Estrategia de testing.

Requisitos del trabajo

Sprint 1:

* Crear casos de prueba de las historias
* testear la API (Postman)

Sprint 2:

* pruebas automatizadas : uso de Selenium de lo desarrollado en sprint 1y2
* Jest, había un opcional y otro obligatorio pero no hicimos ninguno
* Testeo automatizado Postman : Desarrollar los scripts de pruebas automatizados utilizando Postman para las API desarrolladas en este sprint. Agregar las API desarrolladas en este sprint a la colección generada en el sprint I. Ejecutar los scripts con el test runner de Postman y subir a GitLab el archivo JSON con los resultados de la ejecución (debe contener los scripts de sprint I y II).
* Testeo exploratorio Manual : **Testing exploratorio**
* Realizar test exploratorio sobre lo agregado en el sprint. Entregar un documento con notas y forma de la organización del test exploratorio (sesiones, tours, escenarios, workflows, etc.).

**Casos de prueba manuales**

* Agregar a la planilla de casos de prueba sobre las funcionalidades del sprint II.
* Realizar mantenimiento de los casos de prueba manuales generados en el sprint I, en caso de ser necesario.
* Clasificar los casos de prueba en suite de smoke y regression test.
* Generar la suite de prueba con los casos que se puedan ejecutar, luego ejecutar esta suite de prueba.
* Subir a GitLab la planilla de casos de prueba adjunta o el link al archivo.

**Testing estático**

Revisar cada una de las historias de usuario del sprint II y reportar defectos en la planilla correspondiente (por ejemplo, falta de definición de algún componente).

Subir a GitLab la planilla de defectos adjunta o el link al archivo.

Sprnt 3 :

* Implementar testeos manuales:

 Realizar pruebas exploratorias en las páginas de reserva y de confirmación de reserva. Verificar y validar las funcionalidades y la interfaz de usuario.

 Agregar test cases manuales sobre las funcionalidades a entregar en este sprint y ejecutar un ciclo de prueba seleccionando tests cases necesarios, teniendo en cuenta de no duplicar esfuerzo (evitando de ejecutar los que ya se encuentran automatizados).

* Implementar testeos Automatizados:
*  Añadir las pruebas necesarias para los endpoints de productos, reservas y usuarios a la colección creada previamente en Postman (scripts de pruebas automatizados).
*  Añadir / Crear un script relacionado al proceso de reserva en las pruebas de regresión automatizadas creadas con Selenium IDE (o Selenium Web Driver).
*  **Opcional**: Crear y ejecutar las pruebas unitarias (unit testing) de las páginas de reserva y confirmación de reserva a través de JEST framework.

Sprint 4 :

* Implementar test automatizados:

Crear y ejecutar test unitarios utilizando Selenium sobre las funcionalidades de la página de mis reservas (las realizadas por usuario) y sobre la página formulario de producto (la que se ingresa con rol administrador).

Agregar a la colección de Postman script automatizados para probar los métodos agregados en este sprint.

* Implementar testing manual:

Realizar y registrar el testing exploratorio sobre las funcionalidades de la página de mis reservas y sobre la página formulario de producto.

Agregar casos de prueba Manuales para ambas páginas y clasificar los mismos en Smoke y Regression Test.

* Confeccionar el reporte final de testing:

Ejecutar los casos de prueba que pertenecen a la suite de Regresión y generar un **Reporte Final de Prueba** que contenga el siguiente resumen (pueden ser gráficos o tablas agregadas en el template) :

* Cantidad de casos de prueba creados por funcionalidad/historia de usuario tanto manuales como automáticos.
* Cantidad de casos de prueba ejecutados, pasados y fallados.
* Cantidad de defectos por estado y prioridad/severidad
* Como **exit criteria** (criterio para finalizar el testing) no se debería tener defectos en estado abierto de severidad crítica y/o bloqueante.

Estrategia de testing:

En cada Sprint:

* A nivel de back se realizan testeos sobre los Controller y Service desarrollados.
* Se generan y corren los scripts en Postman para testear los endpoints creados.
* Se generan casos de prueba (positivos y negativos) que contemplen los requisitos de las historias de usuarios.
* Se formula un ciclo de testeos incluyendo las nuevas pruebas (Smoke), la repetición de las pruebas fallidas anteriormente (regresión) y/o la repetición de pruebas por modificaciones de funcionalidades (regresión). Se corre dicho ciclo, se reflejan los resultados y se retroalimenta al equipo de desarrollo.
* Se crean y corren en Selenium pruebas sobre los desarrollos del sprint.

Testeos realizados: Tabla

Back-end :

Resultados testeos y errores.