Actividad N°15

# Casos de Prueba

|  |  |
| --- | --- |
| **Identificador** | CP001 |
| **Requerimiento asociado** | REQF 3.2: Los usuarios podrán iniciar sesión con su correo personal o su nombre de usuario que utilicen al momento de registrarse. |
| **Caso de Uso asociado** | CU006, CU012. |
| **Actor** | Usuario. |
| **Nombre** | Datos guardados en la base de datos. |
| **Descripción** | Este caso de prueba desea verificar que los usuarios estén correctamente registrados en la cuenta y puedan ingresar a su perfil. |
| **Prerrequisitos** | Que el usuario tenga conexión a internet, que esté registrado y haya clicado en el botón para *iniciar sesión/registrarse*. |
| **Datos de prueba** | Nickname: Javita2021  Email: javi.espina2021@gmail.com  Password: EsPina21. |
| **Resultados esperados** | Que se inicie correctamente la sesión. |
| **Postcondiciones** | El sistema despliega la interfaz de la página de inicio y muestra un breve mensaje de “Bienvenida” al usuario. |

|  |  |
| --- | --- |
| **Identificador** | CP002 |
| **Requerimiento asociado** | REQF5.1 El sistema dejará que los usuarios puedan editar, añadir o quitar información de su perfil. |
| **Caso de Uso asociado** | CU004, CU005. |
| **Actor** | Usuario. |
| **Nombre** | Modificando el perfil de los usuarios. |
| **Descripción** | Aquí se comprobará que los datos que los usuarios modifiquen o eliminen, se actualicen correctamente en la base de datos. |
| **Prerrequisitos** | Que el usuario esté registrado y haya iniciado sesión, se haya dirigido al botón que lo llevaba a su perfil y cuente con una conexión a internet. |
| **Datos de prueba** | Nickname: Javita2021 Email: javi.espina2021@gmail.com  Password: EsPina21.  Proyectos: ¡Proyectos independientes de surrealismo!  Precios: 20000 CLP  Tiempo de trabajo: 2 semanas.  Dirección: Rembrandt, 1010. Colina, Chile. |
| **Resultados esperados** | Que los datos se realicen correctamente en la base de datos, por los que el usuario ingresa en la tabla de arriba. |
| **Postcondiciones** | El sistema, en caso de que el usuario finalice los cambios de su perfil, se solicitará que ingrese nuevamente su contraseña y si es correcta, será redirigida a su perfil nuevamente con los datos actualizados. Si no, se cerrará su sesión y notificará a su correo electrónico un intento de modificación de sus datos. |

|  |  |
| --- | --- |
| **Identificador** | CP003 |
| **Requerimiento asociado** | REQF 2.3 Se podrá reestablecer la contraseña con el correo personal del usuario y una previa verificación por código que será mandada a este. |
| **Caso de Uso asociado** | CU010, CU012. |
| **Actor** | Usuario. |
| **Nombre** | Olvide mi contraseña |
| **Descripción** | En este caso de prueba se tratará la recuperación de una contraseña y si se actualiza correctamente. |
| **Prerrequisitos** | Que el usuario esté registrado, haya ido al inicio de sesión y clicado en “Olvidé mi contraseña” y puesto su correo con el que se registró. |
| **Datos de prueba** | Email: javi.espina2021@gmail.com |
| **Resultados esperados** | Que la contraseña se modifique correctamente con el código de recuperación. |
| **Postcondiciones** | El sistema le pedirá al usuario que ingrese una nueva contraseña. |

|  |  |
| --- | --- |
| **Identificador** | CP004 |
| **Requerimiento asociado** | REQF 5.3 Al momento de realizar una mantención de seguridad, se creará un registro en la base de datos automáticamente. |
| **Caso de Uso asociado** | CU009 |
| **Actor** | Administrador. |
| **Nombre** | Mantenimiento rutinario. |
| **Descripción** | Verificar que, una vez realizada una mantención, esta se guarde correctamente en la BD. |
| **Prerrequisitos** | Que el administrador haya iniciado sesión en la BD y comience las pruebas pertinentes. Y los cambios se generen automáticamente en la BD, tabla de auditoria. |
| **Datos de prueba** | ID\_Cambio: CH005  Fecha: 02-06-2021  Descripción: Se realiza mantenimiento de la BD.  Usuario: EduardoAGodoy |
| **Resultados esperados** | Que se ingresen los cambios realizados automáticamente una vez se finalicen las modificaciones. |
| **Postcondiciones** | El sistema notifica al administrador que se guardaron los cambios en el registro de auditoria. |

|  |  |
| --- | --- |
| **Identificador** | CP005 |
| **Requerimiento asociado** | REQF 4.2 Desarrollar un “Chat” para que los usuarios puedan comunicarse una vez realicen una compra. |
| **Caso de Uso asociado** | CU008 |
| **Actor** | Usuario registrado. |
| **Nombre** | Mensajería instantánea. |
| **Descripción** | Que los mensajes se envíen correctamente y registren en la BD para que sean mostrados al otro usuario. |
| **Prerrequisitos** | Que el usuario esté registrado y haya iniciado sesión, después de eso, que haya realizado una compra y ahí es redirigido a la ventana de “Compras” donde podrá presionar el botón para abrir el correspondiente “Chat”. |
| **Datos de prueba** | Mensaje: HOLA…  Hora: 9:10am |
| **Resultados esperados** | Que los cambios se guarden correctamente en la BD. |
| **Postcondiciones** | El sistema muestra el mensaje al usuario destinatario con éxito. |

# Tablas requerimientos NO funcionales

|  |  |  |
| --- | --- | --- |
| **Requerimiento no funcional** | **Tipo de Prueba de Software** | **Herramienta de Software** |
| Requerimiento no funcional de **Adecuación Funcional***:* Al terminar la documentación, se comprobará que las necesidades del usuario estén abarcadas o cubiertas, releyendo la documentación. | Prueba de Documentación y procedimiento. | VisualCode |
| Requerimiento no funcional de **Rendimiento**: Capacidad del servidor para sea capaz de soportar máximo 5.000.000 de usuarios conectados simultáneamente. | Prueba de carga. | GRINDER. |
| Requerimiento no funcional de **Rendimiento**: Se utilizarán SSD para el servidor además de un espacio mínimo de 2048 TB, además de 10 GB de RAM dedicada para el servidor. | Prueba de estrés. | Jmeter. |
| Requerimiento no funcional de **Compatibilidad**: El servicio a diseñar será compatible con cualquier navegador en su versión más actual a la fecha, 02-05-2021. | Prueba de compatibilidad. | HTTPunit |
| Requerimiento no funcional de **Compatibilidad**: El servicio a realizar funcionará sin ningún servicio externo, ya que no es necesario para lo que será desarrollada. | Prueba de portabilidad. | Browsera |
| Requerimiento no funcional de **Usabilidad**: El sistema será intuitivo y de fácil uso, evitando llenar la pantalla de cosas innecesarias para el uso del usuario promedio. | Pruebas de interfaz de usuario. | HTTPunit |
| Requerimiento no funcional de **Portabilidad**: El sistema a diseñar no necesitará ninguna instalación si el usuario cuenta con un navegador disponible como los recomendados previamente. | Prueba de rendimiento. | Allmon |
| Requerimiento no funcional de **Portabilidad**: Todo será documentado y separado en clases diferentes, para facilitar una mayor modificación por parte de algún tercero en la página. | Prueba de Mantenibilidad. | VisualCode |
| Requerimiento no funcional de **Seguridad**: El sistema debe permitir que las contraseñas se almacenan en la BD en forma encriptada utilizando el algoritmo de encriptación MD5. | Prueba de integridad de Datos y base de datos. | netsparker |
| Requerimiento no funcional de **Seguridad**: Se harán copias en la BD para evitar posibles fraudes por parte del Cliente Vendedor o Cliente Comprador. | Prueba de integridad de Datos y base de datos. | netsparker |
| Requerimiento no funcional de **Seguridad**: Se baneará permanentemente a cualquier usuario que rompa los términos y condiciones y estos se les serán mostrados a los usuarios al momento de registrarse para que tengan la información de una manera fidedigna y directa. | Prueba de seguridad y Control de Acceso. | netsparker |
| Requerimiento no funcional de **Mantenibilidad**: Facilitar la documentación del proyecto para que terceras personas, que estén encargadas del análisis de las mejoras a desarrollar u otros requerimientos a futuro. | Prueba de Documentación y procedimiento. | VisualCode |
| Requerimiento no funcional de **Mantenibilidad**: Se documentará de manera correcta y organizada todos los cambios en el código, para que cualquier programador pueda utilizar eficientemente el código en caso de requerirse mejoras o cambios, se programará de manera ordenada, añadiendo comentarios antes de clase para ofrecer un mejor entendimiento de la función que realizan cada uno de estos. | Prueba de Documentación y procedimiento. | VisualCode |
| Requerimiento no funcional de **Fiabilidad**: Si llega a ocurrir alguna falla relacionada a la base de datos, además de la copia de esta misma la página cargará un test por defecto, pero las respuestas de los usuarios no se cargarán al resultado global. | Prueba de integridad de Datos y base de datos. | netsparker |
| Requerimiento no funcional de **Fiabilidad**: Para el desarrollo del proyecto se han analizado las necesidades para que todo tipo de persona que lo necesite pueda acceder y buscar información o el servicio de su interés o necesidad.  Además, se contará con medios para poder identificar las debilidades o fallas del sistema una vez este esté listo para su uso para los usuarios. | Prueba de Documentación Y Procedimiento | VisualCode |
| Requerimiento no funcional de **Fiabilidad**: Se realizarán respaldos en la base de datos cada dos semanas cuando las bases de datos tengan un breve mantenimiento.  E Implementar mecanismos que aseguren la integridad de los datos. | Prueba de integridad de Datos y base de datos. | netsparker |
| Requerimiento no funcional de **Usabilidad**: En el servicio a diseñar los usuarios podrán personalizar su perfil, realizar ventas o compras de logotipos, banners, etc. Entre otras funciones nombradas en otros puntos de la documentación. | Prueba de interfaz de usuario e integridad de datos y base de datos. | Browsera |
| Requerimiento no funcional de **Usabilidad**: Al ser un sitio web enfocado en todas las edades será lo más intuitivo y fácil de usar para todo tipo de usuario que lo desee. Es decir, botones que resalten del demás contenido, sin haber ventanas innecesarias, entre otras buenas prácticas.  Mostrar notas sobre qué hace cada botón al poner el mouse encima, esto con la finalidad de que se facilite el entendimiento de cosas que se dificultan para algunos usuarios. Que no sea necesaria una curva de aprendizaje tan grande para utilizar el servicio a diseñar. | Prueba de carga. | Jmeter |
| Requerimiento no funcional de **Usabilidad**: La página debería estar disponible al menos un 99,96% del tiempo y sólo estará de baja en caso de mantención una vez cada 2 semanas, y el mantenimiento durará 1 hora. | Prueba de Integridad de Datos y Base de Datos | Jmeter |
| Requerimiento no funcional de **Usabilidad**: El sistema a diseñar contará con un modo oscuro y posibilidad de cambiar el tamaño de las fuentes para que personas con dificultades visuales puedan utilizar el servicio sin complicaciones. | Prueba de Estilo. | Browsera |
| Requerimiento no funcional de **Usabilidad**: La interfaz será intuitiva y con colores agradables para todo tipo de usuarios, aunque se utilice el modo oscuro que se implementará a futuro. Además de eliminar opciones innecesarias que entorpezcan la usabilidad. | Prueba de interfaz de usuario. | Browsera |
| Requerimiento no funcional de **Usabilidad**: El sistema a diseñar mostrará la función que cumple en un breve apartado en la página de preguntas y respuestas o al poner el mouse encima de algún botón, mostrará una breve descripción de que a donde lo llevará. | Prueba de interfaz de usuario. | Browsera |
| Requerimiento no funcional de **Mantenibilidad**: El sistema a desarrollar será realizado en modular, es decir, con clases separadas para cada función y subfunción, con la finalidad de poder implementar cambios sin entorpecer el cambio sin un gran impacto en el sistema. | Prueba de fiabilidad. | SUCURI |
| Requerimiento de Adecuación **funcional**: El sistema contará con la documentación exhaustiva que permitirá analizar las tareas que se desean implementar y los objetivos que tiene el proyecto, para que estos puedan ser cumplidos y el sistema a diseñar pueda proporcionar todas las funciones solicitadas. | Prueba de Documentación y procedimiento. | VisualCode |

# Diseños de caso de prueba a nivel funcional

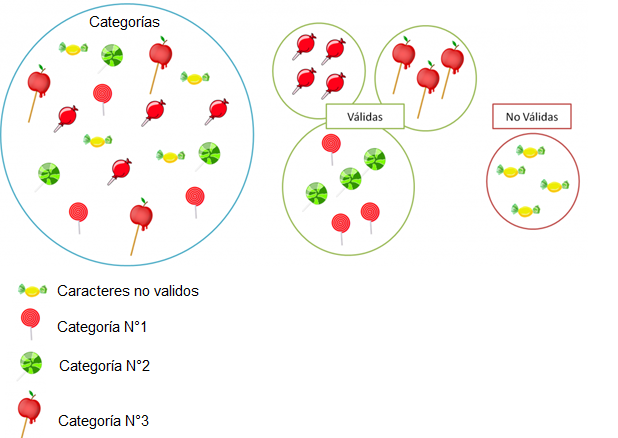
## Particionamiento de equivalencia

Para definir, la división en particiones de equivalencia, también llamadas clases de equivalencia, es una técnica de pruebas de caja negra en la que las entradas del programa se clasifican en clases. Considera como perteneciente al dominio de entrada a todas las entradas que puedan tener efecto en la salida de la prueba, positivo o negativo.

En el caso de nuestro proyecto puede separarse entre las categorías que poseerán los proyectos de los usuarios, por ejemplo:

* Categoría: Arte contemporáneo.
* Categoría: Arte 90’s
* Categoría: Puntillismo
* Etc.

Para poder entender de manera más clara y fácil, se mostrará en una figura como funcionaría un particionamiento de equivalencia, en nuestro sistema:



Para explicarlo, aquí puede verse que ciertos caracteres no son válidos al momento de ingresar una categoría, lo que permite tener un filtro y poder separar entre categorías validas e invalidas. Con invalidas puede ser, intentar ingresar un valor fuera de los permitidos como por ejemplo un “!?” u otros caracteres similares a estos mencionados. Ahora se mostrará cómo se ingresaría un dato para, efectivamente ingresar un caso de prueba de tipo funcional con esta subcategoría.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Caso de prueba | Clase de equivalencia | Condiciones de entrada | | | Resultado |
| Nombre | Descripción | ID |
| CP006 | Nombre\_cat | A - Z | A – Z | Automático | IDcat\_001 |

## Análisis de frontera

Algunas de las funciones de este análisis son las siguientes:

* Parte del principio que el comportamiento al borde de una partición de datos tiene mayores probabilidades de presentar errores (bugs).
* Los valores máximos y mínimos de una partición son sus valores bordes.
* Aplican tanto para datos inválidos como inválidos.
* Al incluirlas en el diseño de casos de prueba, se define una prueba por cada valor borde.
* La capacidad de identificar defectos de esta técnica es alta, ser pueden revisar las especificaciones funcionales para identificar datos interesantes.

Es decir que, para nuestro sistema, esta prueba podría ejercerse al momento de ingresar los montos de los servicios que pongan los usuarios, por ejemplo:

* El valor mínimo que podrá tener un producto es de 1 USD (Actualmente (al día 08/06/2021) es de 717.80 pesos chilenos.)
* Y no podrá tener un monto superior a los 99999 USD (Actualmente (Al día 08/06/2021) el valor es de 71779282,20 pesos chilenos.)

Para ejemplificar de manera más simple, se realizará la siguiente muestra:

Valor menor a (99999 USD)

Valor mayor a (1 USD)

Valor valido

Y si hubiese que mostrarlo en una tabla:

|  |  |  |  |
| --- | --- | --- | --- |
| **N°** | **Caso de prueba** |  | **Representante** |
| 1 | Q < 0 | No valido | * 1 |
| 2 | Q = 0 | No valido | 0 |

## Tabla de decisión

La función o algunas funciones de las tablas de decisión son las siguientes:

* Las tablas de decisión son una herramienta útil para documentar reglas de negocio de alta complejidad que el sistema debe cumplir.
* Las tablas de decisión se crean a partir del análisis de la especificación funcional y la identificación de estas reglas de negocio.
* Las condiciones de entrada y acciones se expresan a menudo en términos de verdadero o falso.
* Las tablas de decisión contienen las condiciones desencadenantes, que son la combinación de valores de verdadero o falso para cada entrada de datos, así como la acción que resulta de cada combinación.
* Cada columna de la tabla corresponde con una regla de negocio que representan la combinación de condiciones, y las acciones que resultan.

El ejemplo más claro, fuera de nuestro servicio a diseñar sería el utilizar un cajero e ingresar correctamente la clave, monto máximo a retirar, etc.

En nuestro caso puede ceñirse a lo que se refiere al inicio de sesión, modificación de datos de los usuarios y si su contraseña es ingresada para hacer efectivos los cambios, entre otras cosas.

En una tabla se vería lo siguiente:

|  |  |  |
| --- | --- | --- |
| Clave: | OK | Nok |
|  |  | **X** |
|  | Redirigido al perfil del usuario. | Notificación error. |

Si el usuario ingresa correctamente, es redirigido a su perfil y sino, se le muestra un mensaje de error, que verifique los datos ingresados.

## Transición estados

Ciertas características de la transición entre estados son las siguientes:

* Un sistema puede presentar diferentes comportamientos según su estado actual o eventos previos.
* Este aspecto del sistema se puede representar en un diagrama de transición entre estados.
* El diagrama de estados, permite al Tester visualizar los estados, transiciones, entradas de datos o eventos que las desencadenan y las acciones que pueden resultar.
* Una tabla de estados, muestra las relaciones entre los estados y las entradas de datos. Puede ayudar a identificar posibles transacciones inválidas.

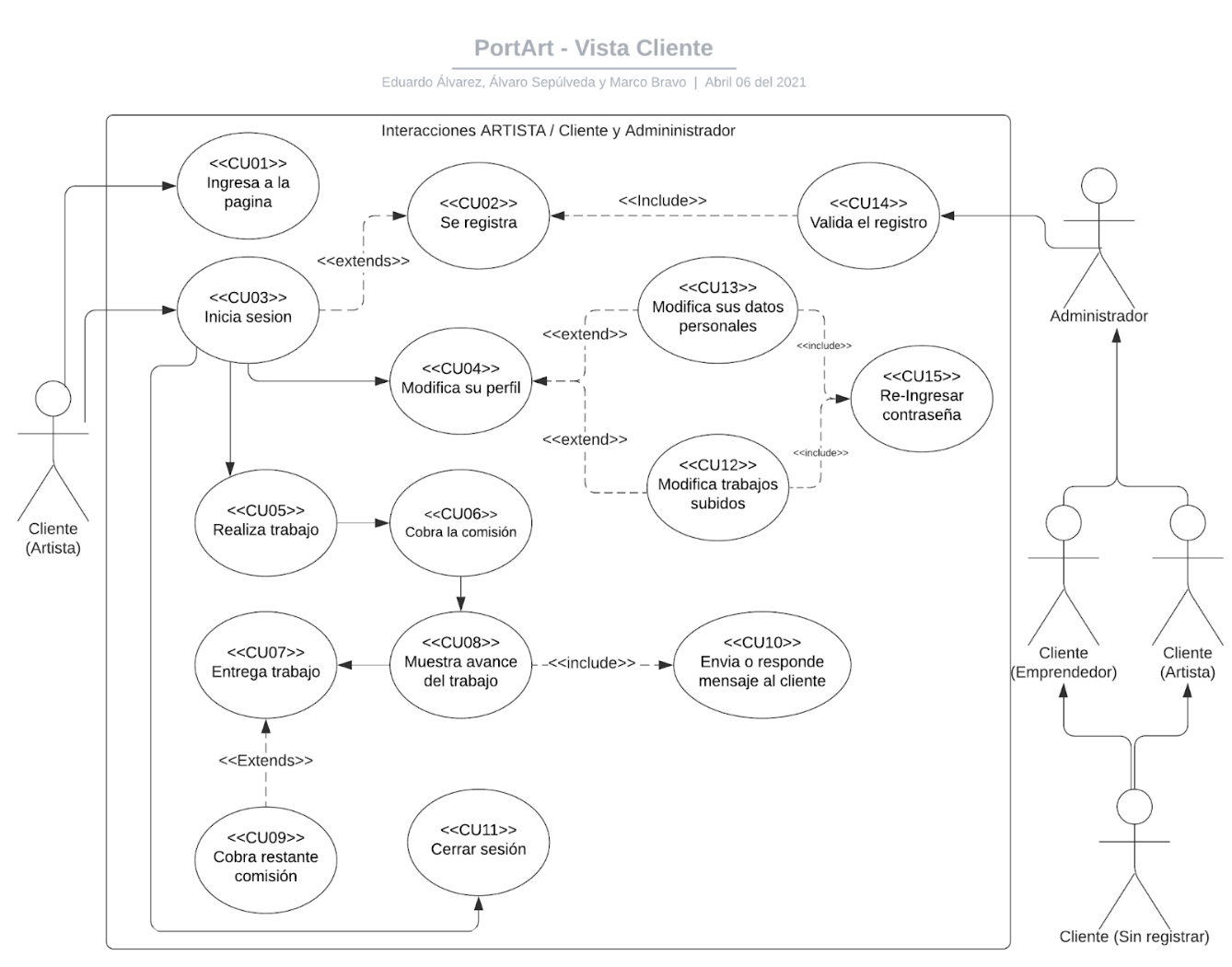
Un ejemplo de uso que tendrían en nuestro sistema sería que, si el usuario permanece cierto tiempo inactivo en la página, su sesión será cerrada por seguridad. Esto podría mostrarse con un diagrama similar a un BMPN.

Se comienza el proceso, el usuario realiza sus actividades en el servicio una vez logeado, si por algún motivo este se va y deja la ventana en segundo plano, su sesión se cerrará después de cinco minutos de espera.

## Casos de uso

Como se ha mostrado anteriormente los Casos de Uso, cumplen la función de mostrar cómo será el funcionamiento de la plataforma, las acciones que realizarán los actores con la finalidad de darle un mayor y mejor entendimiento a los usuarios o programadores encargados de tanto utilizar el sistema como de programarlo o diagramarlo.

Un ejemplo de cómo funcionaría o se vería un CU en nuestro sistema sería el siguiente, que serán las funciones que realizarán los usuarios en el sistema:



# Herramientas utilizadas para pruebas de caja blanca.

# Plan de pruebas

La actividad será desarrollada en un nuevo documento, para facilitar la comprensión.