PROGRAMA PARA LA GESTIÓN DE ACTIVIDADES DEL GIMNASIO BODY ART

Segunda Entrega

Presentado y elaborado por:

JEAN CARLO HERRERA DELGADO

SANTIAGO LÓPEZ GALLEGO

JUAN CAMILO MUÑOZ LÓPEZ

JUAN CAMILO HOYOS PEÑA

PRGRAMACIÓN ORIENTADA A OBJETOS

UNIVERSIDAD NACIONAL DE COLOMBIA

SEDE MEDELLÍN

2020

CONTENIDO

[INTRODUCCIÓN 3](#_Toc36156492)

[DESCRIPCIÓN 4](#_Toc36156493)

[CARACTERISTICAS DE LA INTERFAZ 5](#_Toc36156494)

[Ventana inicial 5](#_Toc36156495)

[Ventana principal de cualquier usuario 6](#_Toc36156496)

[IMPLEMENTACION DE LAS INTERFACES DE USUARIO 7](#_Toc36156497)

[INTERFACES IMPLEMENTADAS 8](#_Toc36156498)

[Ventana Inicial 8](#_Toc36156499)

[Ventana específica e implementación de los métodos de cada usuario 14](#_Toc36156500)

[5 FUNCIONALIDADES IMPLEMENTADAS Y MUESTRA DE RESULTADOS 17](#_Toc36156501)

[Funcionalidades del usuario administrador 23](#_Toc36156502)

[MANEJO DE ERRORES 23](#_Toc36156503)

[ARQUITETURA BÁSICA DEL PROYECTO 28](#_Toc36156504)

[CONCLUSIONES 29](#_Toc36156505)

# **INTRODUCCIÓN**

Después de implementar un sistema donde la teoría de la programación orientada a objetos se aplica de manera adecuada y de hacer que un montón de líneas de código interpreten información ingresada por consola, llega la programación orientada a eventos, ya que usualmente solo los ingenieros de sistemas o programadores conocen los archivos, las carpetas y un sinnúmero de vainas más que tiene por detrás una aplicación. La programación orientada a eventos es en ultimas lo que el usuario final percibe, es cada una de las acciones que el usuario puede interpretar, cada escenario que se le muestra, y de eso se trata el front-end, en la primera entrega no hablamos del nombre que le dábamos a lo que estábamos haciendo, en la programación eso se llama back-end.

La programación orientada a eventos es también programación orientada a objetos ya que se usan clases, atributos, métodos y demás propios de la POO, pero ésta va más enfocada a las interfaces gráficas de usuario; con esto, entra un poco más a la imaginación del programador, las herramientas adquiridas con práctica y lo que un cliente solicita a la hora de pedir que le programen una aplicación. Esto de alguna u otra forma hizo que las GUI desde sus inicios revolucionaron la forma de ver los computadores, los sistemas informáticos e hicieron que los sistemas operativos se volvieran de fácil acceso y uso para todos y no solo para los que saben de sistemas.

Como es de costumbre nosotros como estudiantes, o por lo menos los cuatro miembros de este grupo de trabajo estamos muy conscientes de las actuales coyunturas sociales, políticas y tecnológicas que atraviesa el mundo, porque nos interesa ser partícipes activos de la construcción de país y de sociedad desde la academia; es por esta razón que queremos introducir una reflexión sobre la situación del covid-19 y no específicamente sobre las medidas gubernamentales que tomaron las autoridades, ni tampoco sobre las especificidades sanitarias del virus, sino de la necesidad de quedarse en casa y usar herramientas virtuales para trabajar y estudiar.

Pues aquí defendemos una vez más la necesidad de tener ingenieros de sistemas para poder hacer todo desde casa, aunque ya muchas personas, empresas, colectivos y demás usan esto para hacer eficiente la comunicación, se sabe bien que no se usa muy continuamente, que aún se prefiere la manera presencial de hacer las cosas, pero ahí podemos hacer esta pregunta que puede ser fácil de responder y es ¿entonces que pasa en estos momentos de crisis?, ya la respuesta la tienen ustedes, los sistemas informáticos han sido capaces de hacer todo esto, han sido capaces de transformar un mundo completamente solitario para el que lo vive e hicieron que las personas ya no estén solas y puedan trabajar, estudiar, hacer reuniones y un sinnúmero de cosas más. No siendo más este abrebocas los invitamos a pensar en nuestro argumento, a refutarlo, a dar sus opiniones y sobre todo a inventar, crear e innovar; el mundo en este momento lo necesita, necesita cambiar la forma de ver las cosas, de asumir el trabajo, de educarse y de conectarse con los demás sin dejar de ser humanos.

A continuación, pasaremos a explicar brevemente que contendrá este documento, inicialmente tenemos la descripción de la aplicación, las funcionalidades que se encuentran y se hará una breve introducción a ese front-end o gráficas de usuario que implementaremos dentro de la aplicación asociada al gimnasio BodyArt que en un inicio se creó para que solo sea visualizada por medio de códigos de programación y ahora pase a un entorno más ameno, más cordial y mucho más robusto para una persona que no tienen ni idea de cómo funciona por detrás la aplicación, se incluirá un manual de usuario y por su puesto todo lo asociado con las características de la programación orientada a eventos.

# **DESCRIPCIÓN**

El gimnasio BodyArt recibe aproximadamente 150 clientes al día, ofrece varios tipos de productos: rutinas, dietas y suplementos dietarios. Los clientes pueden por medio de esta aplicación consultar algunos datos básicos, reservar en línea y registrarse.

Nuestros empleados tienen control total de la gestión de actividades que se realizan en el gimnasio, tales como control de clientes, eventos e inventario disponible en la tienda de suplementos.

El administrador puede agregar empleados, puede modificar permisos de algunos de ellos haciendo que la aplicación sea completamente focalizada en grupos, por tanto, es completamente adaptable a las necesidades propias de cada sucursal del gimnasio o de cualquier organización.

# **CARACTERISTICAS DE LA INTERFAZ**

Al crear una interfaz gráfica de usuario, no se tienen en cuenta solo los aspectos básicos de lógica, sino que va más allá de la misma programación y es la imaginación. Esto porque las herramientas que permiten crear dichas interfaces, aparte de tener una estructura clara y definida permiten hacer muy versátil la creación, entonces hace que el diseñador pruebe cosas y materialice algunas que no estaban en mente. A continuación, se muestran las características de las dos ventanas principales, ventana de inicio y ventana de cada usuario.

# **Ventana inicial**

A la hora de iniciar el sistema, se mostrará una ventana que permita que el usuario entienda como ingresar, si es un usuario invitado permita registrarse, consultar información básica de la aplicación, además de ver una serie de imágenes del gimnasio y las hojas de vida de las personas que desarrollamos la aplicación. Esta ventana se asemeja y está diseñada basada en las ventanas que introdujo Windows para la fácil interpretación.

BorderPane principal

Gridpane1🡪 Top

0,0

Gridpane3🡪 Right

0,0

Gridpane2🡪 Left

1,0

1,0

**Panel izquierdo**

El saludo inicial da la bienvenida a la aplicación y está en la parte superior del panel izquierdo, este se creó mediante un Label. En la parte inferior encontramos las opciones de ingreso y registro, por medio dos campos de texto donde el usuario registrado ingresa y contraseña.

Label de saludo

Botón ingreso

TextFields ingreso

Botón registro

**Panel derecho** En el panel derecho se encuentra en la parte superior una serie de imágenes que cambia al pasarle el ratón por encima, y en la parte inferior. Una pequeña hoja de vida de cada uno de los integrantes del grupo.

Label de imágenes, con imageView.

Label de hojas de vida.

**Panel superior**

En la parte superior del panel principal, se encuentra anidado un Gridpane, el cual permite organizar el nombre de la aplicación y los botones de descripción y salir.

Botón Salir

v

Label del nombre

Botón Descripción

# **Ventana principal de cualquier usuario**

**Distribución de ventana para cualquier usuario:**

Esta ventana es la que se muestra después de que cualquier usuario registrado de la aplicación ingrese su cedula y contraseña, y se le muestren las opciones que tienen de acuerdo con el rol de usuario. La estructura básica es la que mostraremos a continuación, ya más adelante se mostrará como quedó la ventana, porque es necesario explicar qué tipo de panel y/o estructura las contiene. Además, es importante saber que los paneles no son posiciones fijas, sino anidados lo cual requiere una precisión en términos de saber en qué parte iría cada botón, label, textfield y/o sub-panel.

Parte 0

Panel Secundario🡪Top

Parte 1

BorderPane Principal

Panel Terciario🡪Center

GridPane 🡪TOP en terciario

Parte 2

**Menú superior**

Parte 0: En la parte 0 está un Label el cual muestra el nombre del usuario, el cual ya ingresó sus datos anteriormente.

Label de nombre de usuario

Parte 1: En esta parte hay tres botones, Archivo, Procesos y Consultas y Ayuda.

Botón ayuda

Botón procesos y consultas

Botón archivo

**Zona de interacción con el usuario**

Y en la zona 2 de interacción con el usuario es donde mostramos los datos solicitados para ejecutar algún método y los resultados de estos.

En esta zona se implementa el formulario genérico solicitado, el cual sea adaptable para cada tipo de usuario y/o método. Se hace por medio de la clase Field Panel

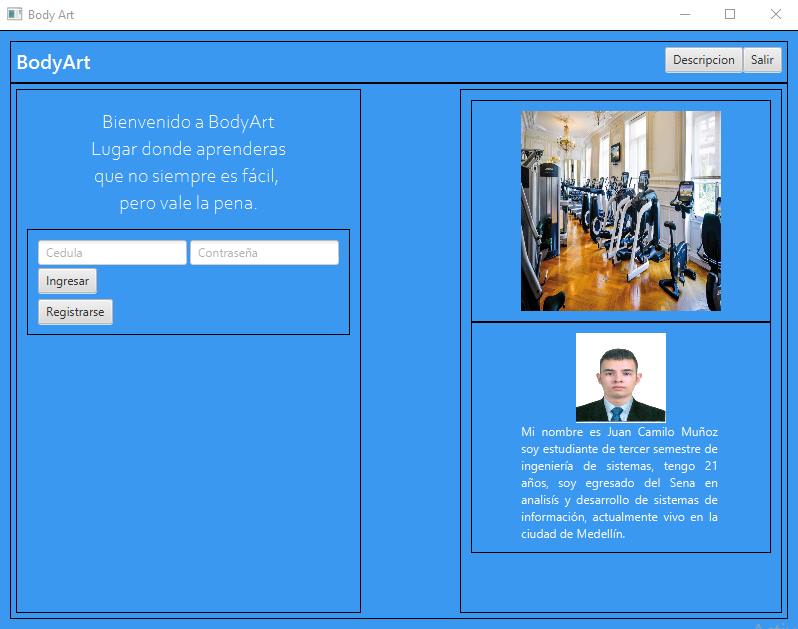
# **IMPLEMENTACION DE LAS INTERFACES DE USUARIO**

# **INTERFACES IMPLEMENTADAS**

En esta etapa del proyecto se muestran las gráficas ya implementadas, y las opciones que existen al dar clic sobre cada uno de los botones y/o eventos que se encuentran programados. Es muy importante, esta etapa ya que es crucial para el entendimiento del funcionamiento de este, a continuación, mostraremos los resultados que muestra la consola al ejecutar el proyecto.

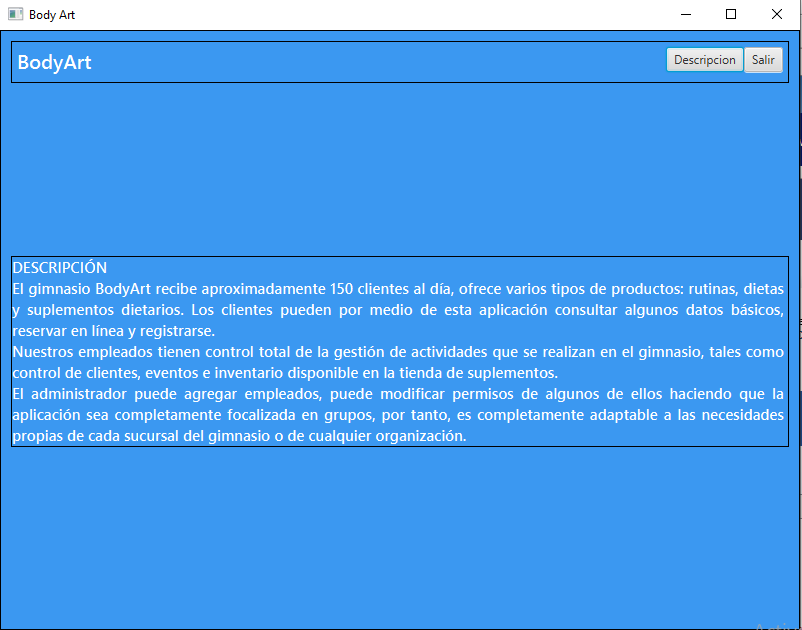
# **Ventana Inicial**

En una etapa anterior del proyecto se muestra la estructura de paneles anidados que tienen cada una de las ventanas, en este caso como podemos observar, ya se encuentran los botones programados, las imágenes agregadas y los eventos listos para ser mostrados. Usualmente todas las aplicaciones hoy en día tienen la posibilidad de contar con estas herramientas, las cuales gráficamente se le muestra al usuario y este sin necesidad de conocer sobre el sistema puede interactuar con él. Esta ventana inicial, es la que se muestra al ejecutar el proyecto y se describirá a continuación.



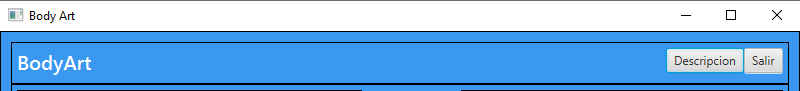
**Botón Descripción**

En este al darle clic, muestra una descripción del proyecto, lo que se puede hacer dentro del mismo y algunas características básicas.



**Botón Salir**

El botón salir, que se encuentra en la ventana inicial interrumpe la ejecución y te saca de las ventanas, en este caso es necesario volver a ejecutar el programa. (Se hizo acorde a los requerimientos del profesor).



**Mensaje de Bienvenida, Ingreso y Registro**

En esta sección se muestra el mensaje de bienvenida, y los campos de texto para poder hacer el ingreso, ya que se requiere que cuando un usuario quiera usar la aplicación deba registrarse o ingresar. Ahora se muestra la forma en que esto se hará, ejemplificándolo con un usuario nuevo y que ese usuario ingrese y muestre la segunda ventana importante de la aplicación.

**5 imágenes**

En la ventana inicial, también se muestran 5 imágenes que cambian al pasar el cursor del mouse por encima de las mismas, esto es una de las funcionalidades de ImageView ya que permite administrar las imágenes y gráficos dentro de una GUI, dentro de botones, etc.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

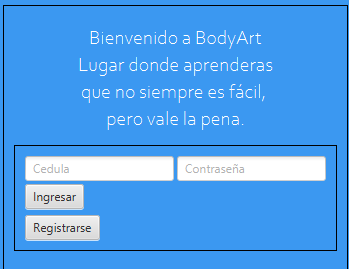
**Hojas de vida**

En la parte inferior se encuentran las hojas de vida de los integrantes del grupo, estas cambian al darle clic con el cursos del ratón. Lo ideal de todas estas pruebas es poder garantizar que nosotros como estudiantes aprendimos que pueden existir muchos eventos dentro de una GUI y no únicamente llenar espacios o dar clic en botones.

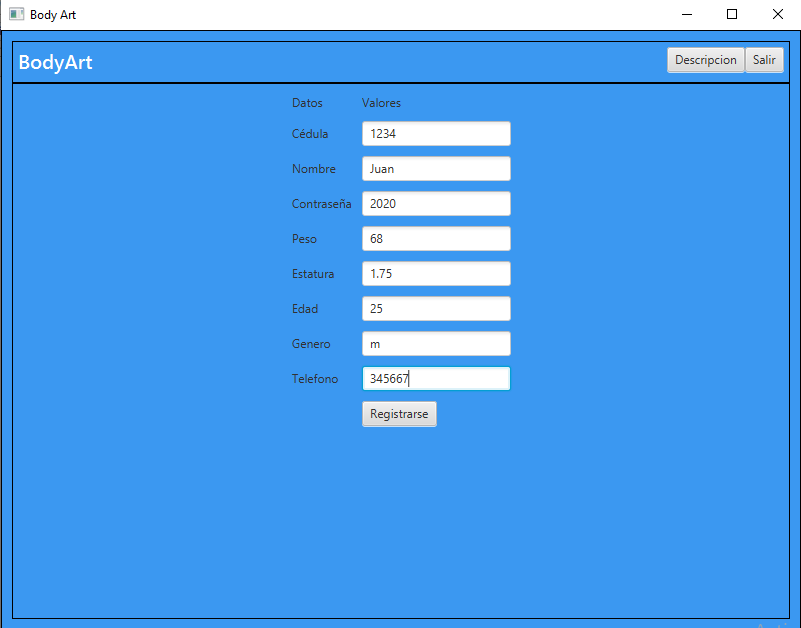
|  |  |
| --- | --- |
|  |  |
|  |  |

**Para Registrarse:**

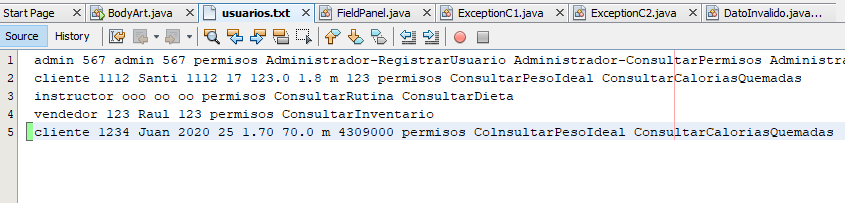
Para hacer uso de la opción de registrarse, y haciendo semejante la forma en como un usuario invitado se registraba en la primera entrega, simplemente se da clic en el botón registrarse. Lo haremos con el usuario **Juan**, con cédula **1234** y contraseña **2020 y** otros datos, que son de interés para la aplicación.



Al ingresar los datos, de nuevo se da clic en registrarse y ya quedan guardados todos los datos del nuevo cliente, recuerde que en la primera entrega los únicos que pueden registrase en este caso son los usuarios de tipo cliente.

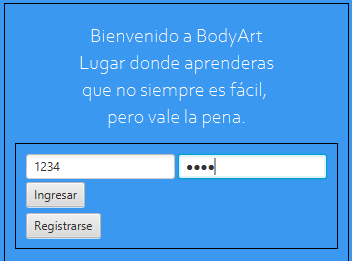


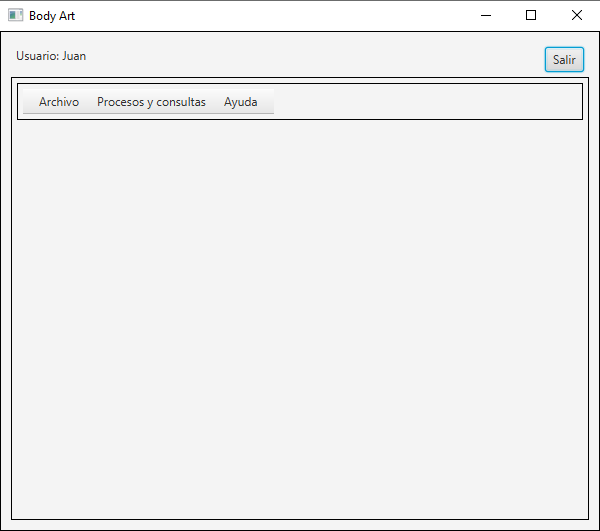
Para verificar el registro se puede ir al botón de ingreso, o en el archivo de texto buscar el nuevo usuario registrado.



**Ingreso para usuario registrado**

Después de estar registrado en la plataforma, se ingresan los valores de cédula **1234** y contraseña **2020** y se da clic en Ingresar:





En la parte superior de la interfaz, le muestra el nombre del usuario actual.

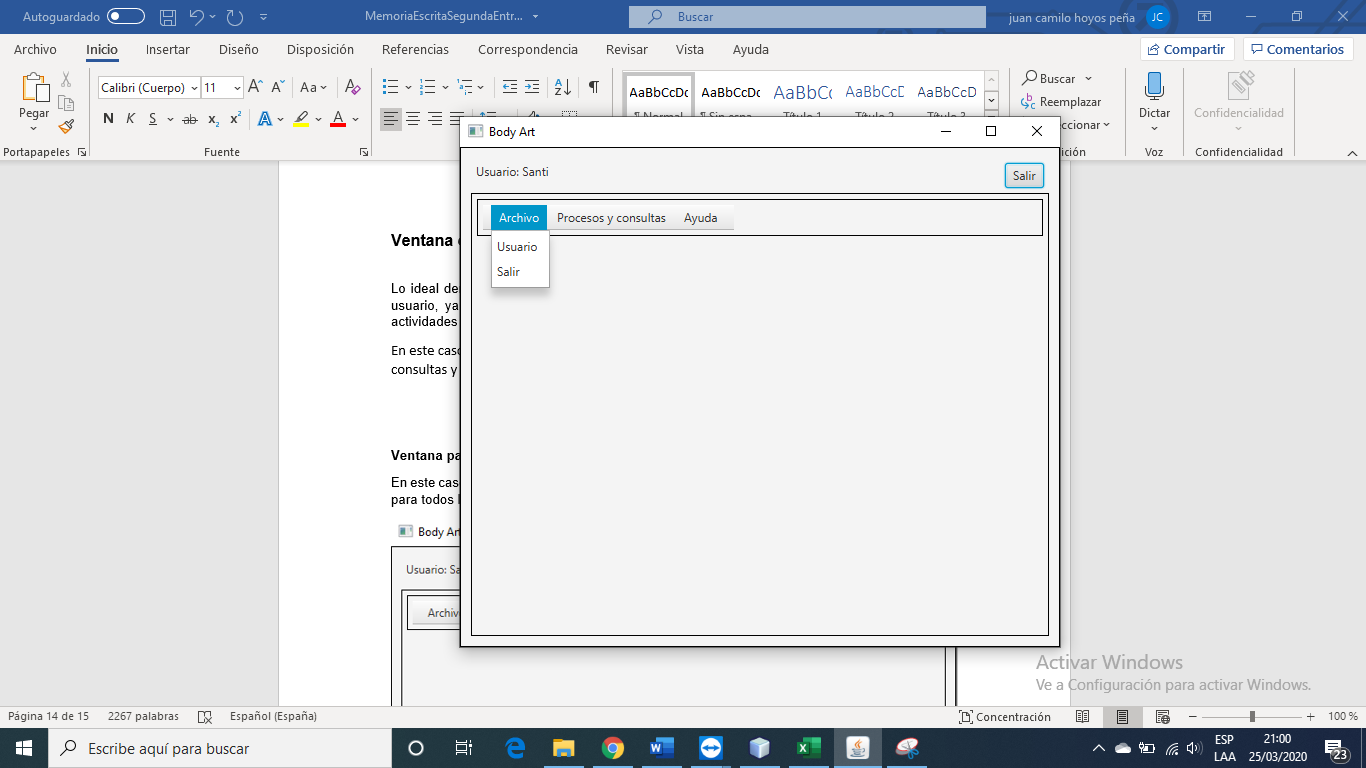
Con la anterior imagen introducimos la segunda ventana importante del sistema, en este caso la que se le muestra a cada usuario después de haber ingresado.

## **Ventana específica e implementación de los métodos de cada usuario**

Lo ideal dentro de una aplicación es que se adapte de acuerdo con las necesidades de cada usuario, ya que dentro del sistema existen varios tipos de usuarios, con roles diferentes y actividades diferentes.

En este caso para todos los usuarios se observan 3 botones, el botón de Archivo, Procesos y consultas y Ayuda. Cada uno muestra lo siguiente:

**Botón Archivo**

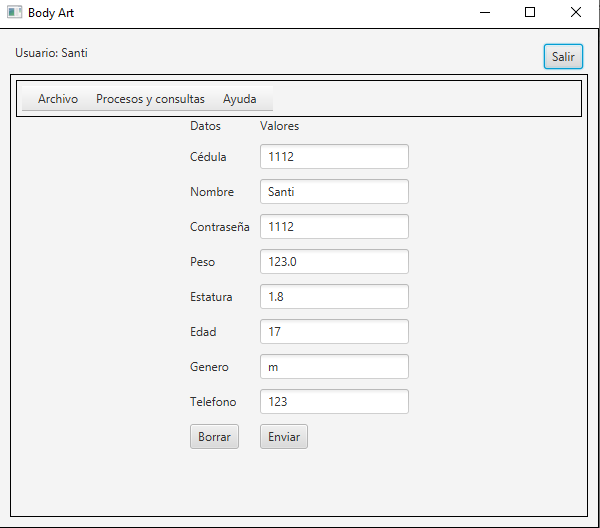


Al dar clic en archivo, se despliega un menú con dos opciones: opción Usuario y Salir.

Al dar clic en el botón **usuario** se despliega la información del usuario actual, ósea los datos que registró al inicio.

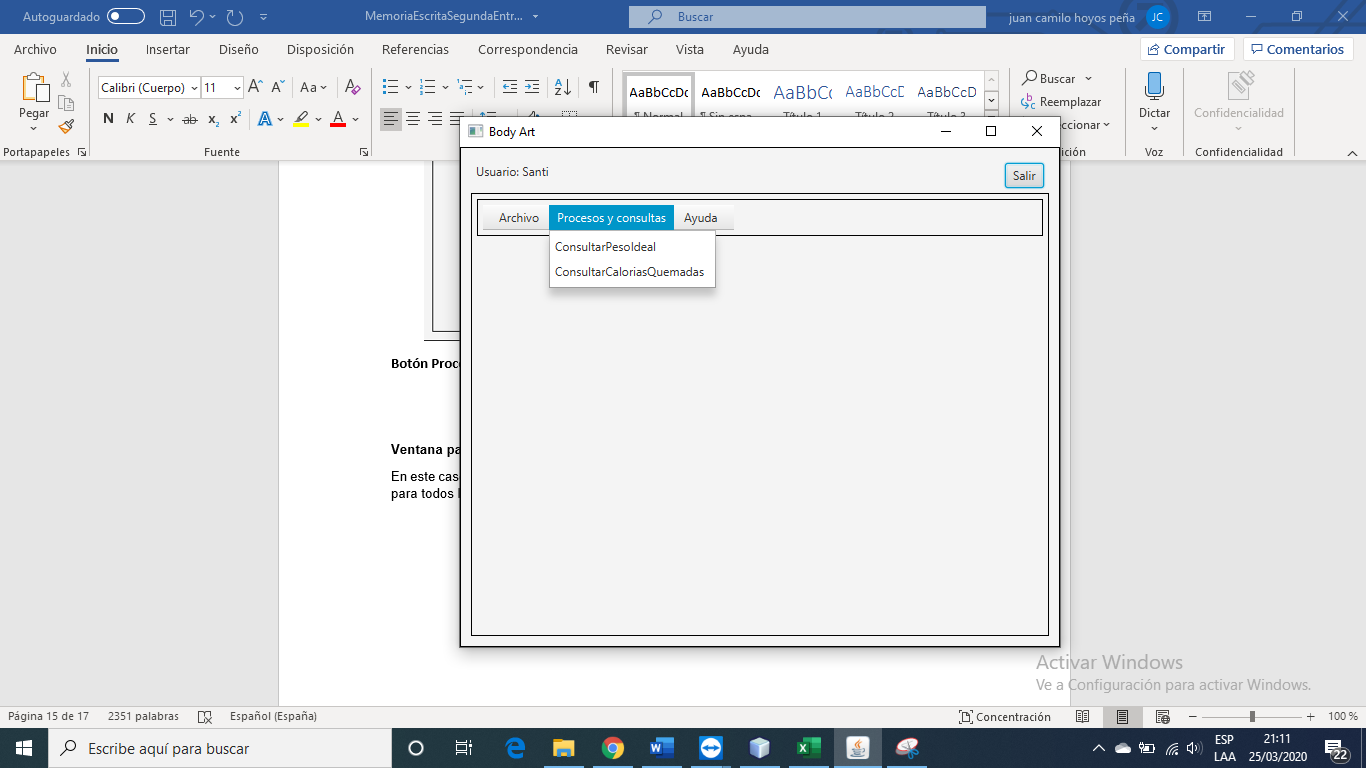
Por tanto, en esta parte es donde se pueden modificar algunos datos, tales como el nombre, teléfono, contraseña, etc. Pero tampoco permite cambiar la cédula ya que eso sería crear otro objeto.

El botón **salir**, devuelve a la ventana inmediatamente anterior.



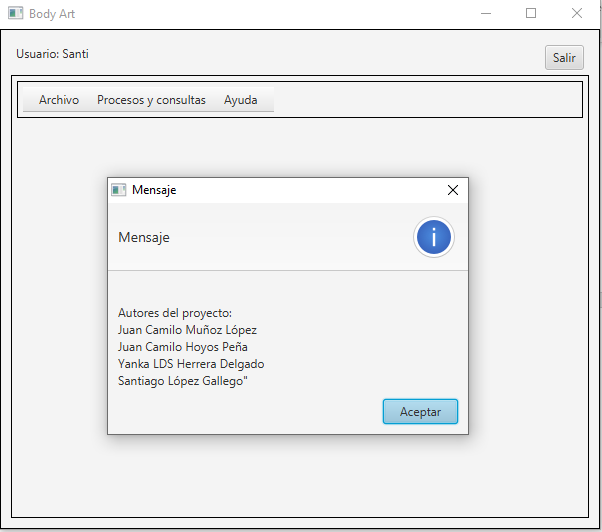
**Botón Procesos y consultas**

Este botón despliega lo métodos que contiene cada usuario, en este caso y para esta práctica escogimos 5 de las 10 anteriormente programadas, ya que esas eran las que se solicitaban desde la primer entrega.



**Botón Ayuda**

Este botón muestra los nombres de los integrantes del grupo, que desarrollamos el proyecto, en un cuadro de texto y/o ventana nueva.



# **5 FUNCIONALIDADES IMPLEMENTADAS Y MUESTRA DE RESULTADOS**

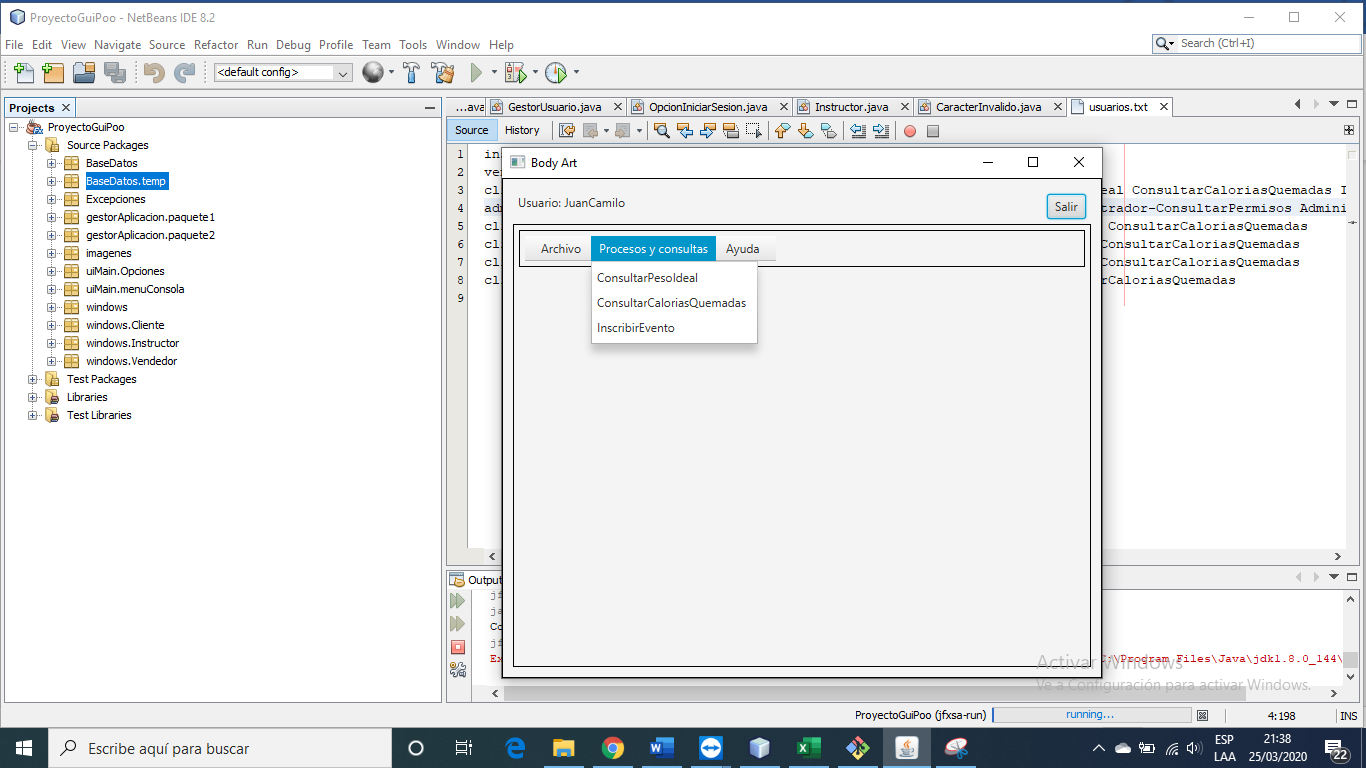
En la entrega pasada, se mostraron algunas funcionalidades propias de cada usuario. Por tanto, mostraremos las 5 funcionalidades que implementamos a cada uno de los usuarios, esto teniendo en cuenta que los tipos de usuarios son completamente distintos y poseen características particulares.

Estas funcionalidades aparte de las básicas, consideramos que son importantes porque permiten relacionar todas las clases y los usuarios presentados anteriormente, además van acorde con el dominio del problema planteado anteriormente introduciendo conceptos que se manejarían en el dominio real. Estas funcionalidades se pueden hacer de dos formas, ingresando por consola los datos o desde el contexto de un objeto

**Usuario Cliente**

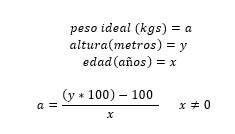
Para ingresar como usuario tipo cliente ingresamos desde la ventana de inicio con la cedula **112233** y la contraseña **123** que es un usuario ya registrado anteriormente.

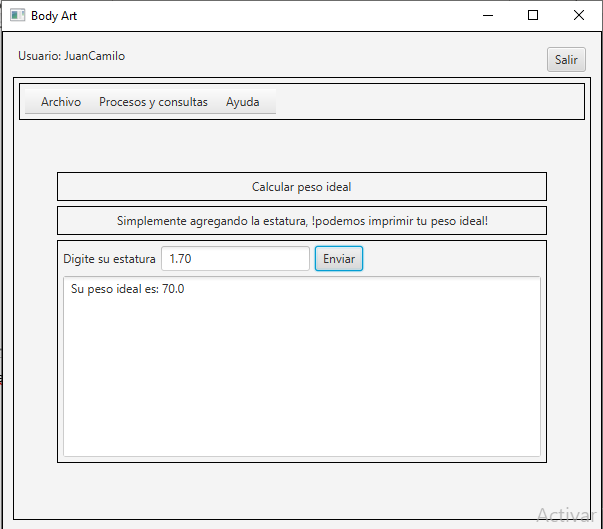
Los usuarios tipo cliente poseen dos funcionalidades:



**ConsultarPesoIdeal**

Este método le solicita, que ingrese por teclado la estatura separada por puntos, de esto en la parte inferior de la pantalla le muestra el resultado, en este caso el peso ideal que debería tener ese cliente.

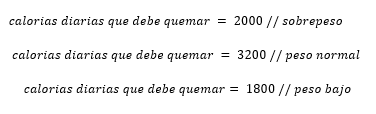


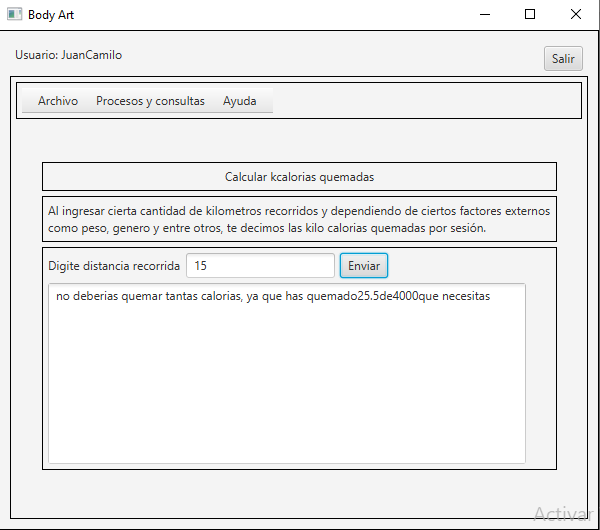


**Consultar Calorías Quemadas**

Esta funcionalidad calcula las calorías quemadas, de acuerdo con las que debe quemar el usuario.

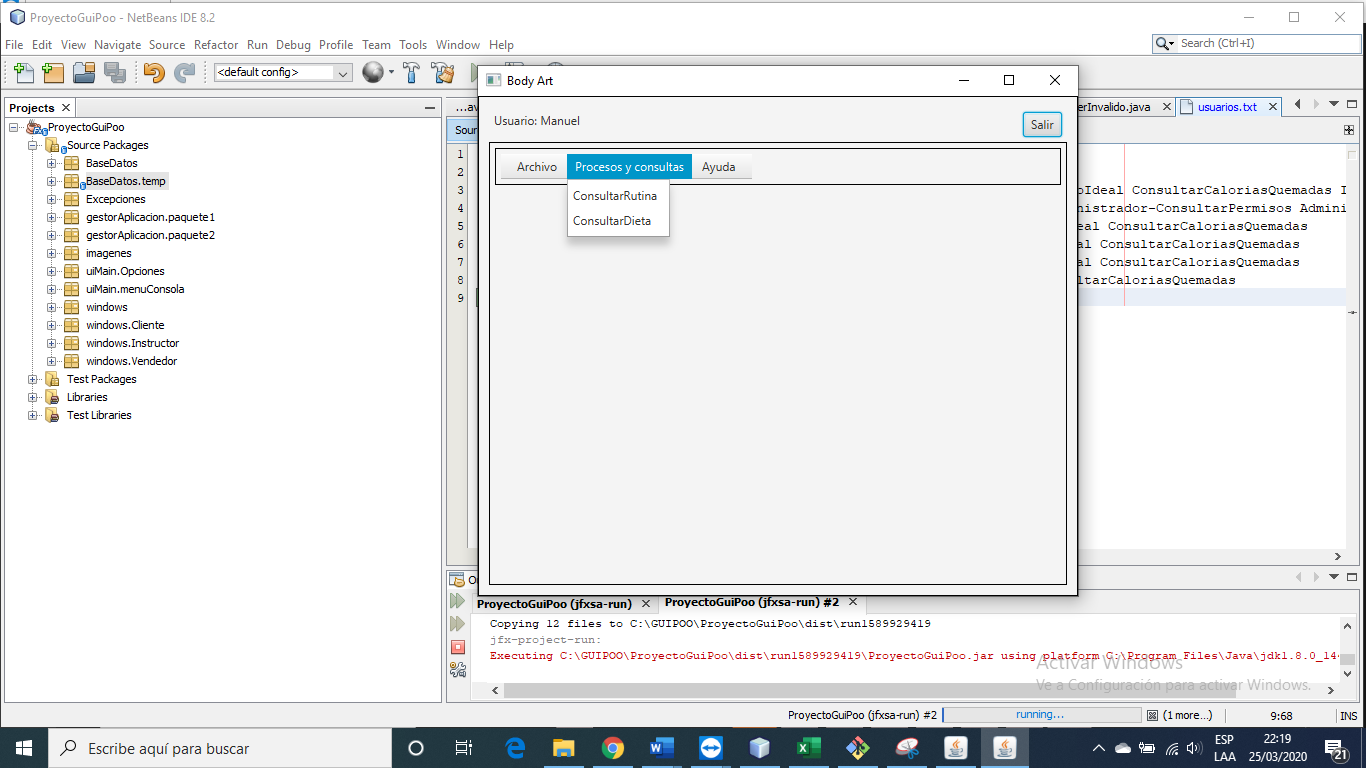
Entonces de acuerdo con el peso que el usuario haya ingresado, o que haya actualizado y la cantidad de kilómetros que recorrió trotando, entonces le va a decir, cuantas ha quemado y cuantas debería quemar.





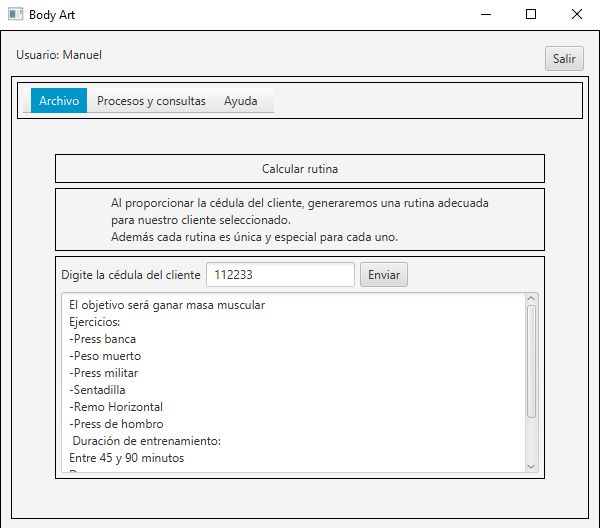
**Usuario Instructor**

Para ingresar como usuario instructor, usamos cedula **2121** y contraseña **4524** esto para Manuel.



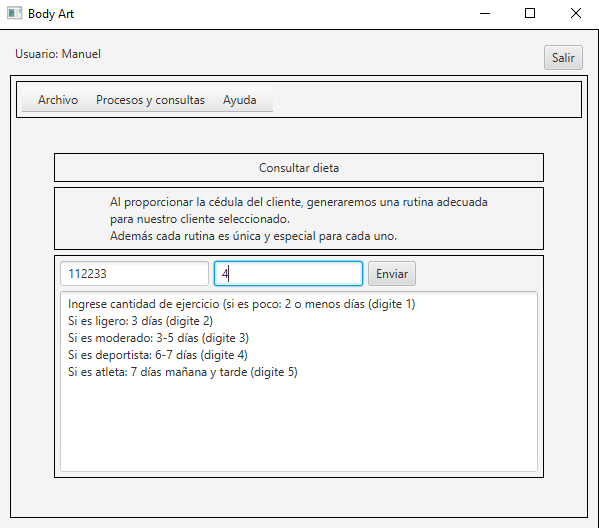
**Consultar Rutina**

El método consultar rutina, propio de instructor. Le solicita por teclado la cedula de un cliente, en este caso le ingresamos la cédula de JuanCamilo **112233,** como resultado le muestra que tipo de rutina debe hacer.

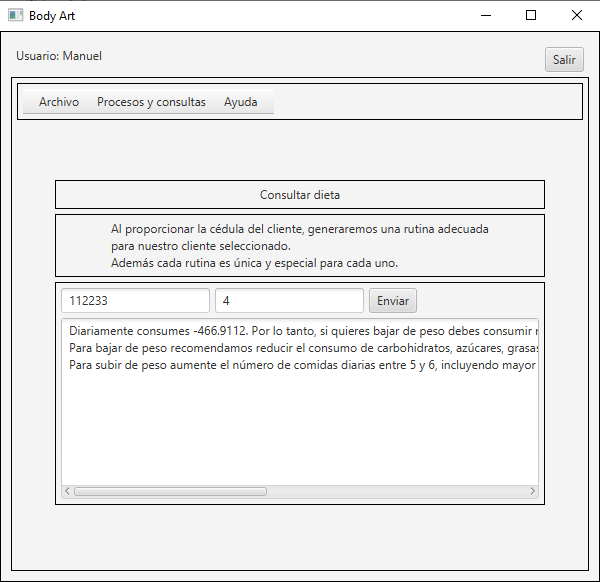


**Consultar dieta**

La funcionalidad de consultar dieta, solicita que ingrese una cédula de un cliente y la cantidad de ejercicio, ya que cada persona es consciente de la cantidad de ejercicio que hará, esto en términos de días.

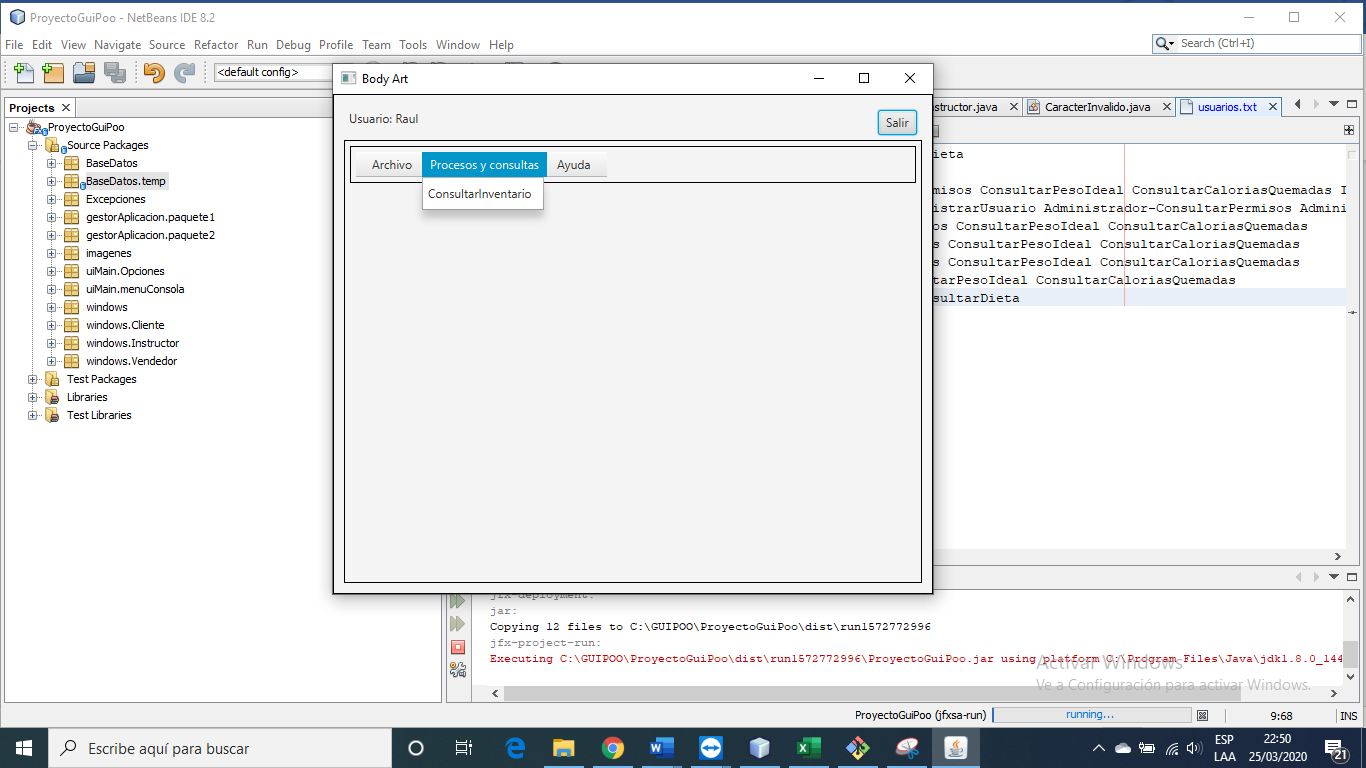


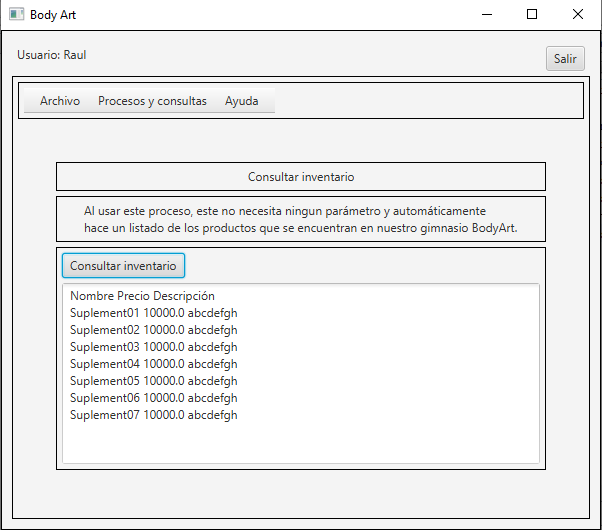
Y el resultado es el siguiente:



**Funcionalidades del usuario vendedor**

Para ingresar como usuario vendedor usamos la cédula **678** y contraseña **000,** el nombre de este usuario es Raul. Los vendedores pueden consultar el inventario disponible.





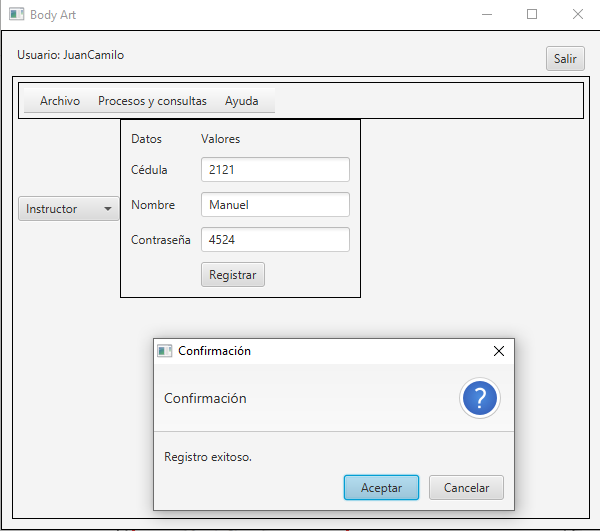
# **Funcionalidades del usuario administrador**

En la primera entrega se tuvieron en cuenta varias funciones que hacía un usuario o super usuario, este se denomina administrador porque posee funciones elevadas y/o de mas alto nivel que los demás tipos de usuarios. En toda compañía los administradores de este tipo de sistemas son personas capacitadas con accesos privilegiados y de mucha responsabilidad ya que recae en sus manos lo concerniente con la página, aplicación, software entre otros.

En nuestro caso el usuario administrador es el encargado de agregar los usuarios de tipo administrador, instructores y vendedores. Pero también otorgarles permisos a dichos usuarios.

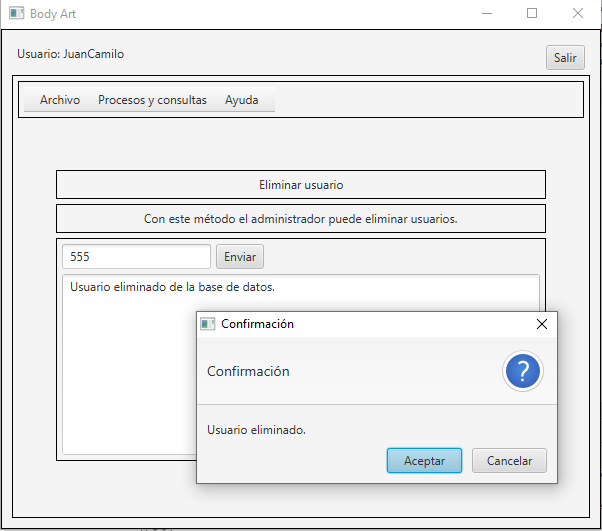
**Opción registrar usuario**

Esta opción permite registrar a cualquier tipo de usuario, como se ha reconocido anteriormente esta es una de las funcionalidades de dicho usuario, puesto que es importante que a los empleados se les asigne un usuario y una contraseña. En este caso el administrador con cédula **567** y contraseña **567** agrega al usuario Raul.



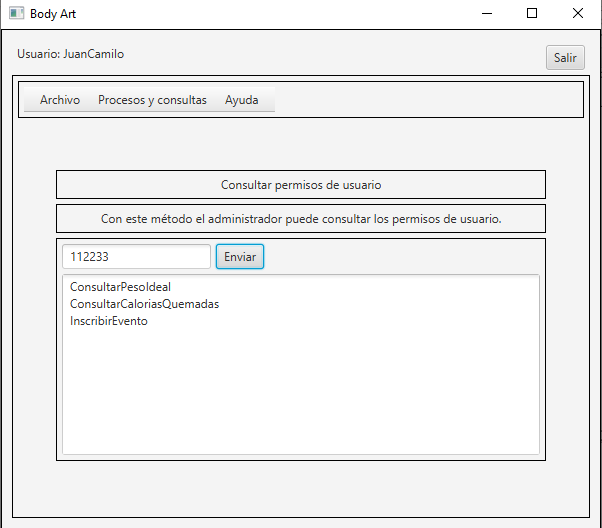
**Opción eliminar usuario**

Como el usuario administrador puede registrar usuarios, de igual forma debería permitírsele borrar usuarios del sistema, esto se aplica generalmente cuando un empleado abandona su puesto de trabajo, cuando un cliente se inactiva por dicho tiempo, todo de acuerdo con las políticas de la organización.



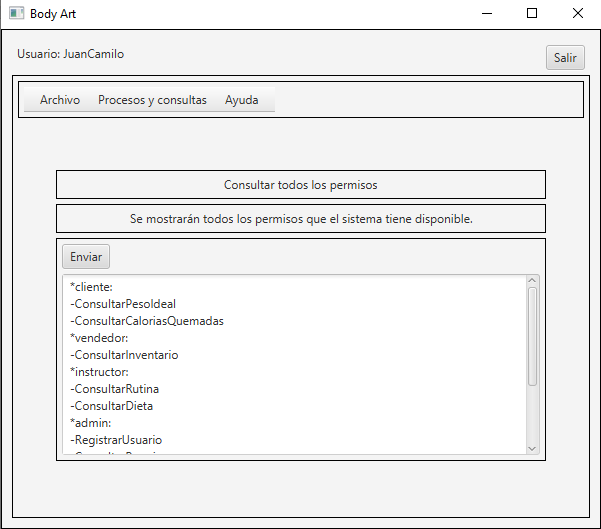
**Opción consultar permiso**

Útil para saber que tipo de permisos tiene dicho usuario, esto se hace por medio de la cedula del usuario. En este caso el usuario con cedula 112233 tiene los permisos listados.



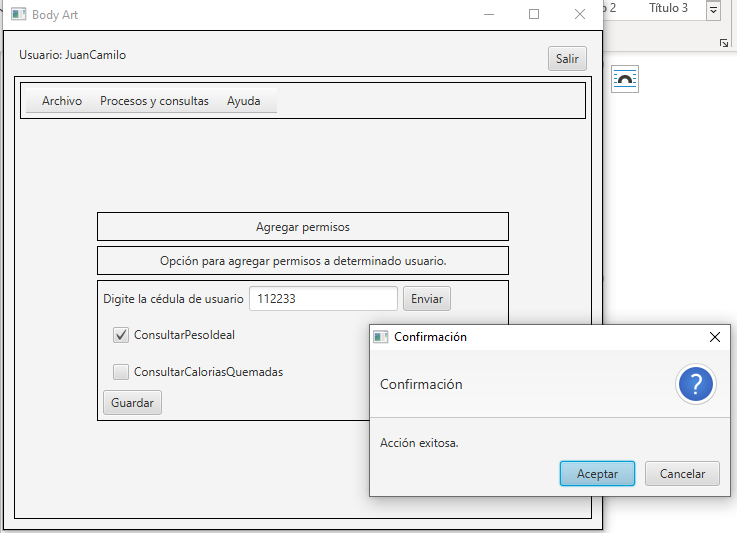
**Opción consultar todos los permisos**

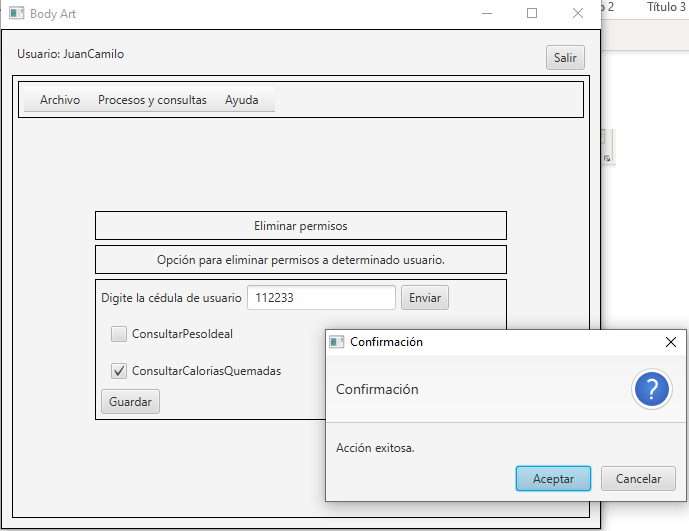
Consultar todos los permisos, es para saber que tipo de permisos puede o no agregar a un tipo de usuario, como administrador no solo es uno, algunos administradores tienen la opción de agregar uno o varios permisos, esto flexibiliza el trabajo del administrador principal entregándole cierto rango a otro administrador, pero no un control total.



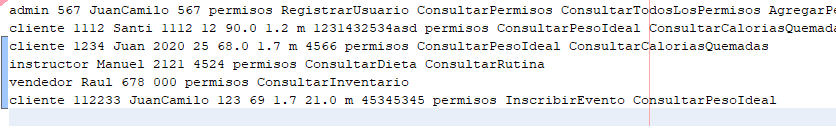
**Opción agregar y eliminar permiso**

Como el administrador tiene control total del sistema, puede agregar o denegar permisos a los usuarios, en este caso serviría esta opción por si un cliente bajó de categoría o un empleado solo se necesita para un área específica.





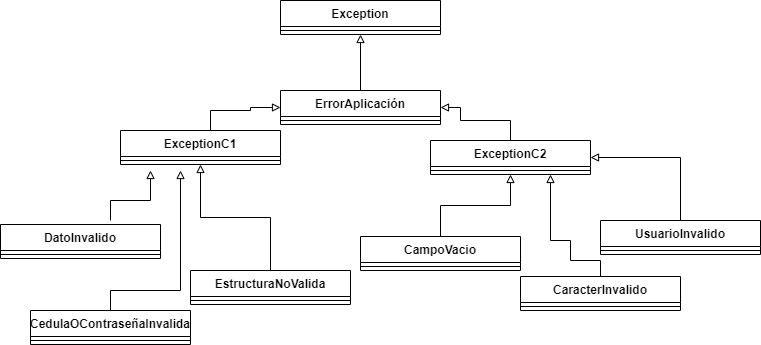
Aquí se evidencia los permisos agregados y denegados



# **MANEJO DE ERRORES**

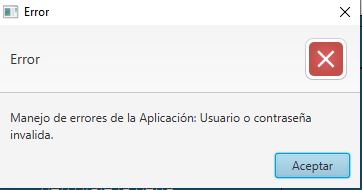
¿Le ha pasado que cuando ingresa en un campo algún dato, este le muestra una advertencia diciéndole que no puede ingresar ese dato, en dicho campo? El manejo de errores se encuentra asociado con ese tipo de advertencias y en la programación se llaman excepciones, puesto que es necesario hacer validaciones para que cuando un usuario ingrese un dato erróneo, la aplicación, el sistema o lo que sea no se dañe y escale un montón de errores. Para nuestra aplicación, lo que queremos mostrar es la ejecución e implementación de algunas excepciones que manejan esos errores, mostrándole por pantalla el tipo de error que se pudo haber encontrado y haciéndole un debido manejo. Las excepciones no dejan de ser programación orientada a objetos, ya que se definen mediante clases, se crean instancias de estos y contienen la misma estructura de una clase.

A continuación, mostraremos la arquitectura y el orden jerárquico de estas, ya que es necesario utilizar el concepto de herencia, etc.



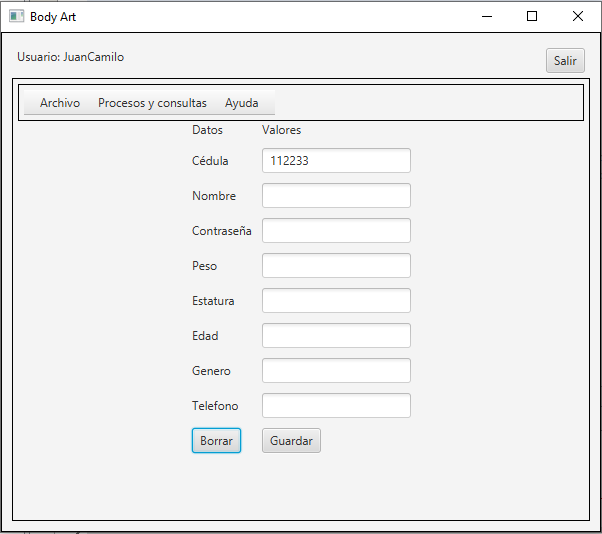
**Error Aplicación**

La clase ErrorAplicación es subclase de la gran clase Exception, esta aporta la primer parte del mensaje de todas la excepciones que se crearon en el proyecto. En este caso aporta el mensaje “Manejo de errores de la Aplicación”.



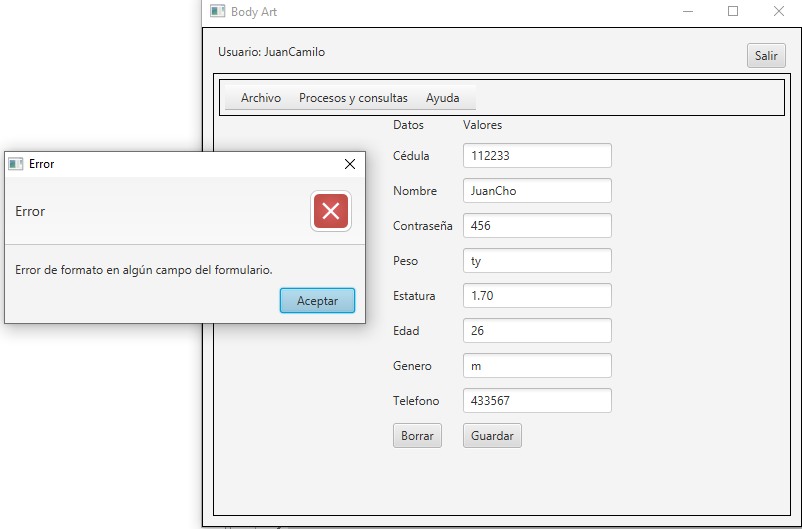
**Dato Invalido**

A la hora de modificar algunos datos del archivo de un usuario, con el botón de borrar se muestra lo siguiente:



Como podemos observar con el botón borrar, se pueden cambiar casi todos los datos de dicho usuario. **Excepto la cédula.**

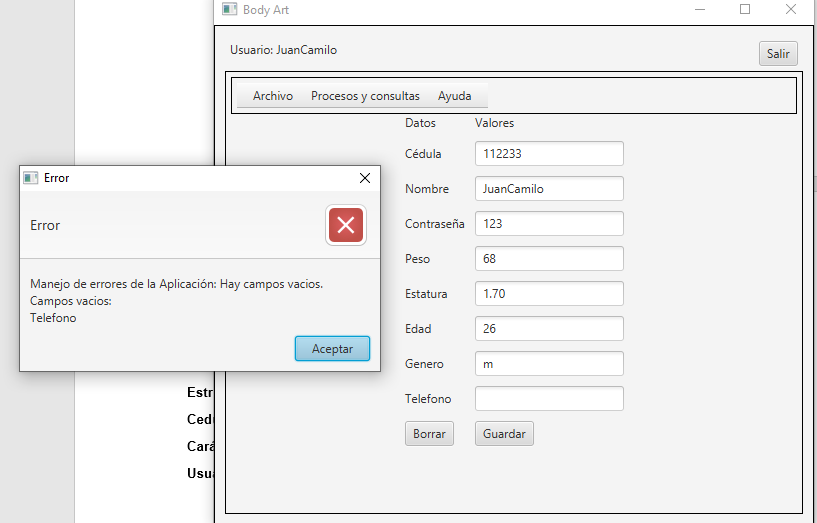
Y al ingresar un datos distinto al preestablecido, muestra la excepción y permite seguir la ejecución



En este caso, tratamos de cambiar algunos datos y le ingresamos datos que no corresponden, por ejemplo, en peso le ingresamos las letras ty y como resultado muestra la ventana de advertencia, esta la saca la excepción DatoInvalido

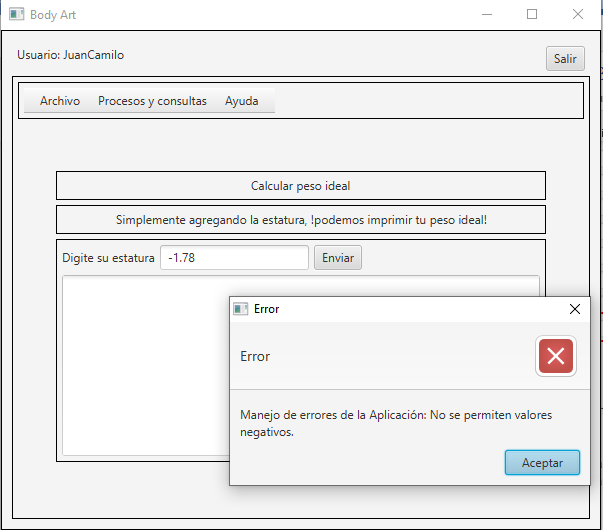
**Campo Vacío**

Esta clase o excepción maneja las excepciones cuando a la hora de guardar, existe un campo vacío.



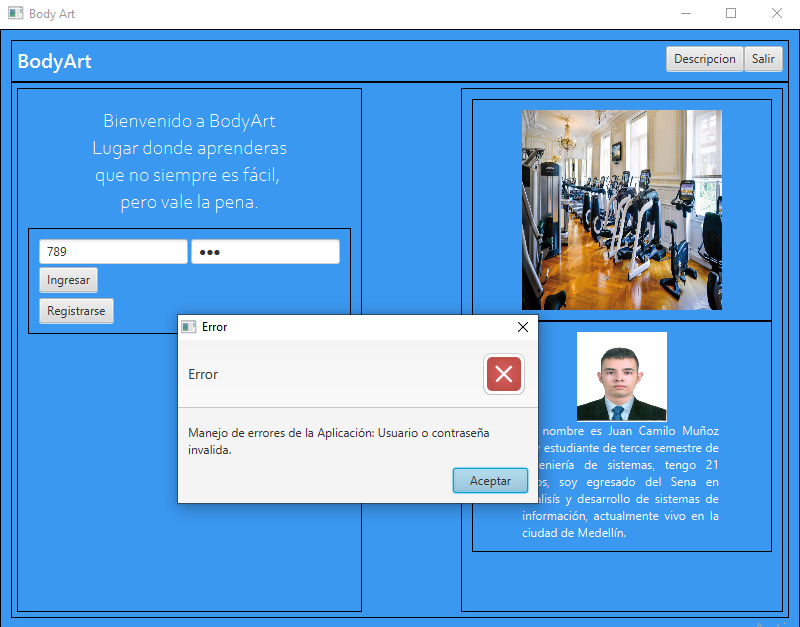
**Estructura No Valida**

La excepción de EstructuraNoValida, permite identificar algunos valores introducidos que no tienen lógica o que difieren de la estructura propuesta en el programa.



**Cedula O Contraseña Invalida**

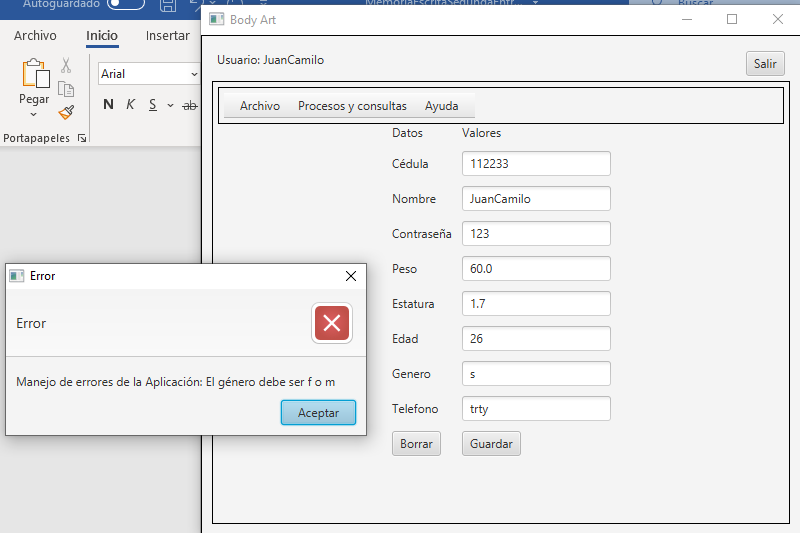
Esta excepción es muy usual, a la hora de un usuario tratar de ingresar a un sistema o aplicación, ya que puede advertir que la cédula no está registrada o si se escribió mal la contraseña.



Como puede observar, estamos tratando de ingresar con un usuario no registrado y el sistema muestra la advertencia.

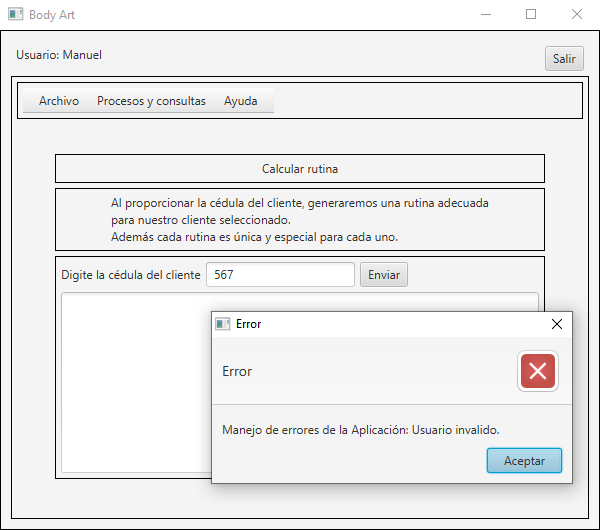
**Carácter Invalido**

Esta excepción se dispara cuando al tratar de ingresar dentro del campo genero un carácter diferente a m o f, que son masculino y femenino respectivamente.



**Usuario Inválido**

Cuando un instructor quiere calcularle una dieta o rutina a un cliente, este debe introducir una cédula de tipo cliente, en caso contrario le genera una advertencia. Este tipo de excepciones es muy común a la hora de ingresar datos ajenos, ya que no sabemos en que bases, permisos y/o condiciones lógicas tienen dichos datos.



En este caso, se está intentando ingresar una cédula de un administrador, el sistema al reconocer que esa no corresponde a un cliente, de una le genera la excepción.

# **ARQUITETURA BÁSICA DEL PROYECTO**

**Paquetes y clases**

Como se puede observar en la siguiente tabla, se crearon nuevos paquetes que no estaban en la primera entrega, estos son todos los de **Windows**, **excepciones** e **imágenes**. Donde se hace la gestión de las GUI, de las imágenes y de la jerarquía de imágenes anteriormente descritas. La clase principal es BodyArt, que es donde se encuentra toda la implementación de los paneles y de algunas clases necesarias para el funcionamiento del programa.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# **CONCLUSIONES**

A manera de conclusión podemos decir varias cosas, la primera es que la programación orientada a objetos y este paradigma es tan importante para la programación y se encarga de darle otra visión al programador, se encarga de hacerle entender que existe un sin número de herramientas para crear un programa desde cero. Java al ser un software de programación fuertemente tipado, permite y exige tener control total de cada uno de los elementos que se presentan en el código, esto es útil para conocer a detalle los tipos de datos cosa que no en algunos software de más alto nivel.

Las interfaces gráficas son de gran utilidad para hacer que un montón de líneas se vean completamente útiles y en definitiva le muestre al usuario final un entorno agradable, robusto y fácil de manejar. No obstante, simplifica de manera abrupta el trabajo que hay detrás de dicha gráfica, pero ese es nuestro trabajo, hacerles la vida más fácil a los usuarios finales.

Los objetos de ahora en adelante los veremos de otra forma, ya que incluso podemos representarlos por medio de un sistema informático y solo nosotros entenderemos ese concepto tan abstracto, que a principios del curso parecía super complejo de entender.

Los ingenieros de sistemas no solo servimos para estar dos horas en una empresa porque se dañó un computador, también hemos revolucionado el mundo con nuestra intrépida personalidad y distinta forma de ver el mundo, no estamos diciendo con esto que somos excepcionales, simplemente estamos valorando nuestro trabajo que, a raíz de un meme, puede ser grotesco e inescrupuloso.

Como estudiantes de ingeniería somos conscientes de los grandes restos que impone la industria, pero también poseemos habilidades críticas que permiten la construcción de país desde la academia, desde la ciencia y desde cada acto social en comunidad, en el salón de clase y en nuestras casas.

De esto también podemos decir que el trabajo en grupo y la disciplina es posible hallarla siempre y cuando, se tengan ganas de aprender puesto que el facilismo de muchas personas impide que saquemos el potencial que tenemos cada uno de nosotros.

Una materia en la cual se exige mucho permite una aprendizaje óptimo de los estudiantes, entonces genera que estos mismos se esfuercen y den todo de sí para hacer un trabajo excelente.