

Prueba Técnica - QA Automation

Te damos la bienvenida al desafío técnico que hemos diseñado para que puedas demostrar todas tus habilidades. Agradecemos el interés en participar en este reto importante. Por favor no usar logos ni imágenes del Banco Caja Social.

Aventura: La Cazadora de Bugs Financieros

Contexto

Eres el o la Cazadora de Bugs Financieros, y tu misión es garantizar que la experiencia de tu producto ficticio Ahorro Digital sea impecable y sin errores antes de salir a producción.

Ahorro Digital consiste en una Aplicación Web en donde los usuarios pueden explorar los productos de ahorro y simular cuánto podrían ganar con sus depósitos.

Tu instinto y tus pruebas deberán detectar errores antes de que los usuarios lo hagan.

Reto Principal

1. Plan de pruebas: objetivos, alcance y riesgos P0/P1/P2.

- Define objetivos, alcance, y criterios de aceptación.
- Identifica riesgos (P0: Onboarding; P1: Simulador; P2: Productos).

Plan de Pruebas – Producto “Ahorro Digital”

Objetivo General

Garantizar que la aplicación web Ahorro Digital funcione correctamente, entregue resultados financieros precisos y ofrezca una experiencia segura, clara y confiable antes de salir a producción.

Objetivos Específicos

- Validar que el proceso de Onboarding permita el registro exitoso y seguro de nuevos usuarios.
- Verificar que el Simulador de ahorro realice cálculos correctos y consistentes.
- Confirmar que la sección de Productos de ahorro muestre información precisa, actualizada y clara.
- Detectar defectos funcionales, de usabilidad y de lógica financiera.
- Reducir riesgos regulatorios y reputacionales asociados a errores en cálculos financieros.

Alcance

- Pruebas funcionales (caja negra).
- Validación de reglas de negocio.
- Pruebas de cálculo financiero del simulador.
- Pruebas de validación de formularios.

- Manejo de errores y mensajes al usuario.
- Pruebas básicas de compatibilidad en navegadores principales.
- Validación de seguridad básica (inputs, campos sensibles).

Fuera de Alcance (para este ciclo)

- Pruebas de performance avanzada (stress/load).
- Auditoría profunda de seguridad (pentesting).
- Integraciones con sistemas bancarios externos (si existieran).

Criterios de Aceptación

Generales

- No deben existir defectos críticos (P0) abiertos.
- Los cálculos del simulador deben coincidir 100% con la fórmula financiera definida.
- Validaciones de campos obligatorios deben funcionar correctamente.
- Mensajes de error deben ser claros y comprensibles.
- No deben existir errores de bloqueo (crashes).

Específicos

Onboarding

- Registro exitoso con datos válidos.
- Validación de correo y contraseña según reglas de negocio.
- Manejo correcto de errores (correo ya registrado, datos inválidos).

Simulador

- Cálculo correcto según:
 - Monto inicial
 - Tasa de interés
 - Plazo
 - Tipo de interés (simple/compuesto si aplica)
- Redondeo correcto a 2 decimales.
- Mensajes cuando el monto ingresado sea inválido.

Productos

- Información visible y consistente.
- Tasas y condiciones alineadas con las reglas de negocio.
- Navegación sin errores.

Identificación de Riesgos (Priorización P0 / P1 / P2)

P0 – Onboarding (Riesgo Crítico – Impacto Alto)

Impacto directo en adquisición de usuarios y seguridad.

Riesgos:

- Registro no funcional.
- Validación incorrecta de identidad.
- Fallos en creación de cuenta.

- Vulnerabilidad en campos (inyección de código).
- Pérdida de datos al enviar formulario.
- Mensajes de error poco claros que generen abandono.

Impacto:

Pérdida de clientes, riesgo de seguridad, impacto reputacional inmediato.

P1 – Simulador (Riesgo Alto – Impacto Financiero/Reputacional)

Impacta la confianza del usuario.

Riesgos:

- Cálculos incorrectos de interés.
- Error en fórmula (interés simple vs compuesto).
- Redondeo incorrecto.
- No considerar períodos correctamente.
- Mostrar ganancias irreales.
- No actualizar resultados en tiempo real.
- Permitir valores negativos o extremos sin validación.

Impacto:

Riesgo reputacional, desinformación financiera, posibles implicaciones regulatorias.

P2 – Productos (Riesgo Medio – Impacto Informativo)

Impacta experiencia y claridad.

Riesgos:

- Información desactualizada.
- Tasas inconsistentes con el simulador.
- Problemas de navegación.
- Mala visualización en dispositivos.
- Errores de contenido.

Impacto:

Confusión del usuario, menor conversión.

Estrategia de Pruebas

- Diseño de casos de prueba basados en reglas de negocio.
- Matriz de trazabilidad (Requisito → Caso de prueba → Resultado).
- Pruebas positivas y negativas.
- Pruebas exploratorias enfocadas en experiencia del usuario.
- Validación cruzada de cálculos con herramienta externa (Excel).

Conclusión

Como Cazadora de Bugs Financieros, priorizo primero la estabilidad del negocio (Onboarding), luego la precisión financiera (Simulador) y finalmente la experiencia informativa (Productos).

El éxito de la salida a producción dependerá de:

- Cero defectos P0.
- Validación matemática certificada del simulador.
- Experiencia de usuario fluida y sin fricciones.

2. Diseña tu estrategia: Casos de prueba: al menos 10 (funcionales, negativos y de validación).

Crea al menos 10 casos de prueba:

- Casos funcionales (campos obligatorios, recaptcha inválido, login fallido).
- Casos negativos (404, 401, monto 0).
- Validaciones de interfaz (botón deshabilitado, mensaje de error visible).

Plan de Pruebas – Ahorro Digital

1. Alcance

El presente documento describe los casos de prueba funcionales, negativos y de validación para la aplicación web Ahorro Digital. El alcance incluye pruebas del módulo de Onboarding (registro y login), Simulador financiero y visualización de Productos de ahorro. No incluye pruebas de performance, pruebas de carga ni pruebas de penetración.

2. Criterios de Aceptación Generales

- No deben existir defectos críticos (P0) abiertos antes de salida a producción.
- Los cálculos financieros deben coincidir con la fórmula definida por negocio.
- Los mensajes de error deben ser claros y no exponer información técnica.
- La interfaz debe responder correctamente a validaciones y estados de error.
- No deben presentarse errores 401/404 sin manejo controlado en UI.

Caso 1 – Registro exitoso

Descripción:

Validar que el usuario pueda registrarse correctamente ingresando datos válidos y superando el recaptcha.

Criterios de Aceptación:

- La cuenta se crea exitosamente.
- Se muestra mensaje de confirmación.
- El usuario es redirigido al dashboard.

Scenario: Registro exitoso con datos válidos

Given que el usuario se encuentra en la página de registro
When ingresa un correo válido "usuario@test.com"

And ingresa una contraseña válida "Pass1234!"
And confirma la contraseña correctamente
And completa el recaptcha correctamente
And presiona el botón "Crear cuenta"
Then el sistema debe crear la cuenta exitosamente
And debe mostrar el mensaje "Registro exitoso"
And debe redirigir al dashboard del usuario

Caso 2 – Campo obligatorio vacío

Descripción:

Validar que el sistema impida el registro cuando el campo correo esté vacío.

Criterios de Aceptación:

- El botón permanece deshabilitado o muestra mensaje de error.
- No se envía solicitud al backend.

Scenario: Campo obligatorio vacío

Given que el usuario se encuentra en la página de registro
When deja el campo correo vacío
And completa los demás campos correctamente
And presiona "Crear cuenta"
Then el sistema debe mostrar el mensaje "El correo es obligatorio"
And no debe enviarse la solicitud al backend

Caso 3 – Recaptcha inválido

Descripción:

Verificar que el sistema bloquee el registro si el recaptcha no es validado.

Criterios de Aceptación:

- No se crea la cuenta.
- Se muestra mensaje de validación de seguridad.

Given que el usuario completa todos los campos correctamente
And el recaptcha no es validado correctamente
When presiona "Crear cuenta"
Then el sistema debe bloquear el registro
And mostrar el mensaje "Validación de seguridad requerida"
And no debe crearse ninguna cuenta

Caso 4 – Login fallido

Descripción:

Validar que el sistema rechace el acceso con contraseña incorrecta.

Criterios de Aceptación:

- Se muestra mensaje de credenciales inválidas.
- No se genera sesión activa.

Given que el usuario ya está registrado
When ingresa su correo válido
And ingresa una contraseña incorrecta
And presiona "Iniciar sesión"

Then el sistema debe rechazar el acceso
And mostrar el mensaje "Credenciales inválidas"
And no debe generar token de sesión

Caso 5 – Error 401 en autenticación

Descripción:

Verificar manejo controlado cuando el backend responde con 401.

Criterios de Aceptación:

- Se muestra mensaje amigable.
- No se exponen detalles técnicos.

Given que el usuario ingresa credenciales inválidas
When el backend responde con código 401
Then el sistema debe mostrar mensaje controlado "Usuario o contraseña incorrectos"
And no debe exponer detalles técnicos del error

Caso 6 – Simulación correcta

Descripción:

Validar que el simulador calcule correctamente el interés compuesto con datos válidos.

Criterios de Aceptación:

- El resultado coincide con la fórmula financiera.
- Se muestran valores con 2 decimales.

Given que el usuario está en el simulador
When ingresa monto inicial "1000"
And ingresa tasa de interés anual "10"
And selecciona plazo "12 meses"
And presiona "Simular"
Then el sistema debe calcular correctamente el interés compuesto
And mostrar el monto final correcto
And mostrar los valores con 2 decimales

Caso 7 – Monto igual a 0

Descripción:

Verificar que el sistema bloquee simulaciones con monto 0.

Criterios de Aceptación:

- Se muestra mensaje 'El monto debe ser mayor a 0'.
- No se muestran resultados.

Given que el usuario está en el simulador
When ingresa monto inicial "0"
And completa los demás campos correctamente
And presiona "Simular"
Then el sistema debe mostrar el mensaje "El monto debe ser mayor a 0"
And no debe mostrar resultados financieros

Caso 8 – Monto negativo

Descripción:

Validar que el sistema rechace montos negativos.

Criterios de Aceptación:

- El campo se resalta como inválido.
- No se ejecuta cálculo.

Given que el usuario está en el simulador

When ingresa monto inicial "-500"

And presiona "Simular"

Then el sistema debe mostrar el mensaje "Monto inválido"

And el campo debe resaltarse como inválido

And no debe ejecutarse el cálculo

Caso 9 – Botón deshabilitado

Descripción:

Verificar que el botón 'Simular' esté deshabilitado con campos incompletos.

Criterios de Aceptación:

- El botón solo se habilita cuando todos los campos son válidos.

Scenario: Botón Simular deshabilitado

Given que el usuario está en el simulador

When no ha ingresado todos los campos obligatorios

Then el botón "Simular" debe permanecer deshabilitado

And debe habilitarse únicamente cuando todos los campos sean válidos

Caso 10 – Error 404 en simulador

Descripción:

Validar manejo de error cuando el servicio de simulación no está disponible.

Criterios de Aceptación:

- Se muestra mensaje controlado.
- Se permite reintentar operación.

Scenario: Servicio de simulación no disponible

Given que el usuario completa los datos correctamente

When el servicio backend responde con código 404

Then el sistema debe mostrar el mensaje "Servicio temporalmente no disponible"

And debe permitir reintentar la operación

Caso 11 – Visualización de productos

Descripción:

Verificar que los productos se muestren correctamente con sus atributos.

Criterios de Aceptación:

- Cada producto muestra tasa, plazo y monto mínimo.

Scenario: Listado exitoso de productos

Given que el usuario accede a la sección "Productos"

When el backend responde correctamente

Then el sistema debe mostrar la lista de productos disponibles

And cada producto debe mostrar tasa, plazo mínimo y monto mínimo

Caso 12 – Producto inexistente

Descripción:

Validar comportamiento cuando se accede a un producto con ID inválido.

Criterios de Aceptación:

- Se muestra página 'Producto no encontrado'.

Given que el usuario intenta acceder a un producto con ID inválido

When el backend responde con 404

Then el sistema debe mostrar página "Producto no encontrado"

And ofrecer opción para regresar al listado

Caso 13 – Acceso no autorizado

Descripción:

Verificar redirección al login cuando un usuario sin sesión intenta acceder a productos exclusivos.

Criterios de Aceptación:

- Se redirige al login.
- Se muestra mensaje de sesión requerida.

Given que el usuario no ha iniciado sesión

When intenta acceder a productos exclusivos

And el backend responde 401

Then el sistema debe redirigir a la pantalla de login

And mostrar el mensaje "Sesión requerida"

Caso 14 – Consistencia tasa producto/simulador

Descripción:

Validar que la tasa mostrada en producto coincida con la usada en el simulador.

Criterios de Aceptación:

- Los resultados coinciden matemáticamente.
- No existen discrepancias entre módulos.

Scenario: Validación de tasa consistente

Given que el producto muestra tasa anual "8%"

When el usuario usa esa tasa en el simulador

Then el resultado debe coincidir con la fórmula financiera definida

And no debe existir discrepancia entre módulos

Caso 15 – Sin productos disponibles

Descripción:

Verificar mensaje adecuado cuando no existan productos activos.

Criterios de Aceptación:

- Se muestra mensaje informativo.
- No se presentan errores técnicos.

Scenario: Sin productos disponibles

Given que el backend devuelve lista vacía

When el usuario accede a la sección productos

Then el sistema debe mostrar mensaje "No hay productos disponibles actualmente"

And no debe mostrar errores técnicos

Ejemplos de diseño:

Adjunto ejemplos en formato Excel, estos formatos varían dependiendo las plantillas de cada organización

"casos de prueba bcs.xlsx" incluida en la carpeta de "**adjuntos y documentos de evidencia de la prueba**" en el repositorio de git

3. Automatización de la detección: con Playwright (UI) o Postman (API). Importante tener en cuenta el Plus “Agrega desarrollos en backend y/o frontend necesarios en la prueba.” Valida el uso de Inteligencia Artificial para crearlos.

Elige tu herramienta:

- Playwright (UI) → flujo de registro completo.
- Postman (API) → prueba endpoints del backend.

Ejecuta 3–5 pruebas automáticas con reporte HTML o JUnit.

Se realizaron 4 casos automatizados que validan la siguiente información:

1. INGRESAR AL PORTAL CLIENTE NO EXISTENTE

Este caso de prueba valida el mensaje de error generado cuando se ingresa una contraseña errada

2. CREAR CUENTA DATOS DE VALIDACION ERRONEOS

Este caso de prueba valida los datos erróneos de validación para el ingreso de generar una cuenta nueva.

3. VALIDACION CODIGO DE RESPUESTA

Este caso de prueba valida el código de respuesta 200 de ingreso a la aplicación de bcs, en ambiente GUI.

4. VALIDACION CODIGO API SIN UI

Este caso de prueba valida el código de respuesta 200 de ingreso a la aplicación de bcs, sin ambiente GUI, consumo del Request URL

Se genera capturas de imágenes paso a paso de cada caso de prueba, las evidencias son almacenadas en el archivo report HTML, y video de ejecución.

Los datos requeridos para los casos de pruebas se almacenan en el archivo .env.qa

Metodología aplica POM

4. Reporte: evidencia de errores, severidad y pasos.

- Documenta bugs encontrados con evidencia (captura o log).
- Prioriza por severidad (Alta/Media/Baja).

Plus

- Agrega desarrollos en backend y/o frontend necesarios en la prueba.
- Ejecución headless con video/captura.
- Casos agrupados por prioridad (P0, P1, P2).
- Recomendaciones para mejorar el flujo.

BUG 1 – Botón “Siguiente” no se habilita correctamente

Módulo: Onboarding (Login)

Prioridad: P0 (Impacta acceso al sistema)

Severidad: Alta

Descripción: El botón “Siguiente” permanece deshabilitado incluso después de ingresar un usuario válido.

Pasos para reproducir:

1. Ingresar a Ahorro Digital.
2. Seleccionar “Ingresar al Portal Personas”.
3. Digitar un usuario válido.
4. Verificar estado del botón “Siguiente”.

Resultado actual: El botón no se habilita de forma consistente.

Resultado esperado: El botón debe habilitarse automáticamente cuando el campo cumple validación.

Evidencia: Captura donde el campo usuario está diligenciado pero el botón permanece disabled.

Impacto negocio: El usuario no puede continuar el proceso → abandono del flujo.

Recomendación: Revisar evento onChange / onBlur que activa validación frontend.

BUG 2 – Lista desplegable Tipo de Documento no carga siempre

Módulo: Apertura de Cuenta

Prioridad: P0

Severidad: Alta

Descripción: Al hacer clic en “Tipo de Documento”, la lista desplegable no aparece de forma intermitente.

Pasos:

1. Ir a “Abrir Cuenta”.
2. Hacer clic en “Tipo de Documento”.
3. Observar comportamiento.

Resultado actual: La lista no se renderiza en algunos intentos.

Resultado esperado: El listado debe mostrarse siempre tras la interacción.

Evidencia: Video donde el combobox se abre sin mostrar opciones.

Impacto: Bloquea creación de cuenta.

Recomendación: Revisar render condicional y dependencia de estados async.

BUG 3 – Validación inconsistente en mensaje de contraseña inválida

Módulo: Login

Prioridad: P1

Severidad: Media

Descripción: El mensaje “Datos incorrectos. Verifique la longitud de su contraseña.” no aparece en todos los escenarios de error.

Pasos:

1. Ingresar usuario válido.
2. Ingresar contraseña inválida.
3. Confirmar mensaje de error.

Resultado actual: En algunos intentos el mensaje no se muestra.

Resultado esperado: Siempre debe mostrarse mensaje claro y visible.

Impacto: Confusión del usuario.

Recomendación: Homologar validaciones backend y frontend.

BUG 4 – Duplicidad en selector botón “Abrir mi Cuenta”

Módulo: Apertura Cuenta

Prioridad: P1

Severidad: Media

Descripción: Existen dos botones con textos similares:

- “Abrir mi Cuenta”
- Botón dentro de acordeón FAQ

Resultado actual: Ambigüedad en identificación.

Resultado esperado: Un único botón principal con identificador único.

Impacto: Riesgo de interacción incorrecta.

Recomendación: Agregar data-testid único y evitar reutilización de textos en componentes secundarios.

BUG 5 – Código de respuesta inconsistente

Módulo: API

Prioridad: P2

Severidad: Baja (si UI funciona)

Descripción: La API devuelve HTTP 200 pero el código funcional esperado no coincide con especificación.

Impacto: Riesgo en monitoreo o integraciones futuras.

Recomendación: Documentar contrato API (Swagger / OpenAPI).

Resumen Ejecutivo

Bug	Prioridad	Severidad	Impacto
Onboarding botón	P0	Alta	Bloquea acceso
Lista documento	P0	Alta	Bloquea registro
Mensaje error	P1	Media	Confusión usuario
Botón duplicado	P1	Media	Riesgo UX
Código API	P2	Baja	Riesgo técnico

PLUS – Mejoras Recomendadas

Backend:

- Estandarizar códigos de respuesta.
- Documentar API.
- Manejo uniforme de errores.

Frontend:

- Validaciones en tiempo real consistentes.
- Evitar render condicional dependiente de eventos async sin fallback.
- Añadir identificadores únicos.

Ejecución Headless con Evidencia (Enfoque Funcional):

- Captura obligatoria en cada P0.
- Video obligatorio en flujos críticos (Onboarding y Apertura).
- Evidencia consolidada en repositorio QA.

Casos agrupados por prioridad:

P0 – Críticos:

- Login
- Apertura de cuenta
- Tipo documento

P1 – Alto impacto UX:

- Mensajes de error
- Botones duplicados

P2 – Técnicos:

- Código API
- Logs

Recomendaciones para mejorar el flujo:

- Reducir dependencias asincrónicas en render de componentes.
- Validaciones desacopladas de eventos blur.
- Mejorar accesibilidad (roles bien definidos).
- Definir contrato API formal.
- Implementar monitoreo de errores en producción.

Enviar solución en repositorios públicos de GitHub.