



**UNIVERSIDAD DE SANTIAGO DE  
CHILE FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE INGENIERÍA  
INFORMÁTICA**

**INFORME LABORATORIO 4**  
**Paradigma Orientado a Objetos y Paradigmas Orientado a Eventos**  
**Paradigmas de programación**

Dyllan Salgado

Profesor: Víctor Flores.  
Fecha de entrega: 04-03-2021

Santiago - Chile  
2021

## Tabla de contenidos

1.INTRODUCCIÓN .....	3
2. DESCRIPCIÓN BREVE DEL PROBLEMA .....	3
3. DESCRIPCIÓN BREVE DEL PARADIGMA .....	4
4. ANÁLISIS DEL PROBLEMA RESPECTO DE LOS REQUISITOS ESPECÍFICOS QUE DEBEN CUBRIR .....	5
5. DISEÑO DE LA SOLUCIÓN .....	7
6. ASPECTOS DE IMPLEMENTACIÓN .....	10
7. INSTRUCCIONES DE USO .....	10
8. RESULTADOS Y AUTOEVALUACIÓN .....	13
9. CONCLUSIÓN .....	15
10. REFERENCIAS .....	16
Anexos .....	17
Programa funcionando: .....	17

## Tabla de Figuras

Ilustración 1: Diagrama de análisis .....	7
Ilustración 2: Diagrama de diseño .....	8
Ilustración 3: Abrir carpeta del proyecto .....	10
Ilustración 4: Ingresando a carpeta SRC .....	11
Ilustración 5: Ingresando a carpeta main .....	11
Ilustración 6: Ingresando a carpeta java .....	11
Ilustración 7: Entrando en consola .....	11
Ilustración 8: Comando javac Main.java .....	11
Ilustración 9: Comando javac Main.java -Xlint .....	12
Ilustración 10: java Main .....	12
Ilustración 11: Interfaz para trabajar .....	12
Anexo1. 1 Mostrando menú inicial .....	17
Anexo1. 2 Registrando a un usuario .....	17
Anexo1. 3 Usuario ya logeado, se muestra menú logeado .....	18
Anexo1. 4 Ingresando a realizar pregunta .....	18
Anexo1. 5 Realizando la pregunta .....	19
Anexo1. 6 Logeando nuevo usuario .....	19
Anexo1. 7 Ingresando pregunta para responder .....	19
Anexo1. 8 Mostrando el menú para realizar respuesta .....	20
Anexo1. 9 Respuesta Realizada .....	20
Anexo1. 10 Logeando con nuevo usuario .....	20
Anexo1. 11 Se realiza una recompensa .....	21
Anexo1. 12 Se añade la recompensa .....	21
Anexo1. 13 Se vuelve a iniciar con usuario que realizo pregunta .....	22
Anexo1. 14 Seleccionar la pregunta del usuario que ha creado la pregunta .....	22
Anexo1. 15 Seleccionamos la pregunta tutorial y la respuesta de Dyllan .....	22
Anexo1. 16 Revisando si se han asignado las recompensas .....	23
Anexo1. 17 Revisando si se han asignado las recompensas .....	23
Anexo1. 18 Revisando si se han asignado las recompensas .....	23
Anexo1. 19 Creando una etiqueta en el stack .....	24
Anexo1. 20 Agregando los datos de la etiqueta .....	24
Anexo1. 21 Visualización de la etiqueta en la pregunta .....	24

## 1.INTRODUCCIÓN

A través de la historia se han presentado una gran cantidad de problemáticas, la cual gracias a la ciencia computacional se han podido resolver a través de distintas abstracciones, generando diferentes soluciones para un mismo problema. A estas soluciones las denominaremos algoritmos, ya que son un método preciso para poder resolver un determinado problema, estos algoritmos serán creados en distintos lenguajes de programación con el paso del tiempo. En el informe se presenta una de las posibles soluciones a la problemática que se nos ha entregado, el cual será resuelto en el paradigma orientado a objetos y en el lenguaje de programación Java. La problemática consiste en la simulación de Stack overflow, pero ¿Qué es Stack Overflow? [1]. Stack overflow es un foro que construye una base de conocimientos sobre programación el cual permite realizar preguntas y respuestas sobre problemáticas que se tengan al momento de programar o en lenguajes de programación en específico. Este foro está dirigido para programadores aficionados y profesionales. Fue creado en el año 2008 por Jeff Atwood y Joel Spolsky.

El objetivo principal del laboratorio consiste en simular el foro stack overflow con requisitos que nos plantea la problemática en el paradigma orientado a objetos, además se debe utilizar el paradigma orientado a eventos para realizar una interfaz gráfica. La interfaz gráfica permite al usuario trabajar con el programa de forma didáctica e intuitiva, permitiendo un trabajo optimo y sencillo.

Para lograr el objetivo principal debemos tener en consideración los objetivos específicos, los cuales son:

- Leer y comprender el enunciado.
- Creación de un TDA correcto que será utilizado para realizar el algoritmo.
- Realizar clases bien diseñadas con sus atributos y métodos.
- Realizar relaciones y clases coherentes en el diagrama de UML.
- Explicar entradas y salidas de las funciones.
- Ordenar y organizar el código para que sea más sencillo de comprender.
- Crear ventanas de interfaz gráfica didácticas y fáciles de entender para el usuario.
- Entregar mensajes de interacción al usuario que utilice el programa.

## 2. DESCRIPCIÓN BREVE DEL PROBLEMA

En la asignatura de Paradigmas de Programación a los estudiantes se les ha presentado la tarea de realizar el Laboratorio Número cuatro, el cual consiste en una simulación del foro Stack Overflow con el paradigma orientado a objetos y el paradigma orientado a eventos, el cual será implementado a través del lenguaje Java y el IDE de Apache NetBeans.

El objetivo principal del foro Stack Overflow, es permitir plantear preguntas que luego pueden ser respondidas por otros usuarios, consta de cinco elementos claves para su funcionamiento, los cuales son:

- **Etiquetas:** Son palabras claves que sirven para categorizar una pregunta, nos permite agrupar preguntas que tengan relación entre sí. Además, cada etiqueta posee una descripción que el usuario puede visualizar, permitiendo ver preguntas y respuestas que tengan relación con ella.

- **Preguntas:** Las preguntas son publicadas por el usuario, las cuales vienen siendo las dudas que este tenga. Las preguntas tienen ciertas características: etiquetas, fecha de publicación, autor, estado, respuestas, votos, visualizaciones, entre otras.
- **Repuestas:** Las respuestas son realizadas por otros usuarios, y permite resolver la duda de la pregunta que ha sido ingresada por un usuario. Las características de las respuestas son: tienen ID propio, que nos permite identificar las respuestas, fecha, ID de la pregunta que ha sido respondida, la respuesta, etiquetas, autor.
- **Usuarios y reputación:** Los usuarios son quienes plantean y responden a preguntas, pero para asegurar la calidad de dichas respuestas el foro nos entrega un sistema de calificación o reputación para usuarios. De esta manera si un usuario tiene alta reputación es porque constantemente participa respondiendo preguntas a usuarios los cuales se sienten satisfechos por la respuesta entregada, generando una mayor fiabilidad a su respuesta.

El problema que se presenta en el laboratorio es simular parte de las características que tiene Stack Overflow y poder ser llevado a cabo por medio de la programación orientada a objetos y orientada a eventos en el lenguaje Java, para poder de esta manera crear una interacción con una interfaz gráfica entre el software y el usuario.

### 3. DESCRIPCIÓN BREVE DEL PARADIGMA

¿Qué es el Paradigma orientado a objetos?

El paradigma orientado a objetos o más conocido como POO, enfrenta los problemas a través de objetos. Esto quiere decir que todo es representado mediante objetos y/o clases. Las clases u objeto es un conjunto de atributos y métodos, los cuáles se pueden ver como datos y comportamiento. Los métodos son procedimientos o funciones que sirven para la manipulación de su información y/o la interacción con otras clases. Algunos de los lenguajes que implementan este paradigma son Smalltalk, C++, Delphi (Object Pascal), Java y C#. [2]

Algunas características del paradigma orientado a objetos son los siguientes:

- **Abstracción:** Representación de las características esenciales de algo sin incluir antecedentes irrelevantes.
- **Encapsulación:** Permite reunir a todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción.
- **Polimorfismo:** Comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre, al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando.
- **Herencia:** Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen.

¿Qué es el Paradigma orientado a eventos?

El paradigma orientado a eventos es aquel en el que la estructura y la ejecución de los programas van determinados por los sucesos o acciones que ocurren en el sistema, definidos por el usuario o por el propio sistema.[4]

Algunas características del paradigma orientado a eventos son las siguientes:

- Dependencia de eventos: El flujo del programa viene dado por eventos que pueden ser acciones del usuario, mensajes de otros programas, etc., separando la lógica de procesamiento de los eventos del resto del código de un programa.
- Controlador de eventos: Es una determinada unidad del programa que se activa para reaccionar ante un evento. Es decir, es un tipo de función o método que ejecuta una acción específica cuando se activa un evento determinado.
- Funciones de activación: Son funciones que deciden qué código ejecutar cuando se produce un evento específico. Se utilizan para seleccionar qué controlador de eventos emplear al producirse un evento.

¿Qué es Java?

Java es un lenguaje de programación y una plataforma informática enfocada en la programación orientada a objetos. Java sirve para crear aplicaciones y procesos en una gran diversidad de dispositivos, ya que nos permite ejecutar un programa en diversos sistemas operativos de manera segura. [3]

#### **4. ANÁLISIS DEL PROBLEMA RESPECTO DE LOS REQUISITOS ESPECÍFICOS QUE DEBEN CUBRIR**

Para poder llevar a cabo un informe y algoritmo óptimo, se debió seguir una serie de requisitos, ya sean de tipo funcionales y no funcionales.

##### Requisitos funcionales

La problemática planteada en este laboratorio consiste en crear una simulación del foro Stack Overflow, en donde un usuario se puede registrar, iniciar sesión, formular preguntas, formular respuestas, aceptar respuestas a preguntas que haya realizado, etc. Estas acciones el usuario las podrá realizar a través de un menú que ha sido implementado en el lenguaje Java, con el paradigma orientado a objetos y eventos, permitiendo una interacción vía interfaz gráfica.

El primer requerimiento funcional que se presenta, es la creación de las clases y estructuras mínimas que van a formar nuestro programa.

Al tener definido las relaciones, clases y estructura del laboratorio anterior que era solo con el paradigma orientado a objetos, se precederá a realizar una adaptación para su uso con el paradigma orientado a eventos, permitiendo un funcionamiento del programa vía interfaz gráfica y no vía menú como el laboratorio anterior. Dicho interfaz nos va a permitir que el usuario sea capaz de manipular e ingresar elementos al stack overflow de forma más intuitiva.

Algunas de las clases y métodos que se han creado son:

- Clase Usuario: Sus atributos son String nombreUsuario, String claveUsuario e int reputacionUsuario. Estos atributos son utilizados para los métodos de registrar usuario y logearlo, producto de que solicita nombreUsuario y claveUsuario. El atributo reputacionUsuario es utilizado para realizar Reward a sus preguntas.
- Clase Etiqueta: Sus atributos son String nombreEtiqueta y String descripcionEtiqueta. Estos atributos son utilizados para el método de crear etiqueta, el cual permite que un usuario logeado pueda crear etiquetas nuevas y estas sean utilizadas. Además, las etiquetas son utilizadas en las preguntas, nos permiten saber a qué tema está relacionada la pregunta.
- Clase Pregunta: Sus atributos son int idPregunta, String tituloPregunta, String contenidoPregunta, String fechaPublicacion, String autorPregunta, ListaDeRespuestas respuestas, ListaDeEtiquetas etiquetas, int estado, int recompensa y String autorrecompensa. Estos atributos son utilizados en los métodos de generar una pregunta, el cual permite que el usuario genere una pregunta que se va a contener en el stack.
- Clase Respuesta: Sus atributos son String contenido, String autor, String fechaDeSubida, int id. Estos atributos son utilizados en el método para generar una respuesta a una pregunta, se necesita el contenido de la respuesta, el autor que genera la respuesta, un id de respuesta y la fecha de publicación. Al tener el autor de la respuesta se puede asignar recompensas con el método Reward si es que su respuesta ha sido aceptada.

### Requisitos No funcionales

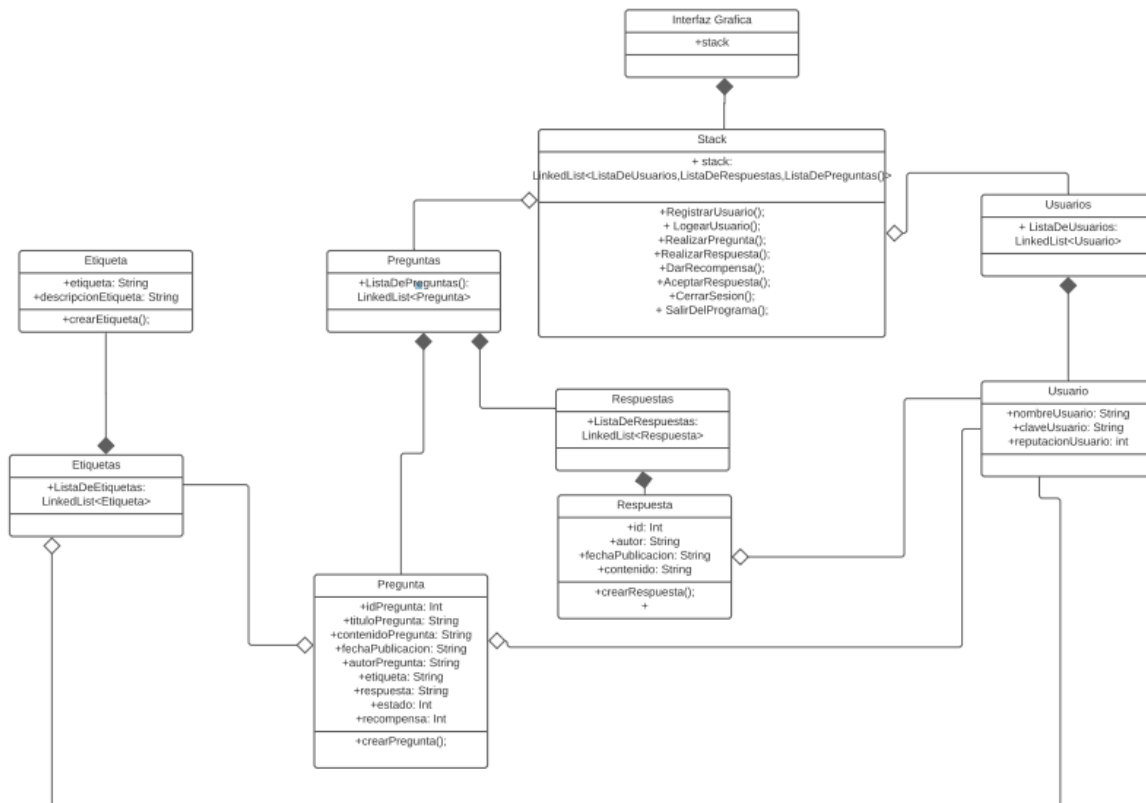
Estos requerimientos son requisitos para la implementación de buenas prácticas y además de generar una estructura al momento de formar el algoritmo. Algunos de estos requerimientos son obligatorios y con puntajes, si no se cumplen los obligatorios el proyecto será evaluado con nota mínima.

Los requerimientos no funcionales son:

- Autoevaluación: Consiste en realizar una autoevaluación a los requerimientos funcionales mencionados anteriormente.
- Lenguaje y versión: Cada laboratorio se debe implementar con un lenguaje distinto y una versión actualizada de este mismo, en este caso el lenguaje utilizado es Java, el IDE utilizado fue Apache NetBeans 12 y el JDK fue OpenJDK versión 11.
- Interacciones con el programa: En este laboratorio todas las interacciones deben ser mediante interfaz gráfica.
- Uso del paradigma: Se debe demostrar la aplicación del paradigma orientado a objetos y el paradigma orientado a eventos.
- Documentación: Se debe documentar el código indicando una breve descripción de las clases creadas, sus atributos, métodos públicos y relaciones.
- Diagrama de análisis y diagrama de diseño.
- Historial de trabajo en Github tomando en consideración la evolución en el desarrollo de su proyecto en distintas etapas. Se requieren **al menos 10 commits** distribuidos en un periodo de tiempo **mayor o igual a 1 semanas**
- Para cada funcionalidad se establecen prerequisites. Estos deben ser cumplidos para que se proceda con la evaluación de la funcionalidad implementada.

## 5. DISEÑO DE LA SOLUCIÓN

Para llegar a una solución y creación del algoritmo, primero se realizó un diagrama de análisis, el cual nos permitió tener una mayor visión del cómo se compone el stack overflow, permitiendo tener una idea más clara y detallada de las clases, atributos y métodos que se debe implementar en dicho stack. Cabe destacar que al momento de implementar el diagrama de análisis a código me he percatado que me faltan métodos, aun así, este diagrama fue de gran ayuda, ya que permitió tener una idea más clara de cómo empezar el proyecto.



*Ilustración 1: Diagrama de análisis*

**Breve explicación del diagrama:**

La imagen anterior corresponde al Diagrama inicial de análisis, se puede observar que los métodos están contenidos en el stack, los cuales podrán ser utilizados por el usuario en la clase Interfaz Gráfica permitiéndoles poder utilizar el programa. Si nos enfocamos en las relaciones se puede observar que preguntas y usuarios se encuentran con la relación de agregación, producto de que puede haber preguntas o no en nuestro stack, lo mismo pasa con Usuarios producto de que pueden existir o no usuarios registrados y este no dependería de los usuarios. Las respuestas están conectadas con composición a las Preguntas producto de que sin preguntas no pueden existir las respuestas. Por el momento se han ingresado la gran mayoría de los métodos en stack.

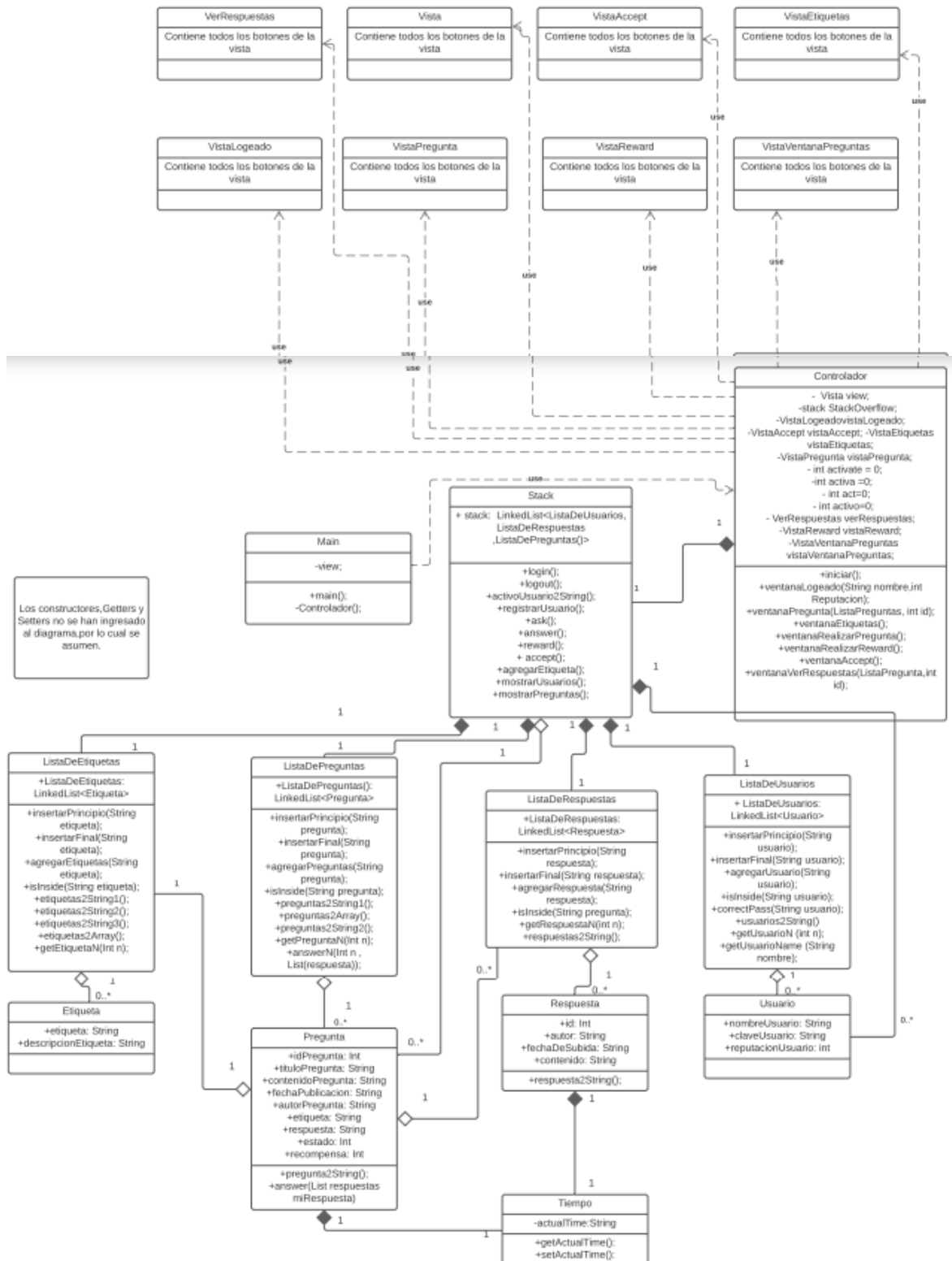


Ilustración 2: Diagrama de diseño



### **Breve explicación del diagrama y cambios realizados:**

Se puede observar que el diagrama aumento su tamaño considerablemente. Se ha agregado al diagrama final la clase Tiempo, la cual nos permite poder obtener la hora y fecha exacta del momento en que se plantea una pregunta o respuesta. Además, se puede apreciar que las ListasDeEtiquetas, ListasDeUsuarios, ListasDePreguntas y ListasDeRespuestas tienen un aumento considerable en sus métodos, producto de que se crearon métodos para insertar al inicio y final de las listas, entre otras. También en este diagrama se han implementado las multiplicidades en las relaciones de agregación y composición. Estas nos permiten entender de mejor manera la relación. Por ejemplo, la ListaDeRespuestas está vinculada con agregación a las preguntas con multiplicidad de muchas a 1, esto quiere decir que puede haber muchas respuestas para una sola pregunta. De igual forma se ha incorporado la clase controlador la cual contiene las ventanas de la interfaz gráfica. Así mismo la ventana main se ha puesto que usa a la ventana controlador, producto que es la inicializadora de todas las ventanas.

Para la implementación del stack se va a realizar con listas enlazadas, las clases a trabajar con este método son las ListasDeRespuestas (contenedora de todas las respuestas), ListasDeUsuarios (contenedora de todos los usuarios), ListasDeEtiquetas (contenedora de todas las etiquetas) y ListaDePreguntas (contenedora de todas las preguntas).

Para la clase usuario se utilizan tres atributos, los cuales son: Username, Password y reputación. El username nos permitirá identificar a cada usuario que se registre, para que de esta manera no existan usuarios con mismo nombre, la password nos permite asegurarnos de que el usuario debe ingresar dicha contraseña para poder logearse, ya que si no calzan no podrá ingresar al stackoverflow y la reputación nos permite saber que tan activos son los usuarios, en donde si realizan preguntas y aceptan respuestas se obtendrán ganancias para aumentar dicha reputación.

Para la clase pregunta, se han utilizado los siguientes elementos:

- idPregunta: Consiste en un numero entero que nos permitirá identificar a la pregunta. Cada vez que realicemos una pregunta nueva este id se irá incrementando,
- tituloPregunta: String con nombre representativo a la pregunta.
- contenidoPregunta: String con el contenido de la pregunta.
- autorPregunta: Nombre del usuario que formulo la pregunta.
- ListaDeEtiquetas: Consiste en una lista de etiquetas que el usuario puede seleccionar.
- ListaDeRespuestas: Consiste en una lista de las respuestas que va a tener la pregunta.
- estado: Si esta respondida o no.
- fechaPublicacion: Consiste en un string con la fecha de publicación de la pregunta.
- Recompensa: Consiste en un entero que nos permite saber si la pregunta tiene recompensa.

Para la lista de respuestas, se han utilizado los siguientes elementos:

- contenido: String con la respuesta que se ha realizado.
- id: Nos sirve para saber a qué número de respuesta corresponde a la pregunta.
- autor: Nombre de usuario que realizo la respuesta.
- fechaDeSubida: fecha en que ha sido respondida la pregunta.

## 6. ASPECTOS DE IMPLEMENTACIÓN

El código realizado en Java está estructurado solo en un archivo llamado “lab4\_StackOverFlow\_20227250\_SalgadoEspinoza”. En el cual dentro tiene un archivo llamado código fuente, el cual tiene los siguientes archivos:

- **gradle:** El cual contiene el wrapper.
- **src:** El cual contiene a los package y archivos.java utilizados en el proyecto.
- **build.gradle**
- **gradle.properties**
- **gradlew**
- **Archivo por lotes llamado: gradlew**
- **Setting.gradle**

Se utilizo Open JDK en la versión 11, el IDE Apache Netbeans y como sistema operativo Windows 10.

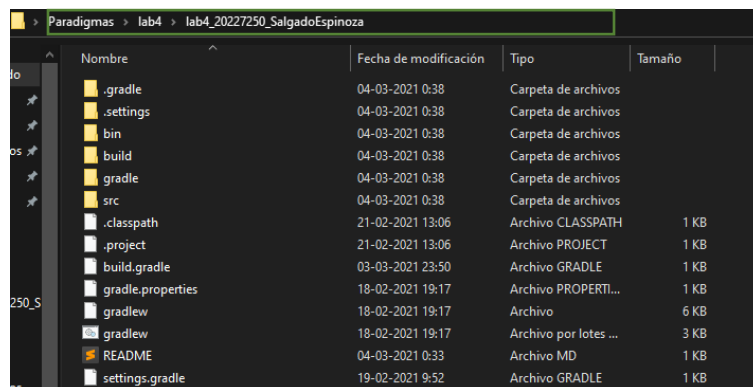
Al momento de realizar los commits se han subido todos los archivos que contenía el proyecto de Apache Netbeans a GitHub por si se me llegaba a borrar alguno sin querer. Esta nube es muy importante, ya que, si se nos llega a borrar algo por X motivo nuestro código o avances que estemos realizando, estarán resguardados, ya que se guardan todos los cambios que se han realizado.

La simulación al foro de Stack Overflow que se ha realizado es muy básica, ya que al ingresar al foro real se pueden apreciar una gran cantidad de funcionalidades que este nos permite realizar. Cabe destacar que de igual manera hay funcionalidades que deben ser muy complejas a nivel de código, provocando que sea más elevado de realizar su simulación.

## 7. INSTRUCCIONES DE USO

A continuación, se va a presentar como se debe compilar el proyecto en la consola/terminal de Windows, cabe destacar que también fue probado en Linux.

**Paso 1:** Abrir Carpeta contenedora del proyecto.



*Ilustración 3: Abrir carpeta del proyecto*

## Paso2: Se debe ingresar a la carpeta SRC

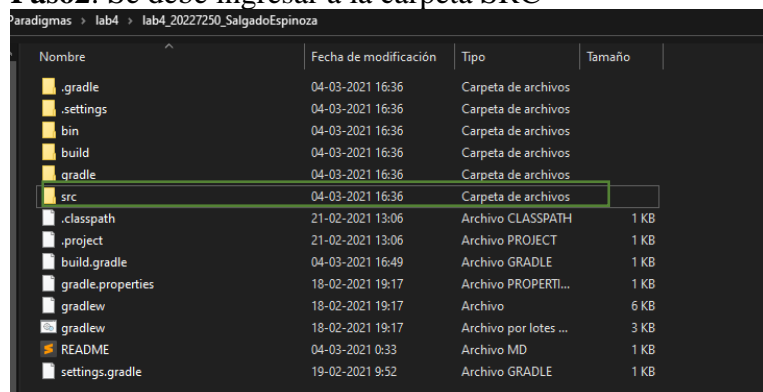


Ilustración 4: Ingresando a carpeta SRC

## Paso3: Ingresamos a la carpeta main.

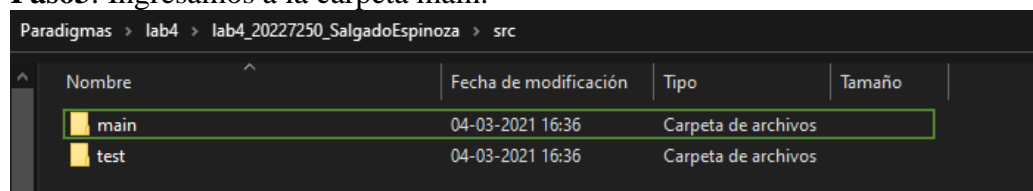


Ilustración 5: Ingresando a carpeta main

## Paso4: Ingresamos a la carpeta java.

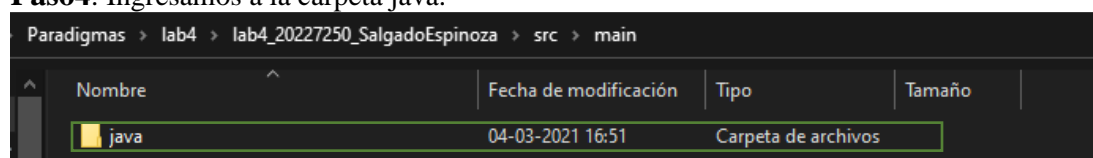


Ilustración 6: Ingresando a carpeta java

## Paso5: Se debe seleccionar el cuadro verde y escribir cmd

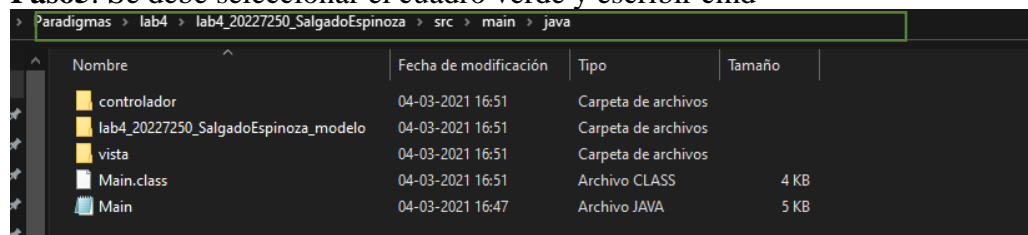


Ilustración 7: Entrando en consola

## Paso6: Se muestra la consola.

Se escriben el siguiente comando: javac Main.java

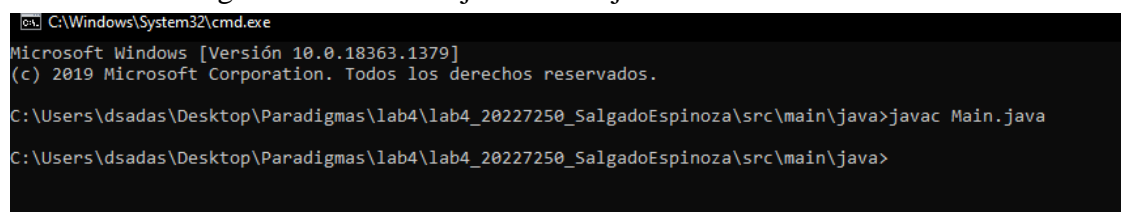
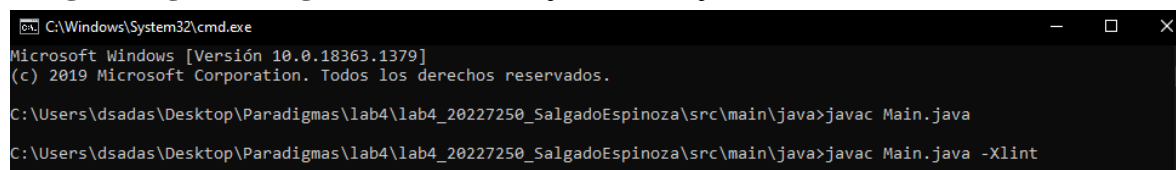


Ilustración 8: Comando javac Main.java

Luego se ingresa el siguiente comando: `javac Main.java -Xlint`



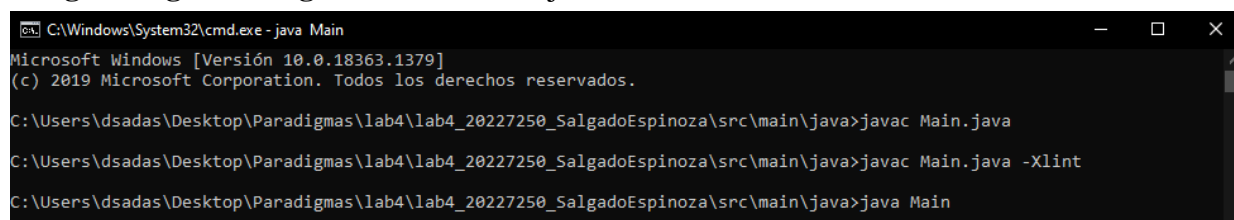
```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.18363.1379]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\dsadas\Desktop\Paradigmas\lab4\lab4_20227250_SalgadoEspinoza\src\main\java>javac Main.java

C:\Users\dsadas\Desktop\Paradigmas\lab4\lab4_20227250_SalgadoEspinoza\src\main\java>javac Main.java -Xlint
```

Ilustración 9: Comando `javac Main.java -Xlint`

Luego se ingresa el siguiente comando: `java Main`



```
C:\Windows\System32\cmd.exe - java Main
Microsoft Windows [Versión 10.0.18363.1379]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\dsadas\Desktop\Paradigmas\lab4\lab4_20227250_SalgadoEspinoza\src\main\java>javac Main.java

C:\Users\dsadas\Desktop\Paradigmas\lab4\lab4_20227250_SalgadoEspinoza\src\main\java>javac Main.java -Xlint

C:\Users\dsadas\Desktop\Paradigmas\lab4\lab4_20227250_SalgadoEspinoza\src\main\java>java Main
```

Ilustración 10: `java Main`

**Paso7:** Al presionar Enter va a salir el siguiente mensaje y se abre la interfaz gráfica.

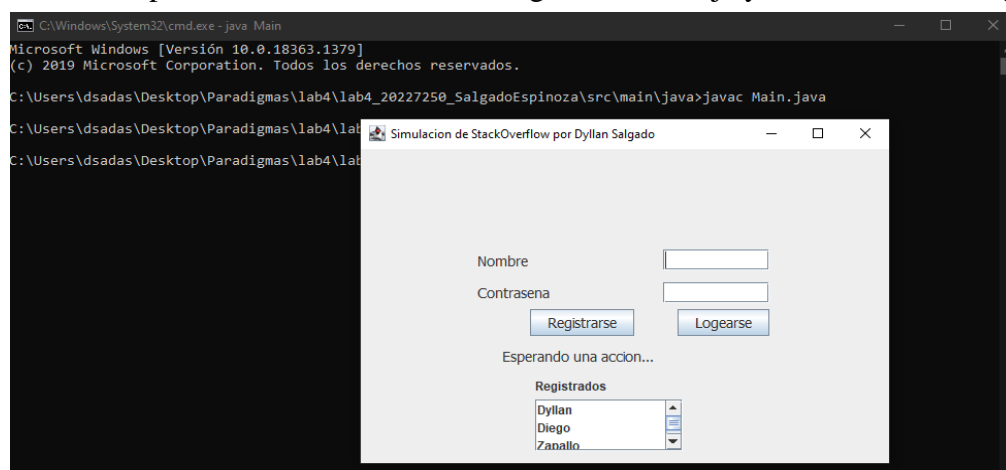


Ilustración 11: Interfaz para trabajar.

Explicación de algunas interacciones. Existe un anexo que contiene imágenes con cada paso de las interacciones.

1. **Registrarse:** Esta opción permite al usuario registrarse, se le solicita el nombre y clave. Si el nombre ya se encuentra registrada muestra en pantalla el siguiente mensaje: Usuario ya se encuentra registrado y se volverá a solicitar los datos. Si no rellena ningún espacio saldrá el siguiente mensaje “rellenar todos los campos disponibles”. Si el nombre se encuentra disponible para ser utilizado y se ha ingresado la contraseña, se muestra el siguiente mensaje: “USUARIO REGISTRADO CON ÉXITO”.
2. **Logear:** Esta opción permite logear a un usuario ya registrado, se le solicita el nombre y clave. Si la clave o nombre de usuario no son correctos se va a mostrar un mensaje para acción.
3. **Crear Etiqueta:** Esta opción permite a un usuario ya logeado crear etiquetas que se utilizan en las preguntas. Se le solicita ingresar el nombre de etiqueta y una breve descripción. Si el nombre de etiqueta ya se encuentra registrada muestra por pantalla

que existe una etiqueta ya con ese nombre.

4. **Realizar Pregunta:** Esta opción permite a un usuario ya logeado crear una pregunta. Se le solicita ingresar Título de la pregunta, Contenido de la pregunta y seleccionar una de las etiquetas que se encuentran disponibles. Al rellenar todos estos se va a mostrar un mensaje diciendo que se agregó la pregunta.
5. **Realizar Respuesta:** Esta opción permite a un usuario ya logeado crear una respuesta a una pregunta. Se va a mostrar todas las preguntas que se encuentran disponibles, al momento de seleccionarla se le pedirá que ingrese la respuesta.
6. **Ofrecer Recompensa:** Nos permite ofrecer recompensas a preguntas. Si tengo menos reputación de la que ofrezco, muestra mensaje en pantalla y vuelve al menú. Si tengo reputación suficiente se realiza la recompensa con éxito.
7. **Aceptar Respuesta:** El usuario que crea una pregunta puede aceptar respuestas de su pregunta, si se asignaron recompensas se le da a la persona que se le acepto la respuesta y si no tiene recompensa la pregunta se le asigna 15 puntos de reputación a la persona que le aceptaron la respuesta y 2 puntos a la persona que crea la pregunta.

### **Posibles errores**

Hasta el momento no se han encontrados errores, he tratado de debugear todos los errores posibles que se puedan encontrar, pero aun así mi código puede tener fallas.

## **8. RESULTADOS Y AUTOEVALUACIÓN**

Se logra simular un foro tipo Stack Overflow. El programa permite registrar usuarios, generar preguntas, responder preguntas, aceptar respuestas de una pregunta, dar recompensa a quien realiza la pregunta y al usuario que se le acepto la respuesta, entre otras cosas. Puede que tenga algunas fallas que no he podido visualizar, pero si es que las tiene no debe ser muy difícil su corrección. No se logró realizar funcionalidades opcionales, se ha preferido realizar las funcionalidades obligatorias y que funcionen bien. Se realizaron bastantes pruebas en las funcionalidades obligatorias, encontrando muchas fallas los cuales se fueron arreglando mientras avanzaba el tiempo.

### **Requerimientos no Funcionales:**

Incluir autoevaluación los requerimientos funcionales solicitados.	Se adjunta un archivo de texto plano con la autoevaluación de los requerimientos.
Lenguaje y herramientas de trabajo:	Se utiliza NetBeans 12 y OpenJDK versión 11.
Interacción mediante interfaz gráfica.	Funciona correctamente
La solución debe demostrar la aplicación del paradigma orientado a objetos y dirigidos por eventos.	Se aplican los conocimientos aprendidos en clases de orientación a objetos y eventos.
Se deben establecer los prerrequisitos para cada	Se han seguido todos los prerrequisitos para una buena

funcionalidad.	implementación del programa.
Separación Modelo-Vista.	Se realiza la separación del modelo -vista, tratando de que no se tenga ningún grado de acoplamiento.
Se debe documentar el código indicando breve descripción de las clases creadas, atributos, métodos públicos y relaciones.	Se comentan todas las clases y métodos.
Organización del código.	Se trata de ser lo más organizado posible. Para mayor orden.
Incluir diagrama de análisis	Se logra implementar.
Incluir diagrama de diseño	Se logra implementar.
Historial de trabajo en Github. Se requieren <b>al menos 10 commits</b> distribuidos en un periodo de tiempo <b>mayor o igual a una semana.</b>	Se cumple con el historial.

#### Requerimientos funcionales:

Clases y estructuras que forman el programa.	Se cumple con las clases y estructuras solicitadas.
Register/login/logout	Se cumple con Register, Login y logout.
Ask	Se cumple con Ask.
Answer	Se cumple con Answer
Reward	Se cumple con Reward
Accept	Se cumple con Accept
Vote	<b>No se logra implementar</b>
Report	<b>No se logra implementar</b>
Ranking	<b>No se logra implementar</b>
Search	<b>No se logra implementar</b>
<b>Funcionalidad Extra creada por mí: CrearEtiqueta</b> Consiste en que los usuarios logeados pueden crear etiquetas que serán contenidas en el stack, permitiendo a cualquier usuario poder utilizarlas. Al momento de crear la etiqueta si ya existe una con ese nombre, no permitirá crearla.	Se logra implementar

## 9. CONCLUSIÓN

A través de los datos adquiridos por el siguiente informe, se puede concluir que: Stack Overflow es un foro muy íntegro en comparación con otros foros de programación, por el motivo de estar dedicado a usuarios profesionales o expertos en el ámbito de programación y para personas que están recién aprendiendo. Al ser un foro que lo utilizan personas expertas, estas mismas pueden ir dando ideas para ir mejorando la página web, prestando servicios de mantención, entre otros. Esto genera, que el Stack Overflow aparte de ser un sistema íntegro como lo dije al inicio, también será muy complejo a nivel de estructuración y código. Esto se experimentó al momento de simularlo, ya que, si bien, era una simulación parcial, tenía que ir pensando en muchas formas en que podría hacer fallar el programa o faltar al momento de generar las funcionalidades en Java.

En el laboratorio, si bien, se ha tomado como base el laboratorio 3 correspondiente al paradigma orientado a objetos, tuve muchas dificultades para llevarlo a cabo, producto de que me he complicado viendo videos respecto de cómo funciona MVC, del acople que debe tener dicho código y también de la cohesión. Asimismo, tuve que estar revisando videos de cómo funcionan los botones en NetBeans y el cómo hacerlos funcionar. Como dije anteriormente, este laboratorio me ha costado realizarlo, pero siento que he aprendido un montón respecto de cómo funciona la interfaz gráfica en Java.

El código creado se puede mejorar considerablemente para que sea mucho más sencillo y ordenado, de igual manera puede pasar con la interfaz, ya que al ser la primera interfaz gráfica que he realizado puede que resulte poco agradable a la vista, entre otras cosas. Con más práctica y tiempo, lo dicho anteriormente puede mejorar considerablemente. En el ámbito de resultados, se pudo haber llegado a uno mucho mejor, no obstante, se llega a una implementación parcial, logrando presentar algunas fallas.

Con respecto a la implementación del paradigma Orientado a Objetos, se trató de seguir fielmente lo aprendido en clases, en el ámbito de relacionar las distintas clases, utilizando correctamente las distintas características y bondades que entrega el paradigma como las que entrega el lenguaje. De igual manera, se ha tratado de poner en práctica la implementación del paradigma orientado a eventos, el cual nos facilitó el trabajo para realizar la interfaz gráfica.

Finalmente, si bien, el laboratorio 4 fue creado en Java me resultó más difícil adaptarme no en el lenguaje, sino, más bien en cómo trabajar con los botones y la interfaz gráfica, igualmente, en comparación con los paradigmas anteriores me resultó más fácil la adaptación, producto de que me confunden demasiado realizar funciones y predicados para todo el código, en cambio, en el actual laboratorio las funciones eran muy similares a cómo se desarrollan en C. Por lo tanto, en este laboratorio se trató de seguir literalmente todas las condiciones que solicitaban las funcionalidades obligatorias.

## 10. REFERENCIAS

- 1) German Escobar, (Sep 21, 2015), ¿Qué es un stack overflow (no el sitio)? (30-11-2020), <https://blog.makeitreal.camp/que-es-un-stack-overflow-desbordamiento-de-pila/>
- 2) Universidad de Antioquia(2015), El paradigma OO: una forma de ver el mundo (06-01-2021)[http://aprendeenlinea.udea.edu.co/lms/men\\_udea/mod/page/view.php?id=19537#:~:text=El%20paradigma%20orientado%20a%20objetos%20\(OO\)%20define%20los%20programas%20en,términos%20de%20comunidades%20de%20objetos.&text=Es%20en%20este%20modo%20de,están%20separados%20y%20sin%20relaci%C3%B3n.](http://aprendeenlinea.udea.edu.co/lms/men_udea/mod/page/view.php?id=19537#:~:text=El%20paradigma%20orientado%20a%20objetos%20(OO)%20define%20los%20programas%20en,términos%20de%20comunidades%20de%20objetos.&text=Es%20en%20este%20modo%20de,están%20separados%20y%20sin%20relaci%C3%B3n.)
- 3) Tokio. (2018, 17 septiembre). Java significado: ¿qué es Java y para qué sirve? (06-01-2021) <https://www.tokioschool.com/noticias/java-significado-que-es-java/>
- 4) Corvo, H. S. (2020, 30 marzo). Programación orientada a eventos: características, ejemplos, ventajas, aplicaciones. (28-02-2021 <https://www.lifeder.com/programacion-orientada-a-eventos/>



## Anexos

### Programa funcionando:

Al iniciar el programa se va a mostrar la siguiente ventana:

Simulacion de StackOverflow por Dyllan Salgado

Nombre

Contraseña

contrasena incorrecta

Registrados

Dyllan	▲
Diego	■
Zanallo	▼

Anexo1. 1 Mostrando menú inicial

**Seleccionando el botón registrarse:** Para cada acción se encuentra un mensaje, por ejemplo, si no rellena los campos solicitados se muestra un mensaje, si ya existe el nombre de usuario a registrar muestra otro mensaje.

Si ingreso un nombre de usuario disponible y la clave se muestra el siguiente mensaje:

Simulacion de StackOverflow por Dyllan Salgado

Nombre

Contraseña

USUARIO REGISTRADO CON EXITO

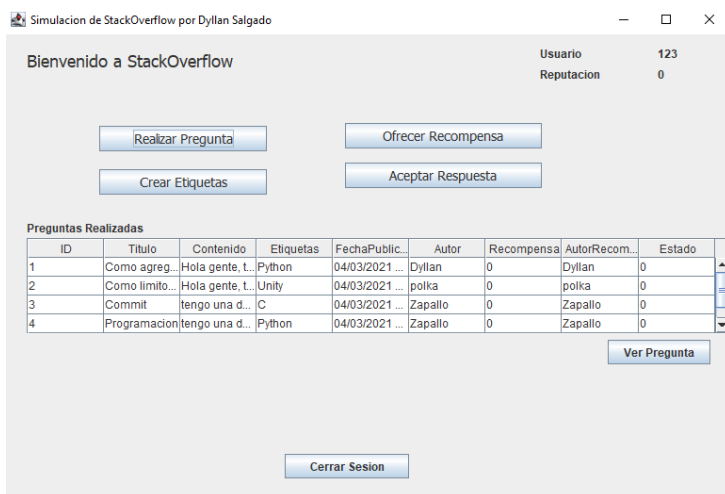
Registrados

Dyllan	▲
Diego	■
Zanallo	▼

Anexo1. 2 Registrando a un usuario

**Seleccionando el botón logearse:** Para cada acción se encuentra un mensaje, por ejemplo, si no rellena los campos para logearse se muestra un mensaje, si el nombre de usuario no se encuentra se muestra un mensaje, si el usuario se encuentra y se ha equivocado en la contraseña se muestra un mensaje.

Si ingreso un nombre de usuario dentro de la lista de registrados e ingreso su contraseña, se muestra la ventana de logeado.



Anexo1. 3 Usuario ya logeado, se muestra menú logeado

La ventana logeado tiene 6 botones, los cuales nos permite realizar una pregunta, ofrecer recompensas, crear etiquetas, aceptar respuestas, ver preguntas y cerrar sesión. Además, nos permite visualizar el nombre de usuario que ha ingresado, la reputación y el stack con preguntas.

**Seleccionando botón Realizar Pregunta:** Se muestra la siguiente ventana.

Usuario: 123  
Reputacion: 0

Realice su pregunta

Titulo:

Contenido:

Etiquetas: C#

Otorgar recompensa: ☐

Publicar pregunta

Esperando una accion...

Preguntas

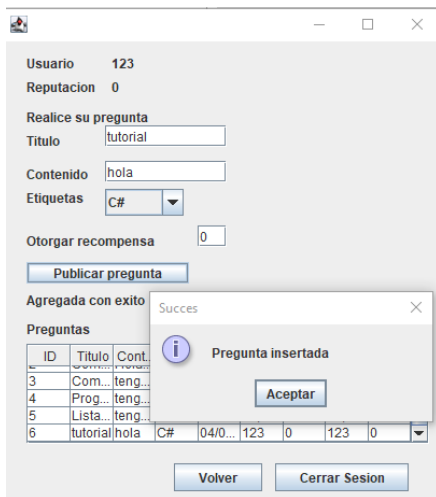
ID	Titulo	Cont...	Etiqu...	Fech...	Autor	Rec...	Autor...	Esta...
1	Com...	Hola...	Pyth...	04/0...	Dyllan	0	Dyllan	0
2	Com...	Hola...	Unity	04/0...	polka	0	polka	0
3	Com...	teng...	C	04/0...	Zapa...	0	Zapa...	0
4	Prog...	teng...	Pyth...	04/0...	Zapa...	0	Zapa...	0

Volver, Cerrar Sesion

Anexo1. 4 Ingresando a realizar pregunta

Si al momento de seleccionar el botón publicar pregunta no se ha relleno ningún campo, se muestra un mensaje, si la recompensa ofrecida es menor a la que tiene el usuario se muestra un mensaje.

Si se han relleno los campos bien y se ha seleccionado publicar pregunta se muestra un mensaje y se añade al stack de preguntas.



Anexo1. 5 Realizando la pregunta

Si se selecciona volver se dirige al menú anterior logeado y si selecciona cerrar sesión se muestra la ventana para registrar y logear usuario. En este caso se selecciona cerrar sesión para ingresar con otro usuario.



Anexo1. 6 Logeando nuevo usuario

**Seleccionando botón Ver Pregunta:** El usuario Dyllan va a responder a la pregunta realizada anteriormente. Para esto seleccionas Ver Pregunta, el cual te va a mostrar un menú para ingresar la pregunta a responder.



Anexo1. 7 Ingresando pregunta para responder

Seleccionamos la pregunta del usuario 123 que es tutorial.

Nos muestra una ventana con la pregunta realizada y todos los datos que contiene.

Para responder solo se tiene que rellenar el campo que dice ingrese respuesta y el botón Responder.

Si no se ha ingresado nada en el campo solicitado y se presiona Responder muestra un mensaje.

Usuario: Dylan  
Reputacion: 10

Pregunta seleccionada

ID: 5  
Titulo: tutorial  
Contenido: hola  
Fecha de Publicacion: 04/03/2021 15:12:47  
Autor: 123  
Estado: 0  
Recompensa: 0  
No existen respuestas a esta pregunta  
Etiquetas: CF

Ingrese la respuesta

Volver Responder

Anexo1. 8 Mostrando el menú para realizar respuesta

Se ha rellenado el campo solicitado y la respuesta ha sido añadida.

Usuario: Dylan  
Reputacion: 10

Pregunta seleccionada

Recompensa: 0  
Respuestas realizadas:  
ID: 0  
Autor: Dylan  
Respuesta: como estas?  
Fecha de publicacion: 04/03/2021 15:16:30  
Etiquetas: CF

Ingrese la respuesta

Volver Responder

Anexo1. 9 Respuesta Realizada

Volvemos a cerrar la aplicación seleccionado volver y luego en la ventana logeado cerramos sesión.

Simulacion de StackOverflow por Dylan Salgado

Bienvenido a StackOverflow

Usuario: Diego  
Reputacion: 10

Realizar Pregunta Ofrecer Recompensa  
Crear Etiquetas Aceptar Respuesta

Preguntas Realizadas

ID	Titulo	Contenido	Etiquetas	FechaPublic...	Autor	Recompensa	AutorRecom...	Estado
1	Como agreg...	Hola gente, t...	Python	04/03/2021 ...	Dylan	0	Dylan	0
2	Como limito...	Hola gente, t...	Unity	04/03/2021 ...	polka	0	polka	0
3	Commit	tengo una d...	C	04/03/2021 ...	Zapallo	0	Zapallo	0
4	Programacion	tengo una d...	Python	04/03/2021 ...	Zapallo	0	Zapallo	0

Ver Pregunta

Cerrar Sesion

Anexo1. 10 Logeando con nuevo usuario

### Seleccionando botón Ofrecer Recompensa:

Ingresamos con Diego y seleccionamos el botón “Ofrecer Recompensa”

Ingresamos el ID de la pregunta del usuario 123 y realizamos la recompensa. Si no tenemos recompensa para dar se mostrará un mensaje, si se ofrece mas de la que se tiene se mostrará un mensaje.

ID	Titulo	Cont...	Etiqu...	Fech...	Autor	Reco...	Autor...	Estado
3	Com...	teng...	C	04/0...	Zapal...	0	Zapal...	0
4	Progr...	teng...	Python	04/0...	Zapal...	0	Zapal...	0
5	Lista...	teng...	C	04/0...	Zapal...	0	Zapal...	0
6	tutorial	hola	C#	04/0...	123	0	123	0

Anexo1. 11 Se realiza una recompensa

### Seleccionamos dar recompensa

Successful

Recompensa entregada con exito

Aceptar

Anexo1. 12 Se añade la recompensa

Cerramos la sesión e ingresamos con el usuario 123 para aceptar la respuesta anteriormente escrita.

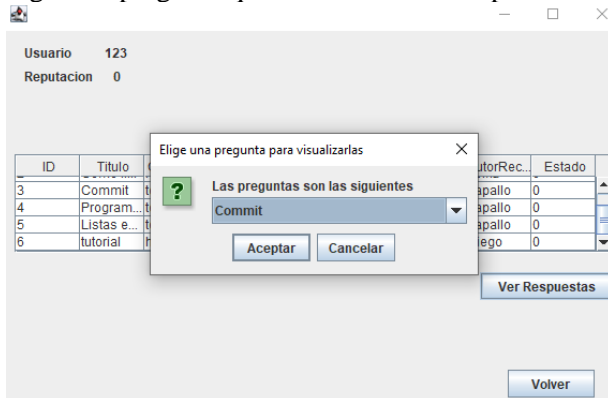


Anexo1. 13 Se vuelve a iniciar con usuario que realizo pregunta

Se puede observar que la pregunta tiene una nueva recompensa de 10 puntos realizada por Diego.

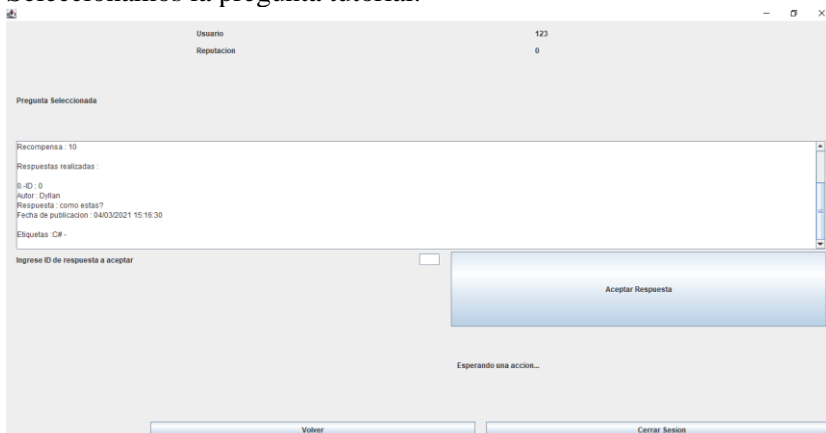
### Seleccionando botón Aceptar Respuesta:

Debes seleccionar Ver Respuestas. Solo se pueden ingresar a preguntas hechas por el autor, si se ingresa a pregunta que no ha sido creada por el usuario logeado muestra un mensaje.



Anexo1. 14 Seleccionar la pregunta del usuario que ha creado la pregunta

Seleccionamos la pregunta tutorial.



Anexo1. 15 Seleccionamos la pregunta tutorial y la respuesta de Dyllan

Ingresamos el ID de respuesta que se quiere aceptar en este caso es 0 y se acepta la respuesta.

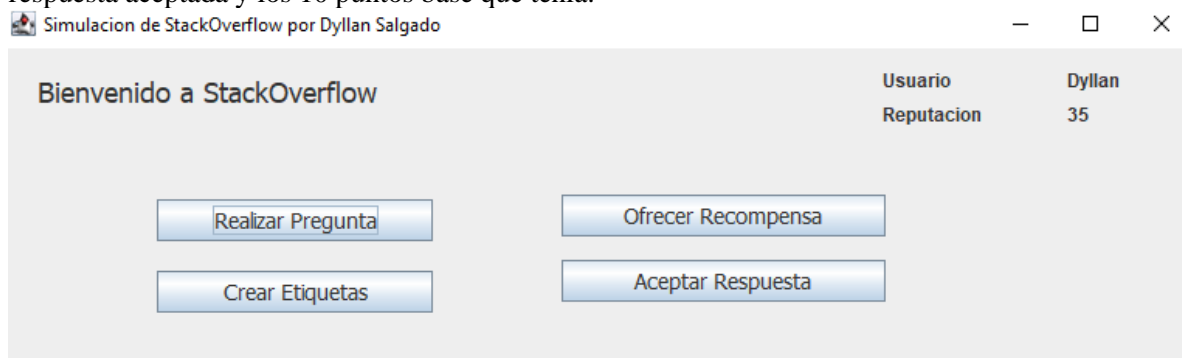
Cerramos la sesión y se puede observar que las recompensas han sido otorgadas.

Al creado de la pregunta se le han otorgado 2 puntos.



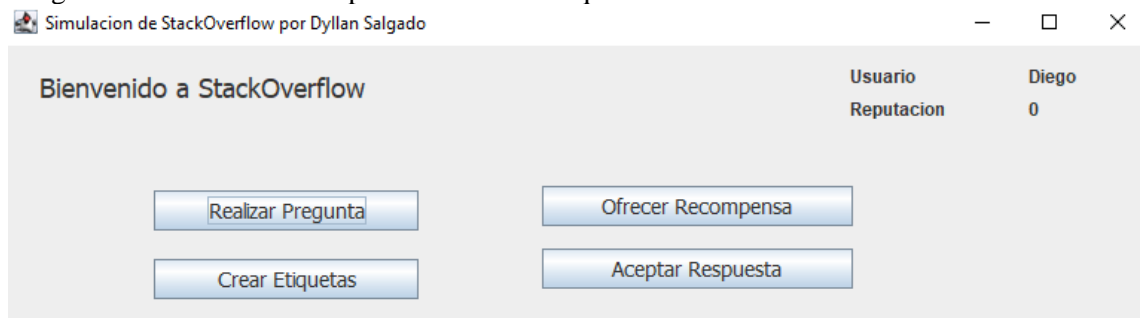
Anexo1. 16 Revisando si se han asignado las recompensas

Al creado de la respuesta se le han asignado los 10 puntos otorgados por Diego mas 15 por ser la respuesta aceptada y los 10 puntos base que tenía.



Anexo1. 17 Revisando si se han asignado las recompensas

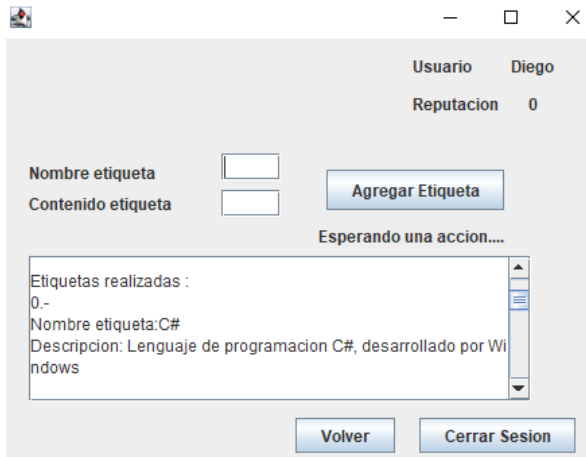
Diego el creador de la recompensa se le resta lo que ha ofrecido.



Anexo1. 18 Revisando si se han asignado las recompensas

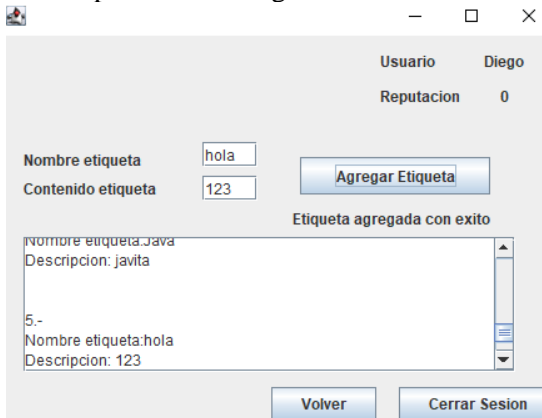
### Seleccionando botón Crear Etiquetas:

Al momento se seleccionar Crear Etiquetas, se muestra la siguiente ventana:



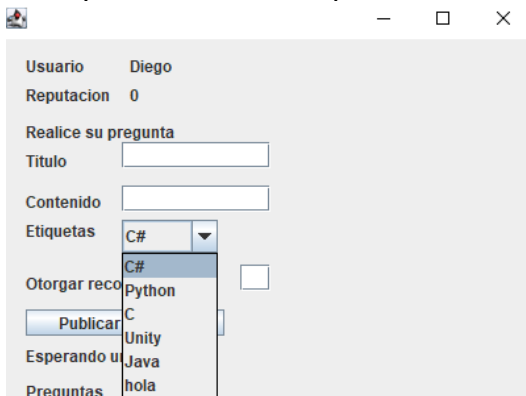
Anexo1. 19 Creando una etiqueta en el stack

Si se rellenan todos los espacios y el nombre de etiqueta no se encuentra registrado se podrá agregar correctamente la etiqueta. Si se ingresan nombres de etiquetas ya registradas muestra un mensaje, lo mismo pasa si no se ingresa nada.



Anexo1. 20 Agregando los datos de la etiqueta

La etiqueta se encuentra disponible en la ventana para crear preguntas.



Anexo1. 21 Visualización de la etiqueta en la pregunta.

Lo anteriormente descrito muestra un breve tutorial de cómo utilizar la aplicación.