



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

**LABORATORIO 3**

**PARADIGMAS DE PROGRAMACIÓN**

Alberto Rodríguez Z.

Profesores:	Roberto González
	Daniel Gacitúa
	Víctor Flores
Fecha de Entrega:	10-08-2018

Santiago de Chile

1- 2018

# **TABLA DE CONTENIDOS**

<b>CAPÍTULO 1. INTRODUCCIÓN.....</b>	<b>4</b>
<b>CAPÍTULO 2. MARCO TEÓRICO.....</b>	<b>5</b>
<b>2.1. PARADIGMA ORIENTADO A OBJETOS.....</b>	<b>5</b>
<b>CAPÍTULO 3. DESCRIPCIÓN DEL PROBLEMA.....</b>	<b>6</b>
<b>3.1. ANÁLISIS.....</b>	<b>6</b>
<b>CAPÍTULO 4. DESCRIPCIÓN DE LA SOLUCIÓN.....</b>	<b>7</b>
<b>CAPÍTULO 5. RESULTADOS.....</b>	<b>10</b>
<b>CAPÍTULO 6. CONCLUSIONES.....</b>	<b>11</b>
<b>REFERENCIAS.....</b>	<b>12</b>

## **TABLA DE FIGURAS**

<b>Figura 1.- Diagrama de las características principales de un objeto.....</b>	<b>5</b>
<b>Figura 2.- Diagrama de clases UML del problema.....</b>	<b>6</b>
<b>Figura 3.- Diagrama de clases UML de la solución.....</b>	<b>9</b>
<b>Figura 4.- Conversación entre el usuario y el chatbot.....</b>	<b>10</b>

# CAPÍTULO 1. INTRODUCCIÓN

El presente informe tiene como objetivo principal ser una referencia a la línea de pensamiento y al contexto de desarrollo del código fuente que lo acompaña para la presentación del laboratorio 3 del curso de Paradigmas de Programación. En esta ocasión se hará uso del paradigma de programación Orientado a Objetos con el lenguaje de programación Java.

En primer lugar, el problema a resolver es la creación e implementación de un chatbot. De este mismo modo, es relevante mencionar que un chatbot se puede definir como programas computacionales que pueden mantener una conversación con un ser humano, sin embargo, esta interacción queda limitada por un contexto/temática y una serie de parámetros. Asimismo, los chatbot resultan ser de utilidad para manejar algunas fases de la conversación como pueden ser saludos de bienvenida, solicitud de datos y responder algunas preguntas específicas por el usuario.

Por consiguiente, en este informe la solución implementada para el chatbot trabaja en base a la temática de atención al cliente de un negocio de comida rápida, en donde se crean flujos conversacionales protocolares, estructuras gramaticales simples y vocabulario conocido por el chatbot. De tal forma dando los parámetros y temática se puede mantener una conversación con el chatbot.

Por lo tanto, se establece como objetivo de este laboratorio, la aplicación de técnicas aprendidas en clases, además de ir mas allá e investigar cuando corresponda, con el fin de crear un algoritmo eficaz que funcione bajo el paradigma orientado a objetos.

## CAPÍTULO 2. MARCO TEÓRICO

### 2.1 PARADIGMA ORIENTADO A OBJETOS

La programación orientada a objetos (POO) es un paradigma que tiene como objetivo el desarrollo computacional basado una colección de objetos que están interrelacionados y trabajan conjuntamente en resolver un problema. Los objetos manipulan los datos de entrada para la obtención de datos de salidas, donde cada objeto tiene distinta funcionalidad.

Uno de los conceptos claves de la POO es la abstracción, que es un proceso de interpretación que implica reconocer y enfocarse en las características importantes de una situación u objeto, deja a un lado los detalles del objeto y definir las características específicas de éste, que los distinguan de los demás objetos, también hay que centrarse en lo que es y lo que hace el objeto, antes de decidir cómo debería ser implementado.

En la POO se trabaja con el concepto de clase, que no es más que un TDA con propiedades adicionales a la hora de implementarse. La clase es un modelo que define las variables y métodos comunes a todos los objetos de cierta clase. Una clase define el estado (atributos) y el comportamiento (métodos) que todos los objetos creados a partir de esa clase tendrán.

Un atributo es una característica de un objeto, que ayuda a definir su estructura y permite diferenciarlo de otros objetos. Se define con un identificador y con un tipo de dato, el cual indica los valores que puede almacenar. El conjunto de valores define el estado actual del objeto.

Los métodos son los comportamientos de un objeto y permite identificar la forma en que actúa respecto a su entorno o respecto a otros objetos. Además, representa una operación o función que un objeto realiza. El conjunto de métodos de un objeto determina el comportamiento general del objeto.



Figura 1.- Diagrama de las características principales de un objeto.

## CAPÍTULO 3. DESCRIPCIÓN DEL PROBLEMA

Se solicita crear un algoritmo en el lenguaje de programación Java que simule la conversación entre el usuario y el chatbot, un chatbot contiene personalidad y un vocabulario. La conversación debe funcionar en base a una temática manteniendo una conversación fluida con un inicio de conversación y un final. La conversación se va almacenando en un registro llamado log, como también una semilla pseudoaleatoria llamada seed que da variabilidad a las respuestas. Al ser POO se trabaja con las siguientes entidades en forma de clase, ya que cada una tiene sus distintas características:

1. **Chatbot**: Clase que interactuará con el usuario respondiendo a sus mensajes. Esta Clase incluya las posibles respuestas del chatbot y la personalidad de este dependiendo de la semilla de entrada seed.
2. **Log**: Clase en la cual se almacena la conversación entre el usuario y el chatbot. Esta además trabaja todo lo relacionado con el log, como guardar datos o mostrar datos de este. Cada mensaje debe incluir la fecha en cuando se llama al objeto de la clase.
3. **Usuario**: Clase que contiene toda la información útil sobre un cierto usuario, esta clase cumple la función como de “memoria” del chatbot ya que almacena los datos de este.

### 3.1 ANÁLISIS

El principal problema radica en la capacidad de darle “sentido” a una conversación, ya que cuando el usuario envía un mensaje, el chatbot debe ser capaz de responderle lógicamente en relación con lo preguntado por este. Por lo tanto, debe existir una gran posibilidad de respuestas para que el chatbot tenga la capacidad de seleccionar la indicada.

De otro modo ya que se trabaja con POO, se debe crear las clases correspondiente y poder relacionarlas entre sí, ósea se instancian objetos de las clases anteriores y se deben establecer relaciones para que una conversación pueda funcionar.

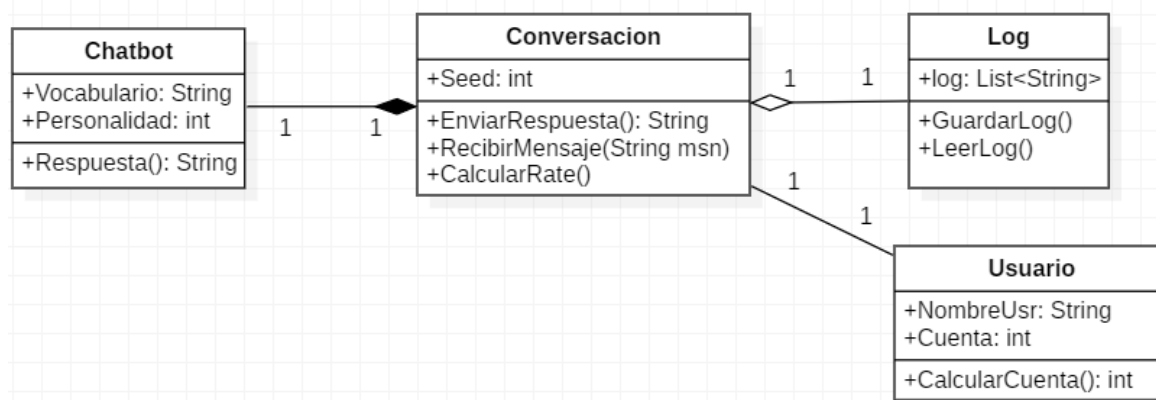


Figura 2.- Diagrama de clases UML del problema.

## CAPÍTULO 4. DESCRIPCIÓN DE LA SOLUCIÓN

Puesto que hay clases ya definidas, además de parámetros mencionados en el enunciado, estos son “!beingDialog”, “!endDialog”, “!saveLog”, “!loadLog”, “!rate”, además se realiza el parámetro extra “!chatbotPerformance”, se debe instanciar objetos de las clases anteriores y relacionarlos para poder desarrollar una conversación.

Primero se crea la clase Principal “Conversacion” la incluye el método “main” en esta es donde se crea toda la conversación entre el usuario y el chatbot y donde se llama a las demás clases, en esta se crean los objetos de las demás clases y donde se relacionan, pero antes de especificar en esta se nombrara los atributos y métodos de las demás clases.

Esta la clase “Chatbot” la cual incluye todas las posibles respuestas de un chatbot respecto a un mensaje específico del usuario, estas respuestas incluyen saludos de bienvenida, despedidas y respuestas al cliente. Estas respuestas son parte de los atributos de la clase, su tipo de dato es un arreglo de String, donde en cada posición del arreglo incluye la respuesta del chatbot según una personalidad determinada por el seed. Otros de sus atributos son los precios de las comidas del restaurant, su tipo de dato entero.

La clase contiene cinco métodos, dos de ellas cumplen la misma función que son los métodos “ObtenerTiempo( )” y “GuardarTiempo( )”, las cuales obtienen el tiempo y la fecha actual en la que se ejecuta el método devolviendo todo en un String, la diferencia entre la dos es que el String que devuelve es distinto, uno se agregar el mensaje del log y el otro para el nombre de un archivo. Otro de los métodos es el de “ObtenerSaludoBienvenida( )” el cual verifica cual es el mensaje correcto que debe enviar el chatbot, según la hora y el seed que se entrega como parámetro, devuelve el mensaje como String. Un método parecido es el llamado “ObtenerDespedida( )”, el cual verifica cual es el mensaje correcto de despedida del chatbot según el seed ingresado, retorna un string con el mensaje.

El método “RespuestaChatbot( )” verifica cual es el mensaje correcto que debe enviar el chatbot según un mensaje enviado por el usuario, este mensaje que es enviado en forma de String se agrega en un arreglo donde en cada posición estará incluida cada palabra del mensaje. Después de esto se compara con cada condicional para ver cual cumple con esta y cuál sería la respuesta más coherente para esta, cuando los mensaje son mas largos se utiliza el método “ComprobarPalabra( )” el cual compara si String determinados se encuentran dentro del arreglo.

La otra clase importantes es la clase “log” la cual se trabaja como un `ArrayList<String>` en la clase “Conversacion” y es donde se ira agregando toda la conversación entre el usuario y el chatbot, al ser `ArrayList` de pueden agregan tantos `String` como sea necesario al ser una lista dinámica. Esta clase no tiene atributos pero si tiene dos métodos los cuales son “`GuardarLog( )`” y “`LeerArchivo( )`”. El primer método tiene la función de crear un archivo de texto y guardar la conversación actual (log), el nombre de este archivo será la fecha en la que es ejecutado el método. El segundo método tiene la función de leer la conversación que se contiene dentro del archivo y continuarla desde ahí, para eso guarda todo en el log y se necesita el nombre del archivo como parámetro, el retorno será mostrar por pantalla la conversación incluida dentro del archivo.

La clase “Usuario” contiene atributos los cuales son el nombre del cliente y la cuenta de los que debe pagar el cliente, ósea el chatbot no solo responde, sino que recuerda lo que le dijo el usuario y va almacenando el valor total de la compra. Tambien se crea un `ArrayList` de esta clase en la cual se van almacenando las evaluaciones del chatbot y el usuario actual, junto con la personalidad específicas. Después de tener ya suficientes conversaciones con distintos usuarios se almacenan todas las evaluaciones, para calcular el promedio la desviación estándar de estas se llama al método de esta clase llamado “`calcularEvaluaciones( )`”.

Como antes mencionado se instancian objetos de las clases anteriores y así se puede desarrollan una conversación con el usuario. La clase “Chatbot” responde a los mensajes, esta es una composición con la clase “Conversacion”, “Log” almacena la conversación y el “Usuario” recuerda los datos del usuario y tambien se puede obtener la nota del usuario y del chatbot en el comando “`!rate`”. Se pueden crear cuantas conversaciones se quiera, hasta que el usuario decida finalizar el programa.



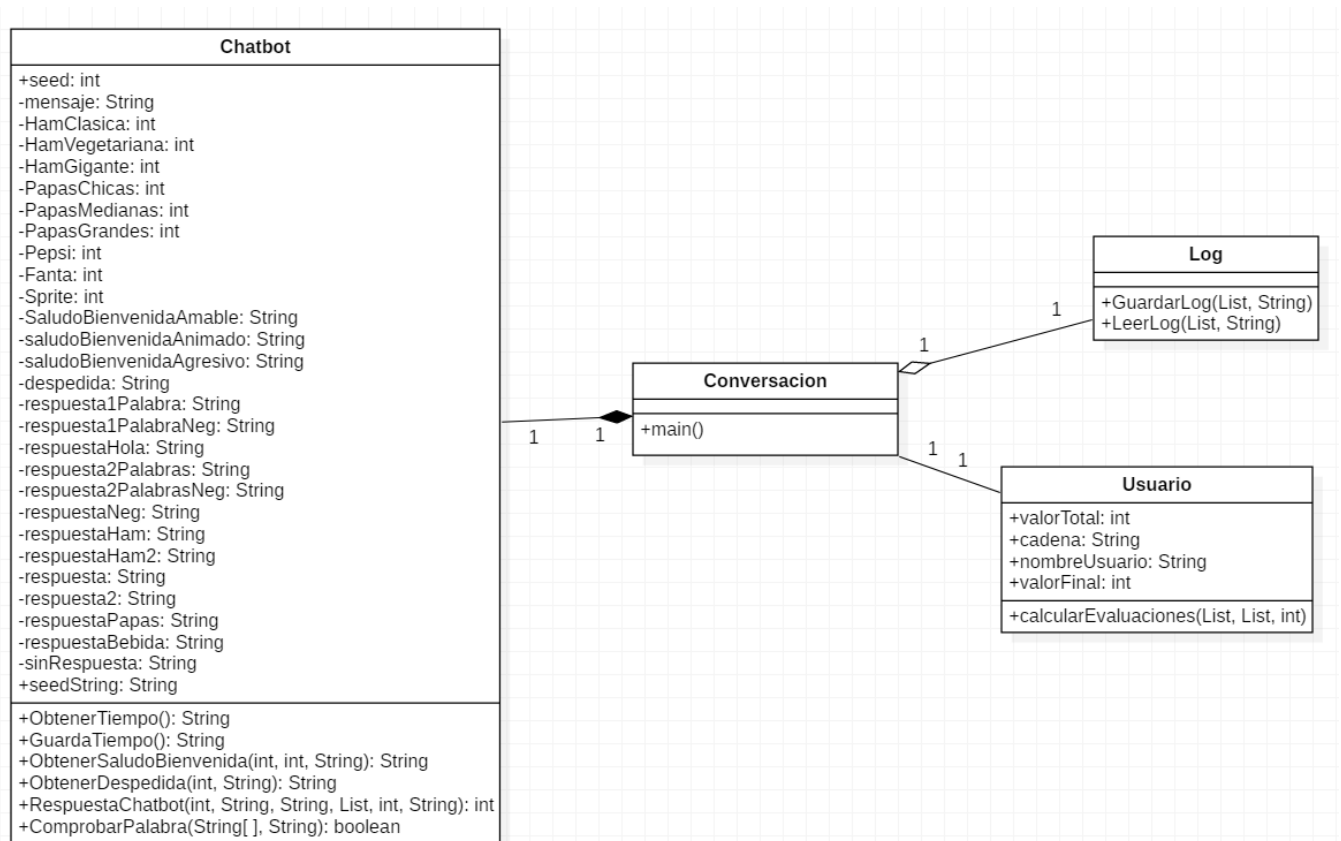


Figura 3.- Diagrama de clases UML de la solución.



## CAPÍTULO 6. CONCLUSIONES

Realizando el laboratorio se pudo concluir que el paradigma de programación Orientada a Objetos tiene las ventajas de ser un paradigma simple de desarrollar, una buena abstracción de las clases, objetos y atributos nos brinda una implementación más detallada puntual y coherente. La reutilización de código es algo que se puede observar en este paradigma, siempre y cuando se desarrollen adecuadamente las clases. La fiabilidad de este paradigma nos permite dividir el problema en partes más pequeñas y probarlas de manera independiente y aislar mucho más fácilmente los posibles errores que pueden surgir.

En comparación a los paradigmas anteriores (Funcional y lógico), el paradigma orientado a objetos se puede trabajar de forma más eficiente, ya que al aislar y probar las clases por separado. Mientras que en los demás paradigmas es más limitando relacionar las funciones o los predicados como se puede hacer con las clases.

En segundo lugar, respecto al problema del laboratorio, ha sido interesante plantear el problema de cómo poder desarrollar una conversación con un chatbot, a través de distintas clases y técnicas que tiene el POO, junto con las implementaciones que contiene la biblioteca estándar de Java. De modo que, se puede afirmar que se realiza un gran trabajo con Java al poder trabajar directamente con las clases y como relacionarlas a través de los objetos.

Finalmente, puesto que se ha logrado desarrollar un chatbot con cierta temática, con el cual uno puede mantener una conversación, es posible concluir que se ha logrado cumplir con el objetivo principal del laboratorio.

## REFERENCIAS

- Laura Álvarez y Helmer Avendaño. *¿Qué es la programación orientada a objetos?* (Colombia) Recuperado de:  
[http://ferestrepoca.github.io/paradigmas-de-programacion/poo/poo\\_teoría/index.html](http://ferestrepoca.github.io/paradigmas-de-programacion/poo/poo_teoría/index.html)
- Depto. De Ingeniería Informática. *Curso de Paradigmas de Programación.* (Chile) Recuperado de: <http://www.udesantia.govvirtual.cl/moodle2/course/view.php?id=7167>