**QA challeng**

**Nombre:** Juan Manuel Paredes Goicochea

**Fecha:** 05/10/2022

**Link challenge:** https://github.com/AdmiPAO/ChallengeQA

**Link github solucion:**

**RETO 1**

En el área de TI de UTP surgió el requerimiento de realizar un nuevo proceso de autenticación. Nuestro scrum master registro los siguientes requerimientos:

* Nueva funcionalidad de login
* Nueva funcionalidad de registro

**Historia de usuario**

Como usuario nuevo deseo poder registrarme en la web y poder loguearme de manera satisfactoria

**Criterios de aceptación**

Los criterios de aceptación para el registro son los siguientes:

* El correo ingresado debe tener un @ para ser válido.
* La contraseña debe tener como mínimo 5 caracteres.
* El correo y la contraseña son requeridos obligatoriamente.
* El sistema debe mostrar un mensaje de error al ingresar campos no válidos.

Los criterios de aceptación para el login son los siguientes:

* El correo ingresado debe tener un @ para ser válido.
* La contraseña debe tener como mínimo 5 caracteres.
* El correo y la contraseña son requeridos obligatoriamente.
* El sistema debe mostrar un mensaje de error al ingresar campos no válidos.

**SOLUCION:**

1. **Identifica los tipos de casos de prueba que estarían enlazadas a los requerimientos propuestos.**

|  |  |
| --- | --- |
| TIPO DE PRUEBA | DEFINICION EN EL CASO |
| Pruebas funcionales | En este caso las pruebas funcionales ayudarían a detectar errores en los flujos de pruebas E2E. Basados en una matriz de pruebas podemos definir diferentes casuísticas |
| Pruebas no funcionales | Probar la estabilidad y tiempos de respuesta del servidor al intentar registrarse y loguearse. Para esta prueba podemos ayudarnos de herramientas como jmeter y postman. |
| Pruebas de caja blanca | Probar la estructura del código y así verificar que se están cumpliendo buenas prácticas en el código desarrollado para el registro y login. |
| Pruebas asociadas al cambio (regresión) | Después de realizar e implementar mejoras al código, podemos pasar a realizar pruebas de regresión a través de la GUI o de APIs tambien. |

1. **Menciona que técnica de diseño de casos de prueba usarías para poder generar dichos casos.**

|  |  |
| --- | --- |
| TECNICA DE DISEÑO DE CASOS DE PRUEBA | DEFINICION Y USO |
| Técnica de casos de prueba basados ​​en la experiencia. | En base a la experiencia y al tratarse de un registro y login de usuario, podemos identificar los happy paths y errores controlados comunes en ese tipo de funcionalidades comunes. |
| Creación de matriz de pruebas | La lista de casuisticas en la matriz de pruebas nos ayudan controlar mas a detalle los resultados esperados vs los resultados obtenidos, para evaluar si se cumplen los objetivos en base a los requisitos o criterios de aceptación. |
| Pruebas de humo | Una prueba rápida de los que es un registro y login de usuario, nos ayuda a probar la estabilidad y funcionalidad del módulo o sistema. |

1. **Identifica, registra y describe los bugs o mejoras que consideres necesarios.**

|  |  |
| --- | --- |
| **Funcionalidad** | **Mejora** |
| Mostrar los mensajes de validación de los campos. | Mostrarlos a través de popups o mensajes visibles de advertencia que adviertan al usuario de algún error en los datos ingresados. |
| Interfaz grafica | La presentación o la interfaz puede mejorar y mostrar un entorno más amigable al usuario. |
| Experiencia de usuario | Visualizar al usuario una cuadro de los datos validos para el correcto registro y login. |

**Reporte de bug y evidencias**

**Pruebas en Backend:**

* 1. **Issue código:** CP001

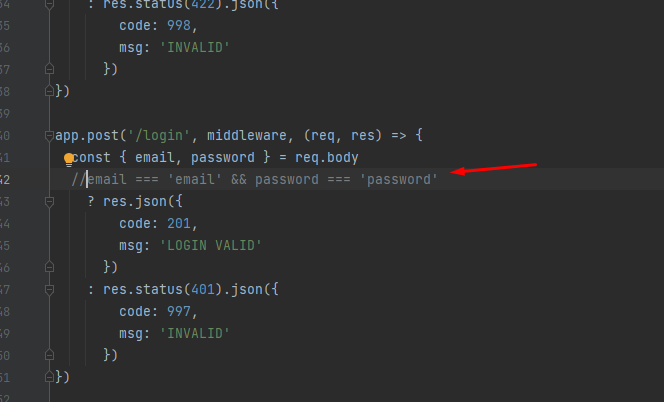
**Título:** Descomentar y comentar las líneas de código.

**Descripción:** en el archivo server.js del back, se encontró una línea comentada de código impidiendo validar los mensajes “LOGIN VALID” en el registro de usuario.

**Tipo de prueba: caja blanca**

**Criticidad:** Alta

**Imagen referencial:**



**Pruebas en Frontend:**

* 1. **Issue código:** CP002

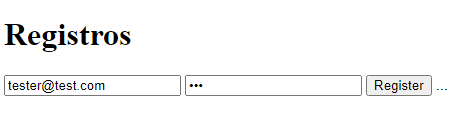
**Título:** Formulario de registro, validación de campos

**Descripción:** En el formulario registro, en el campo password no valida el requerimiento de 5 caracteres como mínimo, el resultado obtenido es INVALID.

**Tipo de prueba: funcional**

**Criticidad:** Media

**Imagen referencial:**





* 1. **Issue código:** CP003

**Título:** Validar el campo correo del formulario Login.

**Descripción: V**alidar que el campo correo cumpla con las validaciones y requerimientos esperados, en este caso no se registro un correo previamente y aun así la respuesta que se obtiene es LOGIN VALID o ingreso valido.

**Tipo: funcional**

**Imagen referencial:**





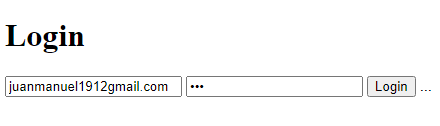
* 1. **Issue código:** CP004

**Título:** Validar el campo email del formulario login con arroba.

**Descripción:** actualmente muestra un mensaje satisfactorio si se ingresa un correo sin arroba.

**Tipo: funcional**

**Imagen referencial:**





* 1. **Issue código:** CP005

**Título:** Validar el campo password del login con 5 caracteres.

**Descripción: según el criterio de aceptación se debe validar que no ingrese menos de 5 caracteres en el campo password. Actualmente acepta un mínimo de 5 caracteres.**

**Tipo: funcional**

**Imagen referencial:**





* 1. **Issue código:** CP006

**Título:** Validar mensajes de error en los campos del formulario login.

**Descripción:** actualmente no muestra una correcta validación de los campos al ingresar cualquier tipo de carácter en el campo correo.

**Tipo: funcional**

**Imagen referencial:**



1. **Implementa los tipos de pruebas automatizadas que consideres necesarios (unitarias, servicios, ui , perfomance), de tal manera que se pueda evaluar los requerimientos solicitados. Se debe crear un proyecto en una carpeta aparte llamada qaChallenge (utiliza el framework de automatización que prefieras y deberas subirlo a tu repositorio).**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tipo de automatización** | **Herramienta y tecnologías** | **Nivel de prueba** | **Patrón de diseño** | **Link del repositorio** |
| **Automatización front-end** | **Selenium, gerkin, java** | **Pruebas a nivel de GUI** | **screentplay** |  |

**Casos de prueba:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **código** | **Caso de prueba** | **descripción** | **Resultado esperado** | **Resultado obtenido** | **status** |
| CP01 | Ingresar correo valido y password valido en el formulario Registro | Validar que ingresando un dato valido en el campo correo y otro valido en el campo password permita el correcto registro. | Que al ingresar un correo con símbolo arroba, permita el registro correctamente | Permite el registro correctamente | satisfactorio |
| CP02 | Ingresar correo invalido y password valido en el formulario Registro | Validar que ingresando un dato invalido en el campo correo y otro valido en el campo password NO permita el correcto registro. | Que al ingresar un correo sin símbolo arroba, no permita el registro. | No permite el registro. | satisfactorio |
| CP03 | Ingresar correo valido y password invalido en el formulario registro | Validar que ingresando un dato valido en el campo correo y otro invalido en el campo password NO permita el correcto registro. | Que al ingresar un pasword de 1 o 2 caracteres no, permita el registro. | No permite el registro | satisfactorio |
| CP04 | Ingresar correo invalido y password invalido en el formulario registro | Validar que ingresando un dato invalido en el campo correo y otro invalido en el campo password NO permita el registro. | Que al ingresar un correo sin símbolo arroba, y un password con 2 caracteres NO permita el registro. | NO permite el registro correctamente | satisfactorio |
| CP05 | Ingresar correo valido y password valido en el formulario Login | Validar que ingresando un dato valido en el campo correo y otro valido en el campo password permita el correcto ingreso. | Que al ingresar un correo con símbolo arroba, y passwor mayor a 5 caracteres, permita el Login correctamente | Permite el ingreso correctamente | satisfactorio |
| CP06 | Ingresar correo invalido y password valido en el formulario Login | Validar que ingresando un dato invalido en el campo correo y otro valido en el campo password NO permita el correcto ingreso. | Que al ingresar un correo sin símbolo arroba, y un password valido con mínimo 5 caracteres, NO permita el Login. | No permite el Login. | satisfactorio |
| CP07 | Ingresar correo valido y password invalido en el formulario login | Validar que ingresando un dato valido en el campo correo y otro invalido en el campo password NO permita el correcto ingreso. | Que al ingresar un password de 1 o 2 caracteres no, permita el login. | No permite el login | satisfactorio |
| CP08 | Ingresar correo invalido y password invalido en el formulario login | Validar que ingresando un dato invalido en el campo correo y otro invalido en el campo password NO permita el login. | Que al ingresar un correo sin símbolo arroba, y un password con 2 caracteres NO permita el login. | NO permite el login | satisfactorio |
| CP09 | Registrarse con campos vacíos en formulario registro | Validar que no ingresando data en el campo correo y password NO permita el correcto registro. | Validar que al NO ingresar data, no permita el registro. | NO Permite el registro | satisfactorio |
| CP010 | Loguearse con campos vacíos en el formulario login | Validar que no ingresando data en el campo correo y password NO permita el login. | Validar que al NO ingresar data, no permita el login. | NO Permite el login | satisfactorio |

**RETO 2**

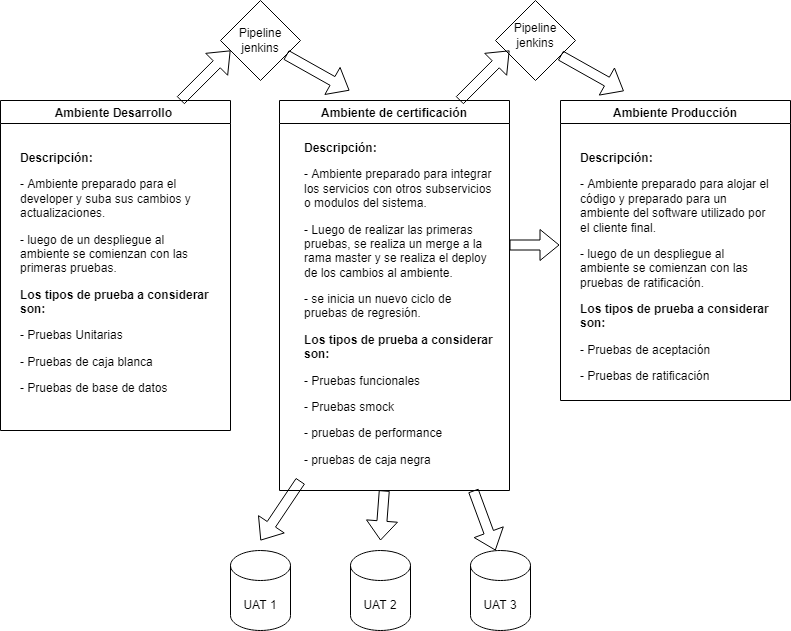
### Entornos de trabajo y niveles de pruebas

El equipo maneja varios ambientes de trabajo (desarrollo, qa, uat, produccion) a lo largo del desarrollo del producto. Consideramos que la calidad debe asegurarse en todo el flujo, por lo tanto, diferentes pruebas debes realizarse en cada uno de los ambientes propuestos.

**Objetivos**

1. Comenta en base a tu criterio que nivel(es) de pruebas aplicaría para cada ambiente (puedes sugerir otros ambientes u estructura si es que lo consideras) (Coloca tus respuestas en el documento ( qa-challenge-reto[nombre].docx ) )

**SOLUCION:**

Ambientes en desarrollo del software:

**RETO 3**

**Priorización de errores**

**Caso:**

Imagina una herramienta web, desarrollada con el objetivo de proporcionar una versión de ancho de banda ligero (se requiere un uso óptimo alrededor de 50 MB) de una herramienta principalmente utilizada por personas de todo el mundo para comprar productos de segunda mano.

Defectos:

1. Los precios no varían cuando una transacción se realiza en diferentes monedas.
2. Las transacciones entre usuarios no se reflejan en un informe de back-end.
3. Hay algunos errores ortográficos en la página de inicio.
4. Las credenciales de usuario no están cifradas en la base de datos.
5. Los foros oficiales de soporte no siempre registran los comentarios del usuario.
6. El ancho de banda necesario para una experiencia de usuario óptima es una velocidad de descarga de 100 MB.

**Objetivos**

Para el siguiente conjunto de defectos, prioriza del 1 al 6, siendo el primero el más crítico y explique brevemente por qué (puedes incluir alguna suposición). Coloca tus respuestas en el documento ( qa-challenge-reto[nombre].docx )

**SOLUCION:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N°** | **Defecto** | **Nivel de criticidad** | **Impacto** | **Criterio o mejora** |
| **1** | Las transacciones entre usuarios no se reflejan en un informe de back-end. | **alta** | Al requerir un informe de log para detectar vulnerabilidades o stoppers en el flujo seria un caos detectar donde esta el bug. | Debe de registrarse cada flujo y acción que realiza el usuario y visualizarse con facilidad para detectar errores tempranos a través de los log. |
| **2** | Las credenciales de usuario no están cifradas en la base de datos. | **alta** | Cualquier persona que hakea la base de datos puede robar información sensible del usuario. | Es importante guardar y encriptar los datos sensibles del usuario que impliquen perdidas y perjudiquen al negocio. |
| **3** | El ancho de banda necesario para una experiencia de usuario óptima es una velocidad de descarga de 100 MB. | **Media** | Si los usuarios detectan demoras de carga de segundos al navegar por la página, fácilmente se irán. | Es importante hacer pruebas de performance y verificar que la pagina responda rápidamente a las peticiones del servidor. |
| **4** | Los precios no varían cuando una transacción se realiza en diferentes monedas. | **Media** | Si los precios se mantienen altos en zonas donde la economía es diferente, fácilmente podría haber perdidas para el negocio. | La página debería colocar sus políticas y términos de condiciones, visibles para que el cliente pueda estar informado de los precios y actualizaciones. |
| **5** | Los foros oficiales de soporte no siempre registran los comentarios del usuario. | **Media** | Si un usuario detecta una mala experiencia en el sitio web, perjudica la imagen de la empresa. | Para mantener una mejora continua, siempre es importante recibir y escuchar la opinión de los clientes a través de la plataforma. |
| **6** | Hay algunos errores ortográficos en la página de inicio | **baja** | Esto no perjudica la funcionalidad, pero si la imagen y la seriedad de la empresa. | Es bueno mostrar profesionalismo en todo sentido al momento de ofrecer los productos o servicios por la página web. |

**RETO 4**

**Agile testing**

**Caso:**

La aplicación web a probar es una plataforma como Mercado Libre. A continuación, podrás encontrar una lista de técnicas y procedimientos utilizados en las pruebas

El equipo de pruebas:

1. Utiliza el mismo entorno que los usuarios finales para realizar las pruebas perfomance.
2. Las pruebas de humo se ejecutan para cubrir todas las posibles opciones del usuario y se realizan en cada implementación de una nueva versión, todas las pruebas se realizan manualmente para reflejar la misma experiencia del usuario.
3. Las pruebas de regresión solo cubren los módulos que probablemente se vean afectados por los cambios realizados en la última versión.
4. Los casos de prueba se crean en Excel para que sean fáciles de editar. Los casos de prueba se eliminan después de la implementación de esa versión, por lo que se crean nuevos completos en cada iteración.
5. Las pruebas comienzan a estar involucradas en el proceso de desarrollo desde la etapa inicial.

**Objetivos**

Evalúa si son los procedimientos de la lista son los más adecuados. Brinda una alternativa / tipo de prueba o mejora para las que no son adecuadas desde tu punto de vista. (Coloca tus respuestas en el documento ( qa-challenge-reto[nombre].docx )

**SOLUCION:**

|  |  |
| --- | --- |
| **Inventario de las pruebas utilizadas por el equipo de desarrollo (Antes)** | **Implementar estas mejoras (Ahora)** |
| Utiliza el mismo entorno que los usuarios finales para realizar las pruebas perfomance. | Deben de utilizar ambientes separados de pruebas, sugiero 3 ambientes (dev, certi, prod), dos de ellos pre-productivos (dev, y certi) donde se pueden realizar pruebas unitarias, funcionales y de performance. |
| Las pruebas de humo se ejecutan para cubrir todas las posibles opciones del usuario y se realizan en cada implementación de una nueva versión, todas las pruebas se realizan manualmente para reflejar la misma experiencia del usuario. | Se deben implementar y desarrollar una serie de casuísticas en una matriz de pruebas, en base a eso comenzar a automatizar algunos casos de prueba que ayuden a las pruebas de regresión donde los flujos repetitivos ya no sean ejecutados manualmente, sino automatizados. |
| Las pruebas de regresión solo cubren los módulos que probablemente se vean afectados por los cambios realizados en la última versión. | Las pruebas de regresión en base a casos de pruebas de cumplir funcionalidades E2E.  De esa manera se asegura una calidad en el software mas optima y más verídica. |
| Los casos de prueba se crean en Excel para que sean fáciles de editar. Los casos de prueba se eliminan después de la implementación de esa versión, por lo que se crean nuevos completos en cada iteración. | Se pueden utilizar herramientas como jira para registrar los casos de prueba y no se eliminados fácilmente, ya que ayuda con el historial de cambios y los diferentes ciclos de prueba que el software requiere. |
| Las pruebas comienzan a estar involucradas en el proceso de desarrollo desde la etapa inicial. | Las pruebas deben iniciar antes, a partir de las historias de usuario y criterios de aceptación, así detectamos defectos tempranamente y aseguramos una mejor calidad desde etapas mas tempranas, mejorando el ROI del cliente también. |