**UNIVERSIDAD PEDAGÓGICA Y TECNOLÓGICA DE COLOMBIA**

**FACULTAD DE INGENIERÍA**

**ESCUELA DE INGENIERÍA DE SISTEMAS**

**PROYECTO FINAL – SEGUNDO CORTE**

Desarrollar una aplicación bajo arquitectura web con JPA e Hibernate que funcione bajo el servidor de aplicaciones Wildfly. Para esto se deberán tener en cuenta las siguientes consideraciones:

Se requiere documentar los requisitos del sistema de información para la administración de propiedades horizontales. Este sistema busca generar valor en la función de la administración de todos los tipos de propiedad horizontal, el mismo les brinda a sus usuarios la capacidad de gestionar todos los gastos y cobranzas dentro de un conjunto residencial, comercial, empresarial y profesional que utiliza la modalidad de condominio para el pago común de los servicios requeridos. A partir de la investigación realizada se conoce la siguiente información:

* El sistema deberá facilitar tareas de administración de propiedad horizontal como la creación y, actualización de los residentes, la generación de los recibos de cobro a los residentes del conjunto, cargar las obligaciones de los residentes, generar paz y salvo de pagos de administración de la Propiedad Horizontal (PH) y poder generar el historial de pago de cada residente.
* La información que se ha determinado que debe tener un residente es su documento de identidad, nombres, apellidos, teléfono celular y por supuesto el bloque y apartamento donde reside.
* Con respecto a los recibos de cobro debe tener adicionalmente la información general de la propiedad horizontal (Nombre de la PH, dirección y teléfono de administrador), la información del detalle del recibo de cobro (Descripción del cobro y valor), la fecha esperada de pago y el estado de cobro. Si hay acuerdos de pago también debe figurar en el recibo de cobro de igual manera que el saldo pendiente.
* El historial de pago deberá contener la información de la propiedad (por ejemplo bloque y apartamento), periodo pagado(mes/año), descripción del pago realizado y el valor del pago. De igual manera deberá contener el saldo adeudado a la fecha.
* El sistema permitirá registrar reservaciones del salón comunal y consultar la disponibilidad del salón comunal para los propietarios por fecha y hora.
* Para registrar la reservación se debe indicar la información del propietario que la solicita (ej bloque y apto), el espacio a reservar (ej Salón, cancha deportiva, etc), la fecha y hora, el número de horas a reservarlo. El número de horas no puede exceder las 5 horas.
* Al registrarse la reserva debe calcular el valor de la reserva, el cual se calcula basado en el número de horas de la reserva. Se calcula de acuerdo al SDMMV (Salario Diario Mínimo Mensual Vigente) que corresponde de dividir el valor del SMMV (Salario mínimo mensual Vigente) entre los días del mes. Este valor de la reserva se cargará como una obligación del propietario.
* Si la reserva se cancela hasta 5 días antes del evento no tiene penalidad, de lo contrario esta penalidad se calcula con la mitad del costo de la reserva, valor que se cargará al propietario

1. La siguiente es la lista de los requerimientos funcionales del aplicativo:

|  |  |  |
| --- | --- | --- |
| Id Requerimiento | Descripción | Rol |
| RNF-01 | Crear un residente | Administrador |
| RNF-02 | Actualizar un residente | Administrador |
| RNF-03 | Generar recibos de cobro de un residente | Administrador |
| RNF-04 | Cargar obligaciones de un residente | Administrador |
| RNF-05 | Generar historial de pago de un propietario | Administrador |
| RNF-06 | Generar paz y salvo de un propietario | Administrador |
| RNF-07 | Registrar reservación | Propietario / Administrador |
| RNF-08 | Consultar la disponibilidad de un espacio físico | Propietario / Administrador |
| RNF-09 | Calcular valor reservación de un espacio físico | Propietario / Administrador |
| RNF-10 | Calcular multa no uso reserva de un espacio físico | Administrador |
| RNF-11 | Cambiar estado de cuenta | Administrador |

**SUPUESTOS Y CONSIDERACIONES**

Los siguientes son los supuestos y consideraciones que debe tener en cuenta al desarrollar la solución propuesta:

* 1. Una vez creado el propietario del inmueble pasa a tener su estado de cuenta “Creado”.
  2. Cuando el administrador carga una obligación al propietario su estado de cuenta pasa a estar “Con deuda”
  3. Si el propietario cancela oportunamente las obligaciones que tiene con la PH su estado de cuenta dentro de las fechas estipuladas en su recibo de cobro pasa a estado “Al día”.
  4. Si por el contrario el propietario no cancela oportunamente durante el presente periodo su estado de cuenta queda “Con deuda”.
  5. Si el propietario lleva tres periodos en los cuales no cancela oportunamente con su obligación su estado de cuenta queda en “Moroso”.
  6. En el caso que el propietario establezca con la administración un esquema de pagos para saldar la deuda, por lo que su estado de cuenta quedaría en “En acuerdo de pago”.
  7. Si el propietario estando en estado de cuenta “En acuerdo de pago” incumple con alguna de las cuotas estipuladas su estado de cuenta pasa a “Moroso”.
  8. Si un propietario lleva más de tres periodos con estado de cuenta “Moroso”, la administración inicia un proceso de cobro coactivo para recuperar el valor adeudado de las obligaciones pendientes. Su estado de cuenta queda en “Cobro coactivo”. Como resultado del cobro coactivo pueden ocurrir dos escenarios posibles: Que pague las obligaciones pendientes, por lo que su estado de cuenta volverá a ser “Al día” o que genere un acuerdo de pago y su estado de cuenta quedaría “En acuerdo de pago” . De lo contrario, si se rehúsa a pagar se inicia un proceso de solicitud de embargo por el monto causado, por lo que su estado de cuenta quedaría como “En proceso de embargo”.

**ACTIVIDADES DEL PROYECTO**

A partir del caso de estudio realice las siguientes actividades:

1. Cree el diagrama de estados UML que refleje la transición de estados que tiene un estado de cuenta.
2. Cree un diagrama de secuencia del requerimiento funcional RNF-11 (Cambiar estado de cuenta).
3. Cree el diagrama de componentes UML de la solución informática del problema planteado
4. Cree el diagrama de despliegue basado en la arquitectura web a la que corresponde la solución informática del problema planteado.
5. A partir de lo descrito en los puntos anteriores realice el desarrollo de la solución bajo arquitecturas web Java considerando la siguiente plataforma tecnológica:
   1. Aplicación web Java mediante el uso de JSP y Servlets o Spring MVC
   2. Utilización de AngularJS (opcional)
   3. Manejo de persistencia mediante JPA y el ORM Hibernate
   4. Servidor de aplicaciones WildFly