NOMBRE: \_\_\_\_ Fiorella Milagros Silva Jara\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Caso de negocio**:

En la página de Choucair Testing, se requiere realizar las pruebas de la sección de “Empleos”. Las pruebas deben verificar diferentes atributos de calidad.

Usted es responsable de las pruebas por ende le piden que las realice. A continuación, se muestran los criterios a cumplir:

Ruta de la aplicación: <https://www.choucairtesting.com/>

1. **Criterios tácticos:**
2. Determine que debe probar (Para esto puede explorar la sección como guste y parta de la condición de que existen errores) para asegurar que se cumple con el mínimo de calidad.

**Usuario**

1. Considere la creación de casos de prueba. (Considerar pasos si es necesario)
2. De estos casos, clasifique que casos son automatizables y no automatizables.
3. Indique si existen y cuáles son las precondiciones para ejecutar los casos de prueba.
4. Presente estadísticas de prueba que permitan describir la conclusión de un indicador que defina.

**Adjunto documento Excel con el desarrollo de los puntos de Criterio 2,3,4 y 5.**

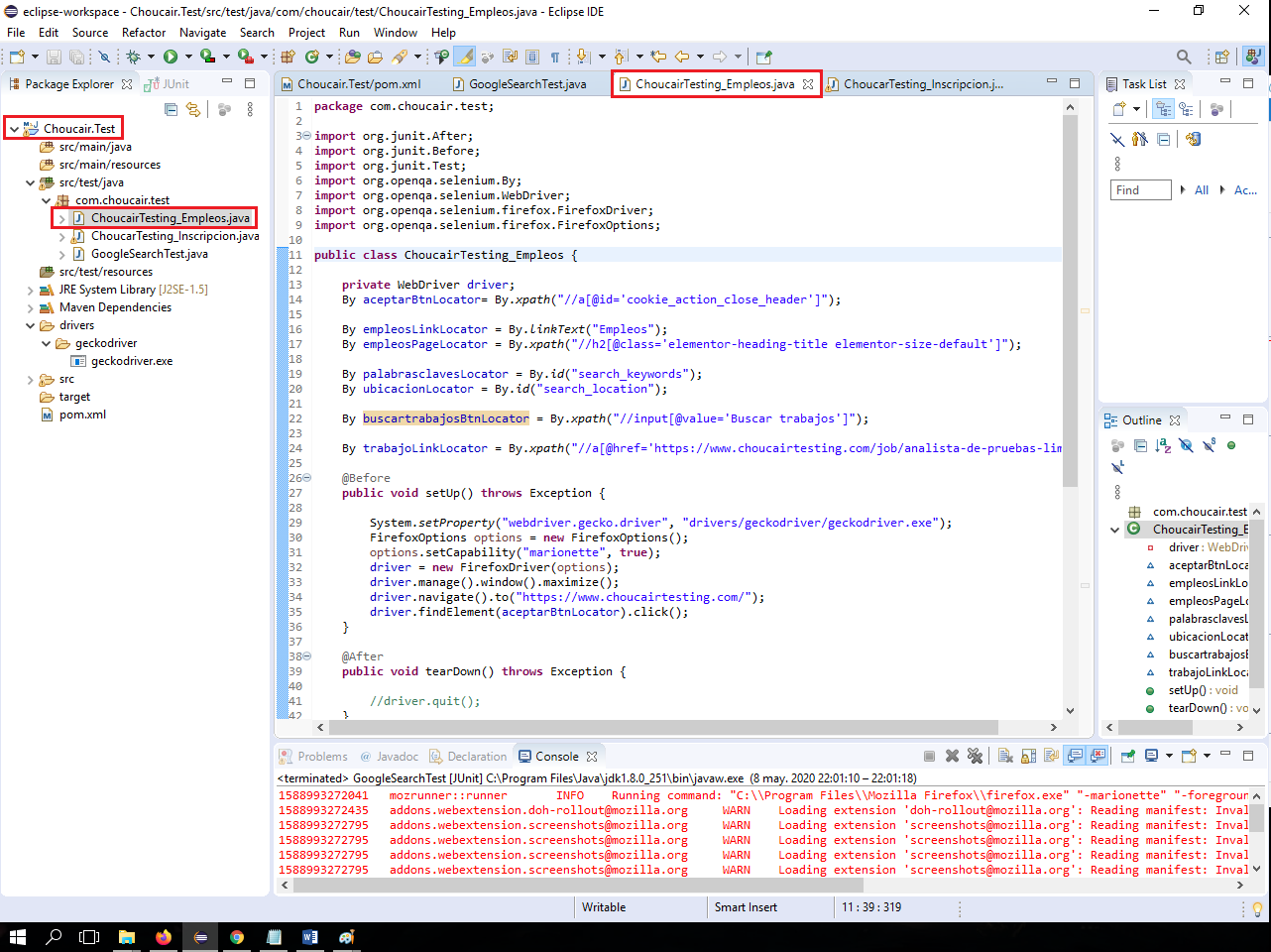
****

1. Sugiera mejoras al módulo probado.

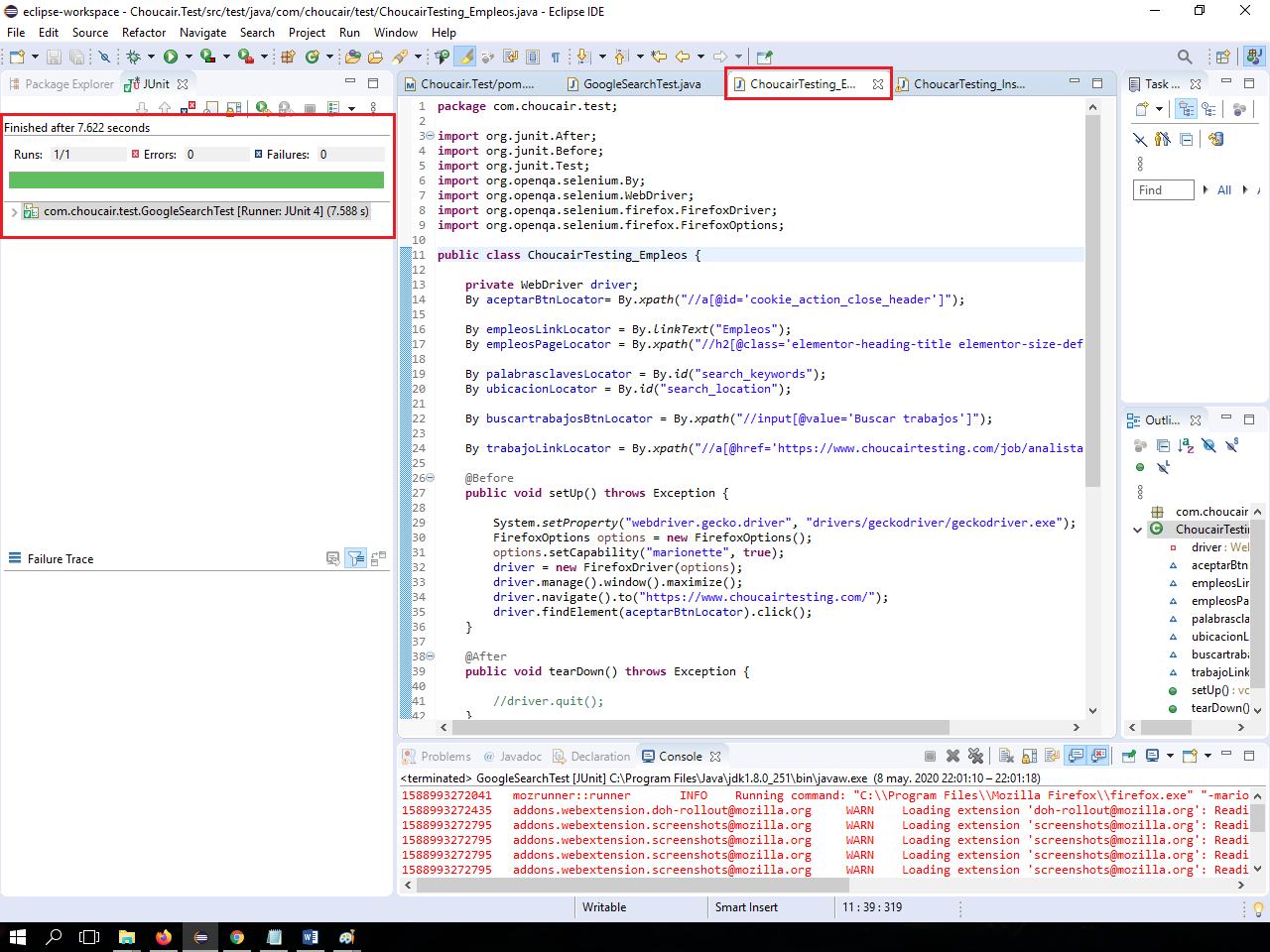
|  |  |
| --- | --- |
| Sugerencias | |
| CUS01.- Buscar empleos disponibles. | |
| · Mejorar el filtro de “Buscar Trabajos” respecto al campo de texto \*Ubicación\*. | El filtro “Ubicación” debería mostrarse como un Listado de Opciones y no como un campo de texto. |
| CUS02.- Generar inscripción de empleo. | |
| · Corregir el campo \*Celular o Teléfono de Contacto\*. | El campo \*Celular o Teléfono de Contacto\* no debería permitir valores negativos, y, además, debería mostrarse como un campo de texto en vez de ser un campo auto-incremental. |

1. **Criterios técnicos:**
2. Realice la automatización (programación) de los casos de prueba que seleccionó como automatizables. Se requiere implementar el uso de las siguientes Herramientas:
   * + Lenguaje: Java
     + Tipo de proyecto: Deseable Gradle
     + Framework de ejecución: Junit
     + Framework de pruebas: Selenium
     + Driver o navegador: Safari
     + Arquetipo: Page Object Model o Screenplay
     + Escribir HU en:
     + Ejecución: Junit o Jenkins Local
3. Realice la ejecución de lo automatizado, con sus respectivas evidencias.

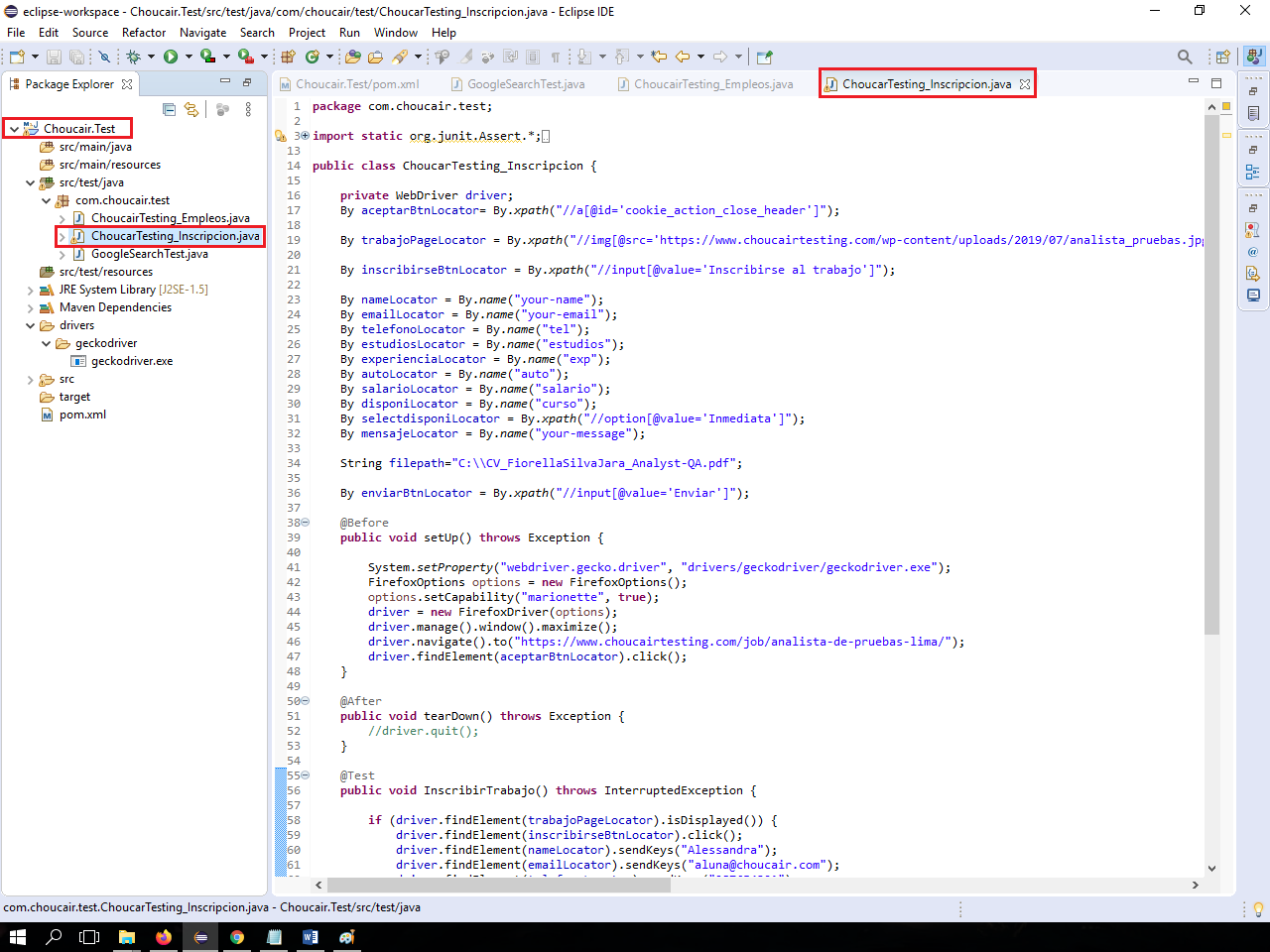
* **CUS01.- Desarrollo\_ChocairTesting\_Empleos**



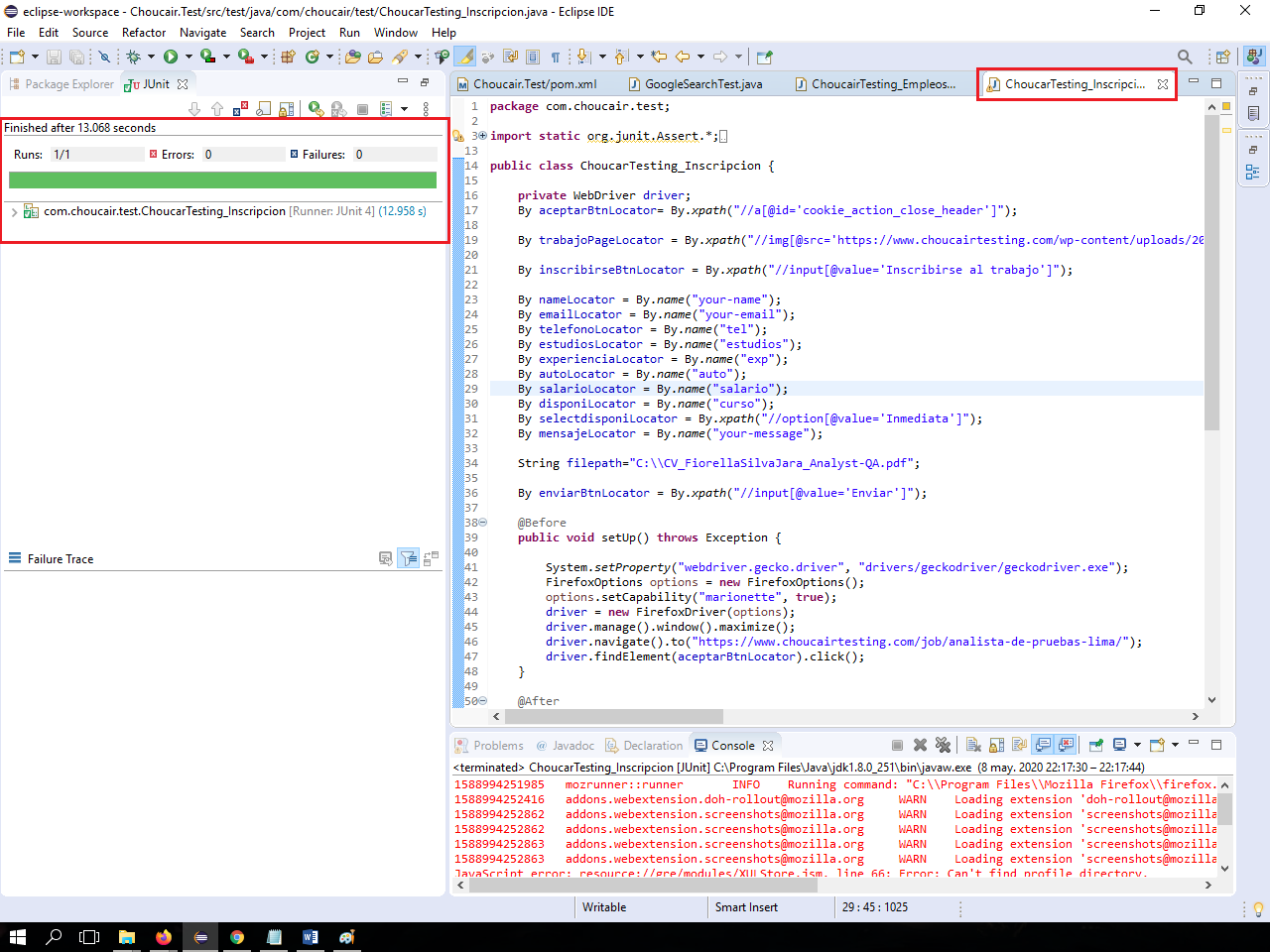
* **CUS01.- Ejecución\_ChocairTesting\_Empleos**

****

* **CUS02.- Desarrollo\_ChocairTesting\_Inscripcion**

****

* **CUS02.- Ejecución\_ChocairTesting\_Inscripcion**

****

1. Colocar el código automatizado en **github** y enviarlo. (Repositorio público)
2. **Agilismo**
3. De las sugerencias de mejora propuestas, defina historias de usuario que plantearía para un nuevo desarrollo, con sus respectivos criterios de aceptación.

|  |  |
| --- | --- |
| Historia de Usuario | Buscar empleos disponibles |
| Descripción | Como usuario, quiero realizar una búsqueda de empleo que se encuentre disponible en mi cuidad para inscribirme al puesto de trabajo. |
| Criterios de aceptación | * Se debe mostrar un listado de opciones con todas las ciudades disponibles para el puesto de trabajo. * Se debe permitir seleccionar una Ciudad. * Se debe permitir dejar de seleccionar una Ciudad. |

|  |  |
| --- | --- |
| Historia de Usuario | Generar inscripción de empleo |
| Descripción | Como usuario, quiero insertar mi celular o teléfono de contacto para inscribirme al puesto de trabajo. |
| Criterios de aceptación | * Se debe permitir datos de entrada numéricos en el campo \*Celular o Teléfono de contacto\* * No se debe permitir datos de entrada numéricos negativos en el campo \*Celular o Teléfono de contacto\* |

1. Teniendo en cuenta que estas historias de usuario se seleccionaron en un sprint, liste que ***tareas de prueba y de desarrollo*** son necesarias realizar estas historias.

|  |  |
| --- | --- |
| HU: Buscar empleos disponibles | |
| Tareas de Prueba | **Tareas de Desarrollo** |
| * Unit testing, de integración, pruebas de aceptación (automáticas o no), pruebas exploratorias, tests de regresión. * Pruebas asociadas a los Refactors/Rebuilds necesarios. * Preparación de la demo (datos de prueba, scripts…). | * Interfaces y métodos necesarios * Refactors/Rebuilds necesarios. * Reparación de incidencias del análisis de código estático. * Corrección de errores. * Actualizar la documentación (técnica, javadoc, de usuario…). |

|  |  |
| --- | --- |
| HU: Generar inscripción de empleo | |
| Tareas de Prueba | **Tareas de Desarrollo** |
| * Unit testing, de integración, pruebas de aceptación (automáticas o no), pruebas exploratorias, tests de regresión. * Pruebas asociadas a los Refactors/Rebuilds necesarios. * Preparación de la demo (datos de prueba, scripts…). | * Interfaces y métodos necesarios * Refactors/Rebuilds necesarios. * Reparación de incidencias del análisis de código estático. * Corrección de errores. * Actualizar la documentación (técnica, javadoc, de usuario…). |

1. **Conceptos Programación / CI / CD**

Describa con sus propias palabras cada uno de los siguientes conceptos:

1. Principios SOLID

Son los principios establecidos para facilitar a los desarrolladores la labor de crear programas legibles y mantenibles. Su aplicación facilita mucho el trabajo, tanto propio como ajeno.

1. Patrón Singleton

Es un patrón diseñado para limitar la creación de objetos pertenecientes a una clase. Su objetivo es el de garantizar que una clase solo tenga una instancia (o ejemplar) y proporcionar un punto de acceso global a ella.

1. Patrón FIRST

Es un patrón diseñado para definir las características que deben tener los casos de prueba para ser consideradas pruebas CON CALIDAD.

Características: Rápido, Independiente, Repetible, Auto evaluable, Oportuno

1. Patrón AAA

Es un patrón aceptado de forma global como forma habitual de escribir pruebas unitarias. Se divide en 3 secciones: Arrange (Organizar/Inicializa), Act (Actuar), Assert (Confirmar/Comprobar).

1. Pull Request

Es una petición de validación que el propietario de un Fork de un repositorio hace al propietario del repositorio original.

1. Release Train

Es un método que busca planificar la entrega de software según un calendario predeterminado y regular, como si de un tren se tratara. El calendario es público para que todos los equipos puedan contribuir con la entrega y representa un compromiso que debe ser cumplido.

1. Quality Gates

Es un hito especial en un proyecto. Se encuentra entre la fase de un proyecto fuertemente dependiente del resultado de una fase previa.

1. Diferencias servicios SOAP / REST

|  |  |
| --- | --- |
| REST | SOAP |
| El formato de mensaje de solicitud consiste en una cadena URI simple con una consulta. | El formato de mensaje de solicitud presenta estructura XML para determinar la operación que debe realizar el servicio web. |
| El formato de mensaje de respuesta devuelve mensajes de respuesta Notación de objeto JavaScript (JavaScript Object Notation, JSON) o XML. El formato de mensaje de respuesta no viene definido por un WSDL ni por un esquema. | El formato de mensaje de respuesta devuelve un formato XML según los parámetros definidos por el WSDL. |
| Pocas operaciones con muchos recursos | Muchas operaciones con pocos recursos |
| Comunicación punto a punto y segura | Comunicación origen a destino seguro |
| HTTP GET, HTTP POST, HTTP PUT, HTTP DEL | SMTP,HTTP POST, MQ |

¡Este es un reto! Por ende, puede consultar toda la información que sea necesaria u otra opción es preguntar las dudas que tenga a personas conocidas.