## 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 catálogoVista, que consta de un cliente, una lista de inmueble y un mapa de inmueble con una 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 | | | | | | |
|
|
| Fecha | Realizado | | A realizar | | Comentarios | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |

## Generar Reserva--------v--------falta!

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Task Card** | Vista alta reserva | |  | Programador |  | |
|  |  |  |  |  |  |  |
| Fecha |  |  |  | Task Points |  |  |
|  |  |  |  |  |  |  |
| Número de Historia | | 7 |  | Número de TaskCard | |  |
|  |  |  |  |  |  |  |
| Descripción:  Los campos a ingresar por el usuario son los siguientes:   * Nombre (string:30) * Apellido (string:30) * Tipo de documento (ComboBox<TipoDocumento>) * Número de documento (String) * Dirección:   + Calle (string:30)   + Número (String:30)   + Piso (String:30)   + Departamento (string:10)   + Localidad (string:30)   + Provincia (string:30) * Teléfono (String) * Email (string:30)   Los siguientes datos son obligatorios: nombre, apellido, tipo, número de documento, calle, número, localidad, provincia, teléfono.  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 interfaz para modificar un propietario es la misma que la interfaz de alta, se pueden modificar todos los campos. Se deben mostrar los datos del propietario en los campos correspondientes.  La UI para la baja de un propietario es un cartel avisando que el propietario va a ser eliminado, y presentará las opciones para aceptar o no la eliminación.  Las UI deben ser similares a los spikes realizados en la fase de exploración de la planificación. | | | | | | |
|  |  |  |  |  |  |  |
| Notas | | | | | | |
|
|
| Fecha | Realizado | | A realizar | | Comentarios | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Task Card** | Lógica alta reserva y generar PDF | |  | Programador |  | |
|  |  |  |  |  |  |  |
| Fecha |  |  |  | Task Points |  |  |
|  |  |  |  |  |  |  |
| Número de Historia | | 7 |  | Número de TaskCard | |  |
|  |  |  |  |  |  |  |
| Descripción:  Para el alta se recibe de la vista un objeto propietario y se debe validar que:   * Nombre sea un string de letras con un máximo 30 caracteres. * Apellido sea un string de letras con un máximo 30 caracteres. * Tipo de documento sea uno de los tipos definidos. * Número de documento sea numérico y corresponda con el tipo de documento. * Dirección:   + Calle sea un string de como máximo 30 caracteres alfanuméricos.   + Número sea String de como máximo 10 caracteres.   + Piso sea String de como máximo 10 caracteres.   + Departamento sea un string de como máximo 10 caracteres.   + Localidad sea un string de como máximo 30 caracteres.   + Provincia sea un string de como máximo 30 caracteres. * Teléfono sea un String de como máximo 10 caracteres. * Email sea un string con formato de correo.   Los siguientes datos son obligatorios: nombre, apellido, tipo, número de documento, calle, número, localidad, provincia, teléfono.  Si todo es correcto se da de alta al propietario en la base de datos, si no se devuelve una excepción  Para la modificación se permitirá cambiar cualquier campo, se deberán realizar las mismas validaciones y si todo es correcto se deberá modificar el propietario en la base de datos. Si algo no es correcto se deberá devolver una excepción.  Para la baja, no se deben tener consideraciones especiales y se debe realizar una baja lógica del propietario.  Se debe implementar un método que permita obtener todos los propietarios de la base de datos.  Debe implementarse métodos para la validación de los campos, pero éstos deben ser privados. | | | | | | |
|  |  |  |  |  |  |  |
| Notas | | | | | | |
|
|
| Fecha | Realizado | | A realizar | | Comentarios | |
| 21/10/16 | Logica de Alta y modificacion | | Baja de un propietario, validar datos | | Se realizaron los casos de prueba antes de codificar | |
| 22/10/16 | Validar datos | | Baja de un propietario | |  | |
| 02/11/16 | Baja de un propietario | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Task Card** | Lógica envío de mail | |  | Programador |  | |
|  |  |  |  |  |  |  |
| Fecha |  |  |  | Task Points |  |  |
|  |  |  |  |  |  |  |
| Número de Historia | | 7 |  | Número de TaskCard | |  |
|  |  |  |  |  |  |  |
| Descripción:  Para el alta se recibe de la vista un objeto propietario y se debe validar que:   * Nombre sea un string de letras con un máximo 30 caracteres. * Apellido sea un string de letras con un máximo 30 caracteres. * Tipo de documento sea uno de los tipos definidos. * Número de documento sea numérico y corresponda con el tipo de documento. * Dirección:   + Calle sea un string de como máximo 30 caracteres alfanuméricos.   + Número sea String de como máximo 10 caracteres.   + Piso sea String de como máximo 10 caracteres.   + Departamento sea un string de como máximo 10 caracteres.   + Localidad sea un string de como máximo 30 caracteres.   + Provincia sea un string de como máximo 30 caracteres. * Teléfono sea un String de como máximo 10 caracteres. * Email sea un string con formato de correo.   Los siguientes datos son obligatorios: nombre, apellido, tipo, número de documento, calle, número, localidad, provincia, teléfono.  Si todo es correcto se da de alta al propietario en la base de datos, si no se devuelve una excepción  Para la modificación se permitirá cambiar cualquier campo, se deberán realizar las mismas validaciones y si todo es correcto se deberá modificar el propietario en la base de datos. Si algo no es correcto se deberá devolver una excepción.  Para la baja, no se deben tener consideraciones especiales y se debe realizar una baja lógica del propietario.  Se debe implementar un método que permita obtener todos los propietarios de la base de datos.  Debe implementarse métodos para la validación de los campos, pero éstos deben ser privados. | | | | | | |
|  |  |  |  |  |  |  |
| Notas | | | | | | |
|
|
| Fecha | Realizado | | A realizar | | Comentarios | |
| 21/10/16 | Logica de Alta y modificacion | | Baja de un propietario, validar datos | | Se realizaron los casos de prueba antes de codificar | |
| 22/10/16 | Validar datos | | Baja de un propietario | |  | |
| 02/11/16 | Baja de un propietario | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 un propietario (si resulta en error lanza excepción SaveUpdateException)  - Modificar un propietario (si resulta en error lanza excepción SaveUpdateException)  - Obtener un propietario (si resulta en error lanza excepción ConsultaException)  - Obtener todos los propietarios (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. | | | | | | |
|
|
|  |  |  |  |  |  |  |
| Notas | | | | | | |
|
|
| Fecha | Realizado | | A realizar | | Comentarios | |
| 31/10/16 | Guardar y modificar un propietario | | Obtener un propietario y listar todos los propietarios | |  | |
| 02/11/16 | Obtener un propietario y listar todos los propietarios | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |