

Algoritmos y programación II-
Tarea Integradora 3.
Residence Management

Integrantes:

Gabriel Alejandro Delgado Álvarez

A00372612

Juan Jacobo García Aristizabal

A00368502

Universidad Icesi.

Departamento de TIC.

Santiago de Cali, 30 de mayo de 2021.

Requerimientos funcionales

RF1. Permite ingresar al administrador a su menú, y permite el ingreso por apartamento a su respectivo menú, un administrador no puede ingresar al menú de un apartamento y un apartamento no puede ingresar al menú de administrador.

RF2. Imprimir y guardar la factura por cada apartamento, el administrador cada mes tendrá la posibilidad de generar la factura de la administración para cada apartamento, cada apartamento en el menú tendrá la posibilidad de imprimir o guardar su factura.

RF3. Generar un csv con los datos de los residentes por apartamento, por parte del administrador se podrá exportar los residentes que tiene cada apartamento.

RF4. Permitir enviar unas notificaciones a todos los apartamentos por parte del administrador. En el menú del administrador habrá una opción para enviar mensajes a todos los apartamentos o alguno en específico.

RF5. Permitir reservar una zona común del complejo, los residentes tendrán la posibilidad a través del menú reservan alguna de las zonas comunes.

RF6. Ver las estadísticas de la unidad. Dentro del menú el administrador tendrá unos gráficos para revisar todas las estadísticas del complejo residencial, cartera y residentes por ejemplo.

RF7. Enviar reclamos por parte de los residentes. En el menú de los residentes habrá una opción para enviar quejas y reclamos al administrador.

RF8. Generar reportes por parte del administrador. El administrador podrá generar pdfs con la información de la unidad para exportar o enviar a otros apartamentos.

RF9. Gestionar los apartamentos y los residentes. Los apartamentos y residentes se podrán editar, eliminar o crear nuevos.

RF10. Dentro del menú del administrador habrá tablas para mostrar los apartamentos, residentes, porteros, personal de servicios, mascotas y carros.

RF11. Gestionar las personas de servicio y limpieza se podrán editar, eliminar o crear nuevos.

RF12. Gestionar los porteros. Los porteros se podrán editar, eliminar o crear nuevos.

Requerimientos completos

RF1. Permite ingresar al administrador a su menú, y permite el ingreso por apartamento a su respectivo menú, un administrador no puede ingresar al menú de un apartamento y un apartamento no puede ingresar al menú de administrador.

RF3. Generar un csv con los datos de los residentes por apartamento, por parte del administrador se podrá exportar los residentes que tiene cada apartamento.

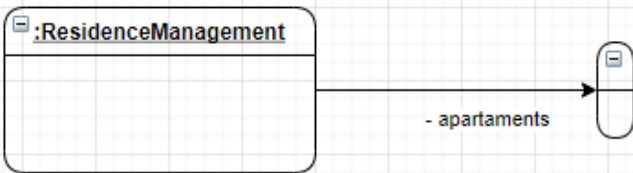
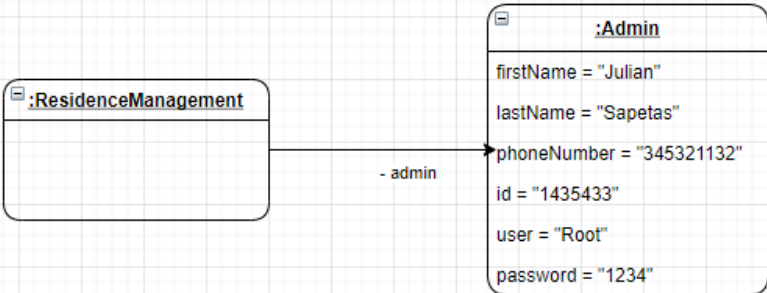
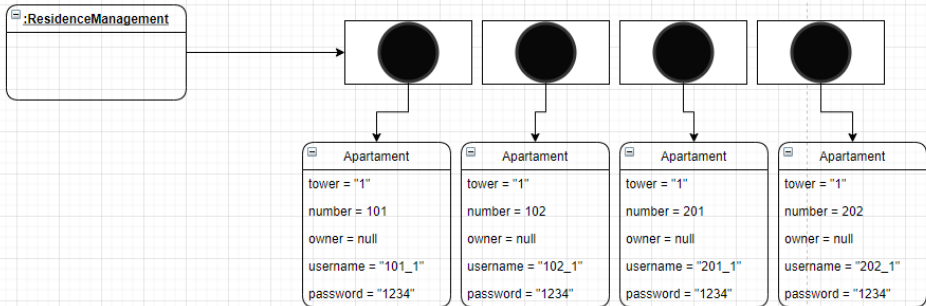
Para las pruebas hemos utilizado una unidad de 4 torres, 10 pisos y 4 apartamentos por torres, para iniciar sesión en un apartamento su usuario es numero de apto + _ y su torre, por ejemplo 403_3 y su contraseña es 1234.

Para el administrador su usuario es root y la contraseña 1234.

Evidencia RF1: <https://youtu.be/iXHHWPllRts>

Evidencia RF3: <https://youtu.be/sShSEbXwy2U>

Configuración de los escenarios

Nombre	Clase	Escenario
scenaryOne	ResidenceManagementTest	
scenaryTwo	ResidenceManagementTest	
scenaryThree	ResidenceManagementTest	

Diseño de casos de prueba

Objetivo de la prueba: Verificar la correcta creación de los apartamentos de un conjunto residencial				
Clase	Método	Escenario	Valores de Entrada	Resultado
ResidenceManagementTest	testCreateApartaments	scenaryOne	N/A	Se han creado correctamente los apartamentos y sus posiciones son diferente a nulas

Objetivo de la prueba: Verificar el correcto inicio de sesión de un administrador con su usuario y contraseña correspondientes.				
Clase	Método	Escenario	Valores de Entrada	Resultado
ResidenceManagementTest	testLoginAdmin	scenaryTwo	String username = "Root"; String password = "1234"; String usernameFail = "Admin"; String passwordFail = "5451234";	El inicio de sesión de un administrador con sus credenciales correctas devuelve un valor de verdadero.

Objetivo de la prueba: Verificar el correcto inicio de sesión con los datos de un apartamento.				
Clase	Método	Escenario	Valores de Entrada	Resultado
ResidenceManagementTest	testLoginApartament	scenaryThree	String username = "202_1"; String password = "1234";	El inició de sesión con las credenciales correctas de un apartamento devuelve un valor de verdadero.

Objetivo de la prueba: Validar la correcta obtención de la posición de un apartamento en un arreglo y verificar que sea igual al pedido				
Clase	Método	Escenario	Valores de Entrada	Resultado
ResidenceManagementTest	testBinarySearchApartament	scenaryThree	String number = "201"; String username = "201_1"; String password = "1234";	El apartamento en el índice devuelto por el método es igual al apartamento creado

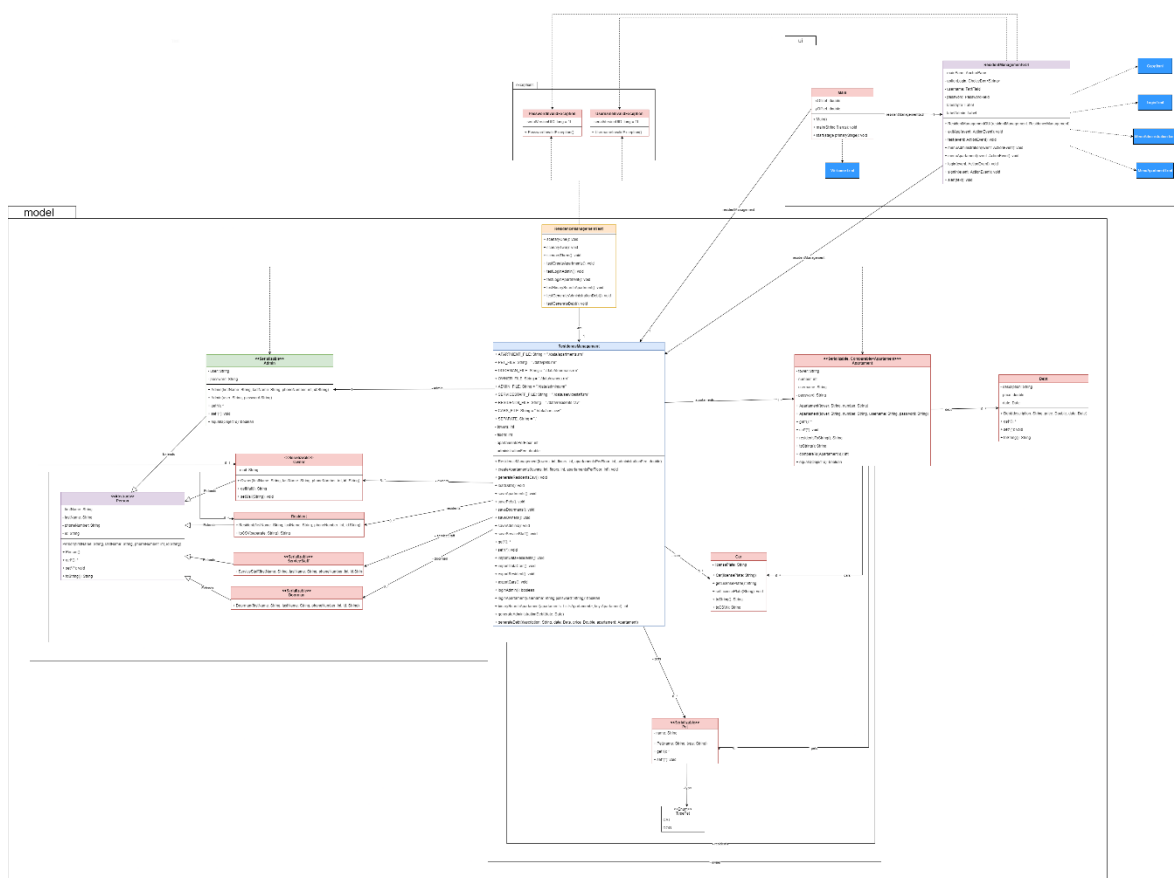
Objetivo de la prueba: Verificar la generación de las deudas de administración de los apartamentos.				
---	--	--	--	--

Clase	Método	Escenario	Valores de Entrada	Resultado
ResidenceManagementTest	testGenerateAdministrationDebt	scenaryThree	Date date = new Date(); double fee = rm.getAdministrationFee(); double fee = rm.getAdministrationFee();	Todos los apartamentos tienen una nueva deuda generada y la lista de deudas no está vacía.

Objetivo de la prueba: Verificar la generación de la deuda de administración en un apartamento en específico.

Clase	Método	Escenario	Valores de Entrada	Resultado
ResidenceManagementTest	testGenerateDebt	scenaryFour	Date date = new Date(); String desc = "Cuota de administracion"; double fee = rm.getAdministrationFee(); double apFee = rm.getApartaments().get(2).getDebt().get(0).getPrice();	La deuda del apartamento seleccionado es igual al precio de la deuda de administración

Diagrama



[Link al diagrama](#)

READ ME