



PLAN DE PRUEBAS

NOMBRE PROYECTO

Versión 1.0

CONTROL DE CAMBIOS AL PLAN

VERSIÓN	FECHA	CAMBIOS RESPECTO DE LA VERSIÓN ANTERIOR	PREPARADO POR	APROBADO POR
1.0	21/08/2023	Versión Inicial	Nombre del QA	Nombre QA Lead

CONTROL DE CAMBIOS TEMPLATE

VERSIÓN	FECHA	CAMBIOS RESPECTO DE LA VERSIÓN ANTERIOR	PREPARADO POR	APROBADO POR
1.0	Agosto 2023	Versión Inicial	Nombre QA	Nombre QA Lead

Tabla de Contenidos

1. INTRODUCCIÓN.....	4
1.1. OBJETIVO DEL PLAN DE PRUEBAS.....	4
1.2. DOCUMENTOS RELACIONADOS.....	4
2. ALCANCE DE LAS PRUEBAS.....	5
2.1. REQUERIMIENTOS DE PRUEBAS INCLUIDOS.....	5
2.2. REQUERIMIENTOS DE PRUEBAS EXCLUIDOS.....	6
2.3. CASOS DE PRUEBAS INCLUIDOS.....	7
2.4. CASOS DE PRUEBAS EXCLUIDOS.....	8
2.5. ENTORNO DE PRUEBAS.....	9
3. ESTRATEGIA DE LAS PRUEBAS.....	10
3.1. CONFIGURACION DE LAS PRUEBAS.....	10
3.2. ORDEN DE EJECUCION DE LAS PRUEBAS.....	11
3.3. CRITERIOS DE INICIO Y TERMINOS DE LAS PRUEBAS.....	13
3.4. EQUIPOS DE PRUEBAS Y RESPONSABILIDADES.....	14
3.5. GESTION DE INCIDENCIAS.....	15
4. ANEXOS.....	16
4.1. ENTREGABLES.....	16

1. INTRODUCCIÓN

1.1. OBJETIVO DEL PLAN DE PRUEBAS

El plan de pruebas tiene como objetivo identificar claramente cuál es el alcance respecto a lo que se incluye y excluye de las actividades de testeo, el orden de ejecución de las pruebas, los recursos necesarios tanto de data como de ambientes o infraestructura, los criterios de inicio y término, participantes del proyecto y todo lo relevante para llevar a cabo la actividad y finalmente aprobar el “paso a producción” de la solución, para así asegurarnos de satisfacer los requerimientos definidos por los usuarios de las distintas áreas

1.2. DOCUMENTOS RELACIONADOS

NOMBRE	DESCRIPCIÓN
Repositorio del proyecto	Link
Issue tracker	Link
Sharepoint	Link
Casos de prueba	Link

2. ALCANCE DE LAS PRUEBAS

2.1. REQUERIMIENTOS DE PRUEBAS INCLUIDOS

Explicación de que es lo que se va a testear. (Alcance/Scope del testing)

2.2. REQUERIMIENTOS DE PRUEBAS EXCLUIDOS

REQUERIMIENTOS	MOTIVO
Performance	Fuera del scope
Seguridad	Fuera del scope
UX	Solo se modificará el backend por lo que no se revisará el UX

2.3.CASOS DE PRUEBAS INCLUIDOS

ID	TÍTULO	ESTADO
TC-1	Verificar login con credenciales válidas	PASS
TC-2	Verificar login con credenciales inválidas	FAIL

2.4.CASOS DE PRUEBAS EXCLUIDOS

Los siguientes casos de prueba quedarán excluidos debido a que las historias que se realizarán en el sprint, no contemplan los siguientes escenarios.

También hay casos de prueba que se crearon en la planificación, pero por cambio de requerimientos, quedaron deprecados.

ID	TÍTULO	ESTADO
TC-10	Registro de cuentas	N/A
TC-11	Registro de cuentas duplicadas	N/A
TC-12	Recuperar contraseñas.	Deprecado
TOTAL TEST		

2.5. ENTORNO DE PRUEBAS

Las pruebas serán realizadas en un ambiente de QA, y posteriormente en un ambiente preproductivo.

QA: <https://ambienteQA.com>

Preprod: <https://ambientepreprod.com>

3. ESTRATEGIA DE LAS PRUEBAS

3.1. CONFIGURACION DE LAS PRUEBAS

1	Data Pre-requisito	- Ambiente de pruebas - Credenciales de ingreso - Documentación del backend (Swagger o similar)
2	Modalidad de ejecución de Pruebas	<i>Se realizarán casos de prueba en QMETRY (Complemento de Jira), los cuales contemplarán las funcionalidades actuales del sistema. Una vez realizado, se ejecutarán en el ambiente de testing.</i>
3	Tipos de Pruebas	<i>Se realizarán pruebas funcionales manuales, ejecutando casos de prueba y se probará el backend utilizando Postman.</i>
4	Integración con otros sistemas	<i>No</i>
5	Pruebas de Regresión	<i>La prueba de regresión se correrá al finalizar el testeo y luego de la salida a producción para garantizar de que el trabajo quedó correctamente realizado</i>
6	Herramienta de Registro y Control de Pruebas	<i>Se utilizará QMETRY, plugin incorporado en el Jira del proyecto.</i>
7	Herramienta de Gestión de Incidencias	<i>Se utilizará Jira como issue tracker.</i>

3.2. ORDEN DE EJECUCIÓN DE LAS PRUEBAS

A continuación se despliega un calendario simple con la programación propuesta para la actividad de ejecución del Testing, de acuerdo a la estimación preliminar realizada:

3.2.1 Cronograma de Actividades

Mes				Agosto												Fechas Estimadas
	Julio							Septiembre				Octubre				
Fases	S 2	S 3	S 4	S 1	S 2	S 3	S 4	S 1	S 2	S 3	S 4	S 1	S 2	S 3	S 4	
Análisis de testing																
Creación de casos de prueba																
Ejecución de las pruebas																
Paso a Producción																
Regresión																

3.3.CRITERIOS DE INICIO Y TÉRMINOS DE LAS PRUEBAS

Criterios de Inicio

Para comenzar la ejecución del Testing, se establece como hitos obligatorios la ejecución sin observaciones o errores de los siguientes pasos:

- Ambiente de prueba
- Plan de prueba actualizado y armado
- Documentación de las APIs
- Los TestCase deben estar diseñados y contruidos en un 100% de acuerdo a lo planificado.
- Datos de acceso al sistema alojado en el ambiente de prueba

Criterios de Término

- La cobertura de ejecución de las pruebas debe ser de un 100% (pruebas pasada exitosamente) de acuerdo a lo definido y planificado.
- No deben existir defectos con severidad crítica (nivel 1), al término de la etapa.
- Si por algún motivo externo o que escapa a lo definido en este plan, se deberá dejar evidencia por escrito del motivo por el cual se finaliza la actividad, con los correspondientes responsables de todas las partes.

3.4.EQUIPOS DE PRUEBAS Y RESPONSABILIDADES

NOMBRE	ROL	RESPONSABILIDAD
Danilo Vezzoni	QA	Creación y ejecución de casos de prueba
Dinno Vezzoni	QA Lead	Supervisar planes y casos de prueba

3.5.GESTIÓN DE INCIDENCIAS

El proceso de gestión de incidentes utilizado corresponde al proyecto, el cual es Jira.

El proceso o Flujo de Incidencia se resumen de la siguiente forma:

1. Equipo de QA genera/reporta la incidencia (con evidencia adjunta), frente a un error en el caso de Prueba ejecutado, éste queda en *Jira* ejecutado y Fallado.
2. La incidencia es derivada desarrollador correspondiente, quién valida si efectivamente es o no incidencia:
 - a. No es Incidencia: Se rechaza el ticket.
 - b. Es incidencia, desarrollo corrige y posteriormente informar por *Jira* al equipo de QA, para su re-testeo, previa habilitación (de la solución) en el ambiente de pruebas.
3. Equipo de QA re-testea o valida nuevamente:
 - a. Sí validación resulta exitosa, Caso de Prueba queda Pasado OK en *X-RAY*, con evidencia respectiva, y la incidencia reportada queda en estado Cerrada, todo lo anterior en *X-RAY*
 - b. Si el error persiste, el equipo de QA Rechaza a Desarrollo la incidencia. El Error reportado queda en estado “Rechazado a Desarrollo”, a la espera que el desarrollador corrija.
4. Ante el punto anterior, el flujo se repite (desde el punto 2) y finaliza cuando la incidencia es corregida Finalmente.

3.6.PLANTILLA REPORTE DE BUGS

ID: El ID de cada BUG debe ser único para poder identificar esa incidencia (Lo pone automáticamente el issue tracker)

Título: [Pantalla] Titulo descriptivo del error

Descripción: Comentar brevemente de que se trata el error y que se logre entender cuál es la falla

Precondiciones: Que es lo que necesito tener previamente configurado para poder ejecutar los pasos.

Test Data

Usuario: admin

Password: 123456

Pasos de reproducción:

1- Paso 1

2- Paso 2

3- Paso 3

4- Paso 4

Resultado Actual: Que es lo que está pasando ahora

Resultado Esperado: Como debería funcionar la aplicación

Screenshot/Video: Captura de pantalla o video del error

Otros campos a completar:

- Asignar siempre a un desarrollador o encargado
- Colocar siempre una criticidad/severidad
- Especificar en que sprint se va a arreglar, de lo contrario, colocarlo en el backlog
- Asociarlo a un test case

(Estos campos no van escritos dentro de la descripción, sino que son campos que trae el issue tracker)

Campos Opcionales:

Versión: Versión de la aplicación (Mobile)

Browser: Especificar en que navegador se encontró la falla

Ambiente: Especificar en que ambiente se encontró el error

Ciclo de vida de un bug:



4. ANEXOS

4.1. ENTREGABLES

ENTREGABLE	DETALLE/LINK
Casos de prueba	LINK
Resultado de la ejecución de pruebas	LINK