**QA CHALLENGE**

**PLAN DE PRUEBAS**

**Proyecto:** Retos QA Challenge

Versión: 1.0

Historia de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Autor(es) | Descripción | Fecha |
| 1.0 | Jair Vega Majin | Actualización de proceso Login y Registro | Jun-21 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Índice

[1 Introducción 4](#_Toc73928256)

[1.1 Alcance 4](#_Toc73928257)

[1.2 Referencias 4](#_Toc73928258)

[1.3 Glosario 4](#_Toc73928259)

[2 Contexto de las Pruebas 4](#_Toc73928260)

[2.1 Prueba (RETO 1) 4](#_Toc73928261)

[2.1.1 Objetivos 4](#_Toc73928262)

[2.1.2 Tipos de Casos de Pruebas 5](#_Toc73928263)

[2.1.3 Técnica de Pruebas 5](#_Toc73928264)

[2.1.4 Registro y Descripción de Bug 5](#_Toc73928265)

[2.2 Prueba (RETO 2) 8](#_Toc73928266)

[2.2.1 Objetivos 8](#_Toc73928267)

[2.2.2 Niveles de Pruebas 8](#_Toc73928268)

[2.3 Prueba (RETO 3) 9](#_Toc73928269)

[2.3.1 Objetivos 9](#_Toc73928270)

[2.3.2 Priorización de Defectos 9](#_Toc73928271)

[2.4 Prueba (RETO 4) 9](#_Toc73928272)

[2.4.1 Objetivos 9](#_Toc73928273)

[2.4.2 Evaluación a los procedimientos 9](#_Toc73928274)

# Introducción

## Alcance

El propósito del QA Challenge es poder resolver y documentar los **retos** propuestos por la empresa y posteriormente ser enviados al repositorio GitHUb.

## Referencias

* Especificación de RETO 1
* Especificación de RETO 2
* Especificación de RETO 3
* Especificación de RETO 4

## Glosario

En este documento se utilizan los siguientes términos abreviados:

* UAT: Pruebas de Aceptación del Usuario (del inglés User Acceptance Test)
* DEV: Desarrollador (del inglés Development)
* QA: Analista de Calidad (del inglés Quality Assurance)
* PROD: Producción (del inglés Production)

# Contexto de las Pruebas

## Prueba (RETO 1)

### Objetivos

* Identifica los tipos de casos de prueba que estarían enlazadas a los requerimientos propuestos.
* Menciona que técnica de diseño de casos de prueba usarías para poder generar dichos casos,
* Identifica, registra y describe los bugs o mejoras que consideres necesarios.
* 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 deberás subirlo a tu repositorio).

### Tipos de Casos de Pruebas

* Pruebas Funcionales
* Pruebas No funcionales de software
* Pruebas de regresión por cambios

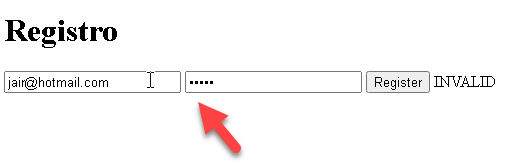
### Técnica de Pruebas

* Prueba y Cobertura de Decisión
* Predicción de Errores
* Prueba Exploratoria
* Requisitos del Cliente
* Objetivos de Pruebas
* Identifica los tipos de casos de prueba que estarían enlazadas a los requerimientos propuestos.
* Tipo de Componente o sistema
* Conocimientos y Competencias del Probador
* Herramientas Disponibles

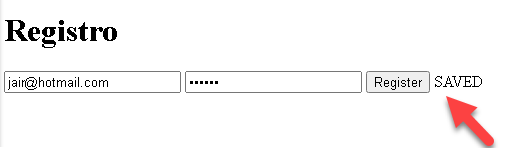
### Registro y Descripción de Bug

#### Defectos Encontrados

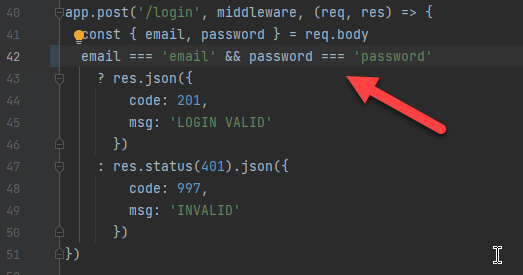
* El **Registro** no valida el criterio donde se debe permitir un mínimo de “5 caracteres” al realizar el registro de un usuario. En la imagen se muestra que con 5 caracteres ingresados el mensaje devuelto es “INVALID”.



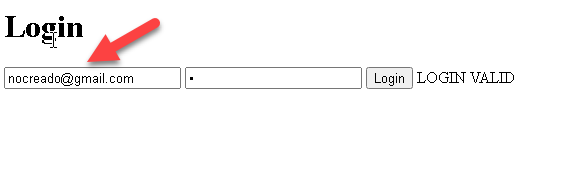
* El **Registro** permite un mínimo de 6 **caracteres** como mínimo diferente al criterio que aceptación que corresponde, devolviendo el mensaje “SAVED”.



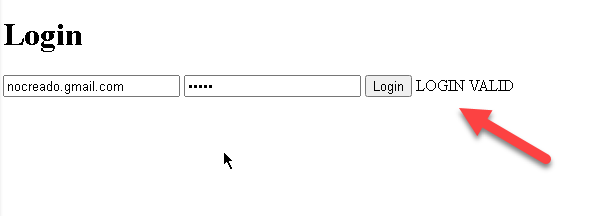
* En el **Login** el archivo “**server.js**” del “**back**” se retira el comentario en la línea “**42**” para que el Login valide el “**email**” y “**password**”.



* El **Login** no está validando si el usuario existe previo registro, en la siguiente prueba se agrego un correo no registrado y el mensaje fue exitoso “LOGIN VALID”.



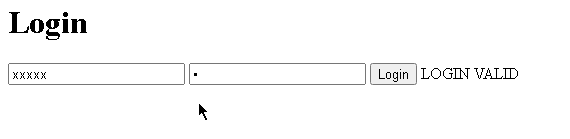
* El **Login** no valida si el correo ingresado tiene un “@”, permite ingresar cualquier correo, mostrando el mensaje exitoso “LOGIN VALID”



* El **Login** no valida el ingreso de al menos “5 caracteres” según el criterio de aceptación se ingresó “1 carácter” y el resultado fue “LOGIN VALID”.



* El **Login** no muestra un mensaje de error al ingresar campos no válidos, solo realiza la validación cuando los 2 campos no están completos.



#### Mejoras Por Considerar

* En la sección de **Login** y **Registro** se deben mostrar con un PopUP o ventana emergente al presionar el botón que corresponde, en vez de mostrar un mensaje al lado del botón.
* Por experiencia de **Usuario**, el TextBox y Button se podrían mostrar mejor de manera vertical y Centrada.
* Se podría agregar un spinner de “**cargando**” al accionar el botón de **Login** y **Register**.
* Por experiencia de **Usuario** se podría agregar mayúsculas en los TextBox de **Email** y **Password**.

#### Pruebas Automatizadas

* Pruebas de Interface subido al GitHub del repositorio.
* Se utilizará las siguientes tecnologías para automatizar el Proyecto:
* VS Code (IDE)
* GitHub (Repositorio)
* Selenium (Suite de Pruebas)
* Java (Lenguaje de programación)
* Maven (Herramienta constructor)
* Gradle (Herramienta constructor)

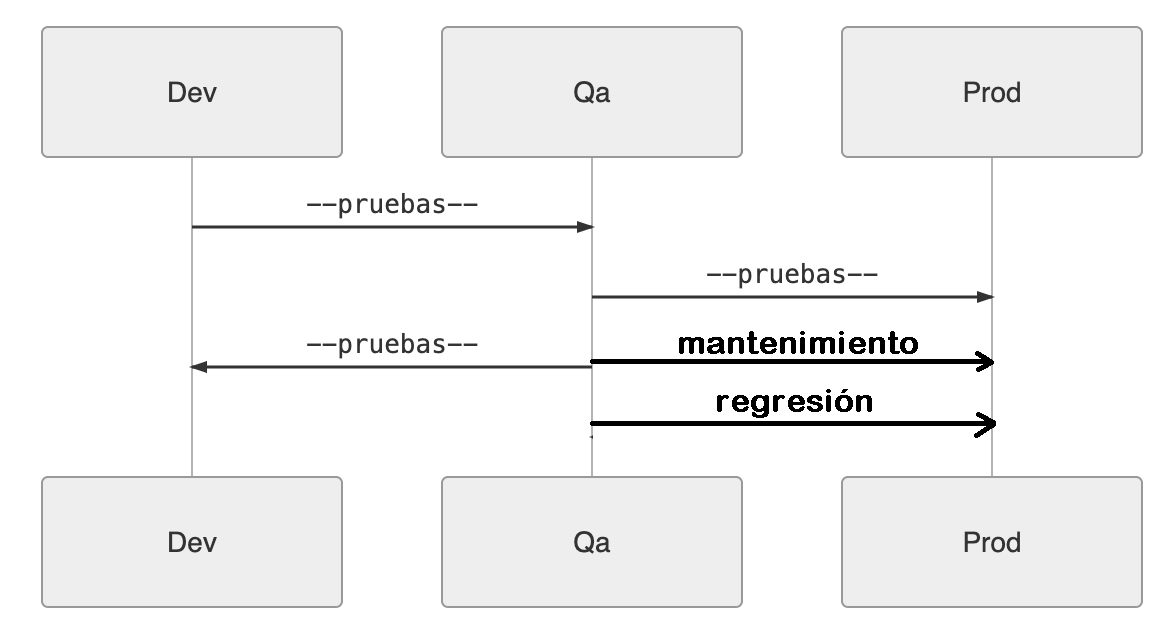
## Prueba (RETO 2)

### Objetivos

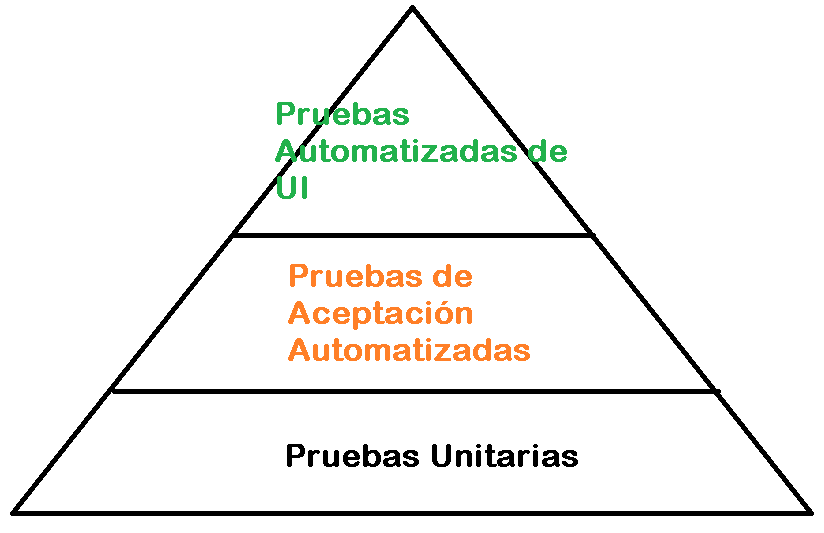
* Comenta en base a tu criterio que nivel(es) de pruebas aplicaría para cada ambiente (puedes sugerir otros ambientes u estructura si es que lo consideras) (Coloca tus respuestas en el documento ( qa-challenge-reto[nombre].docx ) )

### Niveles de Pruebas

* En los ambientes se puede mejorar el flujo de pruebas de la siguiente manera, para evitar en su mayoría o casi nula cantidad de defectos a producción, las pruebas exhaustivas es una falacia principio del ISTQB.



* En los Niveles de Prueba se puede implementar principios y practicas agiles, permitiéndonos crear un software con mejor diseño, entendimiento en general de la mayor calidad posible.



## Prueba (RETO 3)

### 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 )

### Priorización de Defectos

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

## Prueba (RETO 4)

### Objetivos

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

### Evaluación a los procedimientos

* Utiliza el mismo entorno que los usuarios finales para realizar las pruebas performance**. (Riesgo crítico, se debe contar con un ambiente de pruebas es obligatorio para todo proyecto de software)**
* 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. **(Antes de realizar las pruebas de humo es necesario probar todos los criterios de aceptación en el ambiente de CERTIFICACIÓN la técnica a utilizar para cubrir todos los casos puede ser la *Tabla de Decisión*).**
* 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 deben cubrir todos los criterios de aceptación, casos de pruebas creados por el QA y aprobados por el Product Owner en la certificación antes del pase a PRODUCCIÓN).**
* 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. **(En este caso es sumamente importante utilizar una herramienta de pruebas, una de las más completas hoy en día es JIRA que es de pago pero también se puede optar por una OpenSource como Testlink con esto se podrá llevar un historial y control de las pruebas creadas, registradas y ejecutadas a los largo de la vida del software).**
* Las pruebas comienzan a estar involucradas en el proceso de desarrollo desde la etapa inicial. **(Esta es una técnica AGIL, sin embargo, se debe contar con cierta experiencia en testeo de código para poder estar involucrado al 100% desde el inicio del software, la curva de aprendizaje es mayor por ende dependerá del nivel de conocimiento en código del QA a cargo).**