**Escuela Superior Politécnica del Litoral**

**Programación Orientada a Objetos**

**Proyecto Primer Parcial**

**Objetivos:**

Desarrollar en los estudiantes las habilidades de la programación orientada a objetos básica incluyendo la aplicación de los conceptos de manejo de clases y sus relaciones, paquetes, atributos y métodos de instancia y de clase, manejo de listas, visibilidad de atributos y métodos, entrada por teclado y salida por consola

**Descripción:**

Una compañía de bienes raíces le ha pedido que desarrolle un sistema que permita a los interesados en adquirir una propiedad consultar la información de las propiedades disponibles, así como simular el pago de las cuotas basados en el sistema de prestamos francés y alemán.

La compañía de bienes raíces le pide además que exista una opción que le permita a los clientes ponerse en contacto con los agentes vendedores para responder sus inquietudes.

El sistema tiene tres tipos de usuarios

* Clientes: son los usuarios que consultan la información de las propiedades disponibles, simulan prestamos y realizan consultas a los agentes mobiliarios
* Agentes de ventas: son usuarios que contesten las preguntas de los clientes y registran las ventas realizadas.
* Administradores: son los usuarios que manejan el sistema. Ellos deben ser capaces de registrar la información de las propiedades y agentes.

A continuación, más detalles de las diferentes opciones:

**Menú Inicial**

Al iniciar un programa se muestra un menú numérico con dos opciones.

1. Iniciar Sesión
2. Registrarse

La **opción 1** pide el usuario y contraseña del usuario. Si existe un usuario en el sistema con el usuario y contraseña dado se muestra un menú numérico de acuerdo a sus credenciales

La **opción 2** permite a un cliente auto-registrarse. Para el registro del cliente se pide nombre, cedula, correo, fecha de nacimiento. No se pueden registrar personas menores de 18 años.

**Opciones Administrador**

El administrador debe poseer las siguientes opciones:

1. Registrar Propiedad

En esta opción el administrador debe poder registrar las propiedades a las ventas. La agencia tiene a la venta Terrenos y casas. De cada propiedad se debe conocer su precio, sus metros de ancho y profundidad, así como su ubicación (provincia, ciudad, direccion, sector). En el caso de los terrenos se debe conocer si el mismo es de tipo COMERCIAL, VIVIENDA O EMPRESARIAL. En el caso de las casas se necesita conocer el número de pisos y habitaciones. Al crear una propiedad la misma se asigna a un agente de Ventas. Este agente será el encargado de responder cualquier dudad de la propiedad.

1. Registrar Agente

En esta opción el administrador registra la información de los agentes vendedores. De los agentes se conoce su nombre, cedula y correo electrónico.

1. Reporte Contactos y Ventas

Esta opción permite al administrador ver un reporte general de los contactos de los clientes y las ventas realizadas por cada agente. Se debe poder filtrar el reporte por rango de fechas. A continuación ejemplo de ejecución:

|  |  |  |
| --- | --- | --- |
| **Ingrese Rango Fechas:** | **19/07/2021 - 21-07-2021** |  |
| **Agente** | **Numero Ventas** | **Numero de respuestas** |
| 34 | 2 | 6 |
| … | …. | … |
|  |  |  |
| Ingrese código del agente quiere mas detalles o vacio para regresar: 34 | | |
| Ventas | | |
| … | | |
| Consultas | | |
| … | | |

**Opciones clientes**

1. Consultar Propiedades

En esta opción muestra al cliente **las propiedades a la venta** permitiéndole filtrar por precio, tipo (terreno o vivienda), ciudad, sector. Si alguno de los filtros nos es pasado no se considera ese criterio al momento de hacer la búsqueda.

Después de mostrar los resultados que coinciden con la búsqueda el sistema debe permitir ingresar el código de la propiedad de la que se quiere mas detalle. Al ingresar el código se muestra toda la información de la propiedad.

Finalmente se pide al usuario si desea realizar una consulta de la propiedad. Si ingresa que si se le debe permitir al usuario ingresar su consulta y registrarla

**Ejemplo de ejecución:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tipo: | Terreno |  |  |  |  |
| Rango Precio: | 20000-40000 |  |  |  |  |
| Ciudad: | Guayaquil |  |  |  |  |
| Sector: | Norte |  |  |  |  |
|  |  |  |  |  |  |
| código | descripción | precio | Tamaño | ubicación | consultada |
| 34 | Bonito terreno en calle comercial | 30000 | 300 m2 | Guayas, Guayaquil, Norte, Cdla Kennedy norte 123 | NO |
| 76 | Casa al valle | 70000 | 180 m2 | ….. | SI |

Ingrese el código de la propiedad desea mas detalle (o vacío para regresar): 34

Detalles de la propiedad:

….

…

Desea realizar consulta (si/no): si

Ingrese su consulta: Hola aceptan prestamos del Biess?

**Nota:**

* No se debe mostrar ninguna propiedad que ya esta marcada como vendida.
* El campo Consultada muestra si en caso que el cliente haya hecho alguna pregunta de la propiedad.

1. **Buzón de Consultas**

Al seleccionar esta opcion el cliente ve una lista de todas las consultas que ha realizado. Al ingresar el código de una de las propiedades se ve la conversación que ha tenido con el agente que esta atendiendo su requerimiento.

A continuación, se debe permitir contestar y continuar con la conversación o regresar al menú anterior

Ejemplo de ejecución:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fecha Inicio** | **Código propiedad** | **Nombre Agente** | **Pregunta** | **Estado** |
| 19-05-2021 | 34 | Maria | Hola aceptan prestamos del Biess | Respondido |
| 19-05-2021 | 72 | Maria | Hola aceptan prestamos del Biess | Esperando |
|  |  |  |  |  |
|  |  |  |  |  |

Ingrese código de propiedad (o vacío para regresar): 34

Conversación:

19-05-2021: Cliente: Hola aceptan prestamos del Biess

20-05-2021: Agente: Si aceptamos y prestamos hipotecario también

Desea realizar agregar un pregunta o regresar (si/no): si

Ingrese la pregunta

Cuantos anios de uso tiene la casa?

NOTA:

* En la conversación se debe mostrar todas las preguntas y respuestas que se ha llevado a cabo en la consulta.

1. **Crear Alerta**

Esta opción le debe permitir al usuario crear una alerta basada en sus preferencias. Cada vez que se registre una nueva propiedad en el sistema que cumpla con los criterios de búsqueda del cliente se debe mandar un correo con la información de esa propiedad.

1. **Simular préstamo**

Esta opción permite al usuario calcular el valor de las cuotas mensuales de un inmueble en base de su costo, tasa de interés y numero de cuotas y el sistema de amortización seleccionado: francés o alemán. En los enlaces a continuación están los detalles de como hacer el cálculos de las cuotas para cada sistema.

* <https://www.econ.unicen.edu.ar/graduados/index.php/sistema-frances>
* https://economipedia.com/definiciones/sistema-de-amortizacion-frances.html
* <https://www.econ.unicen.edu.ar/graduados/index.php/sistema-aleman>
* https://economipedia.com/definiciones/sistema-de-amortizacion-aleman.html

**Opciones agentes**

1. Revisar buzón

En esta opción al agente deben presentar un listado de todas las consultas hechas por clientes para las propiedades asignadas a él. También debe permitirle responder a los clientes

1. Registrar Venta

Esta opción debe permitir al agente registrar una venta realizad por él. Se debe guardar los datos del cliente y la fecha de la venta

**REQUERIMIENTOS ADICIONALES**

* Al iniciarse el programa este debe crear internamente al menos dos agentes de ventas, un cliente y 4 propiedades: dos terrenos y dos casas. Dos de estas propiedades deben estar en la misma ciudad y sector.
* Debe existir una clase abstracta CalculdoraPrestramo con un método abstracto calcuarPrestamo que tenga los datos necesarios para simular simular un préstamo. Debe existir un clase CalculdoraPrestramoFrances y CalculdoraPrestramoAleman que implemente el método calcuarPrestamo
* Para el manejo de las fechas use las clases LocalDate o LocalDateTime
* El programa en realidad debe mandar el correo electrónico.

**CONSIDERACIONES GENERALES:**

* El proyecto debe ser realizado en grupo de 3 estudiantes.
* Se deben aplicar los Principios de Programación Orientada a Objetos en el diseño de la solución.
* El desarrollo de la solución debe ser en Java 11.
* Realice un diseño en capas, separar la capa de presentación de la capa del modelo.
* Agrupe sus clases en paquetes
* El sistema debe realizar las validaciones respectivas para no permitir el ingreso de tipo de datos que no corresponde con los valores posibles y que no se caiga.
* Si el programa se cae en medio de la revisión este será calificado sobre la mitad de los puntos.
* Todas las opciones del menú deberían tener la posibilidad de regresar hacia atrás.
* Todos los grupos deben usar un sistema de control de versiones para compartir su código con los miembros de su equipo. Use la plataforma GITHUB para guardar el repositorio de sus proyecto.
* Incluir un archivo README con las credenciales de los usuarios del sistema y las instrucciones que usted crea necesarias para poder usar el sistema.
* Todos los métodos deben estar documentados usando la notación de javadoc.

**ENTREGABLES DEL PROYECTO:**

* .zip con las fuentes de su Proyecto con el nombre proyecto1\_grupoNUMEROGRUPO.
* .jar de su proyecto con el nombre proyecto1\_grupoNUMEROGRUPO.
* .pdf con el informe del proyecto (se envía modelo del informe con lo que debe tener).

Javadoc del proyecto

**PUNTAJE DEL PROYECTO**

* El proyecto tiene un puntaje de **25 puntos** del componente teórico del primer parcial de la materia.

Su nota final se calculará de la siguiente forma:

**Puntaje\_Final = Punto\_Proyecto X Sustentacion (0-1) X Participacion (0-1)**

* **Evaluación Participación**

La participación se evaluará en base a lo que cada miembro del grupo contribuya a al repositorio de su proyecto.

**RÚBRICA DEL PROYECTO**

**Funcionalidad (50 %)**

* Se valorará la apariencia del proyecto, sin que esto implique que el estudiante deba utilizar recursos adicionales a los aprendidos en clase. (El proyecto es en consola)
* Se verificará el cumplimiento de toda la funcionalidad requerida, así como las correctas validaciones de los datos.
* El proyecto debe ser debidamente probado antes de presentarlo al profesor. Por lo tanto Se espera que el programa no se caiga al ejecutarlo, ni tenga un comportamiento no esperado (Si se cae en la presentación, son 5 puntos menos por cada caída)

**Documentación (10 %)**

* Javadoc del proyecto generado.
* Documentación interna del proyecto.

**Abstracción y Uso de Objetos (20 %)**

* Creación de Clases pertinentes con sus propiedades y métodos
* Correcta interacción de objetos. Que los objetos se comuniquen entre si y no solo sean llamados todos en el main.
* Usar recursos de herencia e interfaces para la solución.

**Modularidad y Encapsulamiento (20 %)**

* Dividir el problema usando los métodos correspondientes. No escribir bloques inmensos de código.
* Encapsular correctamente el proyecto. Crear paquetes donde se agrupen clases que se relacionen.
* No olvidar que para acceder a las propiedades de las clases se debe proveer los métodos get y set.

**Penalidades**

* **NO SE REVISARÁ PROYECTOS QUE NO CUMPLAN CON EL 60% DE LA FUNCIONALIDAD**
* 10 porciento menos por no seguir buenas prácticas de programación (convención para nombre de clases, nombre de atributos y métodos, falta de documentación de funciones usando Java Doc, uso de constantes cuando es necesario, o algún fallo grave que se detecte en la forma de programar).