## Catálogo de inmuebles

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Task Card** | Lógica alta catálogo de inmueble y generar PDF | |  | Programador |  | |
|  |  |  |  |  |  |  |
| Fecha |  |  |  | Task Points |  |  |
|  |  |  |  |  |  |  |
| Número de Historia | | 5 |  | Número de TaskCard | |  |
|  |  |  |  |  |  |  |
| Descripción:  Se recibe de la vista un objeto catalogoVista, que consta de un cliente, una lista de inmueble y un Map de inmueble con foto.  Para el alta se deberá validar que los inmuebles no se repitan y que haya un cliente asociado al catálogo. También que los siguientes datos de los inmuebles deben existir: código inmueble, tipo de inmueble, localidad, dirección, barrio, precio. Si los siguientes datos no existen, no deben mostrarse: cantidad de dormitorios, baños, garaje, patio, superficie terreno, superficie edificada.  Si todo es correcto se deberá generar un archivo PDF con los datos del catálogo. Luego se genera un catálogo con dicho PDF, el cliente pasado y la fecha actual y se dará de alta en la base de datos.  Si algo no es correcto se devuelve un error que explique la situación. | | | | | | |
|  |  |  |  |  |  |  |
| Notas  La generación del PDF se debe hacer en una clase aparte encargada de esto. | | | | | | |
|
|
| Fecha | Realizado | | A realizar | | Comentarios | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |

## Generar Reserva

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Task Card** | Vista alta, baja y listar reserva | |  | Programador |  | |
|  |  |  |  |  |  |  |
| Fecha |  |  |  | Task Points |  |  |
|  |  |  |  |  |  |  |
| Número de Historia | | 7 |  | Número de TaskCard | |  |
|  |  |  |  |  |  |  |
| Descripción:  En la vista de alta de reserva, los campos a ingresar por el usuario son los siguientes:   * Cliente (ComboBox<Cliente>) * Inmueble (ComboBox<Inmueble>) * Importe (Decimal, 2 dígitos) * Tiempo de vigencia:   + Fecha inicio de la reserva (Date)   + Fecha fin de la reserva (Date)   Si el rango de fechas está al revés, debe invertirse y luego proseguir al alta.  Todos los campos son obligatorios.  Si ocurre un error en la validación de uno o más campos, la interfaz deberá mostrar el error y explicar brevemente que ha sucedido.  La vista de listar reserva puede accederse al seleccionar un inmueble o un cliente. Esta vista lista todas las reservas del cliente o inmueble seleccionado y de esta se puede proceder a ver el PDF de una reserva seleccionada, dar de baja la reserva seleccionada o a crear una nueva reserva para el inmueble o el cliente seleccionado. | | | | | | |
|  |  |  |  |  |  |  |
| Notas | | | | | | |
|
|
| Fecha | Realizado | | A realizar | | Comentarios | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Task Card** | Lógica alta, baja lógica reserva y generar PDF | |  | Programador |  | |
|  |  |  |  |  |  |  |
| Fecha |  |  |  | Task Points |  |  |
|  |  |  |  |  |  |  |
| Número de Historia | | 7 |  | Número de TaskCard | |  |
|  |  |  |  |  |  |  |
| Descripción:  Se recibe de la vista un objeto reservaVista, que consta de un cliente, un inmueble, un importe y un rango de vigencia.  Para el alta se deberá validar que haya un cliente y un inmueble asociados al catálogo. También que el importe sea un número decimal válido mayor que 0 y que el rango de vigencia tenga una fecha inicial mayor a hoy, y que tenga una fecha de inicio anterior a la de finalización.  Si todo es correcto se deberá generar un archivo PDF con los datos de la reserva. Luego se genera una Reserva con dicho PDF, el cliente, el inmueble y el rango de fechas pasado y se dará de alta en la base de datos. También se deberá enviar un correo al mail del cliente con el archivo PDF como adjunto en otro hilo.  Si algo no es correcto se devuelve un error que explique la situación.  Para la baja lógica se debe validar que los campos sean los mismos que en la base de datos, se debe cambiar el estado a BAJA y, por último, se debe persistir. Si algo no es correcto se devuelve un error que explique la situación. | | | | | | |
|  |  |  |  |  |  |  |
| Notas  La generación del PDF se debe hacer en una clase aparte encargada de esto. | | | | | | |
|
|
| Fecha | Realizado | | A realizar | | Comentarios | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Task Card** | Lógica envío de mail | |  | Programador |  | |
|  |  |  |  |  |  |  |
| Fecha |  |  |  | Task Points |  |  |
|  |  |  |  |  |  |  |
| Número de Historia | | 7 |  | Número de TaskCard | |  |
|  |  |  |  |  |  |  |
| Descripción:  Se recibe un objeto mail con tres lista de correos de remitentes (Para, CC y CCO), un String de asunto y una lista de objetos Archivo para adjuntar.  La clase debe armar con estos datos un correo y lo debe enviar a través del servidor de correo y la cuenta del cliente. ([olimpoagilinmobiliaria2016@gmail.com](mailto:olimpoagilinmobiliaria2016@gmail.com)) | | | | | | |
|  |  |  |  |  |  |  |
| Notas | | | | | | |
|
|
| Fecha | Realizado | | A realizar | | Comentarios | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Task Card** | Persistidor y entidad reserva | |  | Programador |  | |
|  |  |  |  |  |  |  |
| Fecha |  |  |  | Task Points |  |  |
|  |  |  |  |  |  |  |
| Número de Historia | | 7 |  | Número de TaskCard | |  |
|  |  |  |  |  |  |  |
| Descripción  Se debe hacer una interfaz con métodos que permitan:  - Guardar una reserva (si resulta en error lanza excepción SaveUpdateException)  - Modificar una reserva (si resulta en error lanza excepción SaveUpdateException)  - Obtener todas las reservas (si resulta en error lanza excepción ConsultaException)  Todas las excepciones mencionadas extienden a PersistenceException.  Se debe hacer una clase que implemente dicha interfaz mediante hibernate y completar las entidades con las anotaciones necesarias.  Implementar la clase Reserva. La clase debe tener los siguientes atributos: id (Integer), cliente (Cliente), inmueble (Inmueble), archivoPDF (PDF), fechaInicio (Date), fechaFin (Date), estado (Estado).  Para cada atributo deben hacerse los métodos set y get. | | | | | | |
|
|
|  |  |  |  |  |  |  |
| Notas | | | | | | |
|
|
| Fecha | Realizado | | A realizar | | Comentarios | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |