**PRUEBA TÉCNICA - FUNCIONAL PARA EL ROL QA ENGINEER**

***Pruebas Manuales/Funcionales:***

1. ¿Qué entiendes por el término pruebas funcionales?
2. ¿Cuáles son los pasos importantes que se deben cubrir en las pruebas funcionales?
3. ¿Cuál es la diferencia entre pruebas funcionales y no funcionales?
4. Define en tus propias palabras qué es un release.
5. Escribe una lista de los posibles estados que puede tener un defecto (escribe una pequeña definición de cada uno).
6. ¿Cuáles puntos consideras importantes a la hora de escribir un caso de prueba?
7. ¿Cuáles puntos consideras importantes a la hora de documentar un defecto?
8. ¿Qué diferencias hay entre probar la respuesta de un API y el esquema de respuesta de un API?

**Análisis y Documentación de Pruebas:**

Imagina que eres parte de un equipo de desarrollo de software que está trabajando en una aplicación de comercio electrónico. La aplicación permite a los usuarios buscar productos, agregarlos al carrito de compras y realizar pedidos. El equipo ha decidido adoptar BDD (Behavior-Driven Development) para mejorar la colaboración entre los desarrolladores, probadores y stakeholders del proyecto.

**Historia de Usuario:**

Como usuario registrado,

Quiero poder ver el historial de mis pedidos,

Para que pueda realizar un seguimiento de mis compras anteriores.

**Criterios de Aceptación:**

1. Dado que soy un usuario registrado, cuando inicio sesión en mi cuenta, debo poder encontrar la opción "Historial de Pedidos" en el menú de navegación.
2. Dado que accede al historial de pedidos, debo ver una lista de todos mis pedidos anteriores, incluyendo detalles como el número de pedido, la fecha y el estado del pedido.
3. Dado que veo la lista de pedidos anteriores, debo poder hacer clic en un pedido para ver más detalles, incluyendo la lista de productos en ese pedido.
4. Dado que estoy viendo la lista de productos en un pedido, cada producto debe mostrar su nombre, cantidad y precio.

**Cobertura de Pruebas Funcionales:**

Crear escenarios de prueba utilizando Gherkin para cubrir los criterios de aceptación mencionados anteriormente. Asegúrate de incluir casos de prueba positivos y negativos, así como casos de prueba límite si es aplicable.

**Instrucciones:**

1. Utiliza el formato Gherkin para escribir tus escenarios de prueba. Puedes organizarlos en un archivo de características.
2. Para cada escenario, identifica claramente los pasos de Given, When y Then.
3. Asegúrate de cubrir todos los criterios de aceptación y proporcionar variaciones que aborden diferentes condiciones y posibles problemas.
4. Incluye comentarios o anotaciones explicativas cuando sea necesario para mejorar la comprensión de tus pruebas.

**Entregables:**

1. Archivo de características que contenga los escenarios de prueba escritos en Gherkin.
2. Informe explicativo que detalle la cobertura de pruebas funcionales, destacando cualquier caso especial considerado.

# Automation Tests

The objective of this test is to evaluate the ability to solve problems, the expected result is to be able to solve all the proposed problems, if not, to make the theoretical approach of the possible solutions.

Please clone the follow public repository: <https://bitbucket.org/josesimetrik/qa-engineer-automation-test/src/develop/>

**PD:** Before starting the solution of the problems raised, study in detail the infrastructure of the framework and its code.

## “Now, to what we came for” 😅

1. Fix all bugs in the framework, these bugs prevent the correct execution of automated tests.
2. Fix bugs in automated tests (respecting the framework architecture)
3. The web application being tested in this case is www.kayak.com, please design and automate a test that meets the following scope:
   1. 1. Navigate to the Kayak home page.
   2. 2. Navigate through each of the right menu options and validate that each page opens correctly (options enclosed in the red box in image 1).

i. This test must be automated by implementing the POM pattern according to the framework infrastructure.

Points to take into consideration:

* Create a branch from the develop branch.
* Create a PR with the proposed solutions, this PR must go to the develop branch (upload changes to the created branch and create the PR 10 minutes before starting the scheduled interview).

○ If these considerations are not followed, the selection process will be canceled unanimously.

RESPUESTAS:

1. **¿Qué entiendes por el término pruebas funcionales?**

Son las pruebas que verifican que el software cumpla con los requisitos y funcionalidades especificadas y que todas las características del sistema operen correctamente según lo esperado.

1. **¿Cuáles son los pasos importantes que se deben cubrir en las pruebas funcionales?**
   * **Identificación de Requisitos:** Entender y revisar los requisitos funcionales del sistema.
   * **Diseño de Casos de Prueba:** Crear casos de prueba basados en los requisitos, cada caso de prueba debe definir los pasos a seguir, los datos de entrada y los resultados esperados.
   * **Configuración del Entorno:** Preparar el entorno de pruebas, ya sea configuración de hardware, software y las herramientas necesarias para realizar las pruebas.
   * **Ejecución de Pruebas:** Ejecutar los casos de prueba. Se observa y se registra cualquier comportamiento inusual o defectuoso.
   * **Registro de Resultados:** Documentar los resultados de las pruebas, incluyendo defectos, errores, resultado real.
   * **Evaluación de Resultados:** Comparar los resultados obtenidos con los requisitos esperados y validar si hay defectos que deben ser abordados.
2. **¿Cuál es la diferencia entre pruebas funcionales y no funcionales?**

La diferencia entre pruebas Funcionales y No funcionales es que la primera se encarga de verificar que el software cumpla con las funcionalidades especificas dentro del sistema, mientras la segunda es la encargada de validar el rendimiento, seguridad y usabilidad del sistema.

1. **Define en tus propias palabras qué es un release.**

Un release es una versión del software con nuevas características, mejoras o correcciones y se encuentra lista para ser entregada a los usuarios finales o para su implementación en un entorno de producción.

1. **Escribe una lista de los posibles estados que puede tener un defecto (escribe una pequeña definición de cada uno).**

* **Nuevo:** El defecto ha sido registrado y se envía a revisión.
  + **Asignado:** El defecto ha sido asignado a un desarrollador para su validación y solución.
  + **En Progreso:** El defecto está siendo trabajado por el desarrollador.
  + **Resuelto:** El defecto ha sido corregido y se implementa correctamente.
  + **Verificado:** La solución ha sido revisada y el defecto se confirma como resuelto.
  + **Cerrado:** El defecto se considera resuelto y no requiere más acciones.
  + **Reabierto:** El defecto ha vuelto a aparecer después de haber sido cerrado.

1. **¿Cuáles puntos consideras importantes a la hora de escribir un caso de prueba?**

Para escribir los casos de pruebas se debe de tener en cuenta lo siguiente:

* + Ser lo más claro posible y especifico en los requisitos.
  + Incluir pasos detallados para la ejecución de las pruebas.
  + Informar si hay condiciones antes de realizar la prueba
  + Especificar los datos necesarios para realizar la prueba.
  + Definir el resultado esperado.
  + Evaluar el éxito de la prueba.

1. **¿Cuáles puntos consideras importantes a la hora de documentar un defecto?**
   * Detallar el defecto de manera precisa.
   * Incluir los pasos necesarios para reproducir el defecto.
   * Comparar el comportamiento esperado con el real.
   * Evaluar el impacto del defecto en el sistema.
   * Mostrar evidencia visual o logs si es posible.
   * Validar el estado del defecto y su prioridad.
2. **¿Qué diferencias hay entre probar la respuesta de un API y el esquema de respuesta de un API?**

La respuesta de un APIverifica que el contenido y los datos devueltos por la API cumplan con las expectativas funcionales, es decir que los valores estén correctos y sean los esperados.

El Esquema de Respuesta de API valida que la estructura y el formato de la respuesta de la API coincidan con el esquema definido.