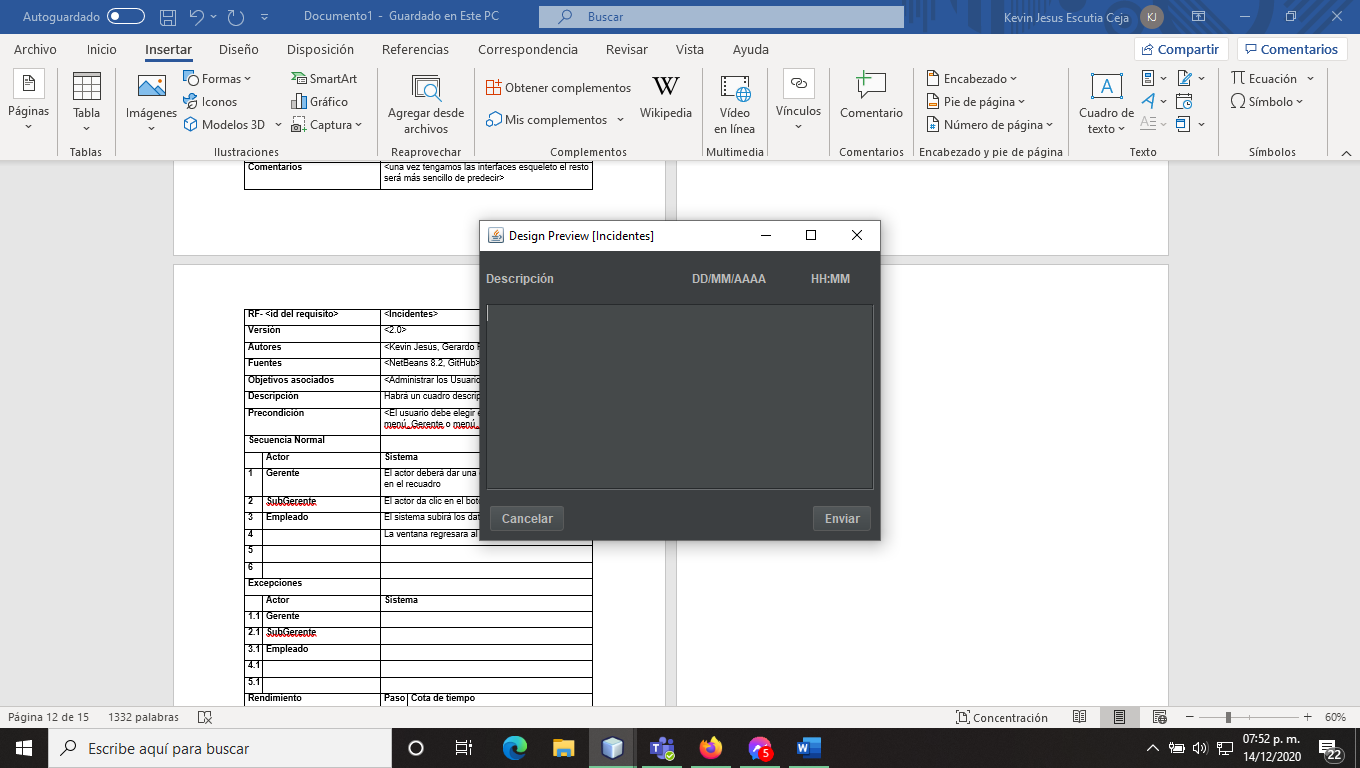
|  |  |  |  |
| --- | --- | --- | --- |
| **RF-** | | <**Consultas\_Usuarios**> | |
| **Versión** | | <2.0> | |
| **Autores** | | <Kevin Jesús, Gerardo Pastrana, Julio Estrada> | |
| **Fuentes** | | <NetBeans 8.2, GitHub> | |
| **Objetivos asociados** | | <Administrar los Usuarios > | |
| **Descripción** | | Habrá una lista de usuarios | |
| **Precondición** | | <El usuario debe elegir el botón de usuarios en el menú\_Gerente o menú\_Subgerentes | |
| **Secuencia Normal** | |  | |
|  | **Actor** | **Sistema** | |
| **1** | **Gerente** | El actor visualizara los datos en la tabla | |
| **2** | **SubGerente** | El actor llena los datos de las cajas | |
| **3** |  | El actor da clic en botón modificar | |
| **4** |  | El sistema revisa que todas las casillas tengan datos | |
| **5** |  | El sistema enviara los datos a la base de datos | |
| **6** |  | El actor da clic en botón regresar | |
| **Excepciones** | |  | |
|  | **Actor** | **Sistema** | |
| **1.1** | **Gerente** | Los datos no corresponden a sus respectivos tipos | |
| **2.1** | **SubGerente** |  | |
| **3.1** |  |  | |
| **4.1** |  |  | |
| **5.1** |  |  | |
| **Rendimiento** | | **Paso** | **Cota de tiempo** |
|  | | 1 | n segundos |
|  | | 2 | n segundos |
| **Frecuencia esperada** | | <1º de veces> veces / <60> | |
| **Importancia** | | { vital} | |
| **Postcondición** | | <Pasa al menú\_Gerentes o menú\_SubGerentes> | |
| **Urgencia** | | {hay presión} | |
| **Comentarios** | | <una vez tengamos las interfaces esqueleto el resto será más sencillo de predecir> | |



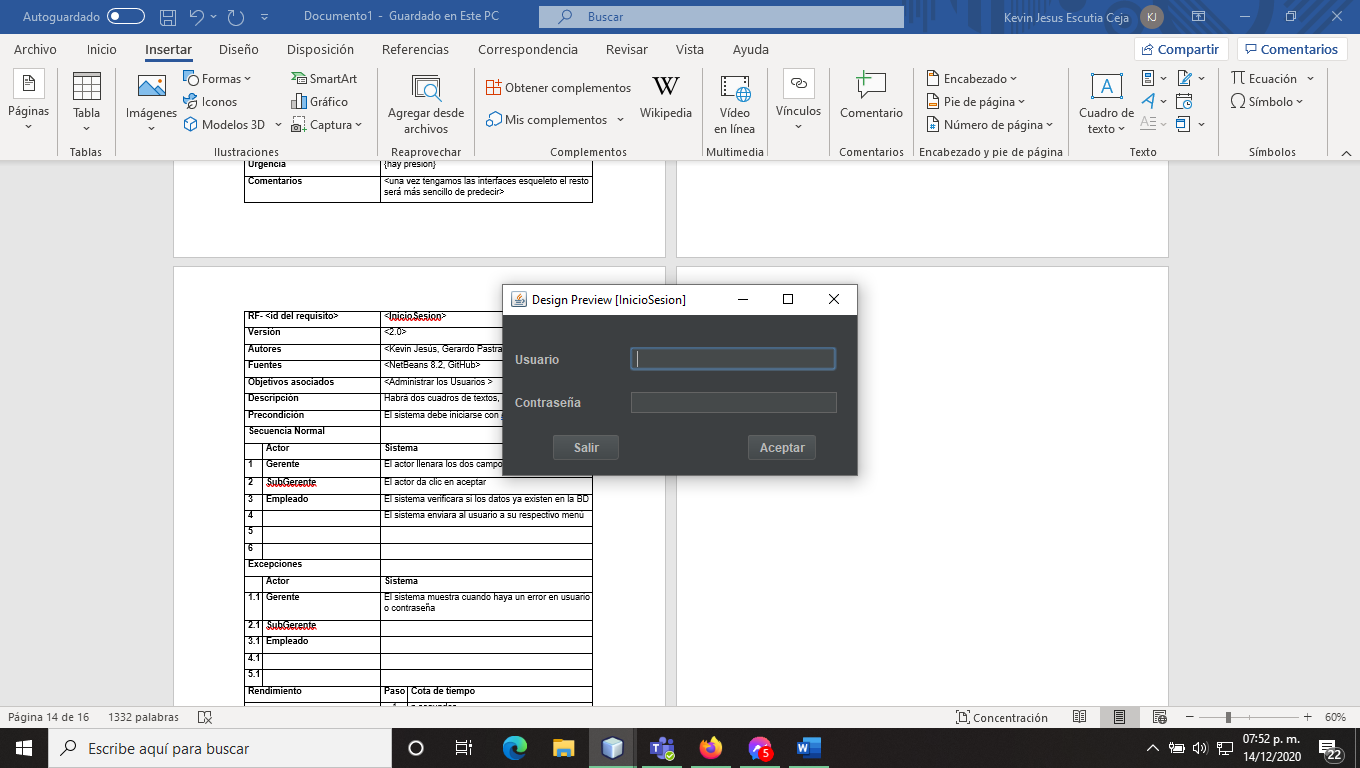
|  |  |  |  |
| --- | --- | --- | --- |
| **RF-** | | <**Consultas\_Ventas**> | |
| **Versión** | | <2.0> | |
| **Autores** | | <Kevin Jesús, Gerardo Pastrana, Julio Estrada> | |
| **Fuentes** | | <NetBeans 8.2, GitHub> | |
| **Objetivos asociados** | | <Administrar los Usuarios > | |
| **Descripción** | | Habrá una lista de ventas | |
| **Precondición** | | <El usuario debe elegir el botón de usuarios en el menú\_Gerente o menú\_Subgerentes | |
| **Secuencia Normal** | |  | |
|  | **Actor** | **Sistema** | |
| **1** | **Gerente** | El actor visualizara los datos en la tabla no puede hacer nada más | |
| **2** | **SubGerente** | El actor da clic en botón regresar | |
| **3** |  |  | |
| **4** |  |  | |
| **5** |  |  | |
| **6** |  |  | |
| **Excepciones** | |  | |
|  | **Actor** | **Sistema** | |
| **1.1** | **Gerente** |  | |
| **2.1** | **SubGerente** |  | |
| **3.1** |  |  | |
| **4.1** |  |  | |
| **5.1** |  |  | |
| **Rendimiento** | | **Paso** | **Cota de tiempo** |
|  | | 1 | n segundos |
|  | | 2 | n segundos |
| **Frecuencia esperada** | | <1º de veces> veces / <60> | |
| **Importancia** | | { vital} | |
| **Postcondición** | | <Pasa al menú\_Gerentes o menú\_SubGerentes> | |
| **Urgencia** | | {hay presión} | |
| **Comentarios** | | <una vez tengamos las interfaces esqueleto el resto será más sencillo de predecir> | |



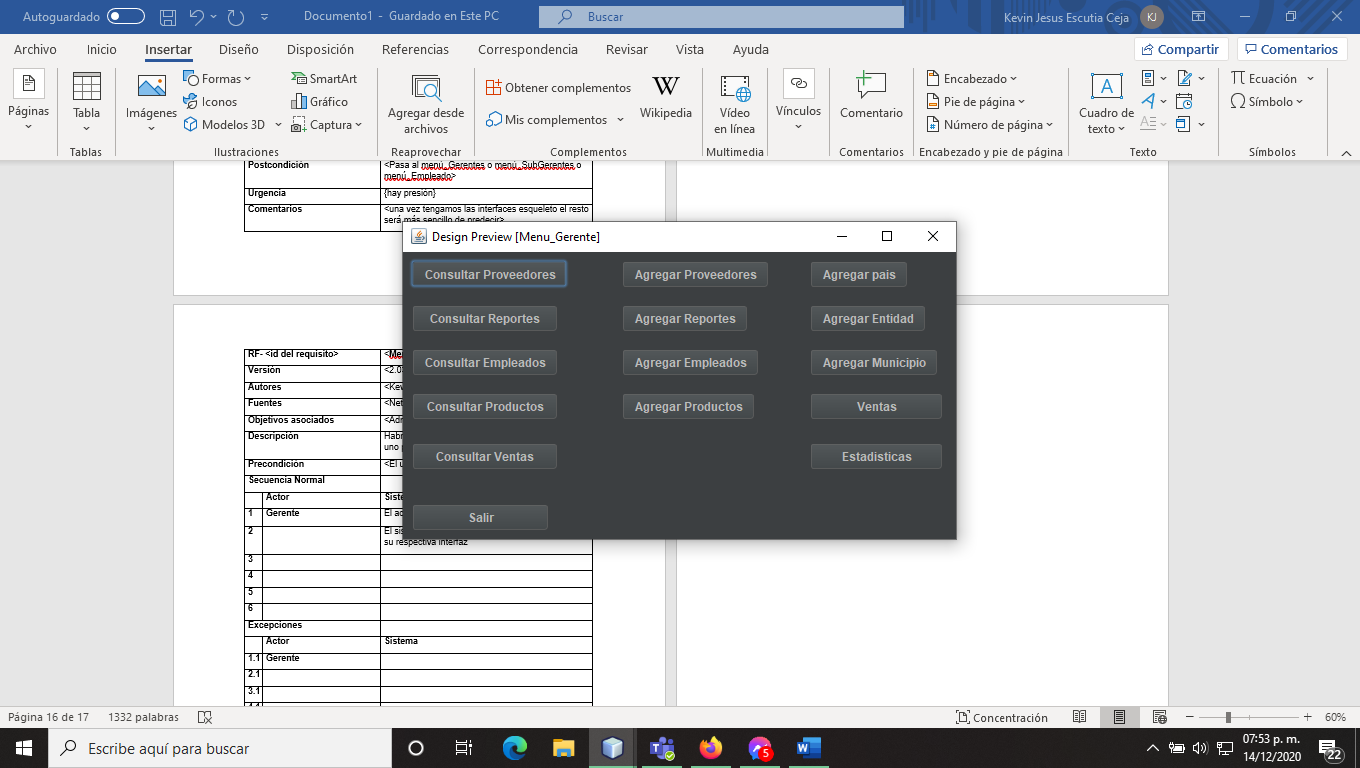
|  |  |  |  |
| --- | --- | --- | --- |
| **RF-** | | <**Incidentes**> | |
| **Versión** | | <2.0> | |
| **Autores** | | <Kevin Jesús, Gerardo Pastrana, Julio Estrada> | |
| **Fuentes** | | <NetBeans 8.2, GitHub> | |
| **Objetivos asociados** | | <Administrar los Usuarios > | |
| **Descripción** | | Habrá un cuadro descripción | |
| **Precondición** | | <El usuario debe elegir el botón de usuarios en el menú\_Gerente o menú\_Subgerentes | |
| **Secuencia Normal** | |  | |
|  | **Actor** | **Sistema** | |
| **1** | **Gerente** | El actor deberá dar una descripción de un incidente en el recuadro | |
| **2** | **SubGerente** | El actor da clic en el botón enviar | |
| **3** | **Empleado** | El sistema subirá los datos a la BD | |
| **4** |  | La ventana regresara al menú del actor | |
| **5** |  |  | |
| **6** |  |  | |
| **Excepciones** | |  | |
|  | **Actor** | **Sistema** | |
| **1.1** | **Gerente** |  | |
| **2.1** | **SubGerente** |  | |
| **3.1** | **Empleado** |  | |
| **4.1** |  |  | |
| **5.1** |  |  | |
| **Rendimiento** | | **Paso** | **Cota de tiempo** |
|  | | 1 | n segundos |
|  | | 2 | n segundos |
| **Frecuencia esperada** | | <1º de veces> veces / <60> | |
| **Importancia** | | { vital} | |
| **Postcondición** | | <Pasa al menú\_Gerentes o menú\_SubGerentes o menú\_Empleado> | |
| **Urgencia** | | {hay presión} | |
| **Comentarios** | | <una vez tengamos las interfaces esqueleto el resto será más sencillo de predecir> | |



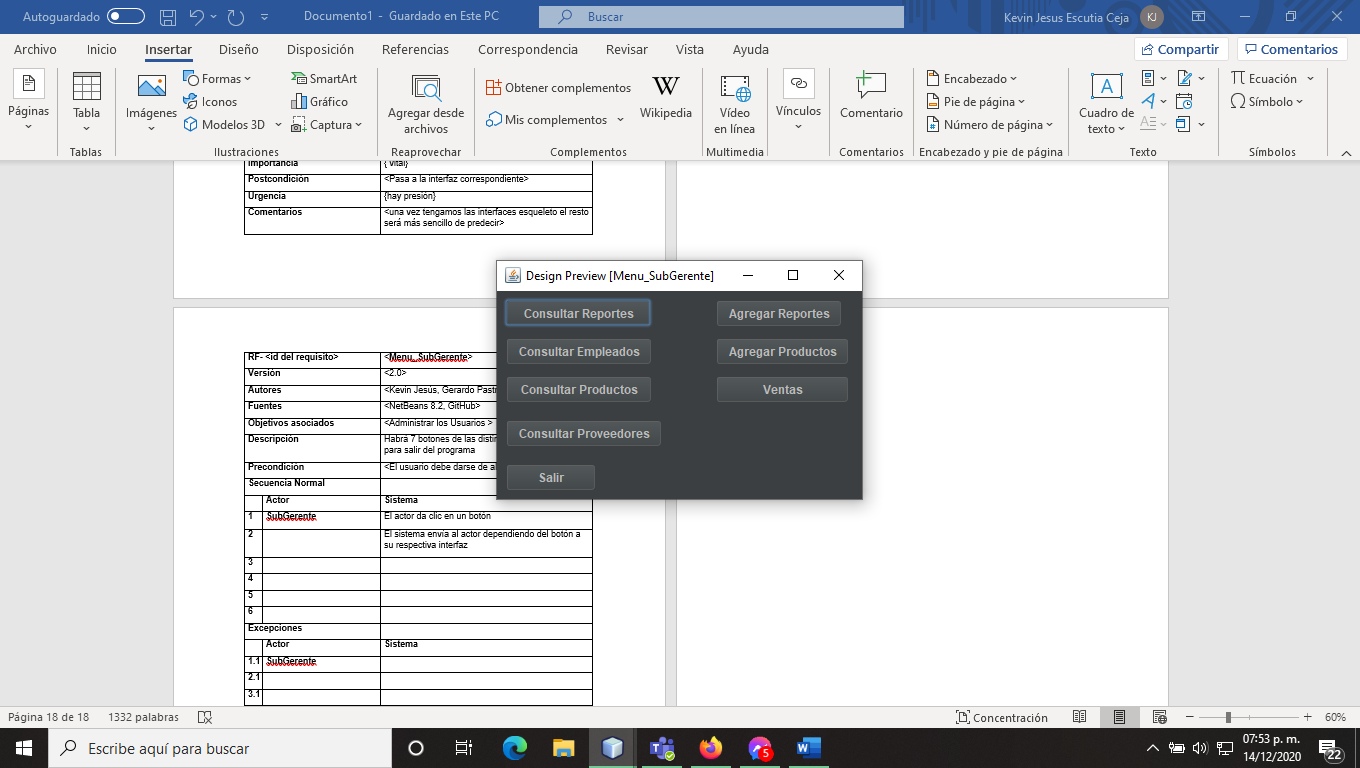
|  |  |  |  |
| --- | --- | --- | --- |
| **RF-** | | <**InicioSesion**> | |
| **Versión** | | <2.0> | |
| **Autores** | | <Kevin Jesús, Gerardo Pastrana, Julio Estrada> | |
| **Fuentes** | | <NetBeans 8.2, GitHub> | |
| **Objetivos asociados** | | <Administrar los Usuarios > | |
| **Descripción** | | Habrá dos cuadros de textos, usuario, contraseña | |
| **Precondición** | | El sistema debe iniciarse con el .jar | |
| **Secuencia Normal** | |  | |
|  | **Actor** | **Sistema** | |
| **1** | **Gerente** | El actor llenara los dos campos | |
| **2** | **SubGerente** | El actor da clic en aceptar | |
| **3** | **Empleado** | El sistema verificara si los datos ya existen en la BD | |
| **4** |  | El sistema enviara al usuario a su respectivo menú | |
| **5** |  |  | |
| **6** |  |  | |
| **Excepciones** | |  | |
|  | **Actor** | **Sistema** | |
| **1.1** | **Gerente** | El sistema muestra cuando haya un error en usuario o contraseña | |
| **2.1** | **SubGerente** |  | |
| **3.1** | **Empleado** |  | |
| **4.1** |  |  | |
| **5.1** |  |  | |
| **Rendimiento** | | **Paso** | **Cota de tiempo** |
|  | | 1 | n segundos |
|  | | 2 | n segundos |
| **Frecuencia esperada** | | <1º de veces> veces / <60> | |
| **Importancia** | | { vital} | |
| **Postcondición** | | <Pasa al menú\_Gerentes o menú\_SubGerentes o menú\_Empleado> | |
| **Urgencia** | | {hay presión} | |
| **Comentarios** | | <una vez tengamos las interfaces esqueleto el resto será más sencillo de predecir> | |



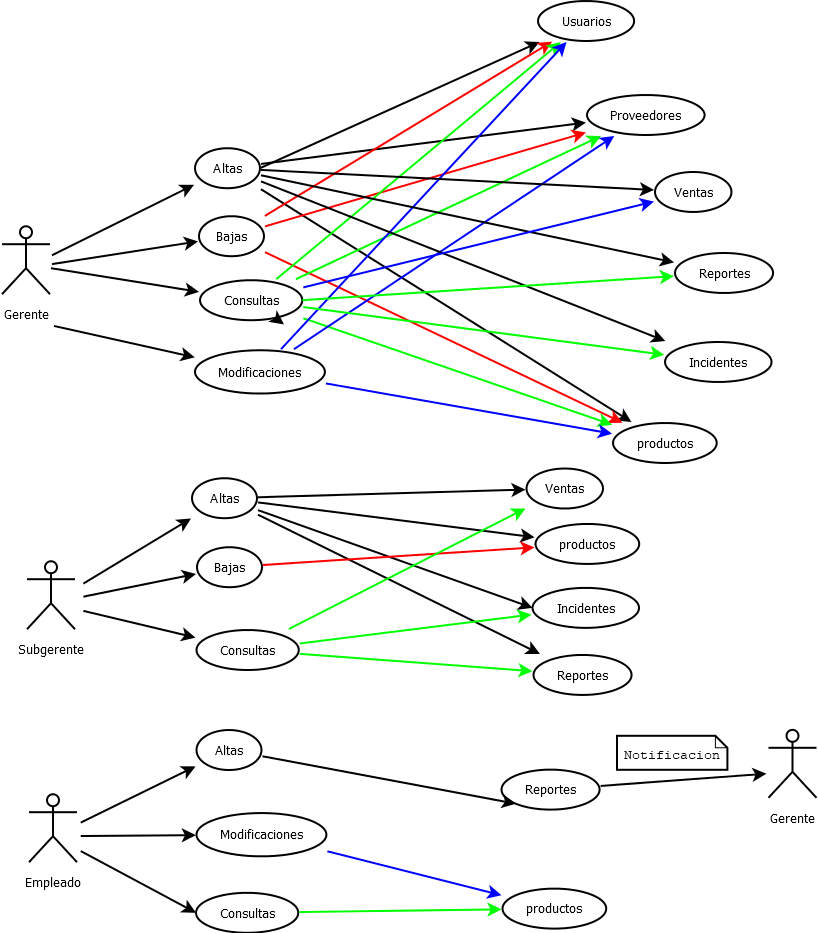
|  |  |  |  |
| --- | --- | --- | --- |
| **RF-** | | <**Menu\_Gerente**> | |
| **Versión** | | <2.0> | |
| **Autores** | | <Kevin Jesús, Gerardo Pastrana, Julio Estrada> | |
| **Fuentes** | | <NetBeans 8.2, GitHub> | |
| **Objetivos asociados** | | <Administrar los Usuarios > | |
| **Descripción** | | Habrá 13 botones de las distintas interfaces más uno para salir del programa | |
| **Precondición** | | <El usuario debe darse de altas> | |
| **Secuencia Normal** | |  | |
|  | **Actor** | **Sistema** | |
| **1** | **Gerente** | El actor da clic en un botón | |
| **2** |  | El sistema envía al actor dependiendo del botón a su respectiva interfaz | |
| **3** |  |  | |
| **4** |  |  | |
| **5** |  |  | |
| **6** |  |  | |
| **Excepciones** | |  | |
|  | **Actor** | **Sistema** | |
| **1.1** | **Gerente** |  | |
| **2.1** |  |  | |
| **3.1** |  |  | |
| **4.1** |  |  | |
| **5.1** |  |  | |
| **Rendimiento** | | **Paso** | **Cota de tiempo** |
|  | | 1 | n segundos |
|  | | 2 | n segundos |
| **Frecuencia esperada** | | <1º de veces> veces / <60> | |
| **Importancia** | | { vital} | |
| **Postcondición** | | <Pasa a la interfaz correspondiente> | |
| **Urgencia** | | {hay presión} | |
| **Comentarios** | | <una vez tengamos las interfaces esqueleto el resto será más sencillo de predecir> | |



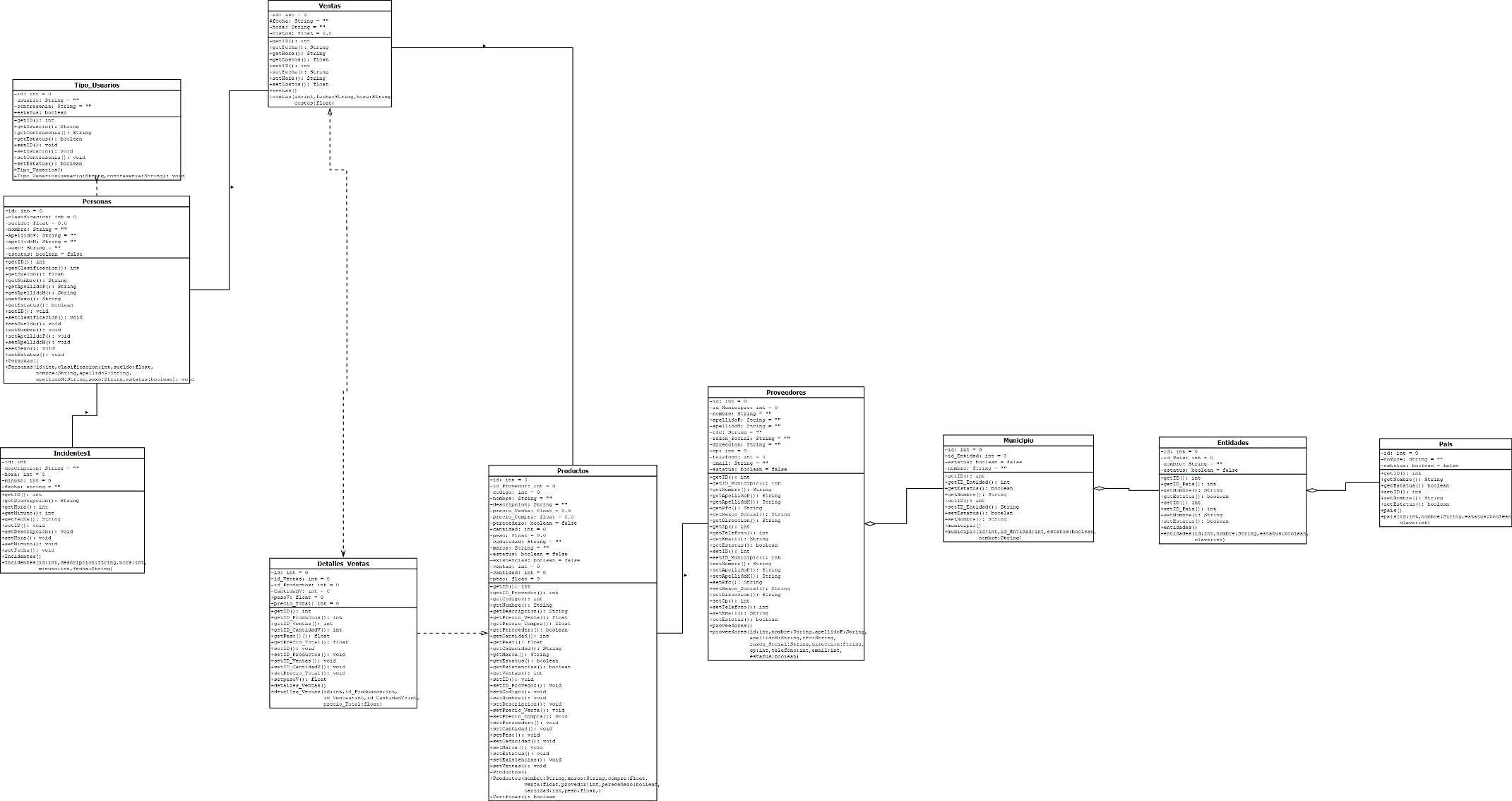
|  |  |  |  |
| --- | --- | --- | --- |
| **RF-** | | <**Menu\_SubGerente**> | |
| **Versión** | | <2.0> | |
| **Autores** | | <Kevin Jesús, Gerardo Pastrana, Julio Estrada> | |
| **Fuentes** | | <NetBeans 8.2, GitHub> | |
| **Objetivos asociados** | | <Administrar los Usuarios > | |
| **Descripción** | | Habrá 7 botones de las distintas interfaces más uno para salir del programa | |
| **Precondición** | | <El usuario debe darse de altas> | |
| **Secuencia Normal** | |  | |
|  | **Actor** | **Sistema** | |
| **1** | **SubGerente** | El actor da clic en un botón | |
| **2** |  | El sistema envía al actor dependiendo del botón a su respectiva interfaz | |
| **3** |  |  | |
| **4** |  |  | |
| **5** |  |  | |
| **6** |  |  | |
| **Excepciones** | |  | |
|  | **Actor** | **Sistema** | |
| **1.1** | **SubGerente** |  | |
| **2.1** |  |  | |
| **3.1** |  |  | |
| **4.1** |  |  | |
| **5.1** |  |  | |
| **Rendimiento** | | **Paso** | **Cota de tiempo** |
|  | | 1 | n segundos |
|  | | 2 | n segundos |
| **Frecuencia esperada** | | <1º de veces> veces / <60> | |
| **Importancia** | | { vital} | |
| **Postcondición** | | <Pasa a la interfaz correspondiente> | |
| **Urgencia** | | {hay presión} | |
| **Comentarios** | | <una vez tengamos las interfaces esqueleto el resto será más sencillo de predecir> | |



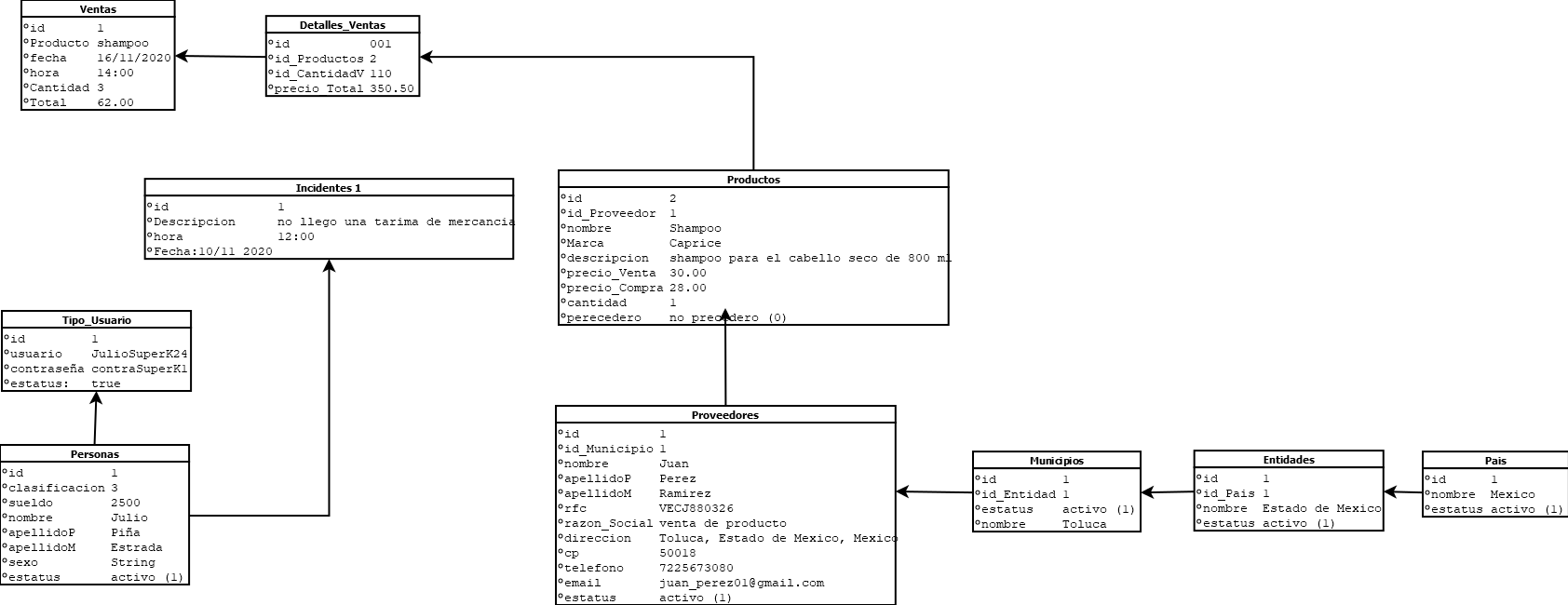
# Diagrama de caso de uso



# Diagrama de clases



# Diagrama de objetos



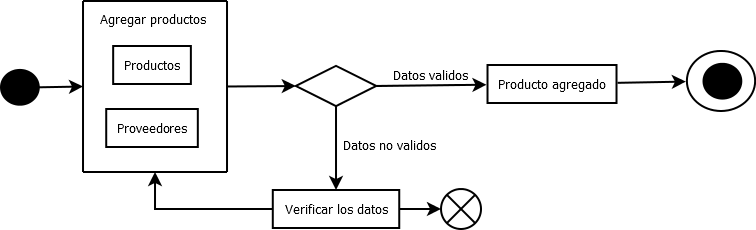
# Diagrama de actividades

Diagrama

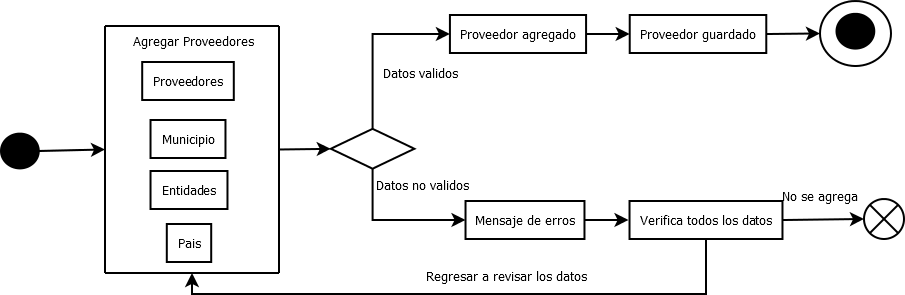
Descripción generada automáticamente

# Máquinas de Estado

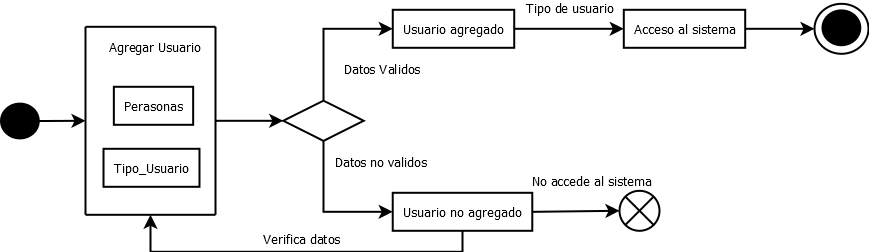
1.-Agregar productos



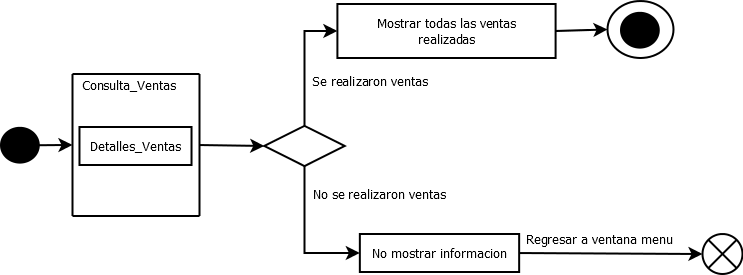
2.-Agregar Proveedores



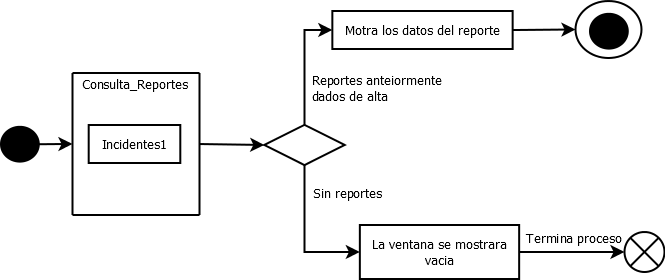
3.-Agregar Usuario



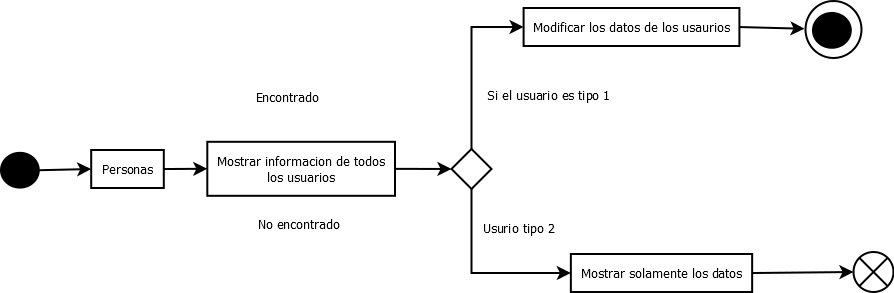
4.-Consultas Ventas



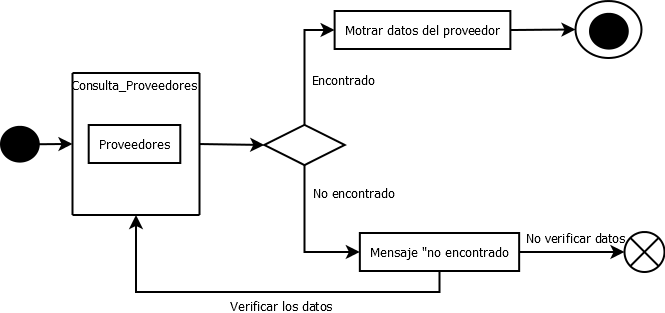
5.-Consulta Reportes



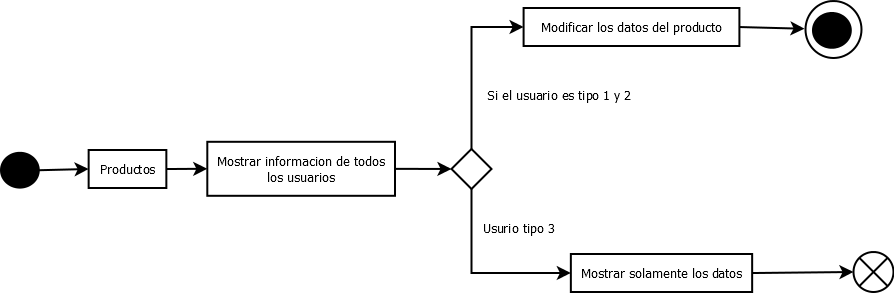
6.-Consulta Usuario



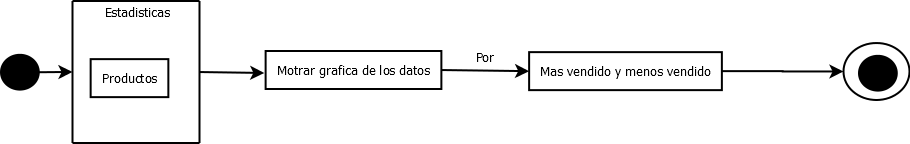
7.-Consulta Proveedores



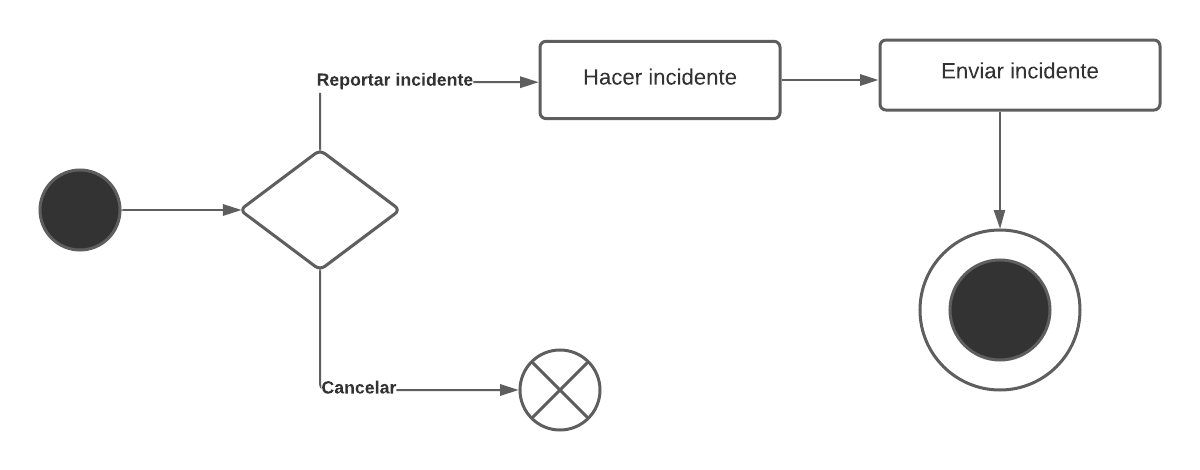
8.-Consulta Productos



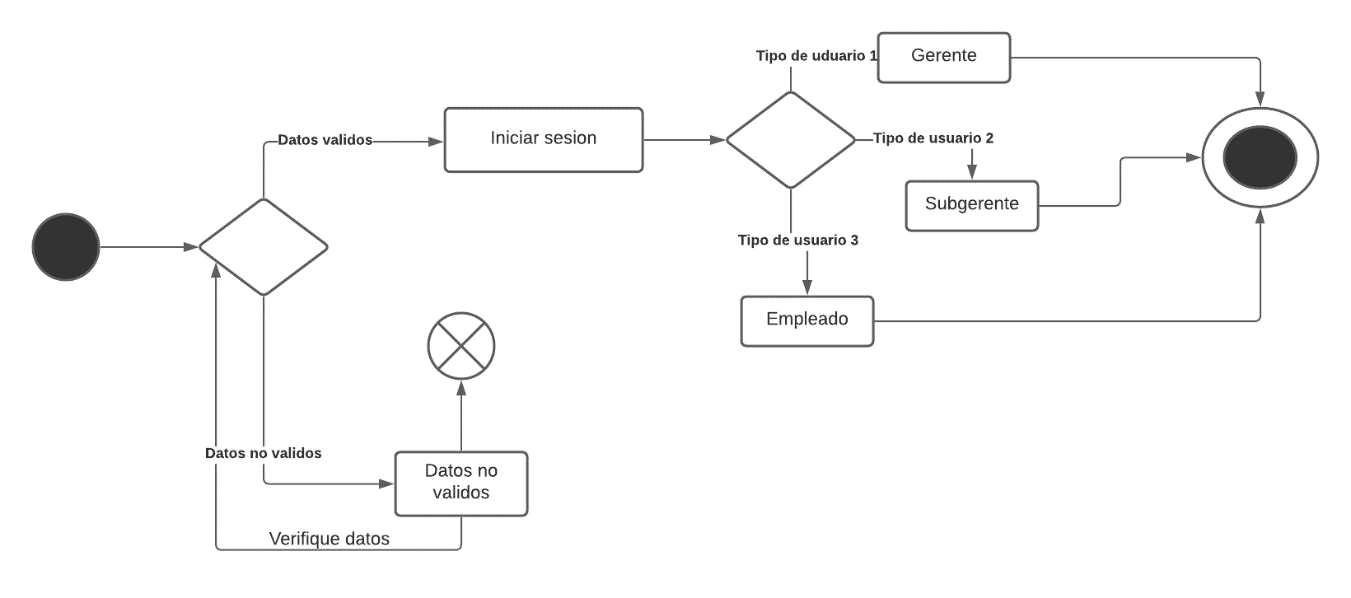
9.-Estadisticas



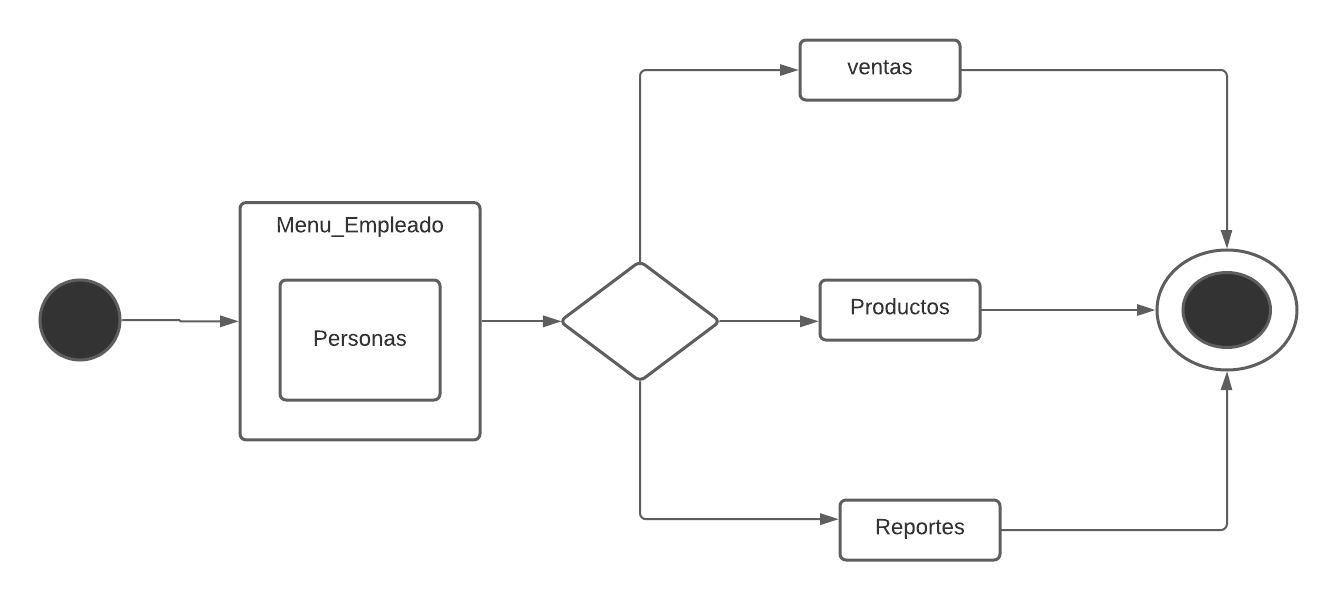
10.-Incidentes

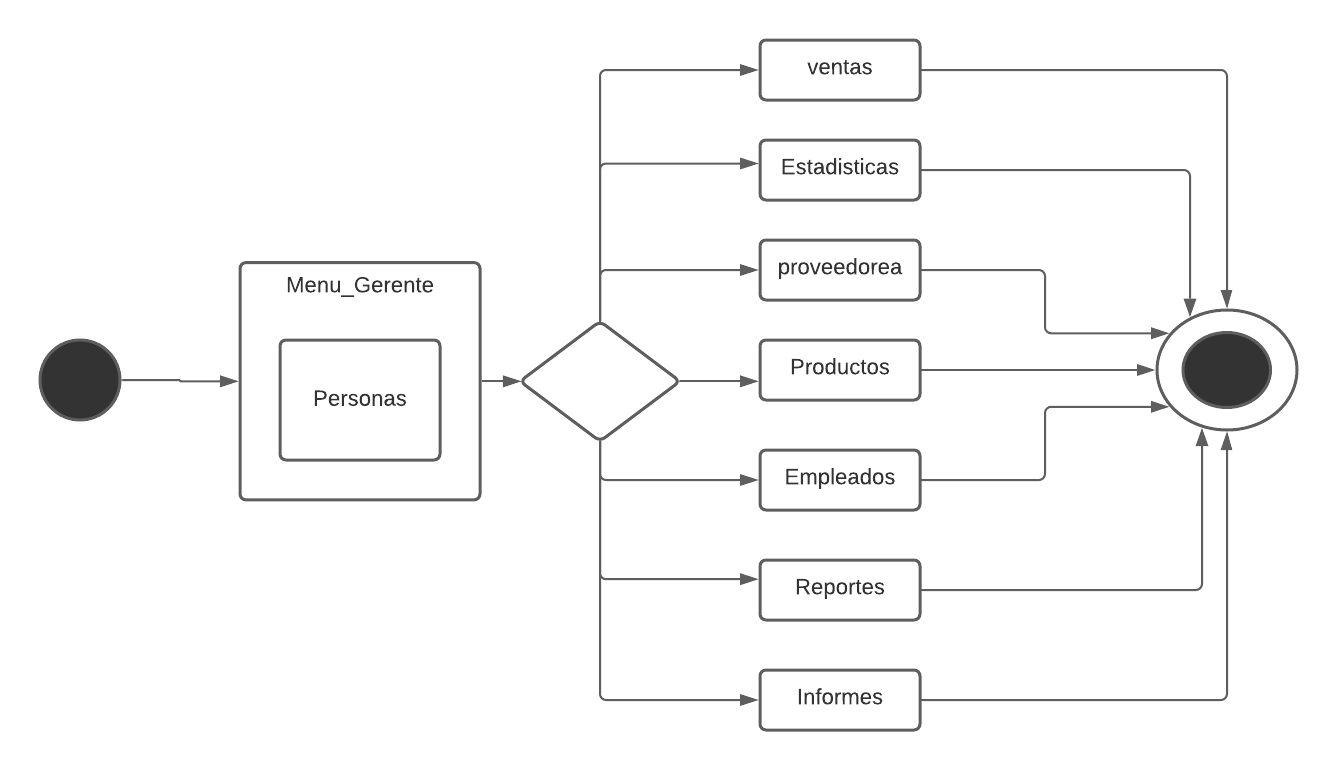


11.-Inicio\_Sesion

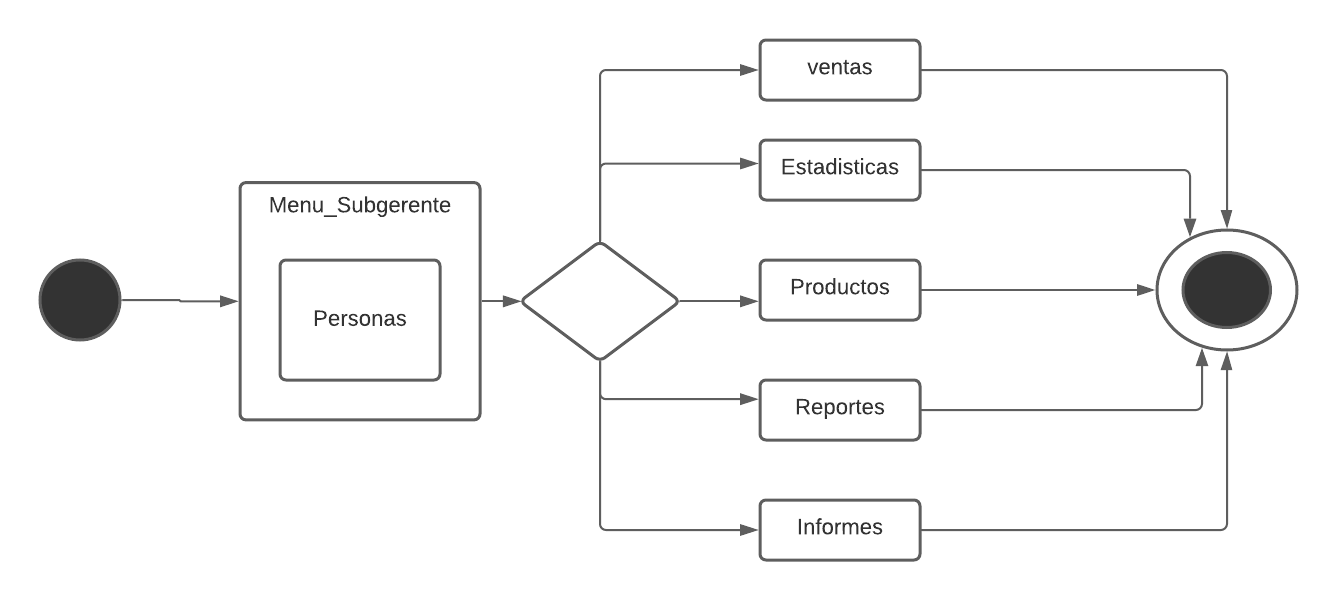


12.-Menu\_Empleado

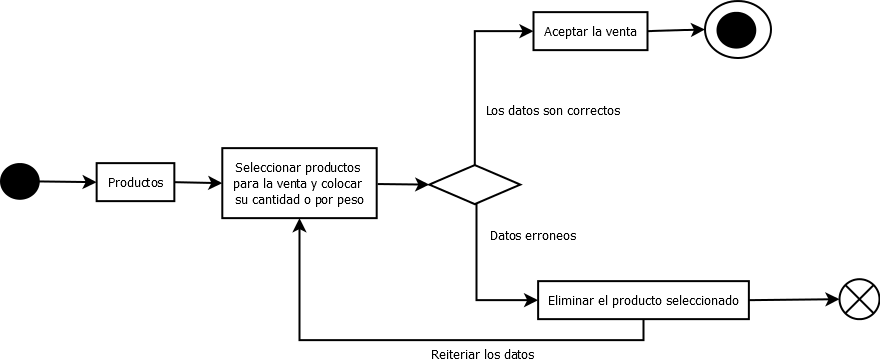


13.-Menu\_Gerente

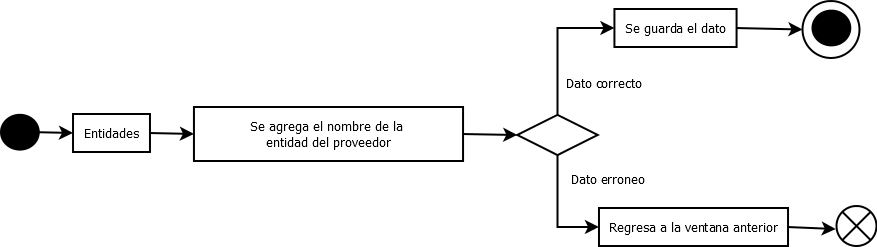
14.-Menu\_Subgerente



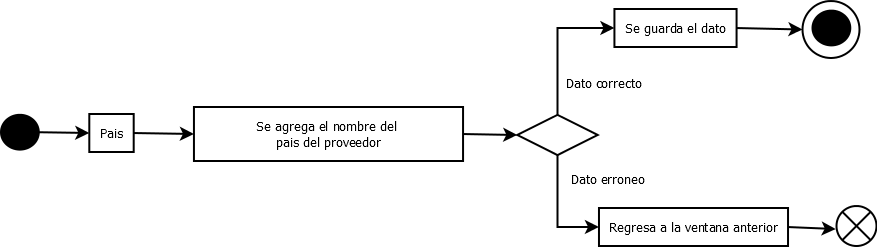
15.- Altas Ventas



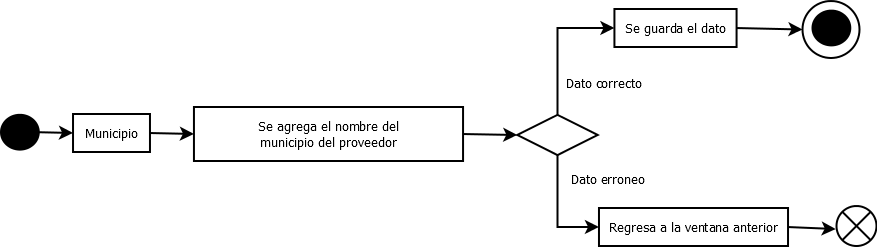
16.- Agregar país



17.- Agregar entidades

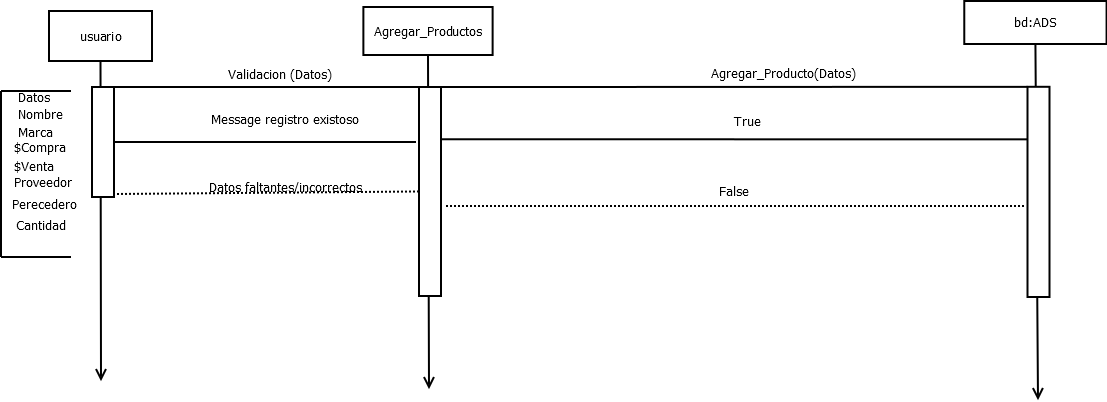


18.- Agregar municipio

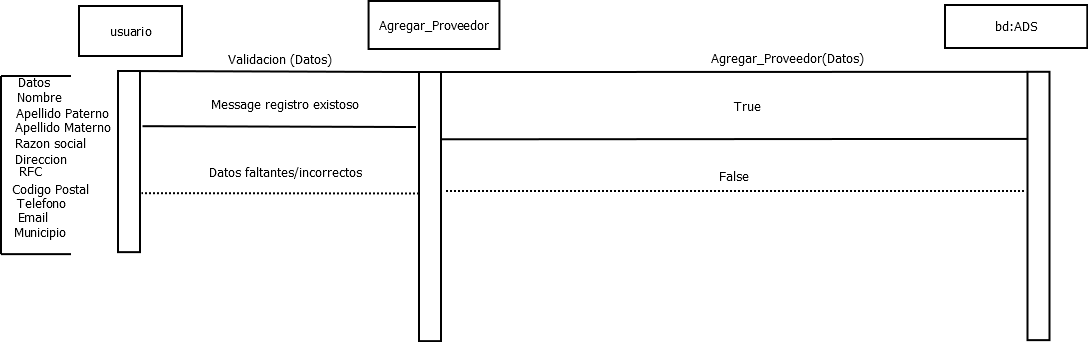


# Diagramas de secuencia

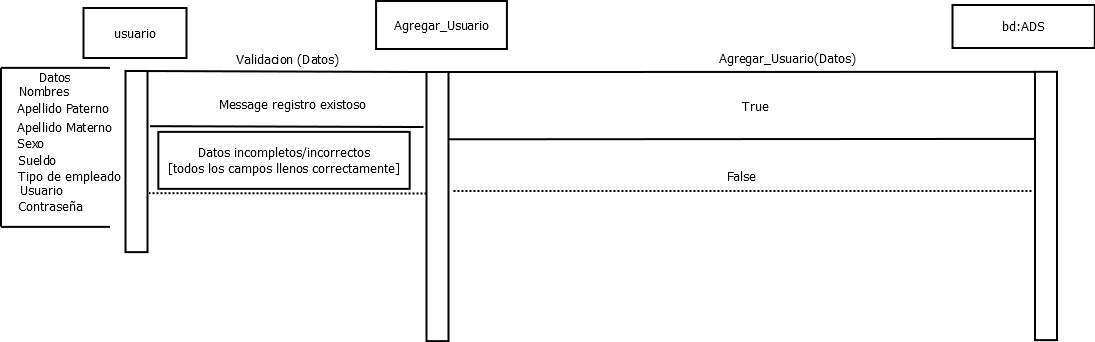
Agregar productos



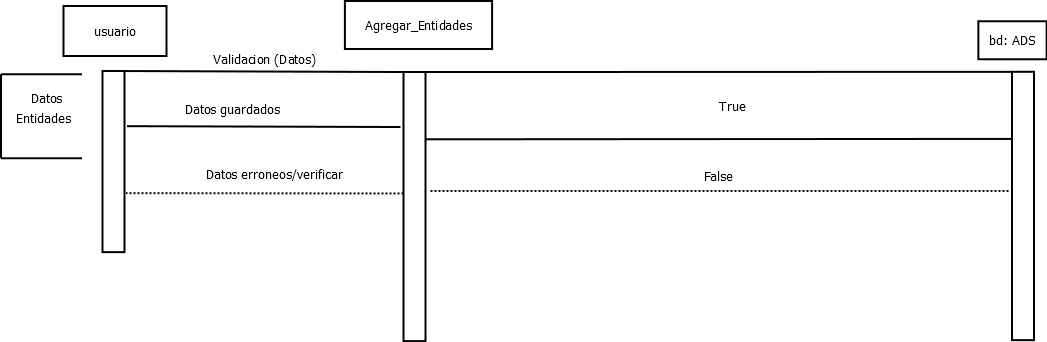
Agregar proveedor



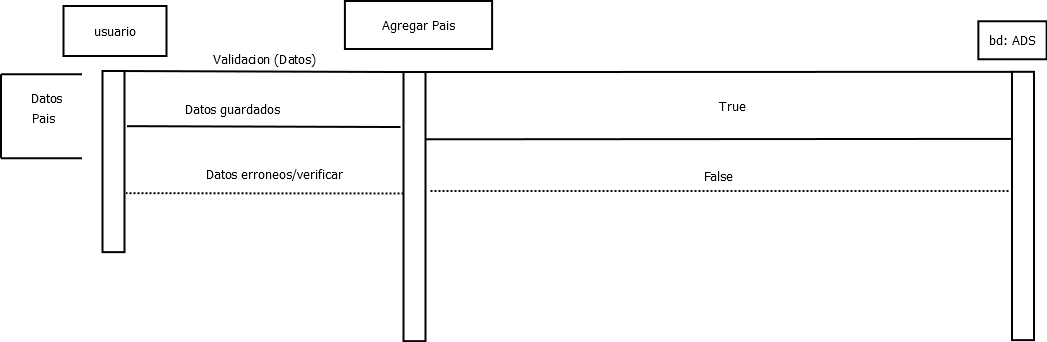
Agregar usuario



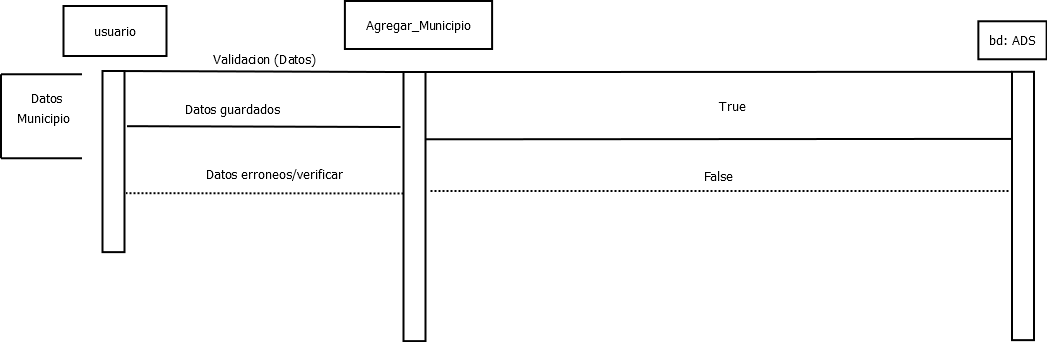
Agregar entidades



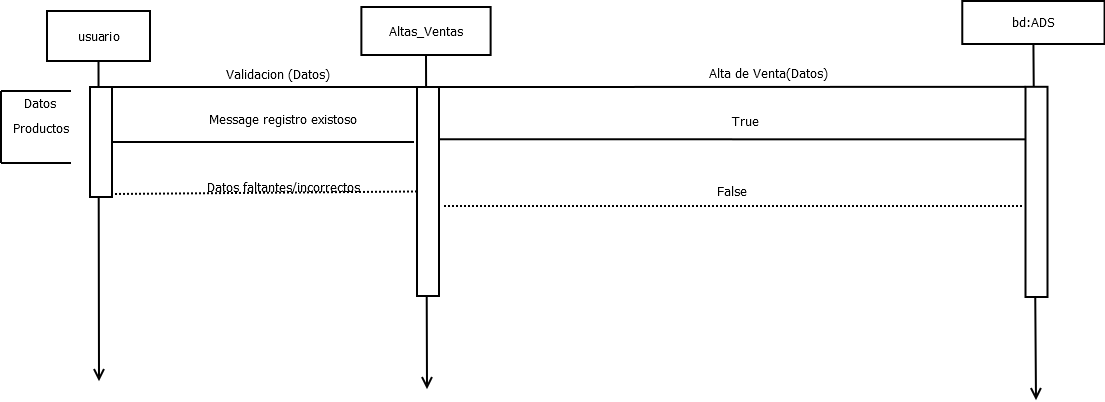
Agregar pais



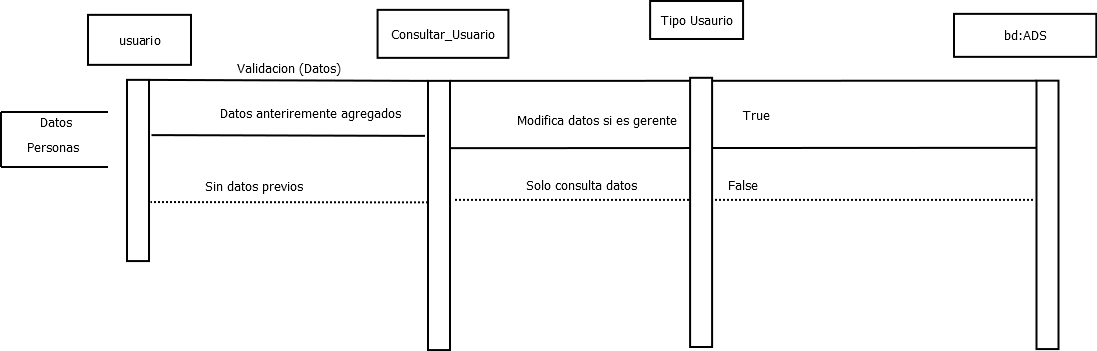
Agregar municipo



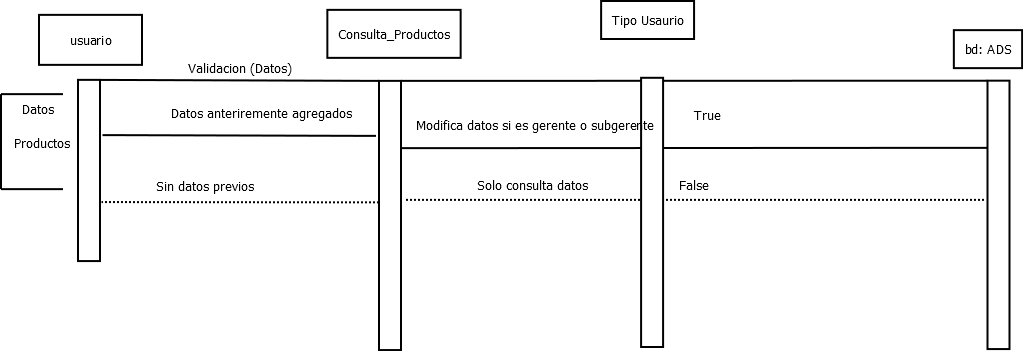
Altas ventas



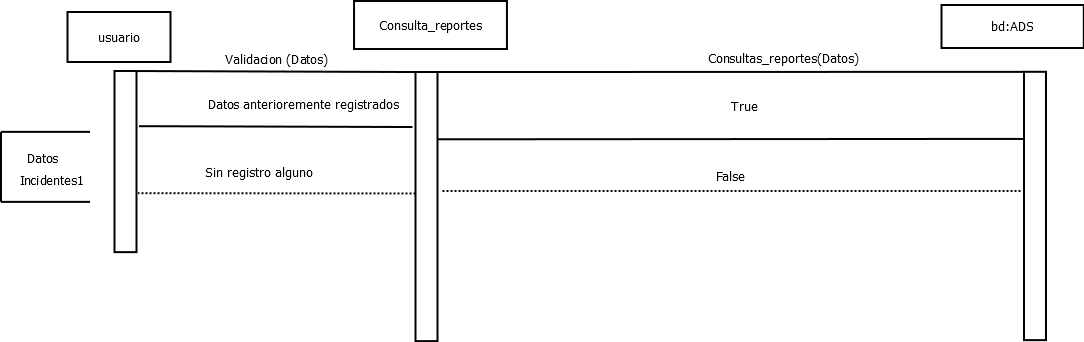
Consulta usuario



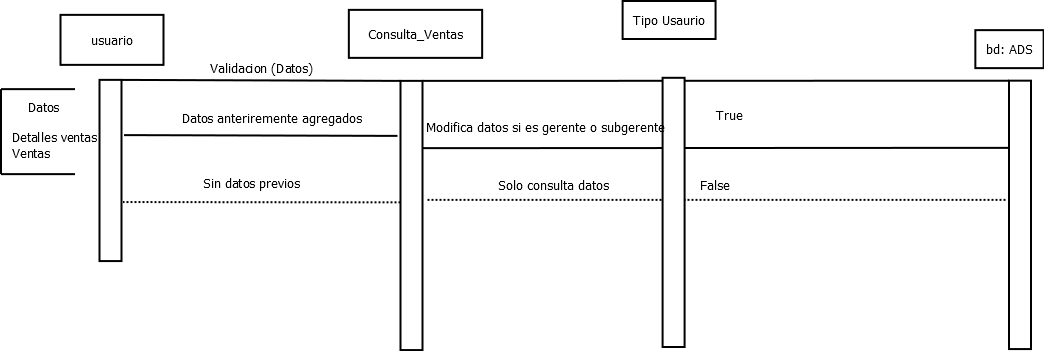
Consulta productos



Consulta reportes



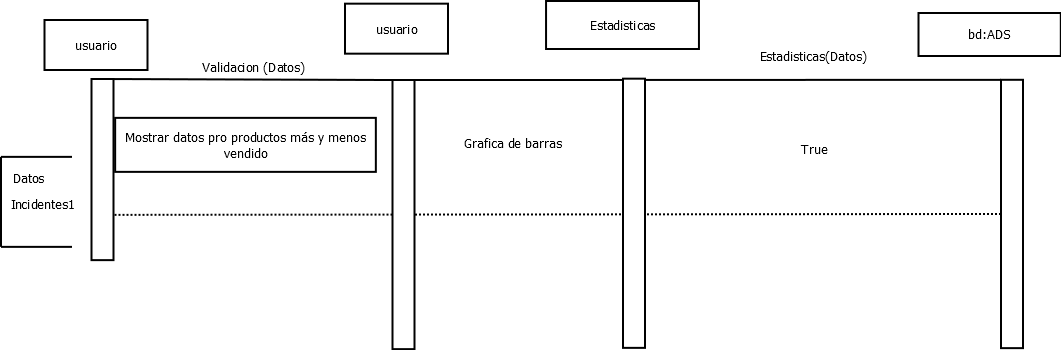
Consulta Ventas



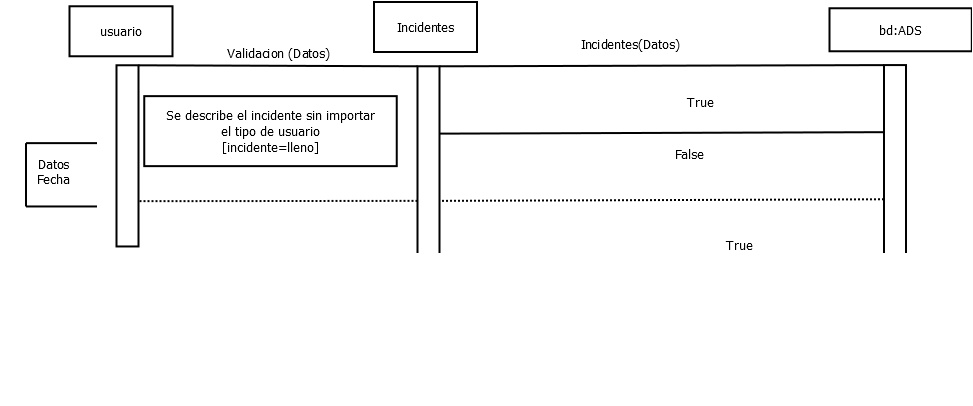
Consulta proveedores



Estadísticas



Incidentes

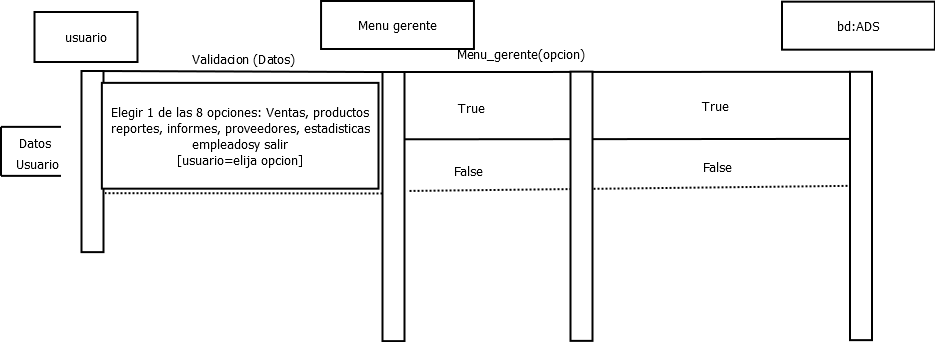


Incidentes 1

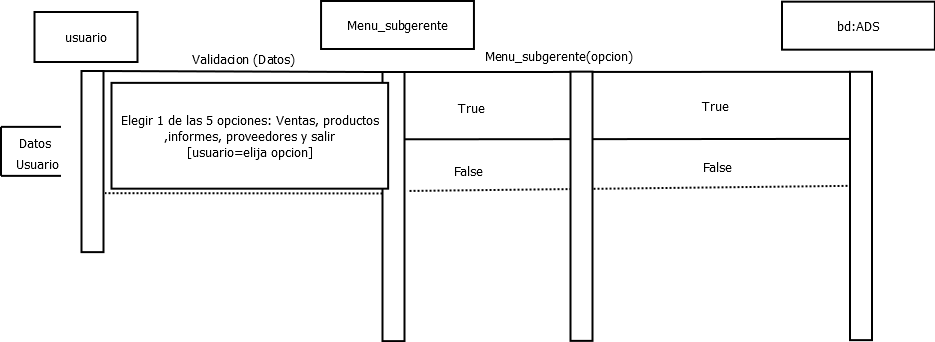
Inicio de sesión



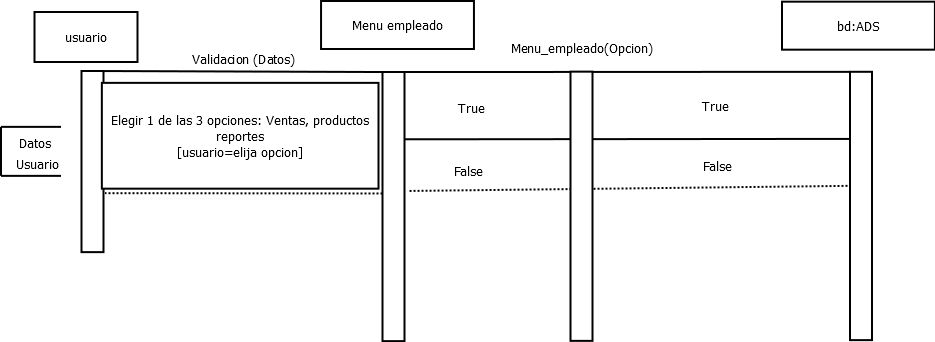
Menú de gerente



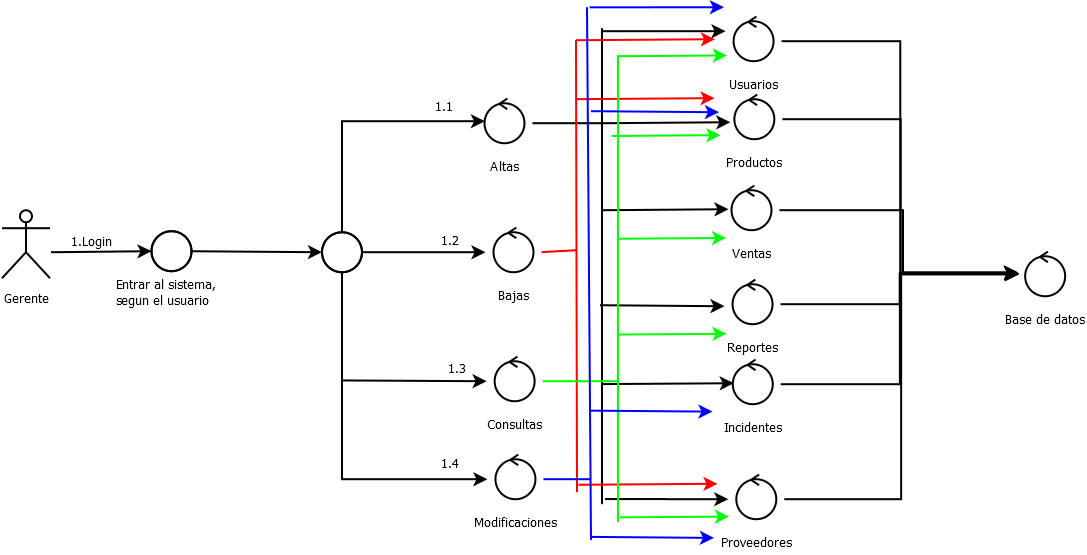
Menú de subgerente

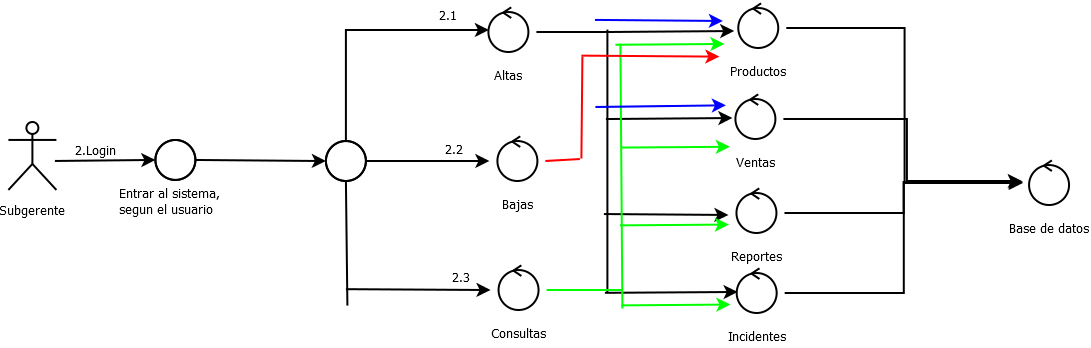


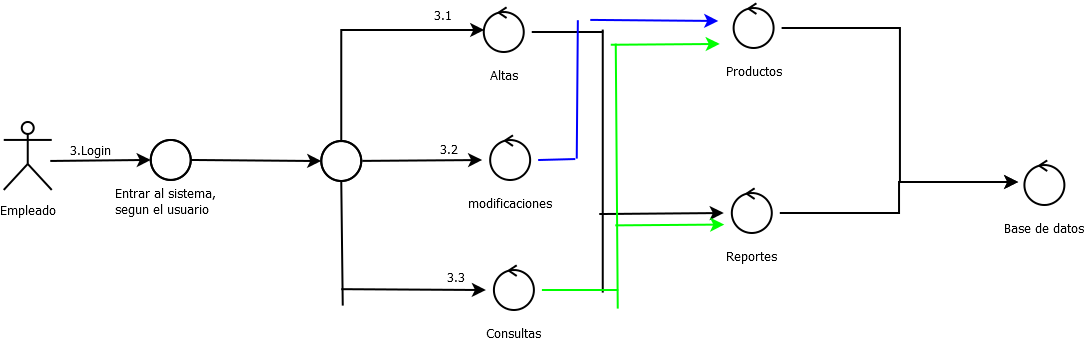
Menú de empleado



# Diagramas de comunicación

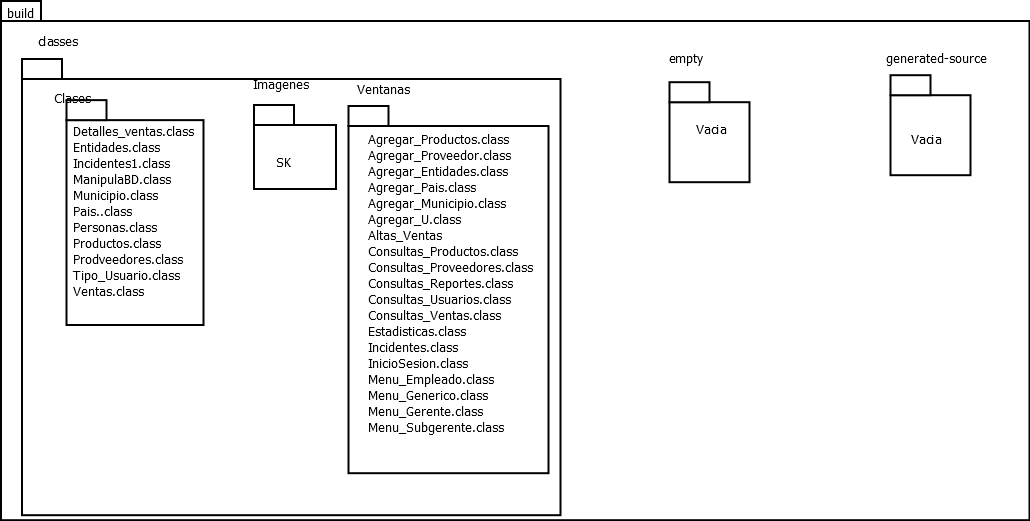


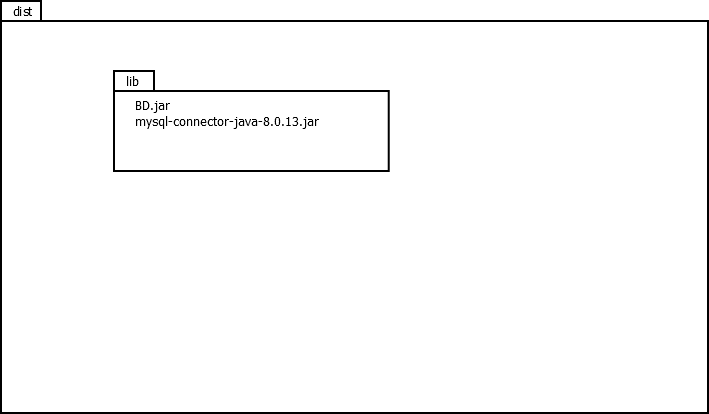


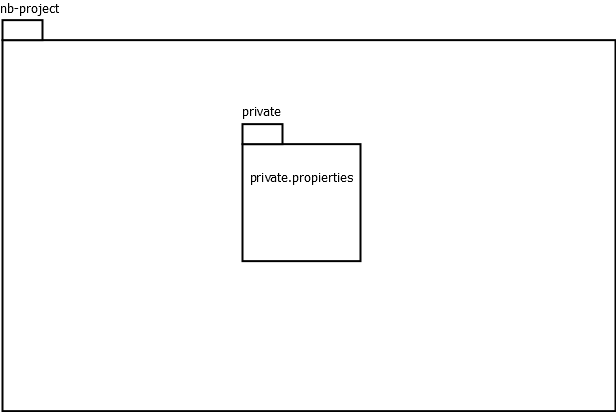


# Diagrama de paquetes

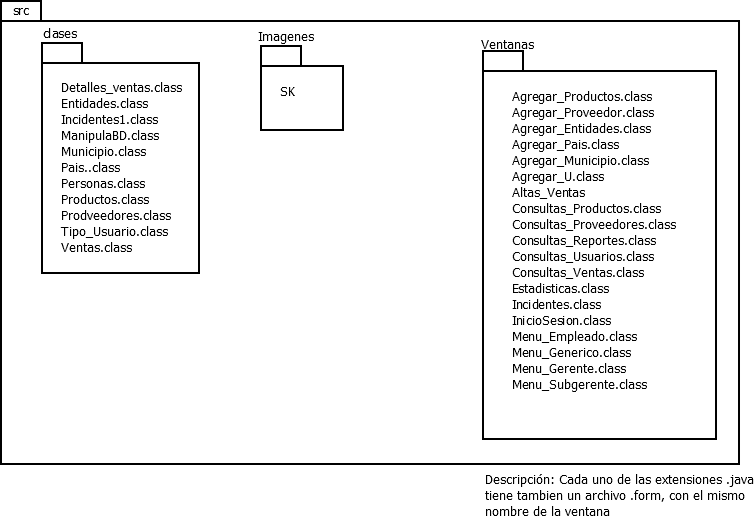
Carpeta: build



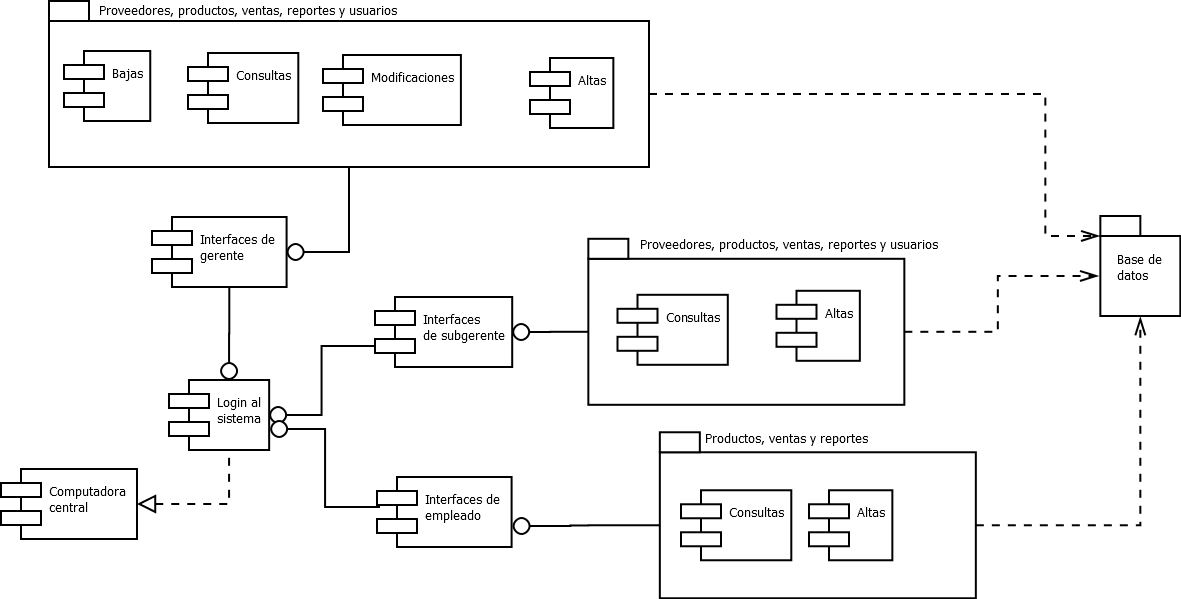
Carpeta: dist

Carpeta:nbprojectt

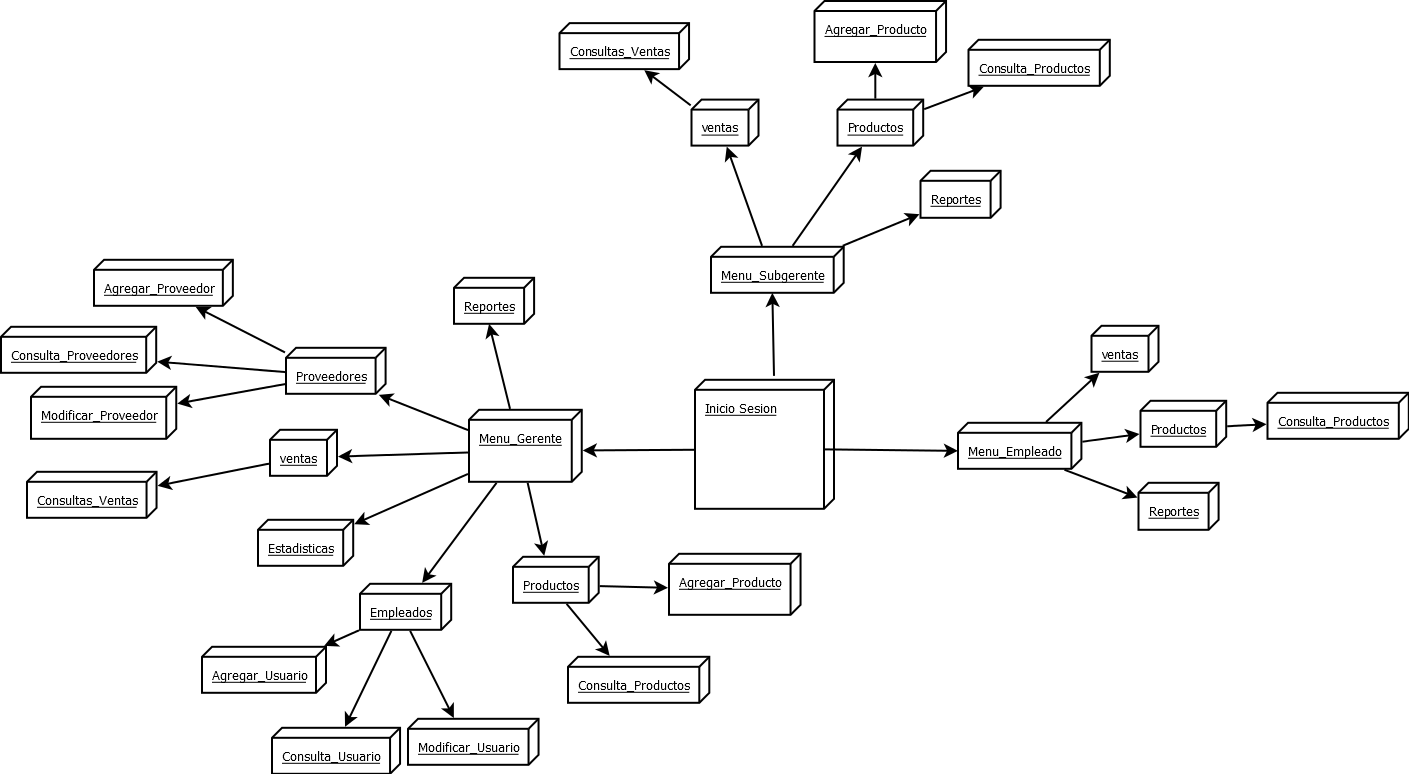
Carpeta src



# Diagrama de componentes



# Diagrama de despliegue



# Conclusiones

Conforme se fue avanzando en el desarrollo del proyecto, se detectaron muchas carencias en diferentes partes de la documentación, por lo que se tuvieron que realizar diferentes revisiones más detenidas para poder detectar los factores que estaban fallando desde el inicio de la documentación, hasta el prototipo de las interfaces. Las revisiones se hicieron en tiempo y forma para poder problemas posteriores en el desarrollo de la documentación, esto se logró gracias al uso de la metodología DorCu. De esta manera se logró detectar las carencias y/o cambios que se necesitaban en el análisis de los requerimientos de sistema y de usuario. Se desarrollaron los diagramas pertinentes para un mejor entendimiento del problema y así seguir con su desarrollo posterior, además de que son parte fundamental de la documentación de cualquier software. En el desarrollo de la base de datos, los integrantes del equipo carecían de conocimientos y dominio del tema, por lo que se tuvo que recabar información de distintas fuentes bibliográficas, para lograr el desarrollo optimo y satisfactorio del proyecto. Cada integrante del equipo logro de manera satisfactoria obtener los conocimientos suficientes para la revisión del proyecto, ya que cada miembro realizo tareas distintas dentro de la documentación, cada integrante pudo encontrar, fallas, carencias, cumplimientos y/o detalles que se encontraban dentro del mismo.

Cada uno de los temas abordados en clases fueron de suma importancia para cada uno de los procesos y tareas que se desarrollaron, ya que en algunas ocasiones varios factores dependen de otro más, y si ninguno se encuentra de la manera correcta, generan más problemas para el desarrollo posterior.

Este fue un proyecto que comenzamos a tomar desde el semestre pasado, en el cual vimos requerimientos, sin embargo tras cursar esta materia nos dimos cuenta de las carencias que tenía nuestro Software, la información que nos hacía falta para que el Software y la documentación funcionaran de la manera adecuada. La creación de los diferentes diagramas nos hizo darnos cuenta de las carencias o falta de razón dentro del Software, cada uno de estos diagramas también ayudaron a que la programación fuese más sencilla de hacer ya que teníamos una mejor idea de cómo es que funciona el proyecto, los tiempos, las clases, los objetos que se utilizan después en la programación. La creación de la base de datos de una manera más “profesional” gracias a todos los diagramas que vimos, así como realmente programar una base de datos para que funcionase el proyecto y no solo quedarnos en textos planos como en otros semestres o con “bases de datos” en Access, sino que realizar una base de datos con todos sus elementos dándola de alta por medio de Xampp y utilizando SQL, para poder crear cada una de las tablas utilizadas. La documentación realizada en este proyecto, ayudo a comprender lo importante que es realizar una documentación adecuada para un buen proyecto, ya que gracias a todos los diagramas es más sencillo comprender como es que funciona el proyecto y que cualquier ingeniero pueda utilizar posteriormente el proyecto y adaptarlo de acuerdo a lo que necesita sin la necesidad que interpretar desde 0

El desarrollo de un sistema de software es uno de los trabajos más importantes de la actualidad, pues de esta depende toda la infraestructura del mundo, y sabes hacer un software de alta calidad esta hiper demandado, por eso este proyecto me ayudo a entender las partes del desarrollo, lo que se debe tener en cuenta, lo que se debe diseñar, y como usarlo para mejorar mis proyectos, para aumentar la calidad y disminuir los errores que pueda tener la aplicación y con esto hacer algo que la gente pueda usar, le guste y recomiende, es un paso más para volvernos ingenieros de software, es por eso que este proyecto a pesar de darme dolores de cabeza y estrés, lo disfrute, pues mejore mis capacidades de análisis de software, también mejore mi trabajo en equipo incluso me puse a investigar herramientas para ser más eficientes, y por eso aprende a usar git y gihub y compartir ese conocimiento con mis compañeros que nos ayudo mucho, pues dejamos de enviar .zip cada vez que actualizábamos el proyecto y lo mejor, podíamos trabajar en paralelo nuestras partes y prácticamente fue un mes de pura documentación y en dos semanas se programo todo, lo cual me sorprendió, fue un sistema difícil para mi nivel, programado de manera tan sencilla que me dio flojera hacerlo, pues ya estaba casi todo, solo era ponerse enfrente de la computadora, abrir el IDE y ponerse a escribir, pues la lógica ya estaba, y lo mejor de eso, si encontrábamos errores, podíamos actualizar ese detalle en la documentación y seguir trabajando es un proyecto que yo en lo personas disfrute, y espero saber cómo aplicarlo en un futuro en mis proyectos personales.

# Bibliografía

Date, C. (s.f.). *Introducción a los sistemas de bases de datos.* 2005: Pearson.

Elmasri, R. (2007). *Fundamentos de sistemas de bases de datos.* Pearson.

Fowler, M. (2003). *UML gota a gota.* Addison Wesley.

Kendall, K. E. (2005). *Analisis y diseño de sistemas.* New Jersey: Pearson.

Nevado, V. (2010). *Introduccion a las bases de datos.* Madrid: Vision libros.

Pressman, R. S. (2006). *Ingenieria del software: un enfoque práctico.* McGraw-Hill.

Seidl, M. (2015). *UML Classroom.* Springer.

# Anexos

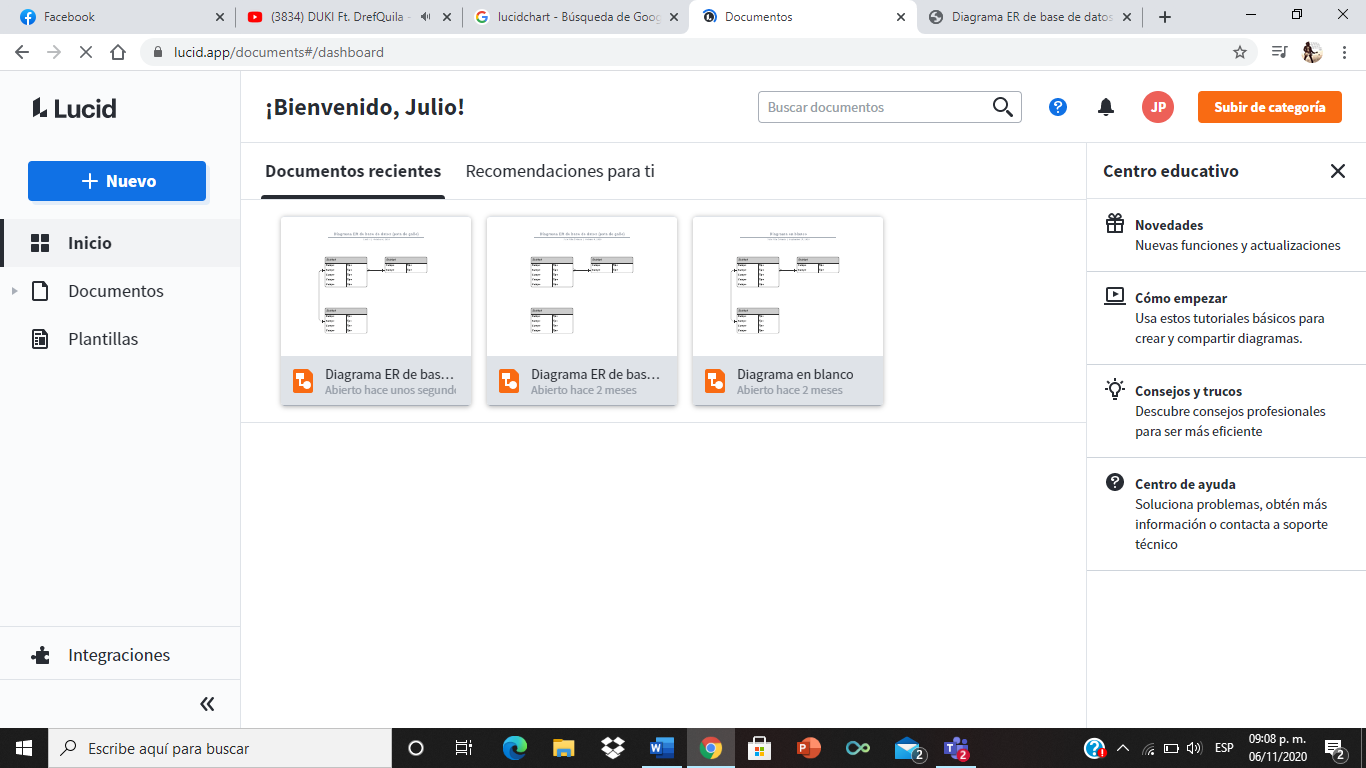
# Enlace de la carpeta con el video y el programa:

<https://drive.google.com/drive/folders/1Cp-Vvr_F_QhpYIL8qPHrVlNjX08F6lCj?usp=sharing>

Herramientas case utilizadas

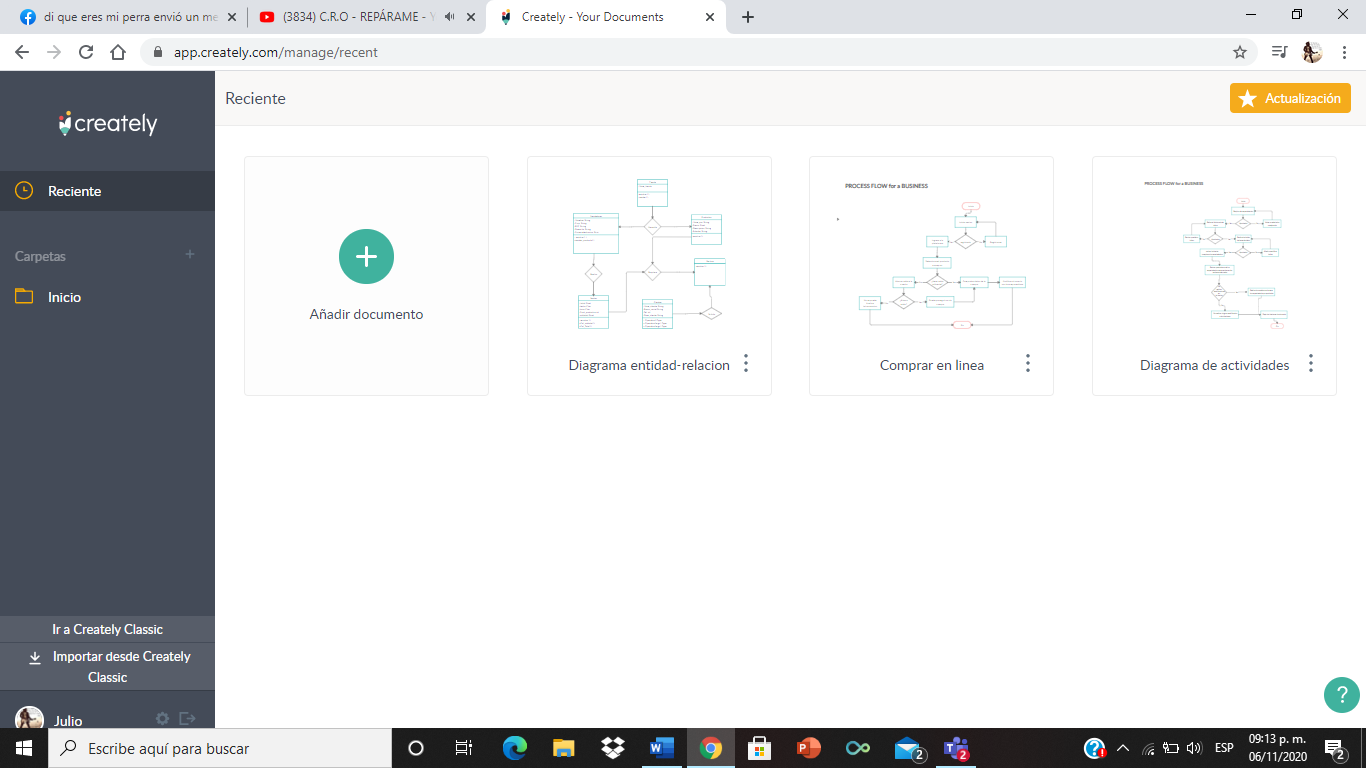
Lucidhart (anexo 1)

Lucidchart es una herramienta de diagramación basada en la web, que permite a los usuarios colaborar y trabajar juntos en tiempo real, creando diagramas de flujo, organigramas, esquemas de sitios web, diseños UML



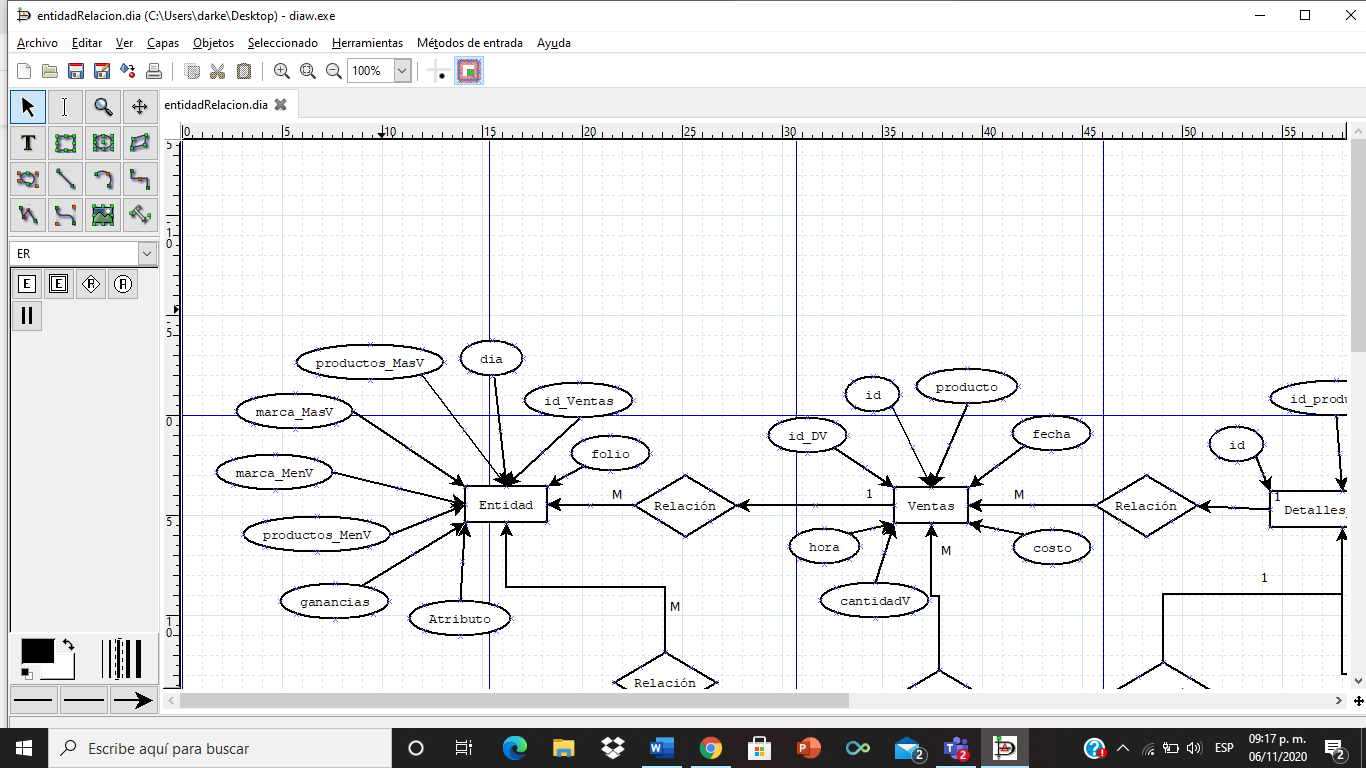
Creatly (anexo 2)

Es una herramienta de diagramación galardonada, reconocida por su facilidad de uso. Especialmente para el desarrollo online de diagramas online



Dia (anexo 3)

Es una aplicación informática de propósito general para la creación de diagramas, totalmente gratuito que no requiere de muchos recursos computacionales.



Netbeans (anexo 4)

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.

