



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

**INFORME LABORATORIO 3**  
**Paradigma Orientado a Objetos**  
**Paradigmas de programación**

Dyllan Salgado

Profesor: Víctor Flores.  
Fecha de entrega: 11-01-2021

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.....	9
7. INSTRUCCIONES DE USO.....	10
8. RESULTADOS Y AUTOEVALUACIÓN.....	13
9. CONCLUSIÓN .....	15
10. REFERENCIAS .....	16

## Tabla de Figuras

Ilustración 1: Diagrama de análisis .....	7
Ilustración 2: Diagrama de diseño.....	8
Ilustración 3: Abrir carpeta del proyecto.....	10
Ilustración 4: Ingresar a consola.....	11
Ilustración 5: Escribir comando para compilar .....	11
Ilustración 6: Menú para trabajar. ....	11

## 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. Nos permite realizar preguntas y respuestas sobre problemáticas que tengamos 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 de este laboratorio consiste en simular el foro stack overflow con requisitos que nos plantea la problemática en el paradigma orientado a objetos.

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.

## 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 tres, el cual consiste en una simulación del foro Stack Overflow con el paradigma orientado a objetos, el cual será implementado a través del lenguaje Java.

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 en el lenguaje Java.

### 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]

¿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]

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.
- Entre otros.

#### 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.

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 la estructura se precederá a realizar el código para que se pueda manipular la clase menú. Dicha nos va a permitir que el usuario sea capaz de manipular e ingresar elementos al stack overflow.

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.

Al tener dichas clases y métodos bien planteadas e implementadas se podrá realizar una breve simulación de un stack overflow.

## 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 consola/terminal.
- Uso del paradigma: Se debe demostrar la aplicación del paradigma orientado a objetos.
- 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 2 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, el diagrama de análisis fue de gran ayuda, ya que permitió tener una idea más clara de cómo empezar el proyecto.

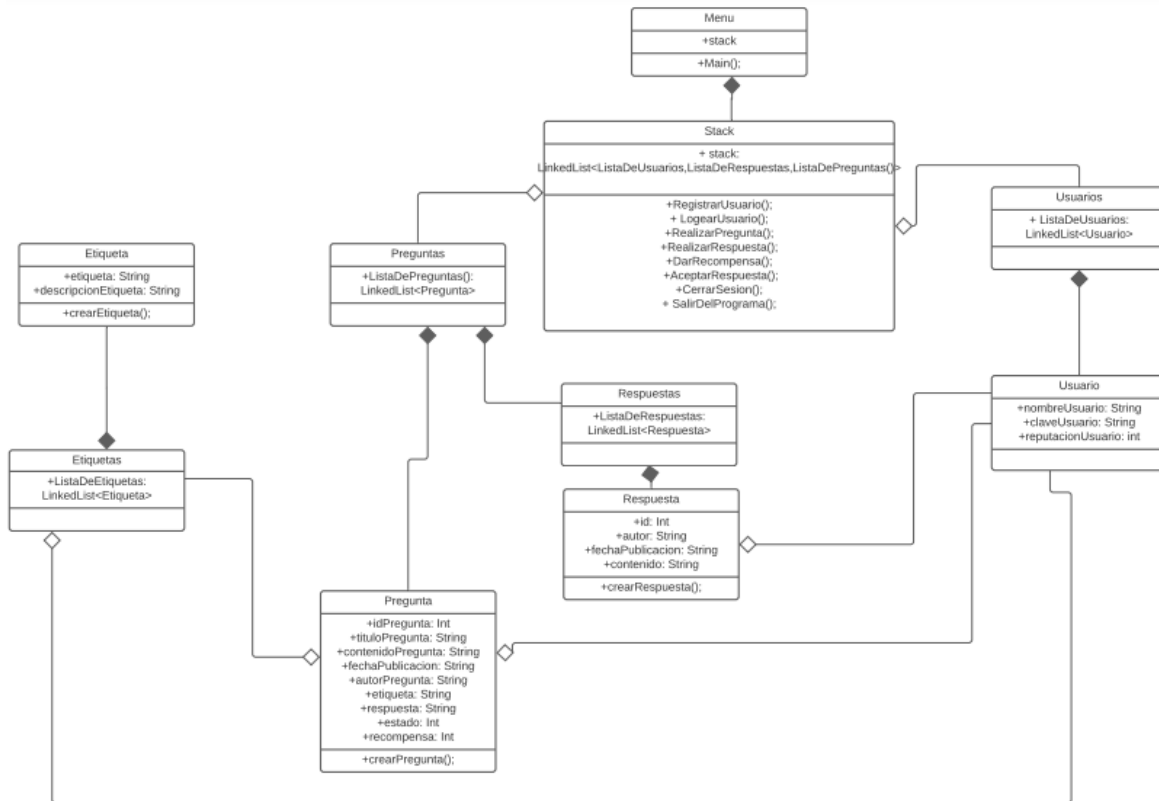


Ilustración 1: Diagrama de análisis

### **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 Menú 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.

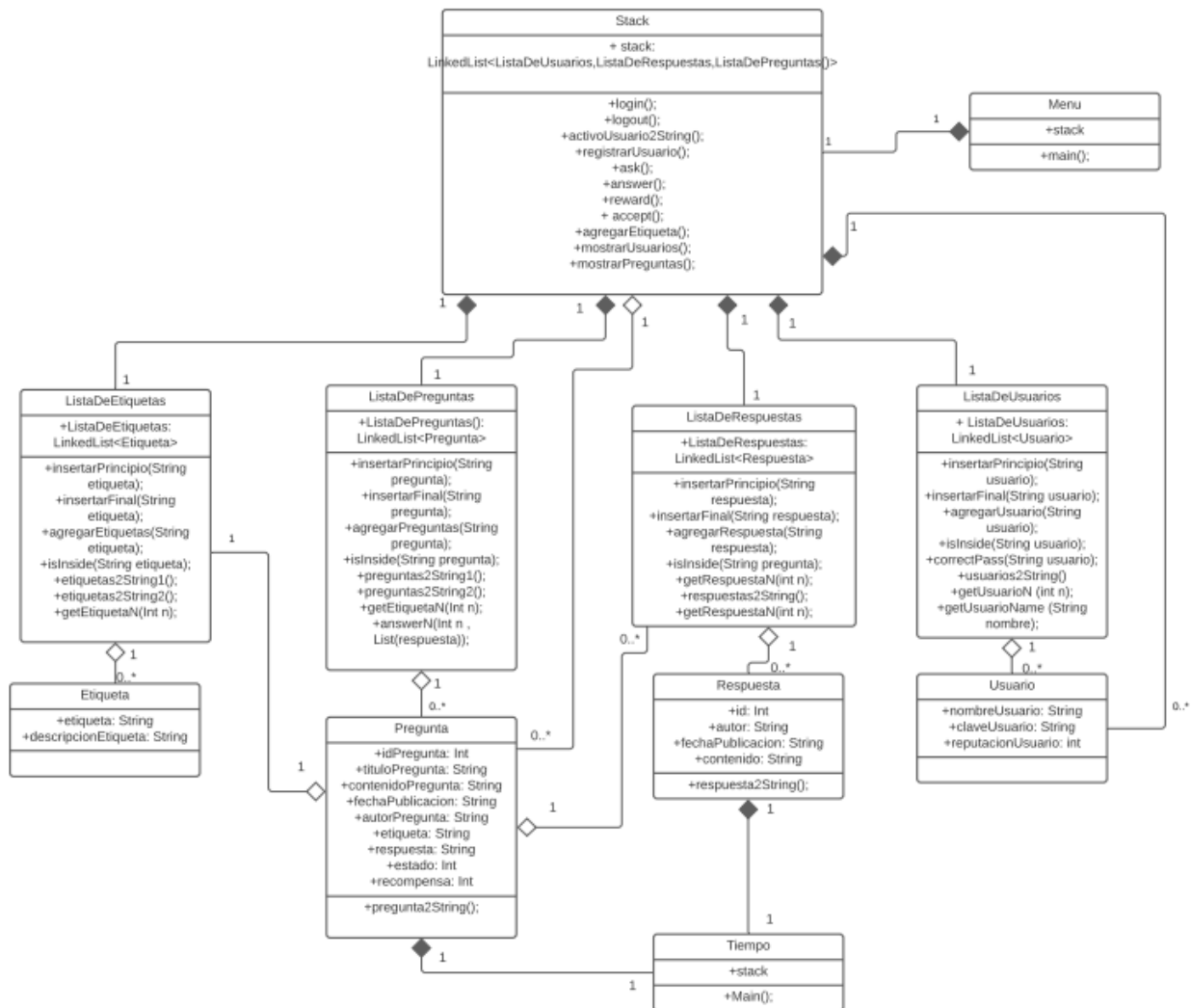


Ilustración 2: Diagrama de diseño

### **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, métodos para agregar elementos a 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.



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 que numero 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 “lab3\_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 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 y el IDE Apache Netbeans.

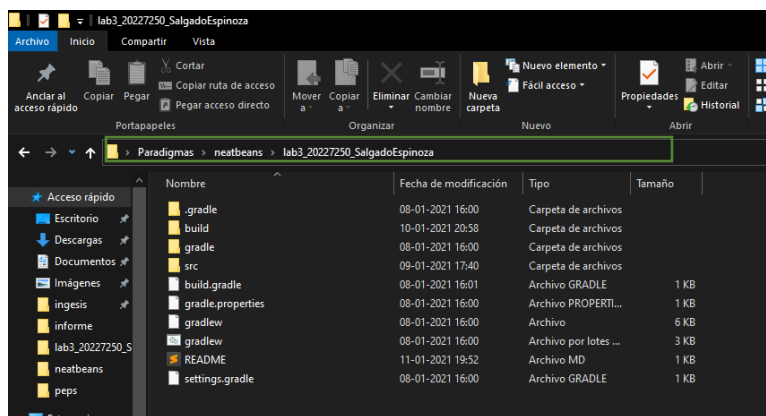
Al momento de realizar los commits se han subido todos los archivos que contenía el proyecto de Apache Netbeans a GitHub por si se me llegaba a borrar alguno sin querer. Esta nube es muy importante, ya que, si se nos llega a borrar algo por X motivo nuestro código o avances que estemos realizando, estarán resguardados, ya que se guardan todos los cambios que se han realizado. Gracias a ir subiendo commits pude rescatar parte de mi proyecto, producto de que al inicio el IDE Netbeans me daba muchos errores, cambie a Eclipse funciono bien un tiempo, subi commits y luego me daba errores, no podía compilar porque faltaba una carpeta. Le pregunte a mi profesor que podía pasar con Netbeans, me explico y solucione la falla, entonces descargue lo que había hecho en Eclipse y lo adapte a Netbeans.

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

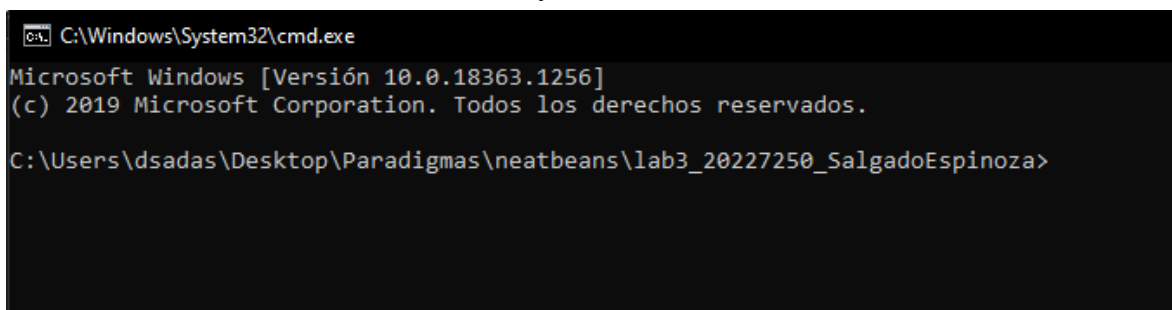
A continuación, se va a presentar como se debe compilar el proyecto en la consola/terminal de Windows, cabe destacar que también fue probado en Linux.

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



*Ilustración 3: Abrir carpeta del proyecto*

**Paso2:** Debe seleccionar el cuadro verde y escribir cmd

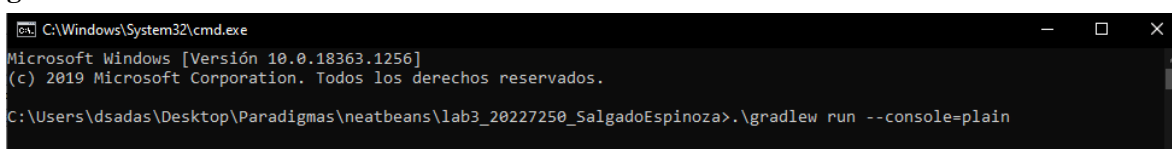


*Ilustración 4: Ingresar a consola*

**Paso3:** Se abre la consola y luego se debe escribir uno de los siguientes comandos. De preferencia el primero, pero cualquiera de los dos funciona.

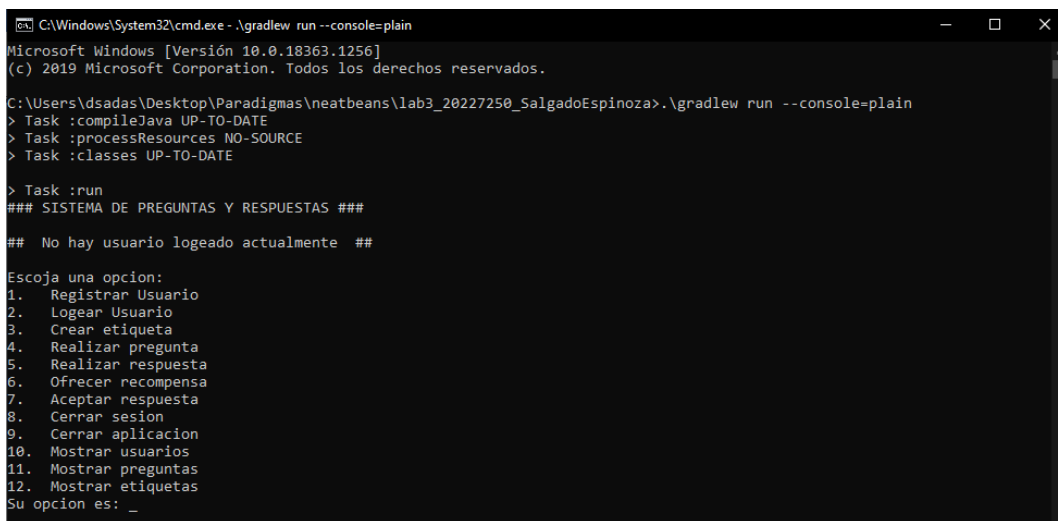
**.\gradlew run --console=plain**

**gradlew.bat run**



*Ilustración 5: Escribir comando para compilar*

**Paso4:** Se muestra el menú del programa y puedes empezar a trabajar.



*Ilustración 6: Menú para trabajar.*

### Explicación de opciones:

- 1. Registrar Usuario:** 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: Ya existe usuario registrado con ese nombre, ¿desea continuar? S/N. Si ingresa S podrá volver a intentar registrar al usuario, si ingresa N va a volver al menú.
- 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 en pantalla diciendo el error que ha tenido.

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 ya existe una etiqueta 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.
  8. **Cerrar Sesión:** Esta opción permite a un usuario que esta logeado poder deslogearse. Se muestra en pantalla que se ha cerrado sesión.
  9. **Cerrar Aplicación:** Esta opción permite cerrar la aplicación. Se cierra el menú.
  10. **Mostrar Usuarios:** Esta opción permite ver a cualquier usuario ver los nombres y reputación de todos los usuarios registrados.
  11. **Mostrar Preguntas:** Esta opción permite visualizar todas las preguntas que se encuentran contenidas en el stack.
  12. **Mostrar Etiquetas:** Esta opción permite visualizar todas las etiquetas y su descripción.
- X>12. Opción que no esta en menú:** Si se ingresa un entero mayor a 12, muestra un mensaje diciendo que ingrese opción válida, si se ingresa un tipo String muestra que algo salió mal.

### **Posibles errores**

Hasta el momento no se han encontrados errores, he tratado de desbugear 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 por el motivo de que tenía fallas en los obligatorios, se ha preferido realizar las funcionalidades obligatorias y que funcionen bien, en vez de realizar los opcionales teniendo fallas en los obligatorios. 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 el lenguaje señalado NetBeans 12 y OpenJDK versión 11.
Interacciones con el programa mediante el terminal.	Funciona correctamente
La solución debe demostrar la aplicación del paradigma orientado a objetos.	Se aplican los conocimientos aprendidos en clases de orientación a objetos.
Se deben establecer los prerrequisitos para cada funcionalidad.	Se han seguido todos los prerrequisitos para una buena implementación del programa.
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 tomando en consideración la evolución en el desarrollo de su proyecto en distintas etapas. Se requieren <b>al menos 10 commits</b> distribuidos en un periodo de tiempo <b>mayor o igual a dos semanas</b>	Se cumple con el historial.

**Requerimientos funcionales:**

Clases y estructuras que forman el programa	Se cumple con las clases y estructuras solicitadas.
Menú interactivo por terminal	Se cumple con el menú interactivo.
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:</b> <b>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.	<b>Se logra implementar</b>

## 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 este laboratorio no tuve muchas dificultades para llevarlo a cabo, producto de que era muy similar a un lenguaje que he utilizado durante un año, el cual corresponde a C. Asimismo, tuve que estar revisando videos y preguntando en clases, por efecto de ser la primera vez que estoy programando con orientación a objetos.

El código creado se puede mejorar considerablemente para que sea mucho más sencillo y ordenado, pero, para eso voy a necesitar más práctica y tiempo. Además, estas semanas largas pasaron la cuenta, si bien, eran días libres no me encontraba en mi casa para avanzar del todo.

En el ámbito de resultados, se pudo haber llegado a uno mucho mejor realizado, pero por temas de tiempo y mal organización no fue posible de esta manera. No obstante, se llega a una implementación parcial, pudiendo 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.

Por último, si bien, el laboratorio 3 fue creado en Java me resultó menos difícil adaptarme al lenguaje, a diferencia de los paradigmas anteriores, 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/>