








| CP-XX | HU o funcionalidad a probar | Descripción de la prueba | Datos de entrada | Pasos realizados | Resultado esperado | Resultado obtenido | Capturas | | | | | | | | | | | | |
|--------------|---|--|---|--|--|---|--|--------|-------|--------|--------|----------|--------|--------------|--------------------|-----------|-------|---|---|
| CP-01 | Registro de usuario | Comprobación de registro de usuarios, se va a probar a crear un usuario con un nombre que ya esté registrado | username: admin | entrar al formulario de registro y rellenarlo con un nombre de usuario que esté siendo utilizado como user o admin | Muestra un error por pantalla y nos dice que ya hay un usuario con ese nombre y por tanto no podemos crearlo | Nos muestra el error esperado y no crea al usuario. | <div><div>¡Vaya! Parece que el nombre de usuario está siendo utilizado.</div><div>Formulario de creación de cuenta</div><div>Nombre de usuario*</div><div>Escriba aquí el nombre de su usuario..</div></div> | | | | | | | | | | | | |
| CP-02 | Acceso al menú de admin | Se trata de entrar en el menú de administrador desde un usuario normal | username: user, password: 1234 y la ruta del menú admin = "http://localhost:9000/admin" | Loguearse como usuario admin, dirigirse al menú admin, entrar en el apartado de servicios y tratar de borrar "Mechas Californianas" que se encuentran presentes en una venta | White-label personalizado por que la seguridad deliene el acceso | White-label personalizado de falta de permisos | <div><div>Se Frodo, no llegamos a la ruta</div><div></div><div>Sam-comido un White-Label</div><div>Volver al inicio</div></div> | | | | | | | | | | | | |
| CP-03 | HU-08 Borrado de servicios | Se trata de borrar un servicio que esté presente en una venta | username: admin // password: admin | Loguearse como usuario admin, dirigirse al menú admin, entrar en el apartado de servicios y tratar de borrar "Mechas Californianas" que se encuentran presentes en una venta | Error en pantalla detallando que dicho servicio no se puede borrar pues está presente en una venta y no se borra | Se muestra el error esperado y no se borra | <div><div>Agregar servicio</div><div>Lo sentimos, este producto se encuentra vinculado a alguna cita y no se puede borrar, hasta que dicha cita no se elimine.</div></div> | | | | | | | | | | | | |
| CP-04 | Borrar usuarios con citas o reclamaciones asociadas | Se trata de borrar un cliente que tiene una cita o una reclamación asociada | username: admin // password: admin | Loguearse como usuario admin, dirigirse al menú admin, entrar en el apartado de gestión de usuarios e intentar borrar al user: rayheredia que tiene citas asociadas. | Error en pantalla que relata que no se puede borrar un cliente que tenga citas asociadas | Se muestra el error esperado y no se borra | <div><div>Agregar cliente</div><div>Lo sentimos, debes borrar las reclamaciones y las citas de este cliente antes de borrarlo.</div><div>Lista de clientes</div></div> | | | | | | | | | | | | |
| CP-05 | Borrar una cita creada por un cliente | Se trata de borrar una cita con un cliente que no se encuentre en el import.sqj sino con uno creado mediante el formulario | username: user2 // password: 1234 | Crear un usuario nuevo con las credenciales deseadas, agregar productos al carrito , pagar la compra y tratar de borrarla desde el listado de citas del usuario | Que se muestre un formulario de confirmación y se borre | No se borra y muestra un error indicando como si no hubiese sido el creador de la venta | <div><div>¡Vaya! Parece que esa cita no está a tu nombre.</div><div>Mis pedidos</div><table><tr><th>Título</th><th>Fecha</th><th>Pagado</th><th>Precio</th><th>Detalles</th><th>Borrar</th></tr><tr><td>Pedido: #951</td><td>19 de mayo de 2024</td><td>No pagado</td><td>75,00</td><td></td><td></td></tr></table><pre>/*public boolean deleteByIdAppointments(UserEntity loggedUser , long appointmentId) { if((appointmentRepo.findById(appointmentId).get().getClient().getId() == loggedUser.getId()) loggedUser.isAdmin()) { appointmentRepo.deleteById(appointmentId); } }</pre></div> | Título | Fecha | Pagado | Precio | Detalles | Borrar | Pedido: #951 | 19 de mayo de 2024 | No pagado | 75,00 |  |  |
| Título | Fecha | Pagado | Precio | Detalles | Borrar | | | | | | | | | | | | | | |
| Pedido: #951 | 19 de mayo de 2024 | No pagado | 75,00 |  |  | | | | | | | | | | | | | | |

| | | | |
|--|--|--|--|
| | <p>Este es el método antiguo, que permitía borrar citas cerradas o abiertas pero solamente de usuarios que estuviesen en el importLog por algo que desconozco, comprobé con la depuración y finalmente se veía como hacía la comprobación de dos valores iguales y devolvía un falso, única y exclusivamente cuando el usuario había sido creado por el formulario, chequeando la base datos, los campos relacionados están completamente en orden, así que sin encontrar otra solución he decidido reescribir el método pero esta vez haciendo la comprobación por medio de una consulta JPQL, aquí dejo el nuevo método y la consulta que lanza está a su derecha.</p> | <pre>return true; } else { return false; } }*/ public boolean deleteSelfAppointments(UserEntity loggedUser , Long appointmentId) { if(appointmentRepo.isAppointmentOwner(loggedUser.getId(), appointmentId) { appointmentRepo.deleteById(appointmentId); return true; } if(loggedUser.isAdmin()) { appointmentRepo.deleteById(appointmentId); return true; } else { return false; } }</pre> | <pre>@Query(""" SELECT CASE WHEN COUNT(a) > 0 THEN TRUE ELSE FALSE END FROM Appointment a WHERE a.client.id = :userId AND a.id = :appointmentId """) boolean isAppointmentOwner(@Param("userId") Long userId , @Param("appointmentId") Long appointmentId);</pre> |
|--|--|--|--|

| | | | | | | | | |
|---------|---|--|--|---|--|--|---|--|
| CP-05.2 | <p>Se repite la prueba 05 referida a borrar una cita creada por un cliente, cambiando al procesamiento del borrado.</p> | <p>Se trata de borrar una cita con un cliente que no se encuentre en el importLog sino con uno creado mediante el formulario</p> | <p>username: user2 // password: 1234</p> | <p>Crear un usuario nuevo con las credenciales deseadas, agregar productos al carrito, pagar la compra y tratar de borrarla desde el listado de citas del usuario</p> | <p>Que se muestre un formulario de confirmación y se borre</p> | <p>Se borra la cita sin mayor problema después de confirmar.</p> | <p>Solución: Luego de comprobar que funciona, concluyo haciendo una PR con relación a la rama usada para corregir el error, commits, mergeo y vuelvo a probar en la main luego de haber hecho pull, este pequeño meme refleja como me siento.</p> | |
|---------|---|--|--|---|--|--|---|--|

| | | | | | | | |
|-------|---|---|--|---|--|---|---|
| CP-06 | <p>Borrar una reclamación creada por un cliente</p> | <p>Se trata de borrar una reclamación con un cliente que no se encuentre en el importLog sino con uno creado mediante el formulario</p> | <p>username: user2 // password: 1234</p> | <p>Crear un usuario nuevo con las credenciales deseadas, rellenar el formulario de reclamaciones, enviarlo y tratar de borrarlo desde el menú de usuario en el apartado de "Mis reclamaciones"</p> | <p>Que se muestre un formulario de confirmación y se borre</p> | <p>No se borra y además no muestra ningún mensaje de aviso.</p> | |
| | | <p>Intentemos solventarlo de la misma forma que la anterior</p> | | | | | |
| | | <p>Issue en la que se deduce la rama para solventar el error: HJ-Y Error de borrar una reclamación con un usuario creado mediante el formulario</p> | | | | | |
| | | | | <pre>//true = lo puede borrar, false = no lo puede borrar public boolean erasableReport(Long reportId , Long clientId) { return (reprepo.findById(reportId).get().getClient().getId() == clientId) ? true : false; }</pre> | | | |
| | | <p>partimos de el siguiente método</p> | | <pre>public void processDeletingSelfReports(UserEntity loggedUser, Long id) { if(erasableReport(id, loggedUser.getId())) { deleteById(id); } }</pre> | | | |
| | | | | <pre>public boolean processDeletingSelfReports(UserEntity loggedUser, Long reportId) { if(reprepo.isReportOwner(loggedUser.getId(), reportId)) { reprepo.deleteById(reportId); return true; } if(loggedUser.isAdmin()) { reprepo.deleteById(reportId); return true; } else { return false; } }</pre> | | | |
| | | <p>Aquí la solución, misma metodología que con el error anterior</p> | | | | | <pre>@Query(""" SELECT CASE WHEN COUNT(r) > 0 THEN TRUE ELSE FALSE END FROM Report r WHERE r.client.id = :userId AND r.id = :reportId """) boolean isReportOwner(@Param("userId") Long userId , @Param("reportId") Long reportId);</pre> |

[illegible]