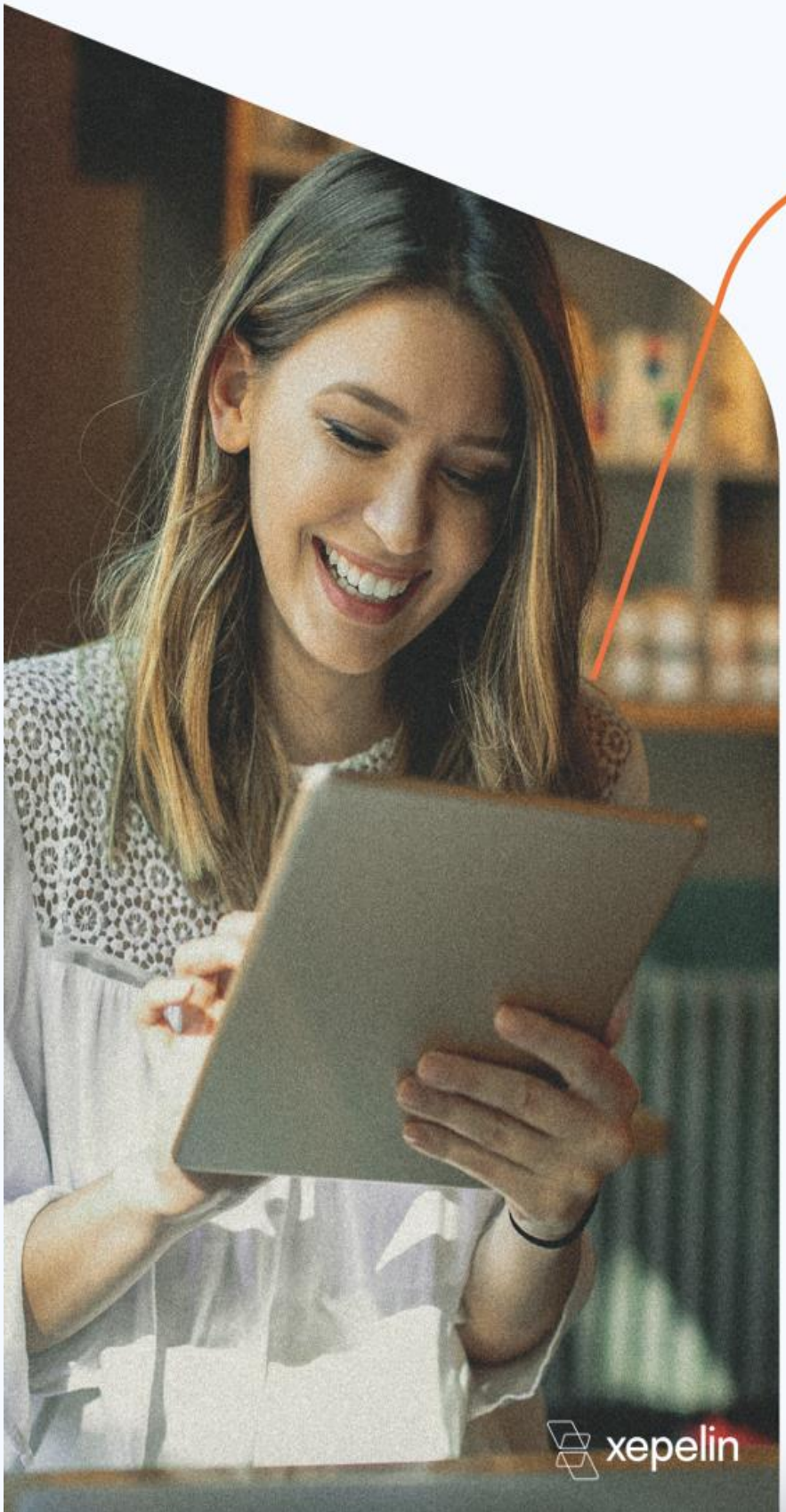
PRUEBA TÉCNICA

Quality Assurance



Indicaciones generales

- **Duración estimada de la prueba:** 60 minutos
- **Requisitos previos:**
  - Clonar el repositorio: Abrir
  - Crear una rama que se llame **test/<su nombre>**
  - Ejecutar el comando **npm install** para descargar e instalar las dependencias.

Ejercicio #1: Análisis de detección de errores y solución de problemas

Ejecutar el comando npm run test1 y buscar donde se encuentran los errores, para luego proceder a corregirlos.

Resultado esperado:
- No se debe presentar ningún error en la consola.
- No se permite cambiar lo que está dentro de console.log ()
- En la consola se espera un 200 como resultado.

Ejercicio #2: Prueba de API y automatización

Automatizar la siguiente prueba funcional de API utilizando la librería de Axios. En el archivo **2-test-api.ts** use los siguientes datos:

- **BASE_URL:** https://fakestoreapi.com
- **ENDPOINT**: /auth/login
- **USERTEST**: mor_2314
- **PASSWORD**: 83r5^_
- **PAYLOAD**:

```
{
    "username":"",
    "password":""
}
```

a. Crear una función asíncrona enviando un **POST** request con Axios. **Syntaxis**: axios.post (*<api endpoint>, <payload>*)
b. Imprimir el **"token"** desde el response.
c. Comparar el **status code** igual a **200**.

Finalmente, ejecute el comando **npm run test2** para validar los resultados obtenidos.

Resultado esperado:
- Se debe ejecutar la prueba con éxito y cumpliendo las condiciones a, b, y c.

Ejercicio #3: Refactorización de código fuente

Buscar, identificar e implementar todas las refactorizaciones y mejoras posibles del código fuente existente en el archivo: **3-test-refactor.ts**

Resultado esperado:
- No es necesario que modifique todo el código, pero sí debes mencionar e identificar todas las posibles mejoras que encuentres.

Ejercicio #4: Prueba automatizada a nivel de Front-end (Web)

Escenario: Validar que aparezcan los salarios de los empleados. Para ello codificar la siguiente prueba en la carpeta ***cypress***

a. Visitar la página: https://ultimateqa.com/simple-html-elements-for-automation/
b. Localizar la tabla de salarios **"HTML Table with unique id"**

HTML Table with unique id

Title	Work	Salary
Software Development Engineer in Test	Automation	$150,000+
Automation Testing Architect	Automation	$120,000+
Quality Assurance Engineer	Manual	$50,000+

Objetivo principal, lograr imprimir todos los salarios de la tabla. Una vez finalizado, ejecute el comando **npm start** para ejecutar el test y validar el resultado esperado.

Resultado esperado:
- Se debe ejecutar la prueba automatizada de forma exitosa, según los criterios de aceptación descritos anteriormente.

Ejercicio #4: Lógica de programación y desarrollo de algoritmos

Completar la prueba que se encuentra en el archivo del proyecto**: 4-test-logic.ts**

a. Debe ordenar los números de menor a mayor.
b. Y no deben venir números duplicados.

Una vez finalizado, ejecute el comando **npm run test4**, para validar su correcto funcionamiento.

Resultado esperado:
- Resultado esperado: [1, 2, 9, 10, 16, 98]

Para finalizar: subir los cambios al branch correspondiente y generar un Pull Request