

Centro Educativo Siloé Amatitlán

4ta. Avenida 0-80

Tel. 6633 2036

centro\_edu\_siloe@hotmail.com

Quinto Bachillerato en Computación con Orientación Científica

**Sistema administrativo de clientes y recursos**

Fatima Stephania Santos Lorenzana

Clave: 16

Profesor Pajoc Raymundo, José Ernesto

Asesor de Práctica

San Juan Amatitlán, Guatemala agosto 2022

**Fase 2**

# Planteamiento del problema

* 1. **Definición**

El proyecto consiste en el desarrollo de una aplicación de escritorio con un menú de funciones administrativas clasificadas por funcionalidad, capaz de generar de forma dinámica distintas interfaces para que el usuario pueda administrar un negocio. Contará con un sistema de carga de archivos, y la capacidad de generar reportes estadísticos con HTML.

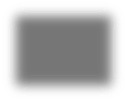
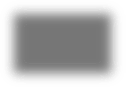
# Aplicación

A continuación, se definen y describen las vistas con las que debe contar la aplicación.

* + 1. ***Autenticación***

# ¿En qué consiste?

La autenticación es parte importante de todo sistema, y cuando se trata de información crucial para el negocio, debe tratarse con particular precaución. Para esto, el estudiante deberá crear un sistema de registro y autenticación de usuarios, donde se permita registrarse y se requiera autenticación para utilizar la aplicación.



# Consideraciones de la interfaz de autenticación.

* La cantidad máxima de usuarios posibles en el sistema es de 10.
* No puede haber dos usuarios con el mismo nombre de usuario.
* Debe existir un usuario maestro capaz de ingresar al sistema sin necesidad de crearse.
* En caso de olvidar su contraseña, el usuario deberá poder recuperarla.

## Ventana principal

Una vez autenticado el usuario, tendrá acceso a la ventana principal del proyecto, la cual debe contar con las siguientes funciones (cada una se describe a detalle más adelante):

* Administración de clientes
* Administración de productos
* Reportes

La ventana principal debe ser capaz de trasladar el foco de la aplicación hacia la vista de la función a la que se acceda. La distribución de los menús y de los componentes propios de cada funcionalidad queda a discreción del estudiante.



## Interfaz de administración de clientes

La administración de clientes permite analizar la interacción e información con el fin de fortalecer la preferencia y mejorar el servicio. Esta área tiene como propósito, brindar al usuario herramientas de creación, actualización, eliminación, y consulta de datos de clientes. Para esto, deben desarrollarse las siguientes funciones:

* + - 1. **Dashboard de clientes en el sistema.** Presenta al usuario la información más

importante de los clientes registrados en el sistema. Debe contar con los siguientes elementos:

* + - * + Tabla con la información de todos los clientes en el sistema.
        + Gráfica de pie de clientes por sexo.
        + Gráfica de barras de cantidad de clientes por rango de edad.
        + Es permitido el uso de la librería llamada JFreeChart.
      1. **Carga masiva de clientes.** Para el ingreso de datos de clientes en el sistema,

deberá cargarse un archivo CSV con la información de los clientes. A continuación, se especifica la estructura del archivo:

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre (String) | Edad (int) | Sexo (char) | NIT (int) |
| ABC | 20 | M | 123 |
| DEF | 40 | F | 456 |

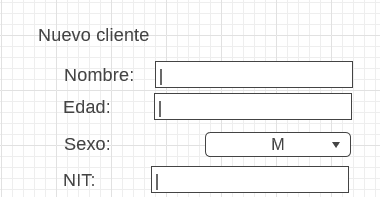
Ejemplo de un archivo que sigue la estructura anterior:

ABC,20

,M,123

* + - 1. **Creación de un cliente.** Esta interfaz permite ingresar un nuevo cliente al

sistema. Deberá validarse que el NIT ingresado no esté registrado con anterioridad, y presentarse un formulario para el ingresode datos como el siguiente:



* + - 1. **Vista informativa de un cliente.** Presenta al usuario la información ingresada

sobre el cliente. Para esto debe realizarse una búsqueda por NIT.



* + - 1. **Modificación de un cliente.** Permitirá modificar los datos ingresados acerca de

un cliente. Debe verificarse que, si se modifica el NIT, el mismo no esté registrado previamente. La interfaz de modificación debe contar con los mismos campos que la interfaz de creación de clientes.

* + - 1. **Eliminación de un cliente.** Eliminará a un cliente del sistema. Para esto, debe

hacerse la búsqueda por NIT. La eliminación puede estar presente en la vista informativa, o realizarse como función individual.

# Consideraciones de la interfaz administrativa de clientes.

* La cantidad máxima de clientes posibles en el sistema es de 100.
* No puede haber valores de NIT repetidos.

## Interfaz de administración de productos

Los productos son la esencia del negocio, y sus datos se encuentran en constante cambio. Por esto, es importante tener un control de lo que sucede en el inventario de la empresa. A continuación, se describen las funciones con las que debe contar el sistema:

* + - 1. **Dashboard de productos.** Presenta al usuario la información más importante de

los productos registrados en el sistema. Debe contar con los siguientes elementos:

* + - 1. **Carga masiva de productos.** Para el ingreso de datos de productos en el sistema,

deberá cargarse un archivo CSV con la información de los productos. A continuación, se especifica la estructura del archivo:

* + - * + Tabla con la información de todos los productos en el sistema.
        + Gráfica de barras de la cantidad de productos por rango de precio.

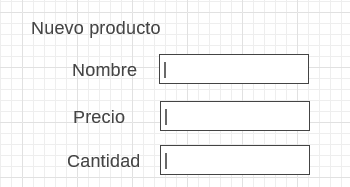
|  |  |  |
| --- | --- | --- |
| Nombre (String) | Precio (float) | Cantidad (int) |
| XYZ | 59.99 | 10 |
| JKL | 199.99 | 200 |

Ejemplo de un archivo que sigue la estructura anterior:

XYZ,59.99,10 JKL,199.99,200

* + - 1. **Creación de un producto.** Esta interfaz permite ingresar un nuevo producto en el

sistema. Deberá validarse que el nombre ingresado no esté registrado con anterioridad, y presentarse un formulario para el ingreso de datos como el siguiente:



* + - 1. **Vista informativa de un producto.** Presenta al usuario la información

ingresada sobre el producto. Para esto debe realizarse una búsqueda por nombre.



* + - 1. **Modificación de un producto.** Permitirá modificar los datos ingresados sobre un

producto. Debe verificarse que, si se modifica el nombre, el mismo no esté registrado previamente. La interfaz de modificación debe contar con los mismos campos que la interfaz de creación de productos.

* + - 1. **Eliminación de un producto.** Eliminará un producto del sistema. Para esto, debe

hacerse la búsqueda por nombre. La eliminación puede estar presente en la vista informativa, o realizarse como función individual.

# Consideraciones de la interfaz administrativa de productos.

* La cantidad máxima de productos posibles en el sistema es de 100.
* No puede haber valores de nombre repetidos.

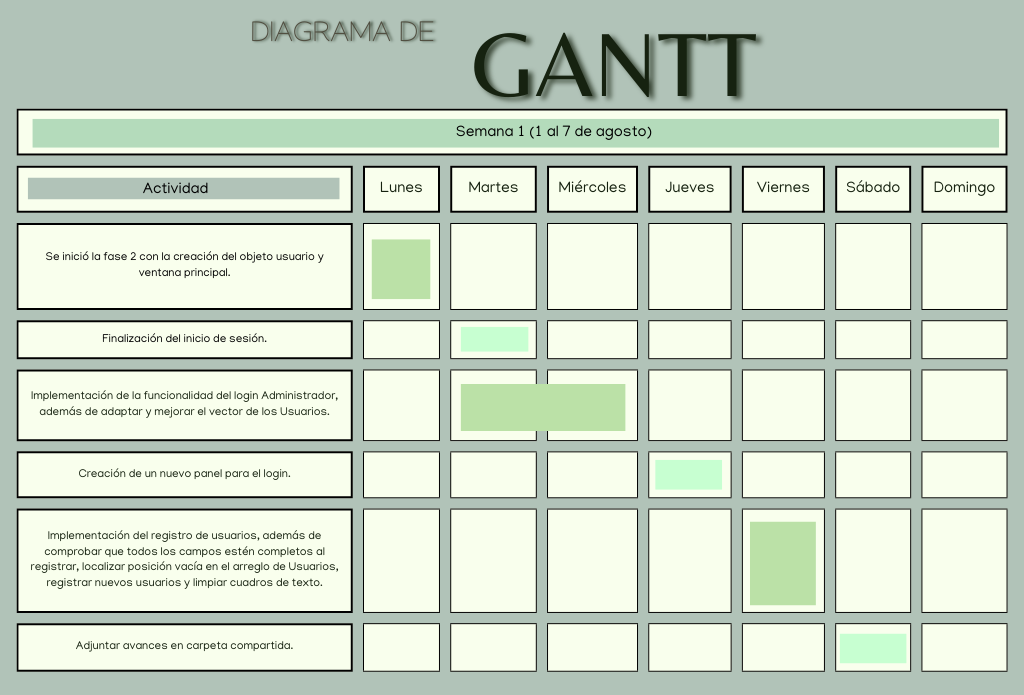
## Reportes

Todo sistema administrativo debe poder generar reportes sobre el estado del sistema. Para esto, se desarrollará un apartado que permita visualizar la información de la aplicación de manera atractiva, haciendo uso del lenguaje HTML. Los reportes son generados por el personal, por lo que deben incluir el nombre del usuario que los ha generado. Los reportes solicitados son los siguientes:

# En HTM.

* Listado de todos los clientes registrados, ordenados por edad de mayor a menor.
* Listado de todos los productos registrados, ordenados por precio de mayor a menor. Si un producto se encuentra agotado, debe distinguirse.

# Cronograma

**Figura 1.**

# Figura 2.

**Figura 3.**

Figura 1,2 y 3:Estas representan las diversas actividades que se realizaron en las semanas correspondientes indicadas al inicio de cada tabla; las actividades se encuentran en el lado izquierdo mientras que los días necesitados para completarlas se encuentran del lado derecho.

1. **Recursos**

**3.1 Computadora**

**3.2 Cable de red**

**3.3 Servicio de internet**

**3.4 Java NetBeans**

1. **Estimación del presupuesto**
2. **Diagramas, algoritmos y bosquejos**

**5.1 Diagramas de clases.**

**Figura 4.**

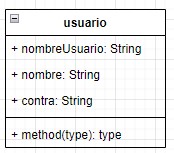


Figura 4: Representa el objeto “usuario”, el cual es un molde que nos facilita manejar diversos usuarios con diversas características; sin embargo, estos utilizan los mismos atributos.

**Figura 5.**

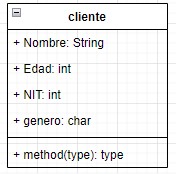


Figura 5: Representa el objeto “cliente”, el cual es un molde que nos facilita manejar diversos clientes con diversas características; sin embargo, estos utilizan los mismos atributos.

**5.2 Algoritmos**

***5.2.3 Inicio de sesión***

* **Paso 1:** En primera instancia debemos de trabajar en la parte lógica del sistema; por lo tanto, debemos de crear un objeto “usuario” para facilitar el proceso de almacenamiento; prosiguiendo con esto, debemos de crear la clase “usuario” en nuestro proyecto y colocar nuestros atributos.

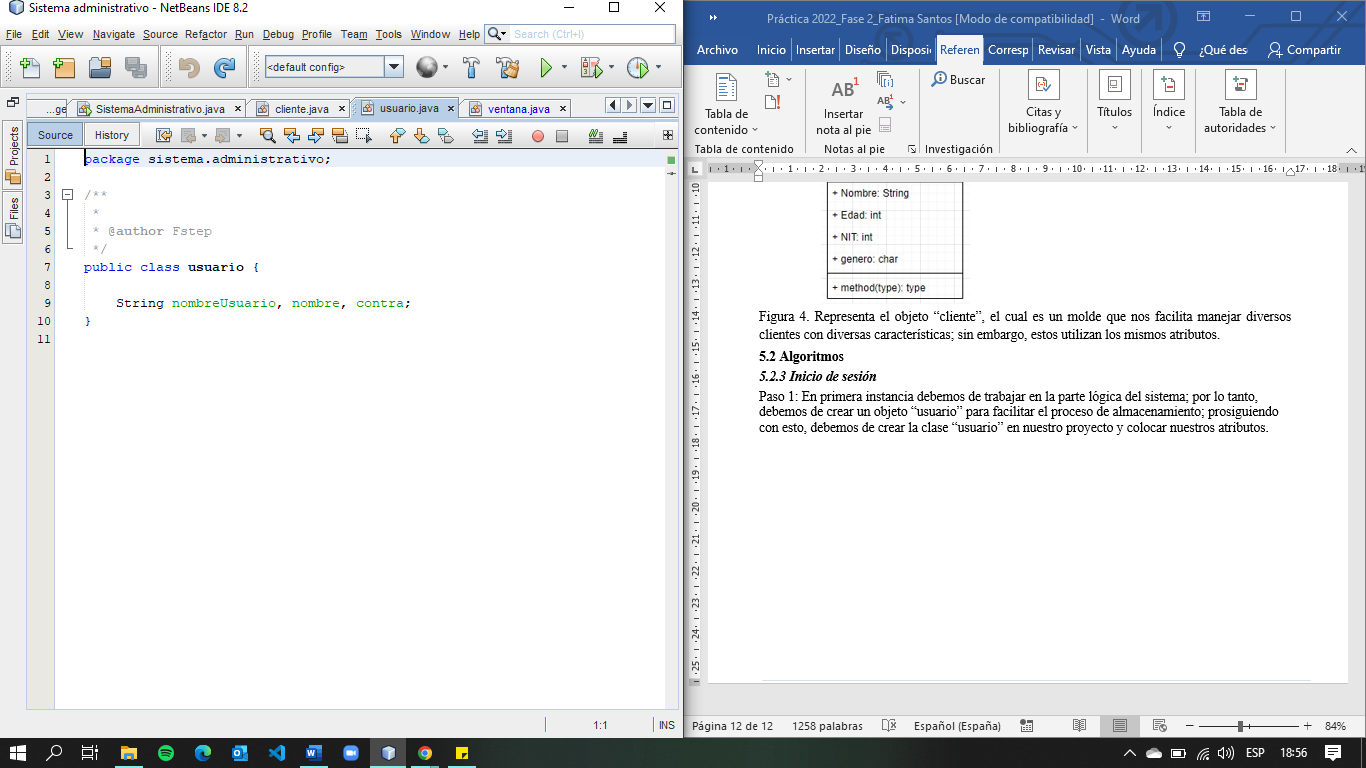
**Figura 6.**

Figura 6: Representa la implementación de la clase “usuario” en el proyecto con sus respectivos atributos.

* **Paso 2:** Crear un vector en el método principal para los usuarios para solventar la cantidad máxima de los usuarios que debe de permitir el proyecto, además de esto debemos de crear un usuario “maestro”; por lo tanto, este usuario maestro debe de presentarse en la posición 0 y colocarle los atributos correspondientes; por otro lado, más adelante debemos de modificar su ubicación para tener un mejor alcance.

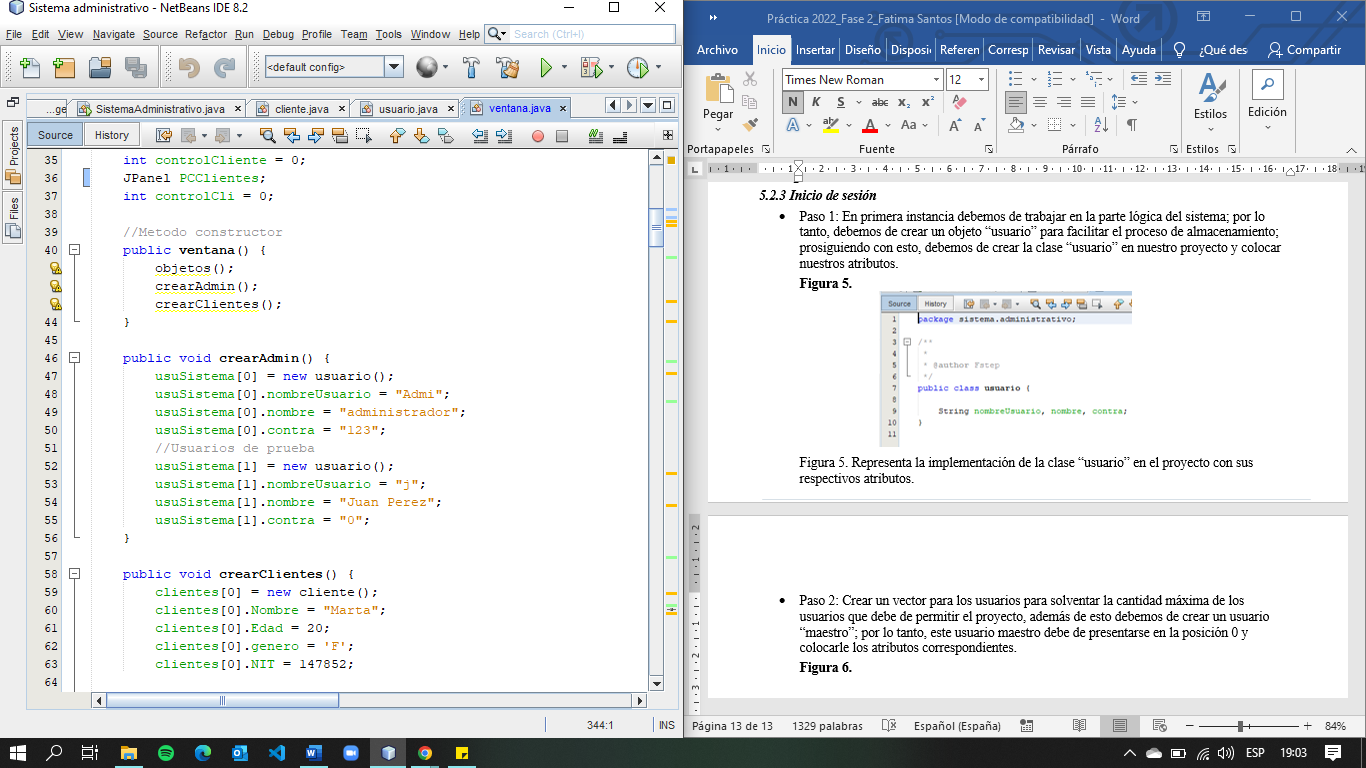
**Figura 7.**

Figura 7: Representa la creación del usuario “master” y un usuario de prueba, para verificar el funcionamiento del sistema.

* **Paso 3.** Para probar su funcionamiento empezaremos a realizar el entorno gráfico, por ese motivo debemos de crear un método llamado “ventana”, para colocar todos los elementos gráficos correspondientes, sin embargo, debemos de heredar características de nuestro método principal.

**Figura 8.**

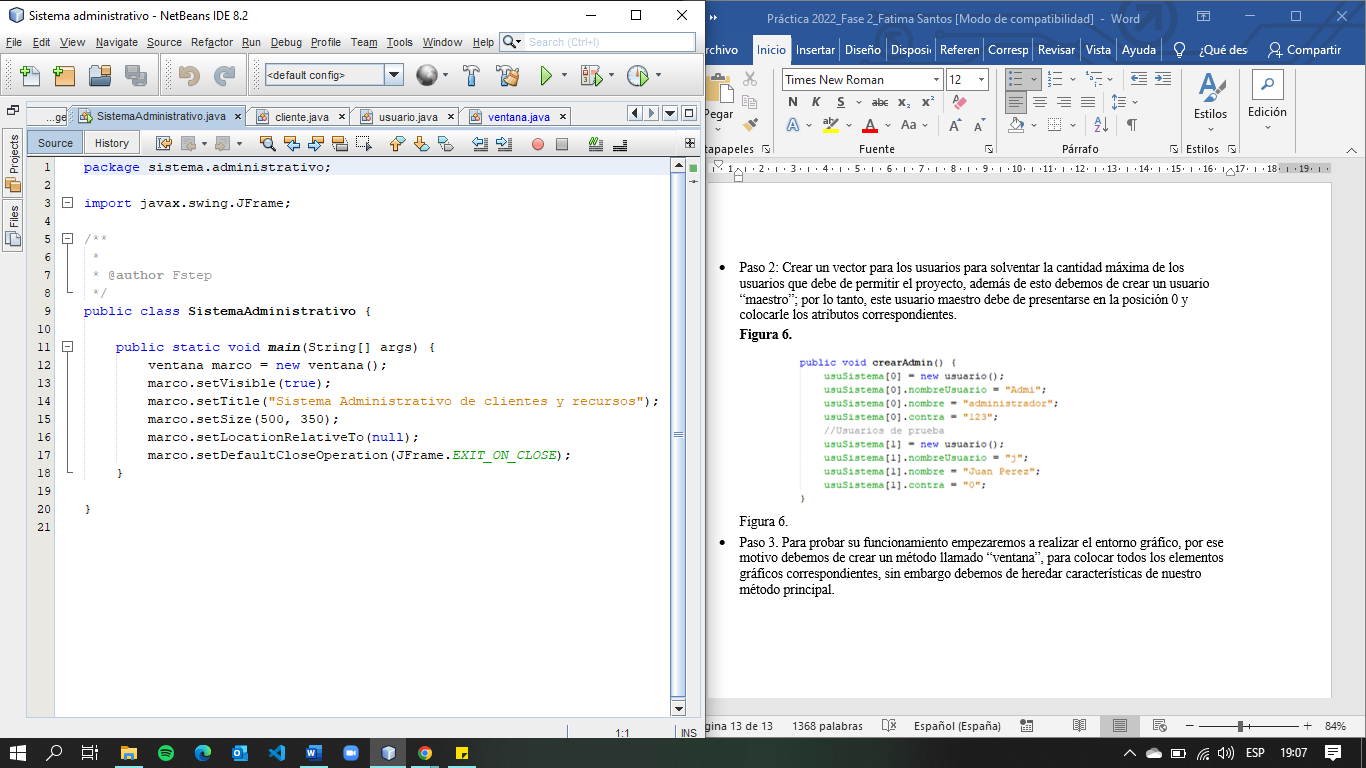
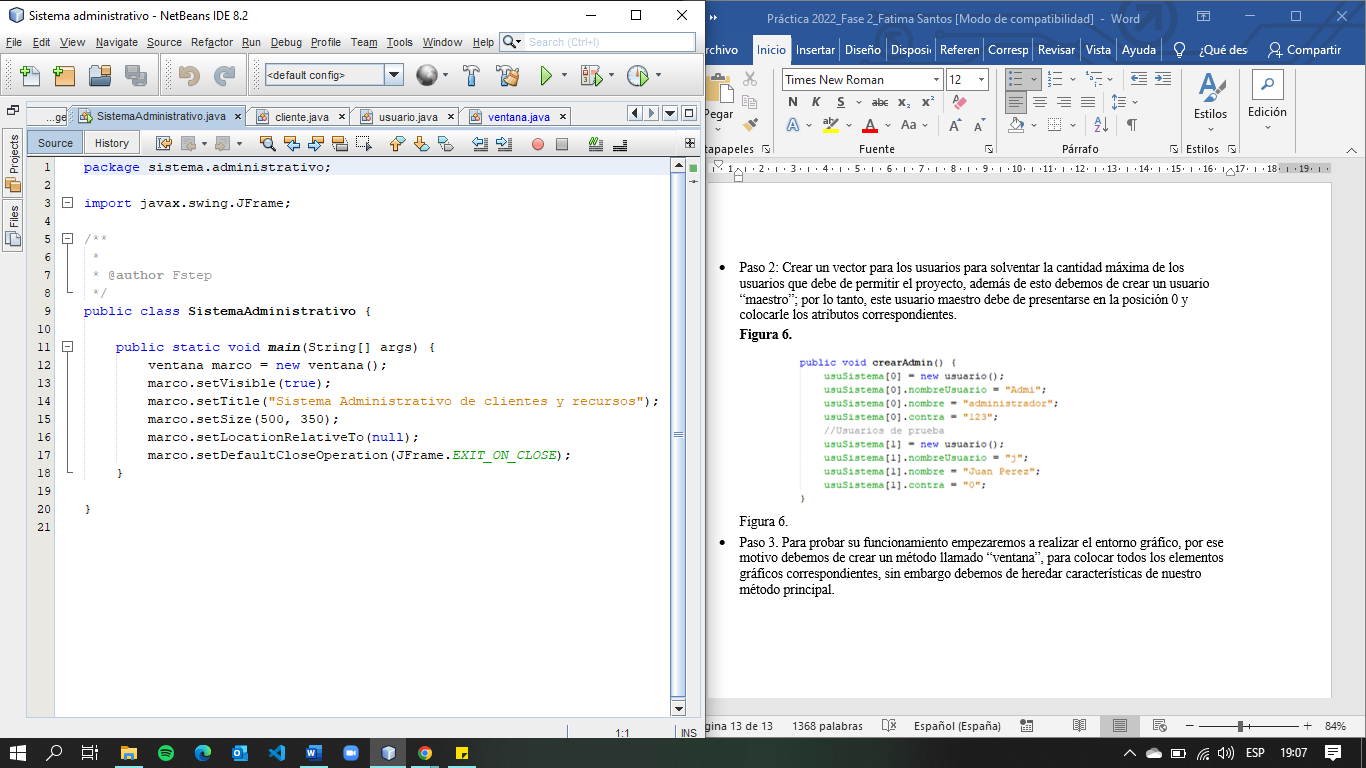


Figura 8: Representa la implementación de las características que necesitará nuestra ventana para gestionar los diversos elementos gráficos.

* **Paso 4:** En la clase ventana debemos de crear nuestro panel, para ello debemos de crear un método “objetos”, el cual nos permitirá programar los diversos elementos gráficos; posterior a esto debemos de crear nuestro panel para ello debemos de importar la librería adecuada, además a nuestra “Ventana” debemos de implementar un método constructor que nos permitirá llamar nuestro “objetos”; además debemos de colocar nuestro método del usuario” maestro”.
* **Paso 5:** En nuestro método “objetos”, debemos de colocar nuestras JLabel, JPassword, TxtField, que le permitirá al usuario poder interactuar con nuestro programa y a nosotros poder almacenar los datos que este ingrese, además debemos de implementar setLayout para que este tome las medidas que nosotros le asignemos.

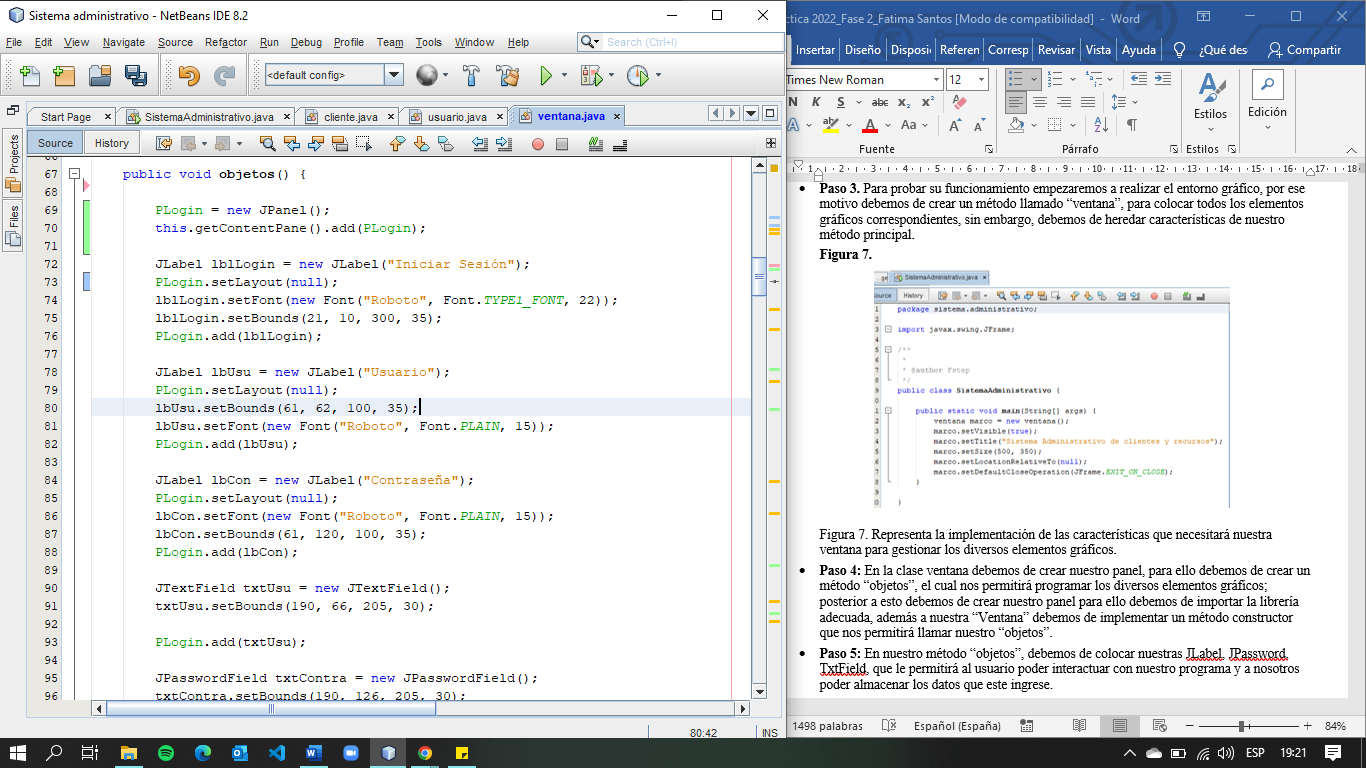
**Figura 9.**

Figura 9: Representa la implementación de las etiquetas entre otros elementos.

* **Paso 6.** Debemos de agregar nuestro botón, el cual permitirá que el usuario pueda ingresar al programa; para ello debemos de crear un evento, con el comando “ActionListener” más el nombre de nuestro evento es igual a “new ActionListener”. Ahora debemos de colocar un “if” para poder comparar los datos que se encuentran almacenados con los datos que el usuario esta ingresando para evaluar si se le da permiso de entrar al programa; además debemos de generar una notificación para que el usuario llene todos los campos.

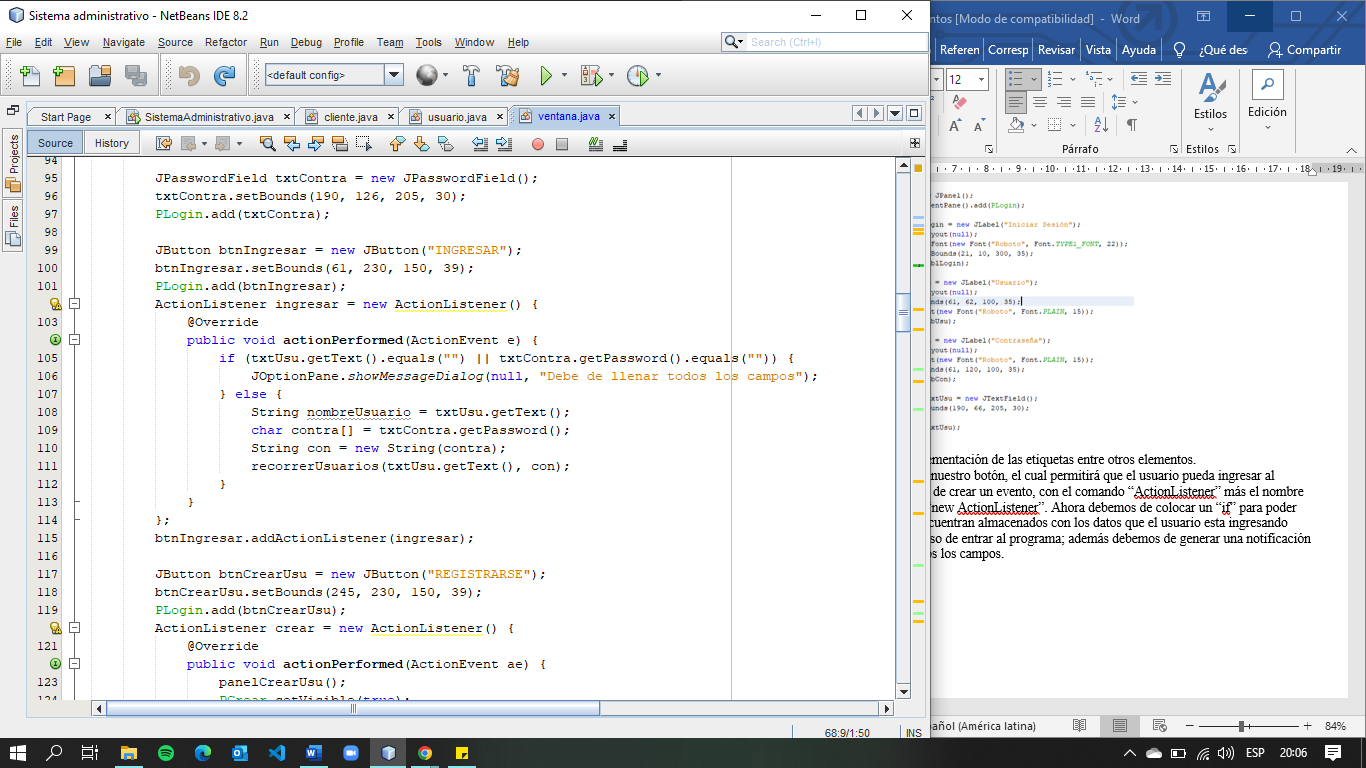
**Figura 10.**

Figura 10: Representa el proceso utilizado para evaluar si se le otorga el permiso de ingresar al usuario y de mencionarle que debe de llenar todos los campos para poder ingresar si es el caso de que deje uno o varios vacíos; por otro lado, debemos de agregar algunas variables nuevas que nos facilitarán el proceso de evaluación (String nombreUsuario, char contra y String con) para que más adelante podamos optimizar el proceso con otro método (“recorrerUsuarios(txtUsu.getText(), con); ”).

* **Paso 7:** Por consiguiente, debemos de crear un nuevo método en el cual se evaluará con mejor detalle los datos ingresados del usuario, no solo los datos del usuario “master” si no que también de los usuarios futuros, para ello debemos de implementar un ciclo “for” para evaluar los datos con los 10 espacios que ocupan los usuarios registrados; ahora que ya recorre este pequeño vector debemos de evaluar, para ello debemos de implementar un “if ” anidado, el cual evaluara si los datos están registrados en el programa o si los usuarios colocaron sus datos de ingreso de manera incorrecta.

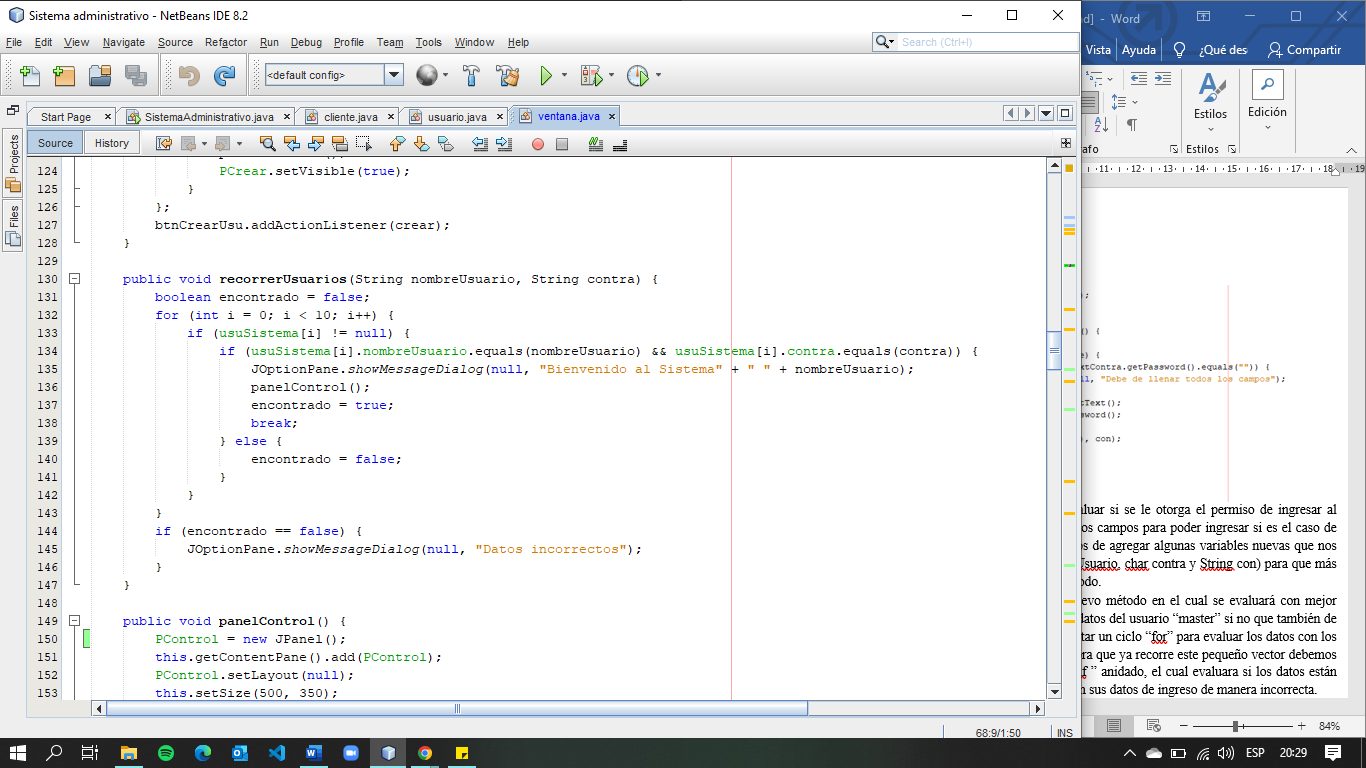
**Figura 11.**

Figura 11: Representa la creación del método, que compara posición por posición los datos que ingresa el usuario con los almacenados, además este método se debe de llamar dentro del evento del botón guardar.