**CORRECCIONES**

**UI FACTURAS**

* Formatos de fecha no homogéneos: se usa uno en los campos de fecha y otro diferente en la tabla.
* Al añadir si se borran todos los campos y luego se pulsa en Crear, los campos se ponen con su borde en rojo (supongo que indicando que son campos requeridos) y al informarlos y pulsar en Crear no ocurre nada, siendo necesario salir de la ventana para poder seguir operando. Además habría que quitar el borde rojo una vez corregido el error o al cambiar el contenido del campo.
* No muestras un mensaje indicando al usuario cual es el error que ha cometido al introducir valores erróneos (no usable).
* El contenido de los campos seleccionar reparación y seleccionar cliente no es usable, ya que se ha de saber el id numérico de ambos para saber que se está asociando.
* Al salir a la ventana Home y volver a entrar en Facturas, desaparecen las facturas creadas, vuelven a aparecer las eliminadas y no se mantienen los datos modificados.
* La ordenación de las columnas no funciona, mejor deshabilitarla si no es funcional.
* Cuando se pone el foco en algunos campos no es fácil ver que lo tiene, por ejemplo la combo del criterio de consulta o en los botones de pulsación en modificación o creación.
* No se lee el texto de la checkbox de pagada.
* El uso del logger para mostrar mensajes es mejorable.
* El javadoc es mejorable.
* Tratamiento de excepciones insuficiente.
* Los nombres de los métodos manejadores de evento es poco indicativo de su función. Además abusas de la derivación de llamadas a métodos. Los propios manejadores podrían realizar la operativa sin derivar a otros métodos o incluso a otros métodos de otros objetos controladores ¿Por qué no llama directamente el manejador de evento de la pulsación del botón de alta o modificación al método facturasLogicController.createFactura/updateFactura, en lugar de hacerlo a través del controlador de la vista de la tabla? Complicas el código innecesariamente.
* Comentarios del código de los métodos insuficientes.

**UI CLIENTES**

* La búsqueda de clientes no es usable: no se sabe cuales son los criterios de consulta (id, dni, nombre, apellidos…) Pon al menos un prompt text para indicarlo. (El foco esta en el campo de búsqueda y si no se quita no se ve el prompt text -.- GG)
* Al salir a la ventana Home y volver a entrar en Clientes, desaparecen los clientes creados, vuelven a aparecer los eliminados y no se mantienen los datos modificados.
* El uso del logger es insuficiente. Es curioso que si borro un cliente en el logger sale que he borrado una factura...
* Al añadir si se borran todos los campos y luego se pulsa en Crear, los campos se ponen con su borde en rojo (supongo que indicando que son campos requeridos) y al informarlos y pulsar en Crear no ocurre nada, siendo necesario salir de la ventana para poder seguir operando. Además habría que quitar el borde rojo una vez corregido el error o al cambiar el contenido del campo.
* No funcionan ni el alta ni la modificación cuando se introducen nombres compuestos o los dos apellidos o alguno de los anteriores tiene tildes.
* No muestras ningún tipo de información al usuario acerca de los errores de formato en los campos a la hora de validarlos. (se muestra en el tooltip después de darle al botón crear)
* Cuando se pone el foco en algunos campos no es fácil ver que lo tiene en los botones de pulsación en modificación o creación.
* El javadoc es mejorable.
* Tratamiento de excepciones insuficiente.
* Comentarios del código de los métodos insuficientes.

GESTIÓN DE TALLER

|  |  |  |  |
| --- | --- | --- | --- |
| *GUI CONTROLLERS* | | | |
| **CLASE** | **GUI** | **FXML** | **AUTOR** |
| HomeController.java | HOM01 | home.fxml | Carlos Santos |
| ClientesController.java | CLI01 | ~~modulo\_facturas.fxml~~ | Ionut Savin |
| ClientesCuController.java | CLI02 | ~~nuevo\_cliente.fxml~~ | Ionut Savin |
| FacturasController.java | FAC01 | modulo\_facturas.fxml | Carlos Santos |
| FacturasCuController.java | FAC02 | nueva\_factura.fxml | Carlos Santos |

|  |  |  |
| --- | --- | --- |
| *LOGIC CONTROLLERS* | | |
| **CLASE** | **DESCRIPCIÓN** | **AUTOR** |
| ClientesManagerTestDataGenerator.java | Generador de datos para pruebas de modulo clientes. | Ionut Savin |
| FacturasManagerTestDataGenerator.java | Generador de datos para pruebas de modulo facturas. | Carlos Santos |

|  |  |  |
| --- | --- | --- |
| *INTERFACES* | | |
| **CLASE** | **DESCRIPCIÓN** | **AUTOR** |
| ClientesManager.java | Interfaz para operaciones con clientes. | Ionut Savin |
| FacturasManager.java | Interfaz para operaciones con facturas. | Carlos Santos |

|  |  |  |
| --- | --- | --- |
| *UTIL* | | |
| **CLASE** | **DESCRIPCIÓN** | **AUTOR** |
| FieldValidator.java | Clase útil para validación de campos. | Carlos Santos, Ionut Savin. |

|  |  |  |
| --- | --- | --- |
| *CONFIG* | | |
| **CLASE** | **DESCRIPCIÓN** | **AUTOR** |
| GTConstants.java | Clase para proveer constantes de la aplicación. | Carlos Santos |

|  |
| --- |
| *PRUEBAS* |

No es necesario conocer ningún dato para pruebas, estos se generaran mediante las clases test data generator.

|  |
| --- |
| *DESCRIPCIÓN DE CASOS DE USO* |

La vía para realizar cada caso de uso es bastante intuitiva, se disponen de varios métodos para acceder a estos.

A cada módulo se puede acceder de distintas maneras:

Haciendo click, haciendo enter con el elemento enfocado, con un acelerador o navegando a través del menú con mnemonics.

Para las operaciones CRUD (Por implementar aceleradores):

Create: haciendo click en el icono crear.

Update, con el ítem seleccionado en la tabla:

* Hacer click en icono modificar.
* ENTER.
* Doble click sobre el ítem.
* A través del menú contextual.

Delete, con el ítem seleccionado en la tabla:

* Hacer click en icono eliminar.
* DELETE (SUPR).
* A través del menú contextual.

|  |
| --- |
| *GITHUB* |

<https://github.com/candi666/GestionTaller.git>