**Escuela Superior Politécnica del Litoral**

**Programación Orientada a Objetos**

**Proyecto primer parcial**

**2021– 1T**

**Objetivos:**

* Diseñar una solución reconociendo los objetos participantes en un problema y la relación entre estos.
* Implementar una solución en un lenguaje de programación orientado a objetos, aprovechando la interfaz gráfica (En JavaFX) y aplicando el concepto de hilos
* Fortalecer el auto aprendizaje y la investigación.

**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:**

En la última década se ha incrementado el número de ciudadelas privadas en el país. Para ingresar al interior de estas ciudadelas residentes y visitantes deben pasar por los puestos de seguridad a las entradas de estas. Por lo general estas ciudadelas tienen dos puntos de ingresos: uno para visitantes y otro para residentes. Cuando un visitante quiere ingresar los guardias toman los datos del visitante y del residente al que visita (nombre, mz y villa) y proceden a confirmar el acceso con el residente a través de llamadas o ir a la casa del residente, este proceso puede demorar dependiendo del tiempo de respuesta del residente lo que causa filas al ingreso de las ciudadelas cuando hay muchos visitantes.

Un grupo de inversionistas ha notado la necesidad que existe en el mercado de automatizar este proceso de control de ingreso de residentes y visitantes al Interior de las urbanizaciones privadas y lo ha contratado a usted y su equipo para desarrollar un sistema que permita lo siguiente:

* La automatización del proceso de ingreso a residentes y visitantes a la entrada de urbanizaciones privadas.
* El registro de residentes por parte de un administrador de la ciudadela.
* El registro de visitantes por parte de los residentes de las ciudadelas.
* La generación de reporte de visitas para los administradores de las ciudadelas.
* La autenticación de los usuarios para el uso del sistema basado en su usuario y contraseña.

El sistema que usted y su equipo desarrollará consistirá en dos interfaces:

* Una interfaz desde la cual el administrador pueda registrar a los residentes y lo residentes puede registrar sus visitas.
* Una interfaz que permita simular el ingreso de residentes y visitantes a la ciudadel.

Graphical user interface, text

Description automatically generated

**Interfaz Usuarios**

**Inicio de sesión**

Para poder usar la plataforma los usuarios deben iniciar sesión con su nombre de usuario y contraseña (los nombres de usuario son únicos). Si las credenciales ingresadas son válidas se debe mostrar un menú numérico con las opciones que este puede realizar.

El sistema debe soportar dos tipos de usuarios: administrador de ciudadela y residente. Cada uno podrá realizar diferentes acciones.

**Opciones Administrador**

**Registro de residentes**

El administrador de la ciudadela es el encargado de registrar a los residentes de la ciudadela en el sistema. El administrador puede ver el mapa de la ciudadela y registrar a los residentes de casa en la ciudadela.

Las casas del mapa se deben cargar desde un archivo que guarda su ubicación en el mapa (coordenada x,y), manzana y villa.

A picture containing toy

Description automatically generated

Al pasa el cursor sobre una villa sale la información de su ubicación y residente:

Diagram

Description automatically generated

Para registrar un residente se da click en la villa y se guardan los datos del mismo: nombre, correo, género. **Cada casa sólo puede tener un residente asociado**. **Si la casa ya tiene un residente en ella no debe permitir registrar otro**.

Si el residente pudo ser creado, se le generará un usuario y contraseña para acceso al sistema que será enviado a su correo. Cada residente además tiene un pin de acceso de 4 dígitos que se requerirá para permitir su acceso a la ciudadela y será enviado también en el correo.

**Reportes**

En esta sección se presenta un panal donde el usuario puede visualizar información estadística acerca del ingreso y salida de visitantes a la ciudadela en un periodo de tiempo. Usted debe generar al menos 2 gráficos estadísticos.

Ejemplos:

* Gráfico de barra con el número de ingresos por hora para residentes y visitantes
* Ingresos residentes vs ingresos visitantes

**Opciones Residente**

**Ver información personal y cambiar pin**

El residente debe ser capaz de ver su información personal: (nombre, correo, manzana, villa) y su pin. En esta pantalla también el residente debe ser capaz de cambiar su pin)

**Registrar Vehículo**

Un residente puede tener uno o más vehículos. El residente debe registrar sus vehículos en el sistema para que estos pueden ingresar por el acceso de residentes. De cada vehículo se necesita guardar el número de matrícula del carro y el nombre del propietario.

**Registro de visitantes**

Para registrar un visitante el residente debe ingresar la siguiente información del visitante: nombre, número de cédula, correo y fecha en la que se hará su ingreso. Una vez hecho el registro el sistema generará un código de acceso único de 8 caracteres que será enviado al correo del visitante.

El código generado tendrá una validez de 10 minutos (5 minutos antes de la supuesta fecha de ingreso y 5 minutos después de la supuesta fecha de ingreso) y sólo puede ser usado una vez para ingresar.

Además, los residentes deben ser capaces de ver el listado de visitantes registrados activos y de requerirlo eliminar el registro de un visitante (siempre y cuando este no haya ya ingresado a la ciudadela).

Graphical user interface, table

Description automatically generated

Registrar Visita

F Fin

F Inicio

Buscar

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre Visita** | **Cedula Vista** | **Código Acceso** | **Fecha** | **Estado** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Para mostrar la información de las visitas use el componente TableView**

**Interfaz Simulación**

**Existe dos puntos de acceso**

Graphical user interface, text

Description automatically generated

**En el punto de acceso de residentes se procederá de la siguiente manera:**

El sistema usará las cámaras en los brazos para detectar si la entidad que pretende ingresar es un peatón o un vehículo (esto lo simulamos preguntado por teclado si es un peatón o vehículo) y en caso de ser peatón o vehículo se realizará lo siguiente:

* **Vehículo**

Las cámaras leerán el número de matrícula del vehículo (esto lo simulamos pidiendo el ingreso por teclado del número de matrícula del vehículo). Si el número de matrícula leído corresponde al número de matrícula de alguno de los vehículos registrados el sistema el brazo procederá a autorizar el ingreso (se muestra un mensaje diciendo acceso concedido).

* **Peatón**

El sistema pedirá el ingreso del número de cédula y pin de acceso de cuatro dígitos del residente. Si existe un residente con los datos ingresados el sistema procederá a autorizar el ingreso (se muestra un mensaje diciendo acceso concedido), caso contrario se niega el acceso (se muestra un mensaje diciendo acceso denegado) y se procede a solicitar nuevamente los datos

A picture containing graphical user interface

Description automatically generated

**En el punto de acceso de visitantes se procederá de la siguiente manera:**

El sistema pregunta al usuario si tiene un código de acceso y de acuerdo con la respuesta del usuario se realiza una de las siguientes acciones

* **Si tiene código de acceso**

El sistema solicita el código de acceso de 8 caracteres enviado al visitante cuando fue registrada su visita en el sistema por parte del residente. Si el código de acceso ingresado es válido (fue creado en los últimos 10 minutos y no ha sido usado) se permite el ingreso del visitante (se muestra un mensaje diciendo acceso concedido). En caso de no ser válido se niega su ingreso (se muestra un mensaje diciendo acceso denegado) y se vuelve a solicitar el código.

* **Si no tiene código de acceso**

Se procede a solicitar el nombre del visitante, número de cédula, el nombre del residente que visita, junto con la mz y villa del residente. A continuación, el sistema envía un correo al residente con los datos de la visita y un código de acceso de 8 caracteres el cual el residente debe enviar al visitante para ser ingresado en el sistema y permitir su ingreso. En caso de no existir un residente con los datos ingresados se debe volver al solicitar la información del al visitante. Una vez enviado el correo el sistema debe solicitar el código de acceso al visitante.

**REQUERIMIENTOS FINALES**

1. Usted debe haber definido un su sistema una función llamada inicializarSistema() que cree al 5 casas, dos residentes y un administrador.
2. El **administrador del sistema** debe tener las siguientes credenciales
   * usuario: **admin**
   * contraseña: **adminn**
3. Para las opciones del administrador y del cliente debe usar el componente de JavaFX **Menu**

<https://docs.oracle.com/javafx/2/ui_controls/menu_controls.htm>

**Entregables del Proyecto**

* .zip con las fuentes de su Proyecto con el nombre proyecto2\_grupoNUMEROGRUPO.
* .jar de su proyecto con el nombre proyecto2\_grupoNUMEROGRUPO.
* .pdf con su reporte

**Rúbrica del Proyecto**

El proyecto tiene un puntaje de **30 puntos de los 100 puntos** del componente teórico del segundo 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)**

**Para que el proyecto sea revisado, este debe cumplir con al menos el 70% de la funcionalidad. Caso contrario el proyecto tendrá una puntación de 0**

**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 %)**

* El programa fuente deberá presentar un código documentado internamente usando Java doc (un correcto uso de comentarios).
* Reporte 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.
* Uso de colecciones apropiadas para el manejo de los datos.

**Modularidad y Encapsulamiento (15 %)**

* 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.

**Manejo de Excepciones (5 %)**

* Uso de excepciones personalizadas y correcto manejo de excepciones.

**Penalidades**

10 porciento menos por no seguir buenas prácticas de programación (convención para nombre de clases, nombre de atributos y métodos, uso de constantes cuando es necesario, o algún fallo grave que se detecte en la forma de programar).

**Consideraciones Generales**

* **El proyecto debe ser realizado usando las librerías JavaFX, Java SE 11 y el patrón de diseño MVC.**
* **Para el desarrollo de la aplicación debe usar JavaFX con fxml**
* Recuerde que debe implementar en su código el manejo de excepciones necesarias en cada caso. Esto quiere decir que su programa no debe caerse y si ocurre alguna excepción debe mostrarse al usuario algún mensaje personalizado más no la excepción en sí. Por ejemplo: “Problemas técnicos. Estamos resolviendo”.
* Sea creativo en el diseño de la interfaz. Trabaje con una pantalla de al menos 700 x 700
* Todo grupo debe tener un repositorio de **GitHub** dónde continuamente suban su trabajo. Con las contribuciones en el repositorio se medirá su desempeño en el proyecto.
* **El proyecto debe cumplir con el 70% de la funcionalidad para ser calificado, caso contrario tendrá una calificación de 0.**
* La nota final del proyecto será igual a: nota x sustentación x participación.
* Recuerde no usar rutas absolutas para referenciar los recursos dentro de su proyecto y probar su proyecto en otros equipos antes de enviarlo para asegurarse que funciona correctamente.
* Debe usar archivos para almacenar los datos de forma persistente en el proyecto.