Modelos de Calidad de Software

Tema Nº11:Introducción a pruebas agiles – Parte 1

Indicador de logro Nº11:Crea casos de prueba a partir de las historias de usuario, empleando los conceptos, roles y ceremonias de SCRUM.

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº11:**

Introducción a pruebas agiles – Parte 1

**Subtema 11.1:**

SCRUM

Scrum es un marco de trabajo de adaptación iterativa e incremental, rápido, flexible y eficaz

diseñado para ofrecer un valor significativo de forma rápida en todo el proyecto.

Scrum es:

•Ligero.

•Fácil de entender.

•Extremadamente difícil de llegar a dominar.

-Ken Schwaber& Jeff Sutherland



Scrum muestra la eficacia relativa de las técnicas de gestión de producto y las técnicas de trabajo de modo que podamos mejorar continuamente el producto, el equipo y el entorno de trabajo.

El marco de trabajo Scrum consiste en Equipo Scrum y sus roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de cualquier proyecto.

Las reglas de Scrum relacionan los roles, eventos y artefactos y rigen las relaciones e interacciones entre ellos.

**Subtema 11.2:**

Ciclo de vida SCRUM

El corazón de Scrum es el denominado Sprint, ciclo o iteración de corta y similar duración, dentro del cual se generan, en pequeña escala, las etapas de creación de un producto: planificación, análisis, creación y validación de la entrega. En cada Sprint el trabajo está enfocado en la entrega de un “producto mínimo viable” o una entrega incremental de producto que se puede usar, inspeccionar y evaluar. El proceso en cada Sprint se repetirá hasta que el cliente de por cerrado el producto.

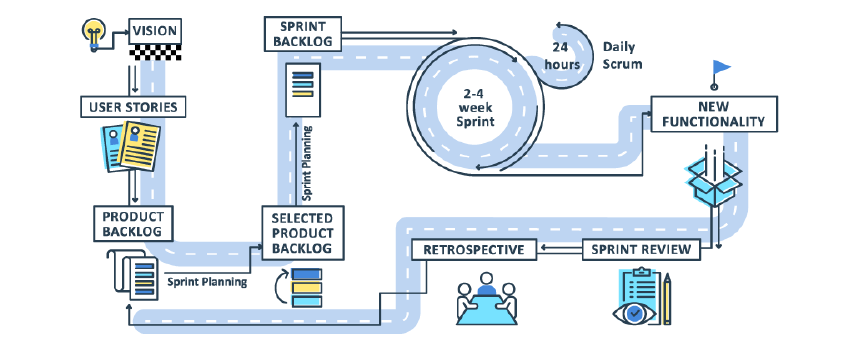


Figura 1

**Subtema 11.3:**

Roles y ceremonias

Un proyecto bajo Scrum se representan a través de roles conformados de la siguiente manera:

* Scrum Master, es el encargado de asegurar la aplicación de las reglas Scrum a lo largo de todo el proceso. Por lo tanto, es su responsabilidad liberar de restricciones al equipo para que puedan concentrarse en el desarrollo de cada incremento de producto.
* Dueño del producto (Product Owner), es la persona que conoce con claridad las necesidades del cliente y en consecuencia, puede tomar decisiones orientadas a maximizar el valor de lo que se debe obtener en cada Sprint.
* Equipo de desarrollo, son las personas que, de forma colaborativa, llevan a cabo las tareas necesarias para obtener un producto mínimo viable en cada Sprint. Su principal característica es que se encuentran auto organizados impulsando la co-localización. Con fines de facilitar la comunicación directa y maximizar la productividad, se sugiere que el equipo esté conformado entre 4 y 8 personas

Tal como se puede apreciar en la Figura 1, un Sprint está compuesto por cuatro tipos de eventos o ceremonias formalmente establecidos que se describen a continuación:

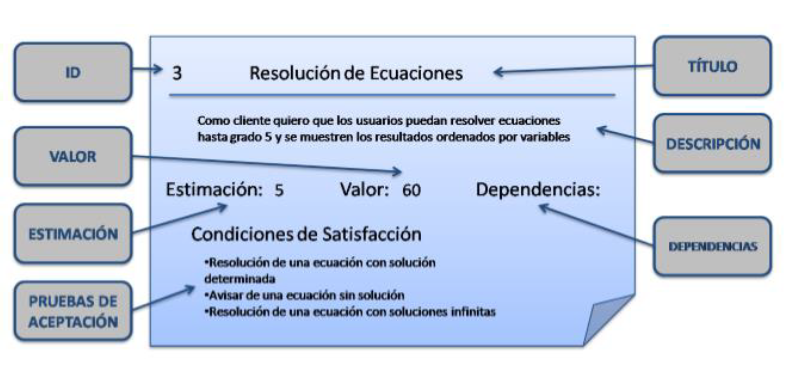
* Planificación del Sprint (Sprint Planning). Esta reunión tiene lugar para definir el entregable de la iteración y las estrategias que se desarrollarán para conseguirlo. Como resultado de ésta reunión se obtiene un artefacto denominado lista de pendientes del sprint o Sprint backlog.
* Scrum diario (Daily Scrum). Se trata de reuniones diarias de 15 minutos y de pie. Se llevarán a cabo para que todo el equipo esté sincronizado en los avances, restricciones y las próximas acciones a realizar.
* Revisión del Sprint (Sprint Review). Es una reunión cuyo objetivo es la validación por parte del propietario del resultado o “producto mínimo viable” obtenido en el sprint. Esta verificación se realiza junto a todo el equipo y los principales interesados del proyecto.
* Retrospectiva del Sprint (Sprint Retrospective). Al finalizar cada Sprint, los miembros del equipo se reúnen para reflexionar sobre la forma en que ha realizado el trabajo y explorar oportunidades de mejora.

**Subtema 11.4:**

Historia de usuarios

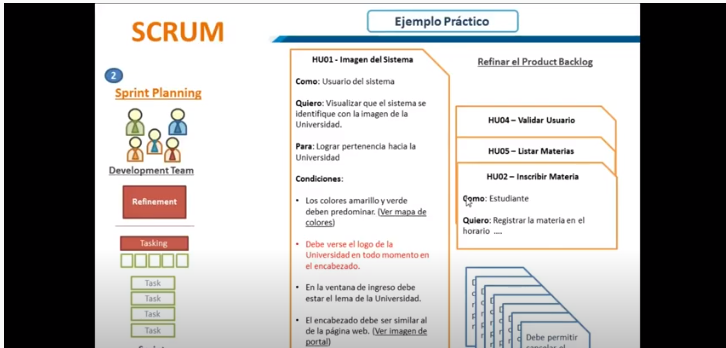
Una historia de usuario debe estar conformada por las 3C:

* Card: Descripción escrita de lo que necesita el usuario
* Conversación: El Product Owner y el Development Team aclaran los detalles
* Confirmación: (pruebas y verificación) Sirve para determinar lo que se espera



El siguiente enlace muestra en detalle y en forma práctica la construcción y manejo del Product Backlog, las Historias de Usuario, valorización y priorización

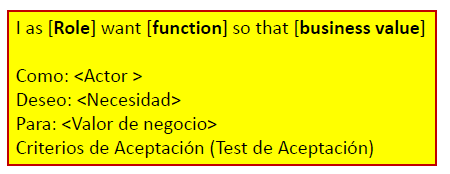
<https://www.youtube.com/watch?v=573jB8DQPEU>



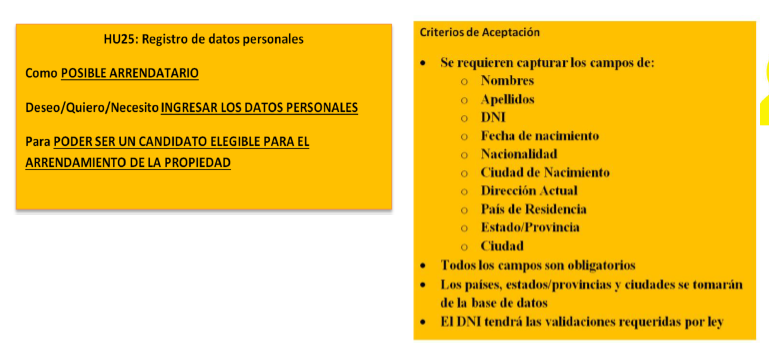
**Subtema 11.5:**

Creación de casos de prueba basados en historias de usuario

Los casos de prueba principales se obtienen de los criterios de aceptación presentes en las historias de usuario.



Ejemplo:



|  |  |
| --- | --- |
| Identificador del caso de prueba | CP\_001 |
| Nombre | Adicionar datos personales de posible arrendatario |
| Precondiciones | Debe estar creada una propiedad en alquiler |
| Entrada | Nombre: Juan  Apellidos: Perez  DNI: 08165233  Fecha de nacimiento: 22/04/1986  Nacionalidad: Peruana  Cuidad de Nacimiento: Lima  Dirección Actual: Av. Precursores 123 – San Miguel  País de Residencia: Perú  Estado/Provincia: Lima  Ciudad: Lima |
| Pasos | 1.- Ingresar Nombre  2.- Ingresar Apellidos  3.- Ingresar DNI  4.- Ingresar Fecha de nacimiento  5.- Ingresar Nacionalidad  6.- Ingresar Cuidad de Nacimiento  7.- Ingresar Dirección Actual  8.- Ingresar País de Residencia  9.- Ingresar Estado/Provincia  10.- Ingresar Ciudad  11.-Hacer Click en el botón “Aceptar” |
| Resultado Esperado | Mostrará un Mensaje “Ok” y salvará al posible candidato |
| Postcondiciones | Se verificará la existencia de un nuevo posible candidato. |
| Estado | No Iniciado |
| Prioridad | Alta |

**Subtema 11.6:**

¿Qué es Cucumber y Gherkin?

Cucumber es una herramienta colaborativa de software que admite el desarrollo basado en el comportamiento para lo cual es necesario desarrollar el mismo con los roles:

* Manager
* Developer
* Business Analyst
* Test Analyst

Cucumber no es una framework para automation es una herramienta para colaborar, para que todos los roles estén al tanto de que se está probando y funcione como una documentación viva. Si usas Cucumber sólo como rol tester no le están sacando el provecho y van a introducir capas que no suman.

Cucumber BDD (Behavior Driven Development) es su analizador de lenguaje ordinario llamado Gherkin el cual permite que los comportamientos de software esperados se especifiquen en un lenguaje lógico que los clientes puedan entender, esto permite obtener reportes por ejemplo para el Rol Manager los cuales son entendible y no output de consola con mucho código.

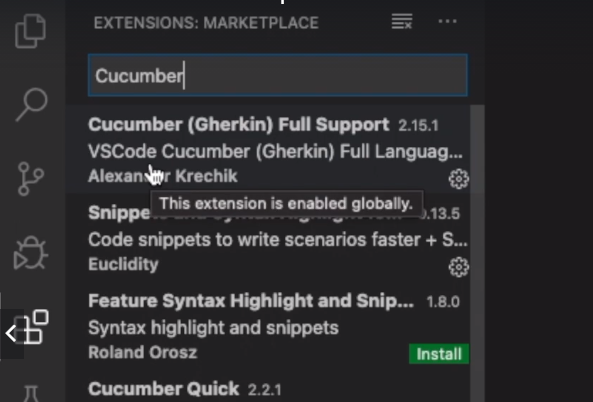


**Subtema 11.7:**

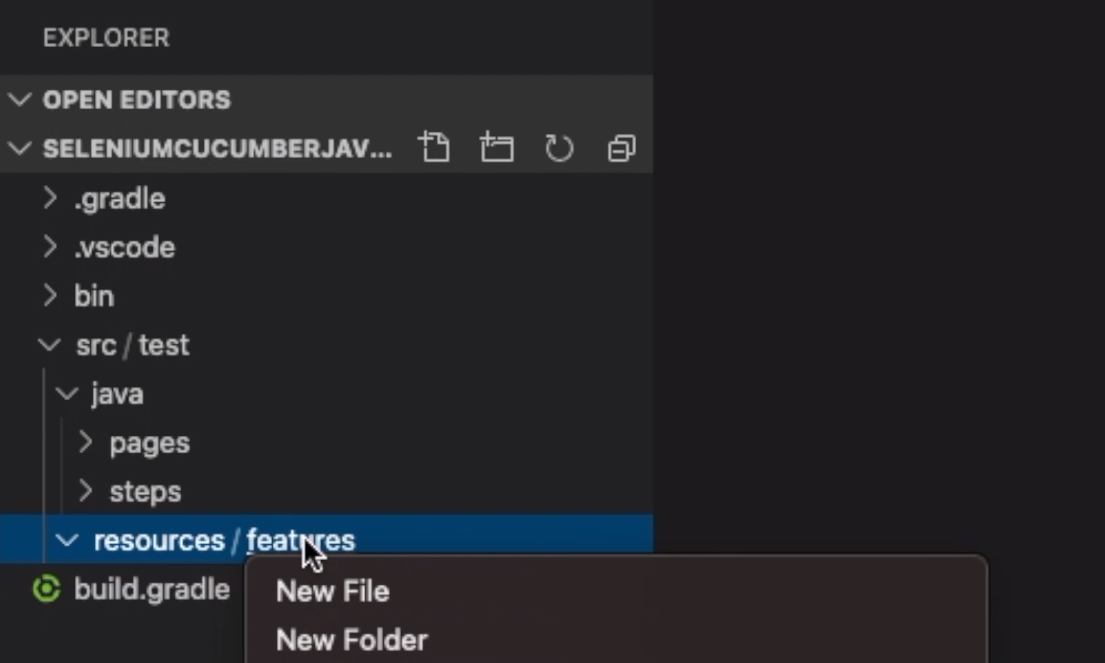
Configuración del entorno de pruebas

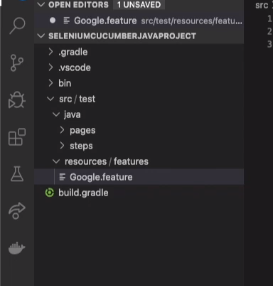
Pasos principales e identificación de variables para la configuración:

* Instalación de extensiones Cucumber (Gherkin) y Snippets and Syntax para un óptimo funcionamiento

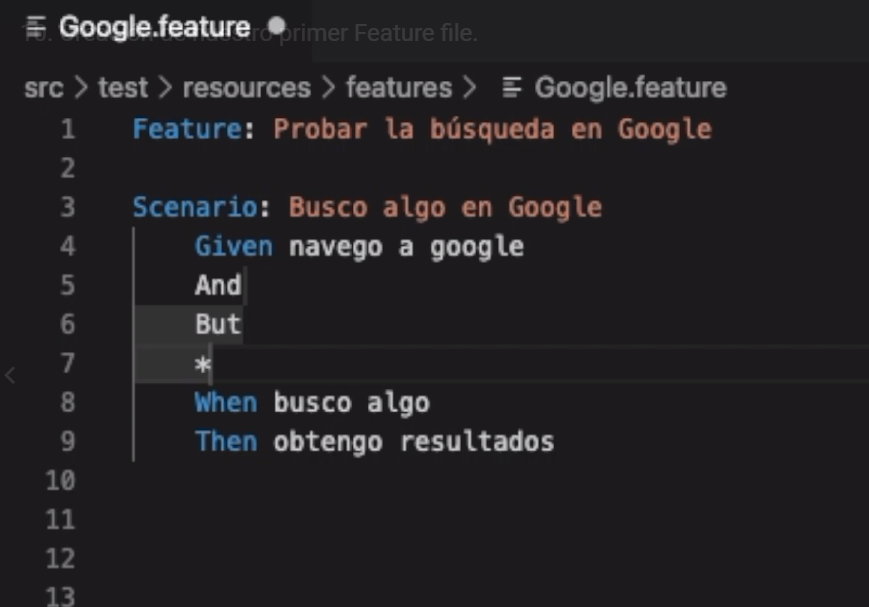


* Crear Archivo principal de Cucumber los Features Files (son Test en lenguaje de código no entendible por el cliente o usuario)

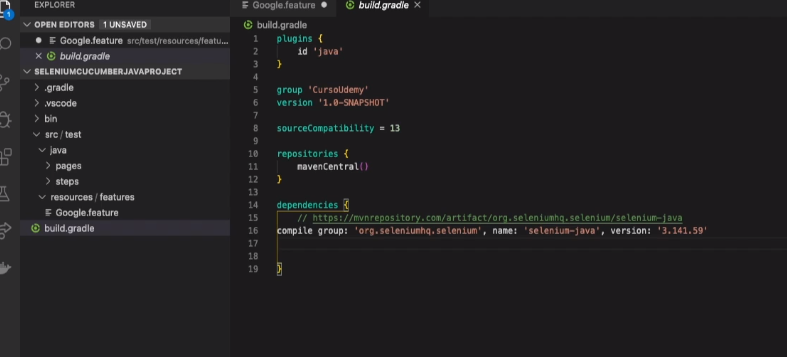


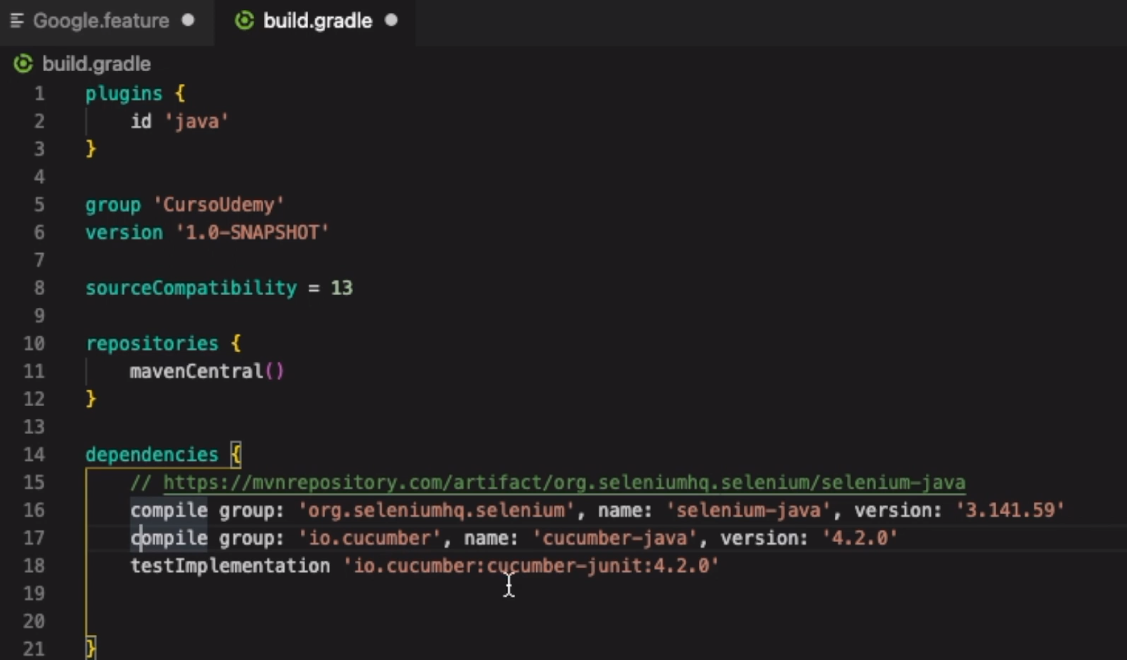


* Estructura básica de Feature

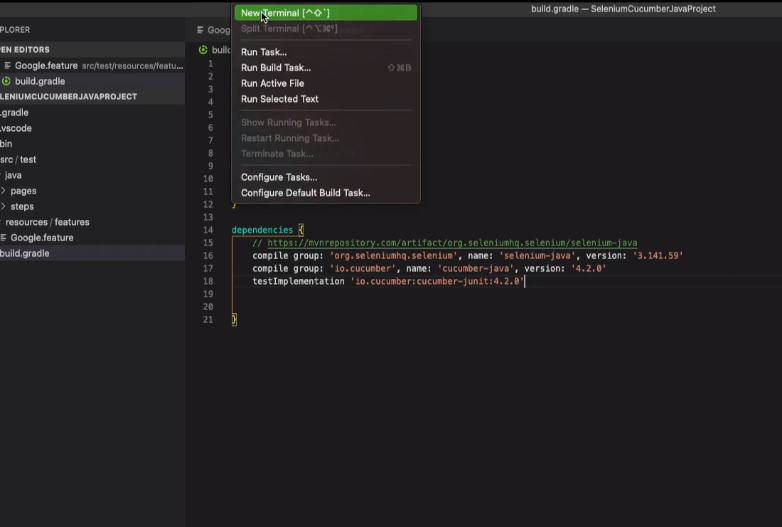


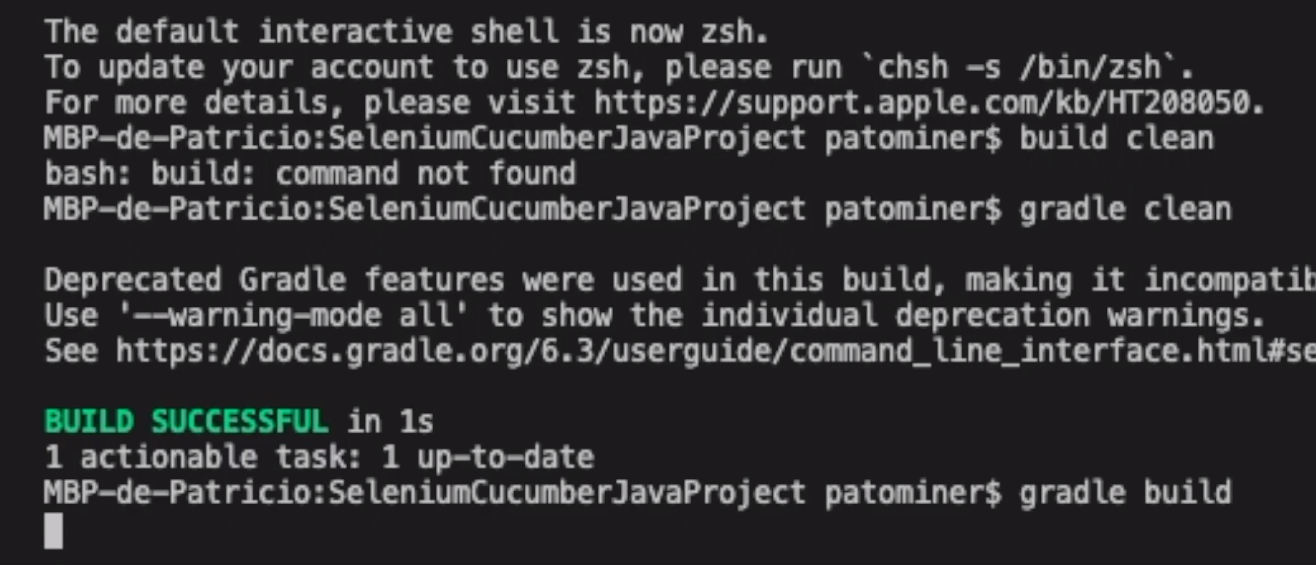
* Para que Cucumber funcione debe estar en Build.Gradle, porque si no está la dependencia no vamos a poder usarlo.





* Vamos a la terminal para ejecutar el gradel clean y gradel build







Actividad:

Revisa y analiza el caso sugerido\* en la plataforma virtual, luego realiza la actividad

Propuesta:

* Según las historias de usuarios brindadas en el caso práctico, crear los casos de prueba.
* Realice la configuración del entorno de pruebas usando gradle, java, selenium web driver y cucumber según lo revisado en clase.

\*Cada docente deberá proporcionar un caso práctico que considere apropiado para la ejecución de la actividad.