
	Reto técnico interno		
	N° Versión: 1.1	Fecha última revisión: 17-05-2024	Página 1 de 6

INTRODUCCIÓN

Este documento tiene como objetivo verificar los conocimientos técnicos de los aspirantes a ingresar al equipo para los roles de QA Senior y Semi - Senior desde el Centro de Excelencia en Calidad de Software de Sofka Technologies. La información declarada en el mismo es de carácter confidencial y no debe ser difundida por terceros no autorizados.


Instrucciones generales

- La descripción del reto técnico consta de dos partes: Metodología de Pruebas Ágiles y Automatización.
- La sección del reto referente a Automatización contiene varios ejercicios, el candidato debe escoger **solo uno** de los enunciados para resolverlo
- La solución al reto deberá ser desarrollada en alguno de los siguientes frameworks o herramientas de automatización de pruebas: Karate, Serenity BDD con Rest Assured o Serenity Rest, Cypress. El framework es a elección del aspirante.
- Bajo cualquiera de los frameworks seleccionados, la solución deberá contar mínimamente con los siguientes puntos:
 - Patrón de diseño o arquitectura claro
 - Reportes de pruebas exportables en formato html, json o csv
 - Diseño de casos de prueba claro, deseable implementación en lenguaje Gherkin
 - Buenas prácticas de programación

	Reto técnico interno		
	N° Versión: 1.1	Fecha última revisión: 17-05-2024	Página 2 de 6

CONTENIDO

1 Descripción del reto:	3
2 Entregables y formatos de entrega:	3

	Reto técnico interno		
	N° Versión: 1.1	Fecha última revisión: 17-05-2024	Página 3 de 6

METODOLOGÍA DE PRUEBAS ÁGILES

1. DESCRIPCIÓN DEL RETO:

El siguiente ejercicio contiene un escenario que describe la situación actual de un cliente en el sector tecnológico. El reto se enfoca en responder las siguientes preguntas en base al contexto dado:

- Si fuéramos nosotros los encargados de identificar y proponer una metodología de trabajo de pruebas, ¿que propuesta plantearías al Banco para mejorar este proceso?
- ¿Identifica qué tipos de pruebas podríamos implementar?
- ¿Qué información adicional consideras que requieres para dar respuesta de los puntos anteriores?
- Generar dos casos de prueba que puedes proponer al Banco para las validaciones. Ejemplo: para funcionalidades de pago, transferencias, consulta de saldo.
- Qué aspectos de la propuesta presentada de pruebas dejarías por fuera y por qué?
- Identifica los riesgos que puedes presentar al Banco, si deciden continuar con su metodología de pruebas actual.

El formato de presentación de esta sección de reto puede ser un archivo word, txt o similares con las respuestas a las preguntas planteadas

2. CONTEXTO DEL EJERCICIO:


Actualmente una empresa del sector Banca solicita nuestros servicios de testing para ejecutar las pruebas en proyectos que tienen en curso de funcionalidades nuevas y actualizaciones de las ya existentes, el Banco tiene un proveedor de desarrollo que se encarga de realizar la entrega en un tiempo estimado de **3 meses de cada funcionalidad completa** y la entrega se realiza en un sesión de 1 hora donde el desarrollador entrega un acta donde documenta los módulos desarrollados y todos los artefactos que le entregará al Banco.

El software del Banco está alojado en la nube (servidores conectados a internet) donde se almacena el código de la funcionalidad entregada por el proveedor, el Banco realiza las pruebas de lo entregado apoyándose de varios de sus colaboradores con más conocimiento en los procesos del Banco, estas personas realizan validaciones para revisar que todo siga funcionando como se espera en el día a día. Los colaboradores **no generan ninguna documentación, ni evidencia de las pruebas o errores que identifican en el proceso de validación.**

Las pruebas que realizan los colaboradores son en **ambiente productivo y en horarios donde no hubiera servicio** del Banco al público.

Cuando se identifica un error en las validaciones, el colaborador envía un correo a la persona encargada de TI del Banco y ésta es la responsable de reportar al proveedor de desarrollo lo identificado, el proveedor recibe el correo con el reporte y procede a dar una **solución en un tiempo estimado de 2 a 3 semanas** dependiendo de la complejidad de la solución.

El Banco ha observado **mucho reproceso** en la entrega del desarrollo y en pruebas, por esta razón han visto la necesidad de contratar nuestra empresa y solicitar una consultoría para saber cómo se puede mejorar el proceso y generar valor constante en el Banco.

	Reto técnico interno		
	N° Versión: 1.1	Fecha última revisión: 17-05-2024	Página 4 de 6

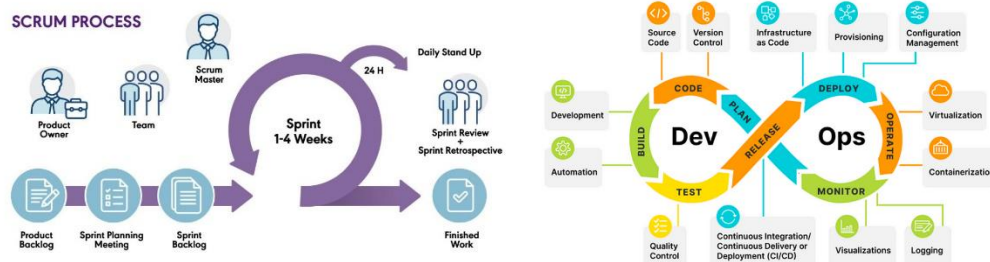
Propuesta de la solución

1. Metodología de Trabajo de Pruebas

Necesitamos un marco de trabajo como Scrum y Devops adaptable. Es decir, necesitamos Integración Continua (CI) y Entrega Continua (CD):

CI: Implementar herramientas de CI como Azure Devops, Jenkins o GitLab CI para ejecutar pruebas automáticas en cada cambio de código.

CD: Automatizar despliegues con pipelines CI/CD para reducir tiempos y errores.



2. Tipo de pruebas a implementar

Pruebas Automatizadas:

Unitarias: Pruebas unitarias para asegurar que cada componente funcione correctamente de manera aislada.

Integración: Pruebas de integración para validar la interacción entre diferentes componentes del sistema.

Regresión: Pruebas de regresión para asegurar que nuevas funcionalidades no rompan las existentes.

Aceptación (UAT): Pruebas de aceptación para validar que el sistema cumpla con los requisitos del usuario.

Seguridad: Para descubrir vulnerabilidades, amenazas, riesgos en una aplicación de software y prevenir ataques maliciosos.


Rendimiento: Para someter el sistema a una carga de trabajo con el fin de medir su velocidad, fiabilidad y estabilidad en esas condiciones de trabajo.

Gestión de Pruebas y Documentación:

Utilizar herramientas como Azure, JIRA, TestRail o Zephyr para gestionar casos de prueba, capturar evidencias y generar reportes.

Ambiente de Pruebas:

Configurar un entorno de preproducción idéntico al de producción para realizar pruebas exhaustivas antes de desplegar en producción.

	Reto técnico interno		
	N° Versión: 1.1	Fecha última revisión: 17-05-2024	Página 5 de 6

3. Información importante para dar respuesta a los puntos anteriores serian

- Detalles completos de las funcionalidades y sus requerimientos técnicos.
- Calendario de entregas y plazos para pruebas y despliegues.
- Infraestructura actual y herramientas disponibles
- Claridad de roles o del rol de prueba

4. Casos de prueba:


- Validar intentar una transferencia a otro banco con fondos insuficientes y su mensaje de alerta
- Validar que la consulta exitosa de saldo de un cliente que previamente inicia sesión en el aplicativo.

5. Dejaríamos por fuera lo siguiente:

- Las pruebas manuales en producción pueden ser un poco riesgosas y mas si hay en medio datos sensibles, además la no documentación de casos de prueba es desacertado. De manera que seria en un ambiente pre-productivo QA o DEV.
- Los tiempos de entrega para agregar valor al producto, ya que son demasiado extensos, por ende, se consideraría 15 días enfoque ágil PMV.

6. Principales riesgos de seguir trabajando de la misma manera

- La falta de trazabilidad de pruebas dificultaría lograr replicar escenarios que generar bugs o errores.
- Afectar a algún cliente de la vida real en caso de tomarlo para pruebas, en datos sensible o dinero.
- Afectación en la confianza del cliente, satisfacción baja y por ende el cambio de banco por un mal servicio.

	Reto técnico interno		
	N° Versión: 1.1	Fecha última revisión: 17-05-2024	Página 6 de 6

AUTOMATIZACIÓN

1 DESCRIPCIÓN DEL RETO:

Ejercicio 1

Considere como SUT la API <https://regres.in/api-docs/>, el cual es un servicio de gestión de usuarios. Implementar los casos de pruebas asociados a las siguientes funcionalidades:

- Consultar un usuario creado por ID
- Actualizar los datos un usuario y verificar la correcta actualización de los datos por medio de consulta
- Eliminar un usuario del sistema
- Consultar la lista completa de usuarios y verificar que el usuario eliminado no se encuentra en el sistema

Ejercicio 2


Considere como SUT la API <https://petstore.swagger.io/> el cual es un servicio destinado a la gestión de una tienda de venta de mascotas. Implementar los casos de pruebas asociados a las siguientes funcionalidades:

- Consultar una orden de compra
- Buscar la orden de compra creada
- Verificar el inventario de ventas
- Eliminar una orden de compra

Ejercicio 3

Considere como SUT la API <https://openweathermap.org/current> la cual emite información del clima en tiempo real. Implementar los casos de pruebas asociados a las siguientes funcionalidades:

- Obtener la información del clima consultando por nombre de ciudad
- Obtener la información del clima consultando por latitud y longitud
- Obtener la información del clima en formato Json

	Reto técnico interno		
	N° Versión: 1.1	Fecha última revisión: 17-05-2024	Página 7 de 6

- Obtener la información del clima en formato XML

2 ENTREGABLES Y FORMATOS DE ENTREGA:

- Código de pruebas en formato .zip o url de repositorio público de github
- Archivo README.md con instrucciones para ejecutar el proyecto (incluir versiones de dependencias a instalar, paso a paso, comando de ejecución)