

**Escuela Superior Politécnica del Litoral**

**Programación Orientada a Objetos**

**Sistema diseñado a objetos**

**Grupo: 2**

**Integrantes:**

- **Yanaleen Britney Plúas Macias**
- **Issac Alexander Maza Punine**
- **Christopher Geovanny Tomalá Drew**

**Periodo: 2021 1T**

## Table of Contents

<b>1. Antecedentes .....</b>	<b>3</b>
<b>2. Descripción .....</b>	<b>3</b>
<b>3. Diseño del Sistema .....</b>	<b>3</b>
<b>4. Discusión de la Solución .....</b>	<b>9</b>
<b>5. Evidencias de Participación .....</b>	<b>10</b>
<b>6. Evidencias de ejecución del programa .....</b>	<b>¡Error! Marcador no definido.</b>
<b>7. Referencias Bibliográficas.....</b>	<b>¡Error! Marcador no definido.</b>

## **1. Antecedentes**

Tener un sistema que cumpla la función de gestionar las actividades de una empresa, es indispensables para cualquier empresa hoy en día. En este caso tenemos a una empresa que está dentro del mundo de bienes raíces, que desea implementar un sistema que le permita registrar a sus agentes inmobiliarios, a sus potenciales clientes, así como designar a una persona encargada del uso correcto de este sistema que tendrá la función de administrador, estos permiten una gestión correcta de la información entrante, como la información de las propiedades(casa, terrenos) que se encuentran disponibles para la venta, también a las que ya vendidas. Igualmente permitiéndoles a los potenciales clientes realizar préstamos, así como consultar información de la propiedad de su interés y poder ponerse en contacto con los agentes inmobiliarios para ver si se realiza una compra o no.

Para así poder llegar a una mayor cantidad de clientes posibles, y poder aumentar sus ventas.

## **2. Descripción**

Desarrollo de un sistema orientado a objetos de una compañía que permite a los interesados en adquirir una propiedad consultar la información acerca de las propiedades disponibles, así como simular el pago de las cuotas que registra el sistema de préstamos francés y alemanes. Este sistema ayuda a los clientes de a ponerse en contacto con los agentes-vendedores para realizar consultas y responder inquietudes, operaciones de venta, registro de pagos entre otras funcionalidades.

## **3. Diseño del Sistema**

- Casos de Uso de los usuarios del sistema

### **1. Administrador:**

- Maneja el uso general del sistema
- Inicia Sesión en la plataforma.
- Registra Propiedad.
- Ilustra la información de las propiedades
- registra los agentes.

### **2. Agente:**

- Inicia sesión en la plataforma.
- Registra las ventas realizadas

- Recibe el buzón(ve sus consultas y las responde) .

### **3. Cliente:**

- Consultas propiedades
- Consultas agentes
- Simula el proceso de préstamos.
- Inicia sesión en el sistema.
- Se autorregistra en la plataforma.

Los casos de uso pueden tener uno o varias situaciones.

#### **1. Caso: Cliente**

##### **a) El cliente inicia sesión en la plataforma:**

- **Caso 1:** Ingresa su usuario y contraseña , si los datos son los correctos, se muestra la página de inicio del cliente.
- **Caso 2:** Ingresa los datos, pero estos no son correctos, se muestra un mensaje de error y esto lo regresa al caso 1.

##### **b) El cliente se autorregistra:**

- **Caso 1:** El cliente ingresa datos válidos, se crea un nuevo usuario en la plataforma llamada “Cliente”.
- **Caso 2:** El cliente resulta ser alguien que no cumple los requisitos como por ejemplo ser menor de edad, y muestre un mensaje “No se puede registrar en el sistema” , y lo regresa al caso 1.

##### **c) El cliente consulta información de una propiedad:**

- **Caso 1:** El cliente ingresa el id de la propiedad que desea saber, si el dato egresado en valido muestra la información de la propiedad.
- **Caso 2:** El cliente no ingresa la información correcta, se notifica que el dato es incorrecto, lo regresa al caso 1.
- **Caso c:** El cliente ingresa un id inexistente, se muestra un mensaje que dice “Dato Inexistente” y esto se repite hasta que se dé el caso 1.
- **Caso d:** El cliente no tipea nada, y regresa a la página principal de **Cliente**.

##### **d) El cliente simula un préstamo:**

- **Caso 1:** Ingresa datos correctos.
  - **Caso 1.1:** El usuario selecciona un sistema de amortización francés y realiza una estimación de su préstamo.
  - **Caso 1.2:** El usuario selecciona un sistema de amortización alemán y realiza una estimación de su préstamo.
- **Caso 2:** El usuario ingresa mal los datos y muestra un mensaje de error “Datos incorrectos”, esto se repite hasta que realice el caso 1.
- **Caso 3:** El usuario decide regresar al menú principal.

**e) El cliente decide crear una alerta:**

- **Caso 1:** Ingresa sus preferencias para que este sea notificado, si los datos son correctos se crea una alerta.
- **Caso 2:** Ingresa sus preferencias para que este sea notificado, si los datos llanamente son incorrectos, se divisará un mensaje de error “Datos incorrectos”. Esto se repite hasta que el usuario realice el caso 1.
- **Caso 3:** Decide regresar a la página principal.

**2. Caso: Agente de ventas.**

**a) El agente de venta inicia sesión en el sistema:**

- **Caso a:** Ingresa sus credenciales(usuario y contraseña) y son correctos, se muestra la página de inicio de Agentes en la plataforma .
- **Caso b:** Ingresa sus credenciales y no son válidos, regresa a la pagina principal sin iniciar sesión.

**b) El agente registra una venta al sistema:**

- **Caso 1:** Ingresa los datos correctos de la venta y los registra.
- **Caso 2:** Ingresa datos incorrectos, muestra un mensaje de error “Dato/s incorrecto/s”, esto se repite hasta que se dé el caso a.
- **Caso 3:** Decide volver a la página principal.

**c) El agente revisa su Buzón de mensajes:**

- **Caso 1:** No tiene consultas por responder, muestra un mensaje “El buzón esta vacío” .
- **Caso 2:** Obtiene consultas.

- **Caso 2.2:** Responde a la pregunta y le envía al cliente una respuesta.
- **Caso 2.2:** Decide no responder a la pregunta.
- **Caso 3:** Regresa a la página principal de Agentes.

### **3. Caso: Administrador.**

#### **a) El agente de venta inicia sesión en el sistema:**

- **Caso 1:** Ingresa sus credenciales(usuario y contraseña) y son correctos, se muestra la página de Administrador en la plataforma.
- **Caso 2:** Ingresa sus credenciales y no son válidos, regresa a la página principal sin iniciar sesión.

#### **b) El agente registra una propiedad en sistema:**

- **Caso 1:** Ingresa los datos correctos de la propiedad.
  - **Caso 1.1:** Si es un terreno debería ingresar el tipo(Comercial, Vivienda o Empresarial).
    - **Caso 1.1.1 :** Si es correcto el dato, se le asignara a un agente en ventas.
    - **Caso 1.1.2 :** Si no es correcto, se mostrará una imagen diciendo “Dato erróneo”. Lo cual lo regresara al caso 1.1.
  - **Caso 1.2:** Si es una casa, se deberá ingresar el número de pisos, habitaciones.
    - **Caso 1.2.1 :** Si es correcto el dato, se le asignara a un agente en ventas.
    - **Caso 1.2.2 :** Si no es correcto, se mostrará una imagen diciendo “Dato erróneo”. Lo cual lo regresara al caso 1.2.
- **Caso 2:** Ingresa datos incorrectos, muestra un mensaje de error “Dato/s incorrecto/s”, y vuelve a pedir datos.
- **Caso 3:** Decide regresar al menú principal para Administrador.

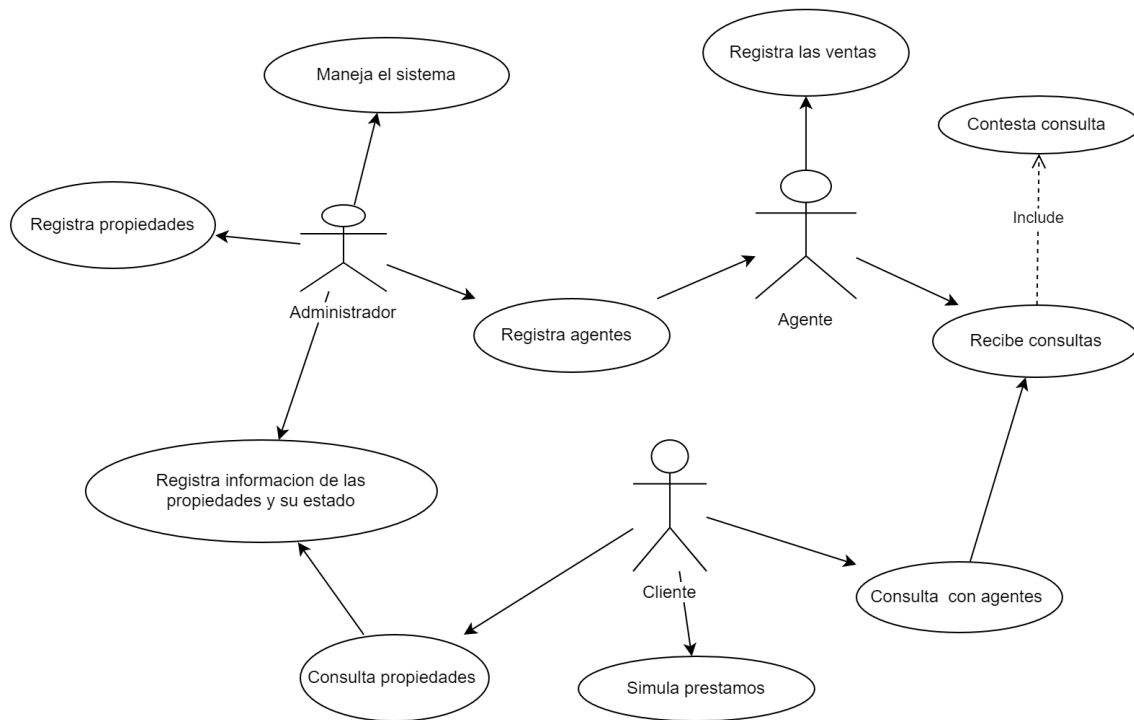
#### **c) El administrador registra a un agente al sistema:**

- **Caso 1:** Ingresa los datos correctos del agente, se registra un agente más.
- **Caso 2:** Ingresa daros incorrectos al sistema, se muestra un mensaje de error “datos incorrectos”, y se vuelve a pedir los datos .
- **Caso 3:** Decide no ingresar nada, regresa al menú inicio de Administrador.

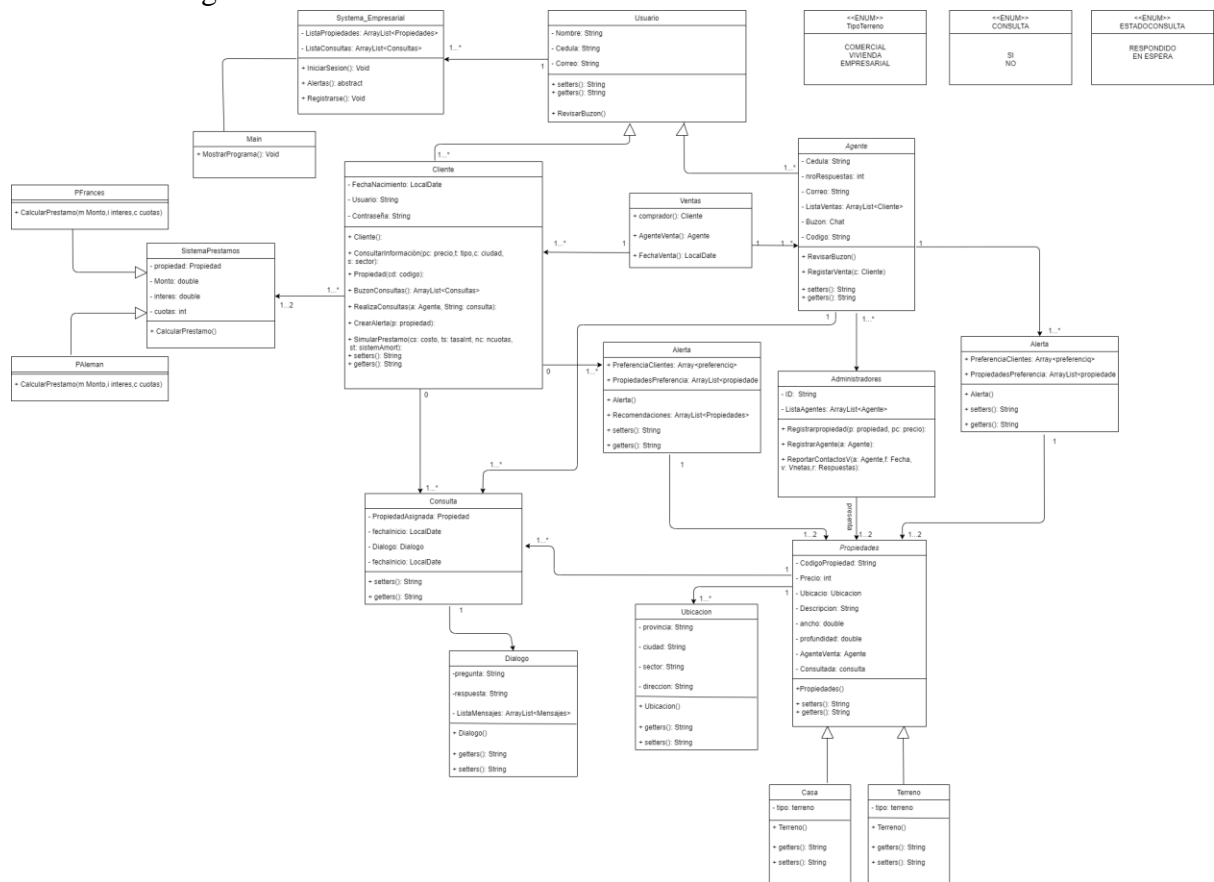
**d) El agente realiza un reporte de los contactos y ventas:**

- **Caso 1:** Ingresa un rango de fecha(de donde, hasta donde) para filtrar las ventas, el ingreso es correcto, se muestra el reporte de ventas que previamente han sido filtradas.
- **Caso 2:** Ingresa un rango de fecha(de donde, hasta donde) para filtrar las ventas, el ingreso no es correcto, se muestra una notificación diciendo “Datos ingresados incorrectos” y volverá a pedir datos.
- **Caso 3:** Regresa a la página principal de Administrador.

- Diagrama de casos de usos



- Diagrama de clases





- Breve descripción de la responsabilidad de las principales clases.

Paquete	Clase	Responsabilidad
Usuario	Administrador	Los administradores son aquellos que registran las propiedades con su respectivo precio, recibe los nombres de los agentes y los registra en una lista y reportan los contactos con sus respectivas ventas.
Usuario	Cliente	El cliente es aquel que consulta la información de las propiedades (precio, tipo, ciudad y sector), además realiza consultas a los agentes, y simula prestamos estos pueden ser de estilo francés y alemán.
Usuario	Agente	El agente es aquel que revisa las consulta de los clientes y las contesta, además de eso registra las ventas realizadas.
Propiedades	Propiedades	En Esta clase se registran todas las propiedades que están en ventas con sus respectivos datos.

#### 4. Discusión de la Solución

El programa se realizo con el fin de resolver una problemática cotidiana, en la que se realizan ventas de propiedades inmobiliarias. Las ventas se generan con actitud e interacción, y nuestro programa propone interactuar de forma funcional y eficaz para los clientes. Aplicamos los conceptos de:

- I. Abstracción  
Las clases captan toda la esencia de proyecto, estas recopilan y omiten detalles innecesarios para nosotros que solamente mostramos lo que en si es relevante. Las mayorías de abstracciones que realizamos fue por bloques de código para que el problema sea algo sencillo de manejar, ya que cada clase y métodos tienen sus respectivas funciones en todo el programa.
- II. Encapsulamiento  
Fue primordial y critico el uso de encapsulamientos en la interfaz y comandos de cada clase para modelar las entidades revelantes, ya que controlábamos lo que podíamos ver y utilizar los distintos módulos internos de nuestro sistema. Usamos modificadores de acceso; privados y públicos.
- III. Herencia  
Realizamos un sistema que permite al sistema trabajar como un solo algoritmo entre distintas clases. Utilizamos las clases de base y las derivadas, las importadas y las de superposición. Usamos el método para calcular valores correspondientes de otras clases derivándolas con el *import*, *super()*, etc. También hicimos uso de clases abstractas y enum, que sirven como clases derivadas o de implemento para un caso específico, sin una clase base.
- IV. Polimorfismo  
El software puede usarse sin problemas, debido a que todos los objetos trabajados fueron de tipo primitivo, puede usarse libremente y las veces que sea necesario. Cuando hablamos de herencia, también hablábamos de polimorfismo, cuando

recibíamos un parámetro de una clase base. El sistema que diseñamos se basara en acciones e interacciones que tengan esos métodos para desernir el problema.

## 5. Evidencias de Participación

### Distribución de Tareas

Nombre	Tarea
<b>Yanaleen Britney Pluas Macias</b>	Diagrama de casos de usos, diagrama de clases, documento
<b>Issac Alexander Maza Punine</b>	Diagrama de casos de usos, diagrama de clases, documento
<b>Christopher Geovanny Tomalá Drew</b>	Diagrama de casos de usos, diagrama de clases, documento

### Evidencias de contribución en el repositorio

- Link del repositorio:
- <https://github.com/ChristopherTomala02/PROYECTO-1P>
- Capturas de pantalla de al menos dos commits hechos por cada integrante del proyecto, de los integrantes del proyecto y grafico de contribución (esto lo puede sacar de la Sección Insigths -> contributors).

