Modelos de Calidad de Software

Tema Nº12:Introducción a pruebas agiles – Parte 2

Indicador de logro Nº12:Automatiza los escenarios, usando Cucumber.

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

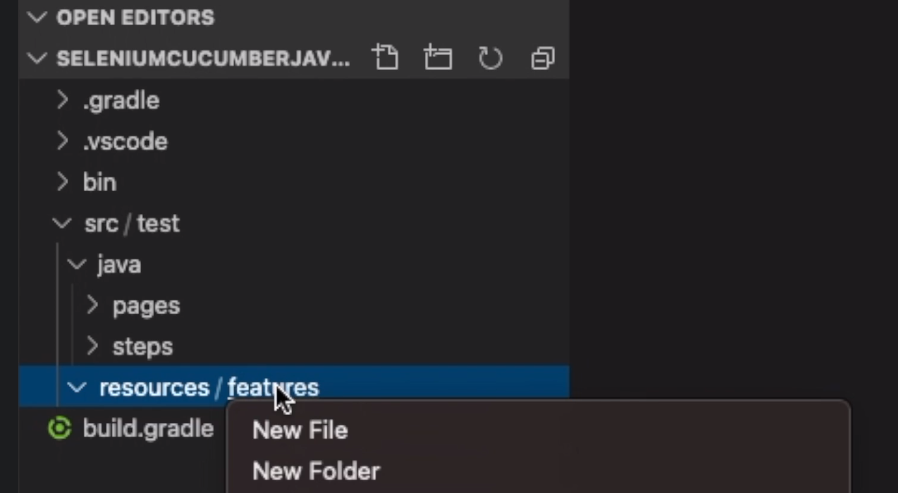
**TEMA Nº12:**

Introducción a pruebas agiles – Parte 2

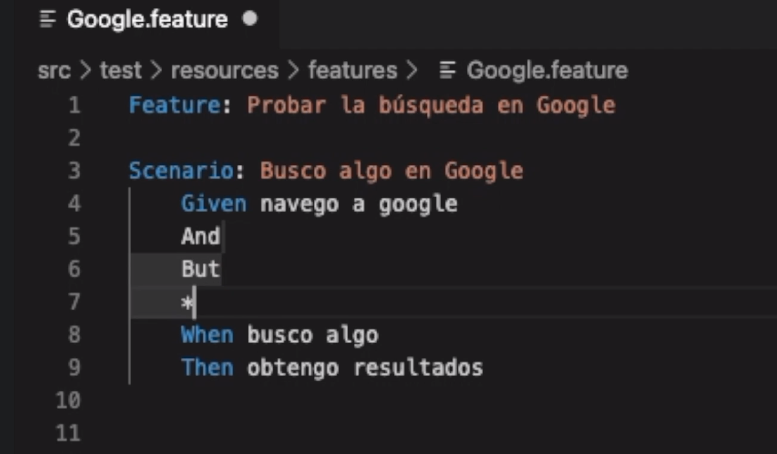
**Subtema 12.1:**

Creación de proyecto java con cucumber.

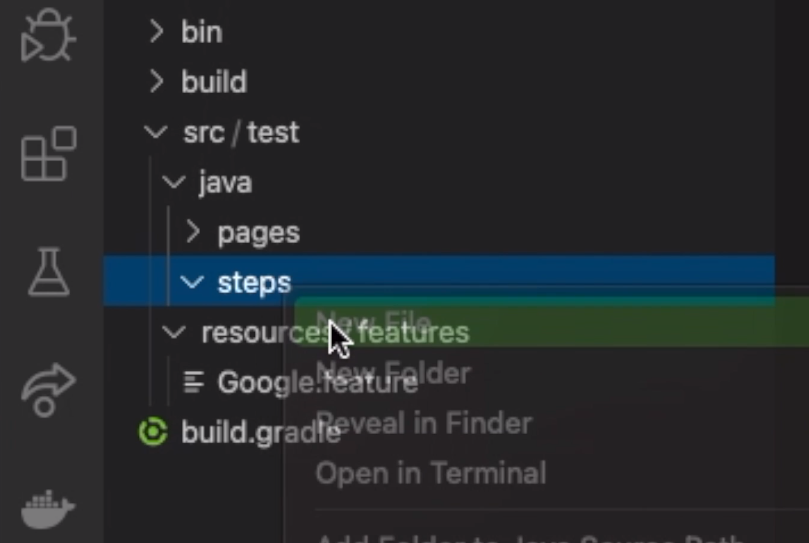
Iniciamos creando nuestro archivo Feature File de extensión feature el cual es el archivo típico y principal de Cucumber, y donde lo creamos



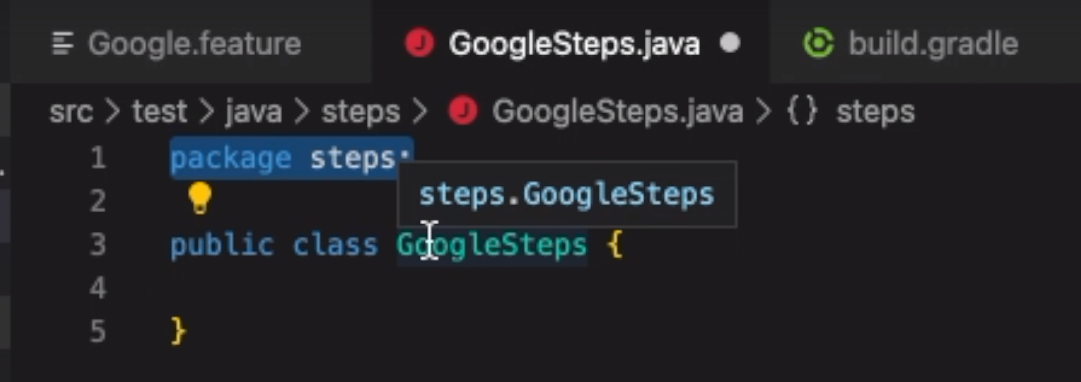
Definimos el Feature y Escenario



Estos pasos por si solos hacen nada, ahora hay que definir los pasos, creamos clase en Stpes.

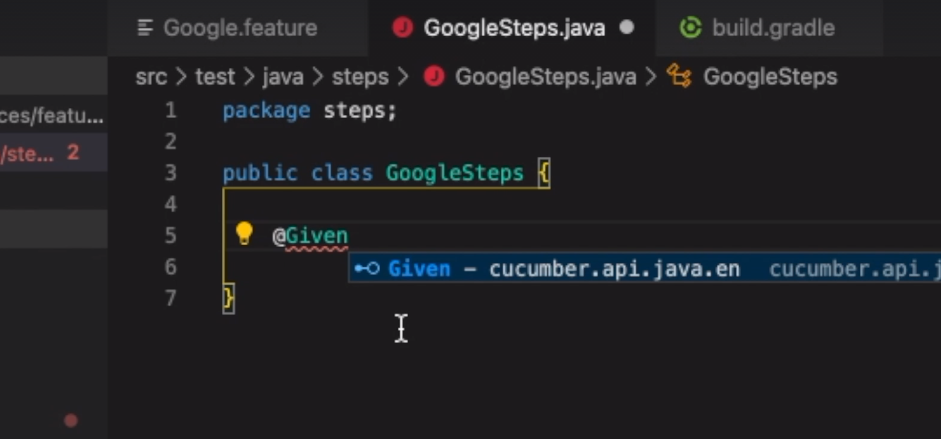


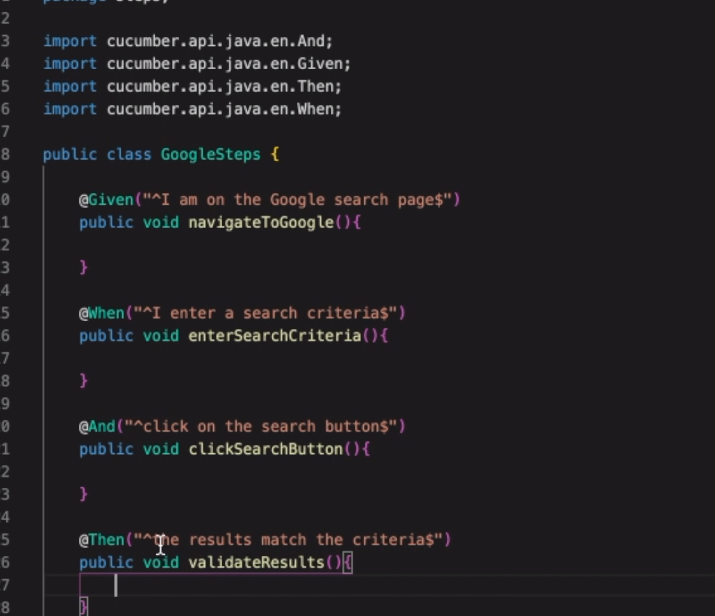
Sugerencia cada página debe tener una clase, motivo más fácil su mantenimiento.



Cada que creamos los pasos automáticamente Java pide importar las librerias, vamos a acompañar con una función public en cada paso, mantenemos la descripción del Feature.

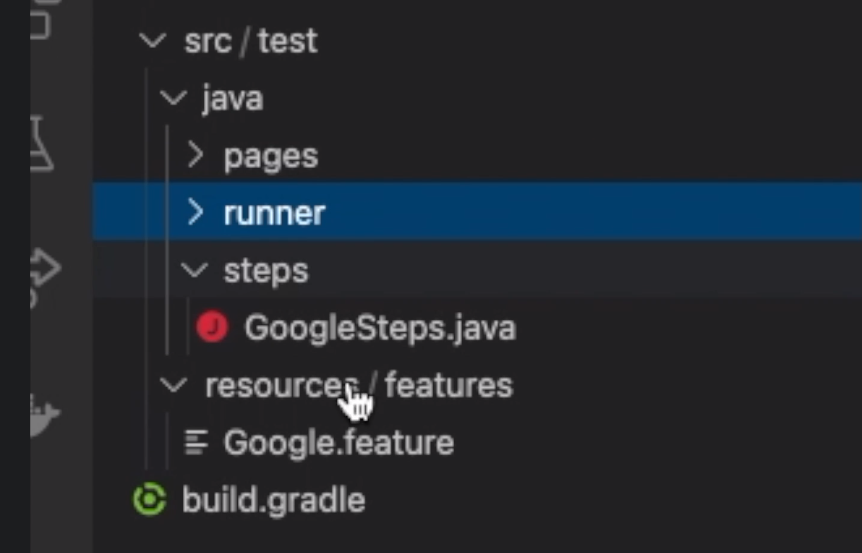
Tip: La descripción iniciar con simbolo de potencia y terminar con $ para evitar entendimientos ambiguos.

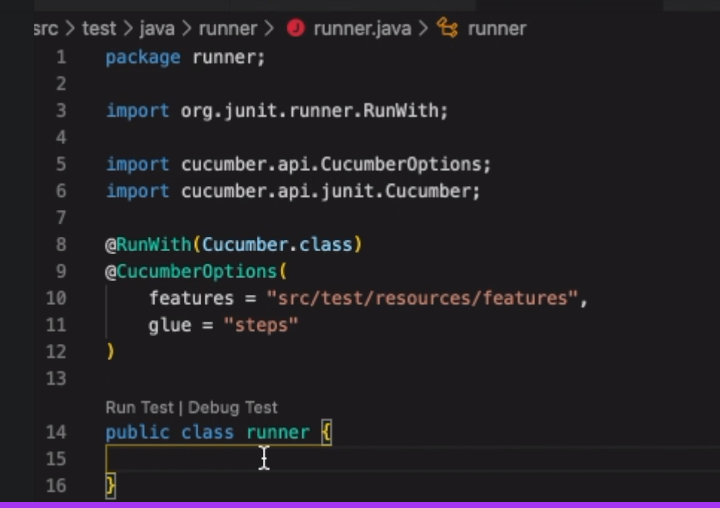




Luego de crear el step definition class, procedemos a crear el directorio Runner y un nuevo archivo runner.java, el cual es un archivo de configuración de Cucumber.

Indicamos donde están los Features en nuestro proyecto y donde están las definiciones para los steps de esos features (glue)



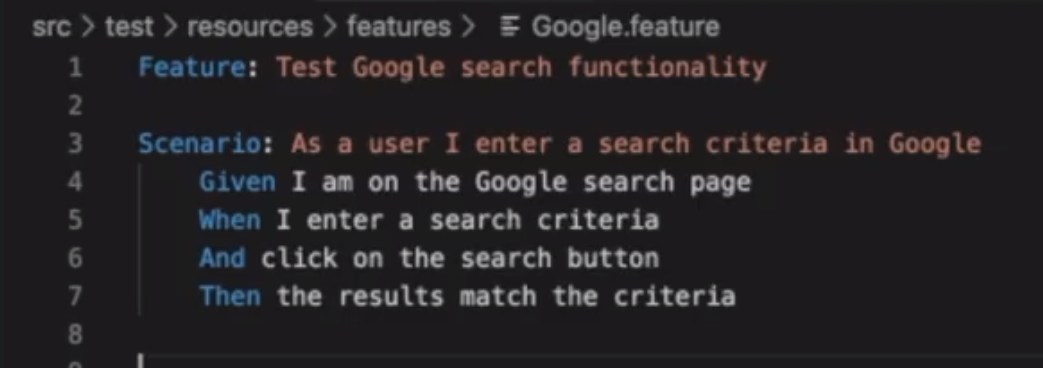


Importante usar el patrón POM (Page Objecto Model) donde indica usar una clase para cada página (Alineado a principio SOLID de responsabilidad única)

**Subtema 12.2:**

Estructura de un archivo Feature

El archivo Feature tiene la siguiente estructura Básica:

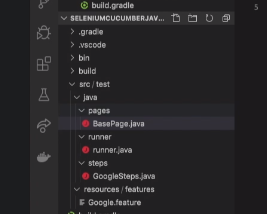


Este archivo de Cucumber mediante el cual vamos a escribir escenarios en el lenguaje Gherkin

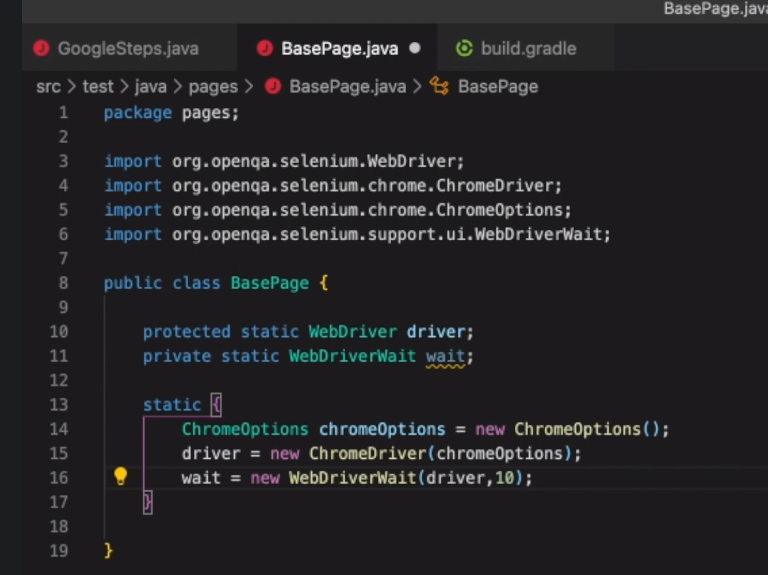
**Subtema 12.3:**

Automatización básica empleando selenium, cucumber y java

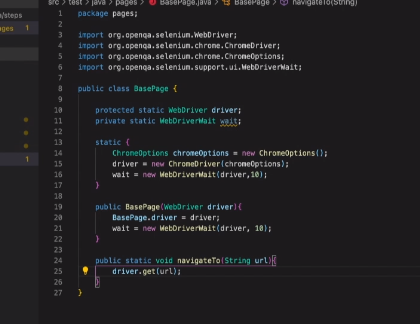
Reusando el Feature creado así como los Step Definition y pasos del Subtema 1.1 , ahora vemos una clase Base Page donde creamos una instancia única del web driver el cual se va compartir con las otras clases de páginas que van a heredar esta funcionalidad, en resumen aplicar herencia y encapsulamiento.



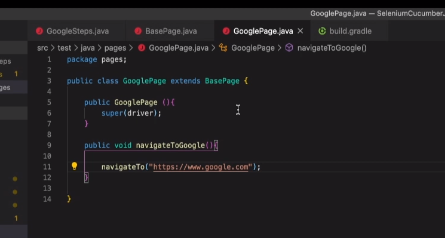
Considerar que el ejemplo que estamos haciendo es con Chrome y asignando un tiempo de espera de 10 segundos con bloques estáticos



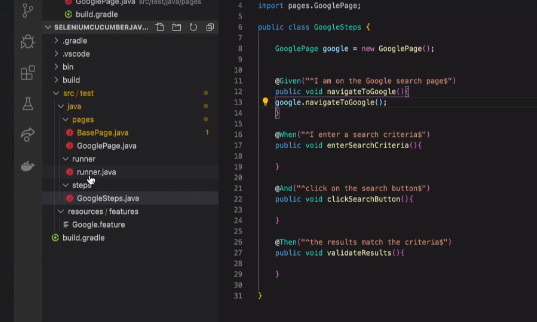
Ahora adicionamos un constructor (public BasePAge) y una función para navegar



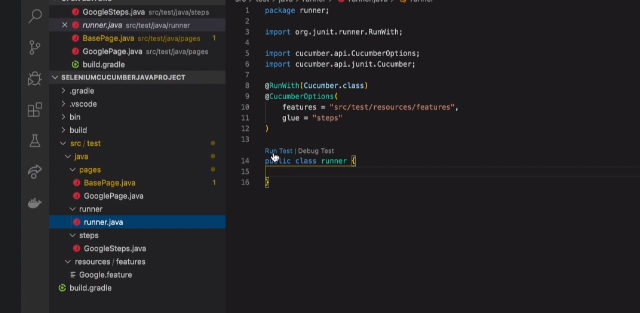
Ahora creamos una página más GooglePAge.java con uso de herencia

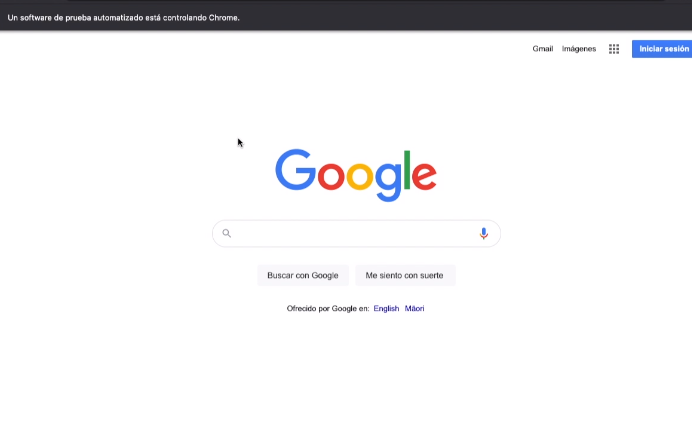


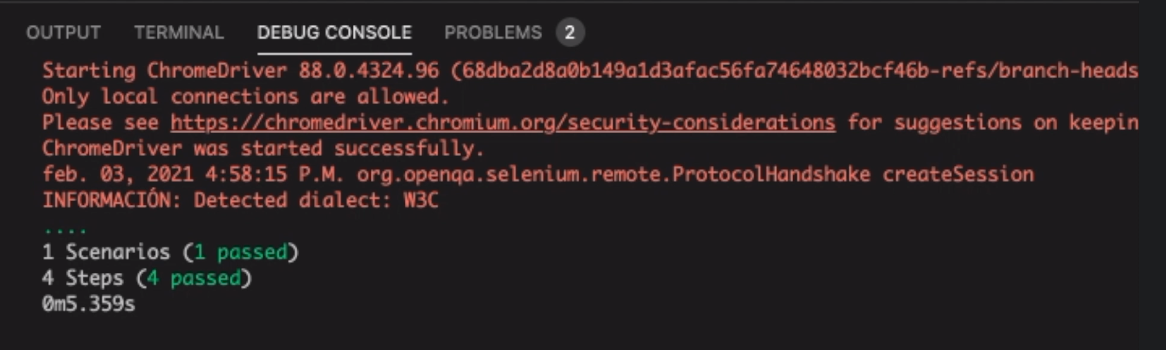
Ahora completamos GoogleSteps y abrimos una instancia del webdriver



Ahora ejecutamos Run







Actividad:

Revisa y analiza el enlace sugerido\* en la plataforma virtual, luego realiza la actividad propuesta:

* Efectuar la automatización de las historias de usuarios creadas en la sesión del tema 11 empleando cucumber.

\*Cada docente deberá emplear el enlace que considere apropiado para la ejecución de la actividad.