

# API - Automation Test

# Prueba técnica perfil funcional.

Visita la web de **–TU-EMPRESA–** y crea 1 Escenario de prueba de aceptación más importantes que harías en la web para asegurarte de que la subida a producción de la nueva versión ha sido correcta.

Utiliza el siguiente formato:

- Título del test (Resumen).
- Descripción.
- Etiqueta: de ser necesario agrupar los test cases.
- Tipo de prueba: Pruebas de Remojo, Pruebas de Stress, Prueba Carga, Prueba Smoke
- Tecnología: Android, iOS, Web...
- Pre-condiciones: si necesito cumplir alguna.
- Pasos a reproducir con resultado esperado (detállalos lo máximo posible).

Por ejemplo:

PASO DE LA PRUEBA	RESULTADO ESPERADO
Ir a la web de –TU EMPRESA–	La web se muestra correctamente en el navegador

# Prueba técnica.

Después de haber definido los casos de prueba de aceptación mínimos que necesitas, crea un proyecto Node con las siguientes características:

- Proyecto **Npm** con **Node.js**
- Añade **Mocha, Chai, Chai-HTTP, Axios** y las dependencias necesarias para ejecutar el proyecto.
- Utiliza uno de los siguientes patrones de diseño **Modelo Creacionales, Estructura, Comportamiento** para organizar la estructura de tu proyecto.
- Desarrolla los **5 test de aceptación** que has definido anteriormente de forma automática, no te olvides de:
  - Si haces una comprobación (**check**).
  - Utiliza el **JSdoc** en cada uno de los métodos.
- Crea una suite en para lanzar todas las pruebas.
- Cuando termines el proyecto, súbelo a **GitHub** y comparte el repositorio.
- Crear un fichero **README.MD** en el repositorio explicando cómo lanzar los test y la estructura del proyecto.

# Puntos extras(no obligatorios)

Los puntos extras ayudaran a tu evaluación utilizando las siguientes herramientas y tecnologías.

- Añadir sistema de *reportería* (**Allure report**)
- Utilización de API datafake (<https://fakejson.com/> , <https://github.com/DiUS/java-faker> o similar)

FALABELLA

