

Requisitos y pruebas en proyectos ágiles

Dirección General de Sistemas de Información y Comunicaciones

Áreas de Gobierno Tecnológico de SSII y del Dato



Servicio Andaluz de Salud
Consejería de Sanidad, Presidencia
y Emergencias

Contenido

- [Dirección General de Sistemas de Información y Comunicaciones](#)
 - [Áreas de Gobierno Tecnológico de SSII y del Dato](#)
- [Histórico de cambios](#)
- [Introducción](#)
- [Definición de requisitos. Historias de usuario](#)
 - [Historia de usuario: Acceso a PIT](#)
 - [Narrativa:](#)
 - [Criterios de aceptación:](#)
 - [Historia de usuario: Crear apartado de datos de identificación de usuario](#)
 - [Narrativa:](#)
 - [Criterios de aceptación:](#)
- [Casos de prueba: precondiciones y escenarios](#)
 - [Archivo .feature](#)
 - [Estructura y definición](#)
 - [ID de la historia de usuario](#)
 - [Feature](#)
 - [Background](#)
 - [Scenario](#)
 - [Etiquetado de los escenarios](#)
 - [Scenario Outline](#)
 - [Ejemplos](#)
 - [Buenas prácticas](#)

Resumen



- **Versión:** v02r02
- **Fecha publicación:** 10 de Febrero de 2023
- **Entrada en vigor desde:** 10 de Febrero de 2023

Histórico de cambios

Todos los cambios en el documento vendrán acompañados de un registro de las modificaciones. De este modo se podrá realizar un seguimiento y consultar su evolución.

Versión	v01r02	Fecha publicación	10 de Febrero de 2023	Fecha entrada en vigor	10 de Febrero de 2023
- Se actualiza el apartado "Archivo .Feature" incorporando aspectos a tener en cuenta en la redacción para la correcta importación de los mismos. - Actualización del apartado "Entrega de los features", incluyendo enlace al anexo correspondiente sobre la estructura del repositorio testprojet. - Se elimina el Anexo I: Gestión del ciclo de vida del software, dado que esta información ya se ha añadido en el anexo referenciado en el punto anterior.					

Versión	v02r01	Fecha publicación	3 de Mayo de 2022	Fecha entrada en vigor	3 de Mayo de 2022
Alcance					
- Cambio de título. Anteriormente el artículo estaba publicado dentro del marco agile como "Documentación a entregar proyectos ágiles" - Actualización del apartado de buenas prácticas. - Se simplifica el título de los escenarios y se añaden aclaraciones para ayudar en su redacción. - Se modifican las reglas de etiquetado para adaptarla al nuevo modelo de gestión basado en aplicación /componente. - Inclusión del Anexo I: Gestión del ciclo de vida del software.					

Versión	v01r01	Fecha publicación	29 de Octubre de 2020	Fecha entrada en vigor	29 de Octubre de 2020
Alcance					
- Definición de Historias de usuarios y requisitos. - Formalización de los casos de pruebas y precondiciones en Gherkin. Ficheros .feature.					

Introducción

Con este artículo se pretende abordar la cuestión de la documentación a entregar en los proyectos gestionados bajo un marco de metodología ágil. Las historias de usuarios y las pruebas que la validan serán el pilar sobre el que sostener el conocimiento de las diferentes aplicaciones.

Actualmente la STIC dispone de un conjunto de herramientas destinadas a la gestión de los proyectos y del conocimiento generado como consecuencia del trabajo diario. JIRA y Confluence se utilizan habitualmente para este fin. Pero, tal y como se indica en el [DoD](#) estandar, el formato para los planes de pruebas de las historias de usuario funcionales serán archivos de extensión .feature que se subirán al repositorio de código Git, junto con el código a probar.

Esto nos permitirá trabajar en la generación de unos planes de prueba vivos, dinámicos, que se puedan organizar por funcionalidades, y que estén preparados para su automatización. La redacción de pruebas en este formato y la utilización de Gherkin como lenguaje para su redacción se extiende a todas las pruebas funcionales.

Cualquier otro medio o formato de entrega deberá ser previamente consensuado con la STIC y definido en el [DoR](#) y/o [DoD](#) del proyecto.

Definición de requisitos. Historias de usuario

La herramienta destinada a la gestión del proyecto es JIRA. En ella se darán de alta los diferentes requisitos que seguirán la siguiente organización jerárquica:

- Mejoras: Siendo una entidad propia del JIRA de la STIC, una Mejora es una representación de los objetivos /requisitos de negocio a alto nivel. Esta entidad representa el concepto de Épicas en proyectos ágiles (definición épica según Agile Alliance). Las Mejoras son subdivisibles en Historias de Usuario.
- Historias de usuario: una historia de usuario es una explicación general e informal de una función de software escrita desde la perspectiva del usuario final cumpliendo los requisitos INVEST para asegurar la calidad en su redacción. Su propósito es articular cómo un elemento de trabajo o una función software proporcionará valor al usuario final.
 - I - Independent (independiente)
 - N - Negotiable (negociable)
 - V - Valuable (valiosa)
 - E - Estimable (estimable)
 - S - Small (pequeña)
 - T - Testable (comprobable)

El formato en el que se redactará seguirá la siguiente narrativa:

Como <rol: ¿Para quién vamos a construir esto?>

Quiero <acción: ¿Qué vamos a construir?>

Para <valor de negocio: ¿Por qué vamos a construirlo?>

Las historias incluirán sus criterios de aceptación en la propia historia de usuario de JIRA en lenguaje natural con formato Gherkin (tal y como se indica en el [DoR](#) y [DoD](#)), priorizando siempre la claridad y enfoque funcional al formato . Estos son fundamentales para validar que cumple con las necesidades de negocio, además de ayudar a los desarrolladores a entender el funcionamiento del producto y servir como guía durante la implementación de la funcionalidad.

En el caso de equipos con una madurez ágil baja o no familiarizados con los criterios de aceptación o con su correcta redacción y enfoque funcional, el uso de Gherkin ayuda y guía en la redacción paso a paso de los mismos, sin olvidar detalles fundamentales de la aplicación y descripción de las pruebas. El objetivo final es definir el funcionamiento de forma declarativa en base a ejemplos.

Plantilla de CAs en formato Gherkin:

Nº (de escenario)	Título (del CA)	Contexto	Evento	Resultado/Comportamiento
1	Título	Dado...	Cuando...	Entonces...

A modo ilustrativo, se añaden a continuación una serie de ejemplos sobre la definición de las historias de usuario con distintos niveles de complejidad utilizando para ello historias de usuario definidas en el ámbito del proyecto Estación Clínica Unificada (ECU):

- Historia de usuario simple

Historia de usuario: Acceso a PIT

Narrativa:

Como profesional que atiende a un paciente
Quiero acceder a la aplicación de Procesos de incapacidad temporal
Para gestionar la situación de incapacidad de mi paciente

Criterios de aceptación:

Feature: Acceso a PIT

N º E s c e n a rio	Criterio de Aceptaci ón (Título)	Contexto	Evento	Resultado / Comportamiento esperado
	Backgro und	Dado un profesional autenticado y un paciente identificado por su NUHSA y el icono de PIT pintado		

1	Acceder al módulo incapacidad temporal		cuando el layout está pintado y el icono de PIT está pintado y el usuario hace "clic" sobre el icono del estado de incapacidad temporal	entonces se abrirá el módulo de PIT en una ventana modal según prototipo.
2	Comportamiento modal de la ventana abierta		cuando se abre el módulo de PIT	entonces solamente la ventana conteniendo al módulo abierto estará activa, impidiéndose cualquier interacción fuera del contenedor de PIT.

Feature: Cierre de PIT

Nº	Criterio de Aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
	Background	Dado un profesional autenticado y un paciente identificado por su NUHSA y una instancia de PIT abierta		
1	Cierre de la ventana de PIT		cuando se cierre la ventana de PIT	entonces se desactivará el comportamiento modal que impide las interacciones fuera del contenedor de PIT.

2	Actualización información asociada al estado de incapacidad temporal del paciente en ECU		cuando se cierre la ventana de <i>PIT</i>	entonces se notificará dicho cierre al layout para que fuerce el refresco de la información asociada al estado de incapacidad temporal del paciente.
---	--	--	--	---

- Historia de usuario compleja

Historia de usuario: Crear apartado de datos de identificación de usuario

Narrativa:

Como profesional que atiende a un ciudadano utilizando su historia clínica

Quiero tener sus datos de identificación siempre visibles

Para asegurar la correcta identificación en todo momento para mejorar la seguridad del paciente

Criterios de aceptación:

Feature: Mostrar Datos identificativos del paciente

N º E s c e n a rio	Criterio de Aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado

	Background	Dado un profesional autenticado y un paciente identificado por su NUHSA		
1	Pintar la sección de datos de paciente		cuando se dispongan de los datos del paciente	entonces se pintará un elemento en la pantalla informando los mismos en base al prototipo acordado.
2	Formato de nombre y apellidos		cuando se pinten el nombre y apellidos del paciente	entonces el nombre y apellidos se mostrarán con un tamaño de letra más grande y llamativo que el resto de la información incluida en el elemento de visualización.
3	Desbordamiento de nombre y apellidos		cuando se pinten el nombre y apellidos del paciente y estos no quepan en una sola línea	entonces se mostrarán ocupando un máximo de dos líneas con un tamaño de letra inferior al utilizado para una sola línea.
4	Política de no ocultación del nombre y apellidos		cuando se pinte cualquier elemento adicional flotante o no flotante en la interfaz de usuario	entonces el nombre y apellidos estarán siempre visibles no pudiendo ser ocultados de ningún modo de la interfaz de usuario.

5	Información de la edad del paciente		cuando se informe la edad del paciente	entonces se calculará en base a la fecha de nacimiento mostrándose <edad mostrada> <table><tr><th>Tiempo transcurrido desde fecha de nacimiento hasta <today></th><th>edad mostrada</th></tr><tr><td>< 1 mes</td><td>tiempo en días</td></tr><tr><td>>= 1 mes y <24 meses</td><td>tiempo en meses</td></tr><tr><td>>=24 meses</td><td>tiempo en años</td></tr></table>	Tiempo transcurrido desde fecha de nacimiento hasta <today>	edad mostrada	< 1 mes	tiempo en días	>= 1 mes y <24 meses	tiempo en meses	>=24 meses	tiempo en años		
Tiempo transcurrido desde fecha de nacimiento hasta <today>	edad mostrada													
< 1 mes	tiempo en días													
>= 1 mes y <24 meses	tiempo en meses													
>=24 meses	tiempo en años													
6	Dinámica del tooltip para la edad		cuando me posiciono sobre el dato relativo a la edad	entonces se mostrará la fecha de nacimiento del paciente.										
7	Informar fecha de nacimiento		cuando se informa la fecha de nacimiento del paciente	entonces esta se mostrará en formato dd-mm-yyyy										
8	Identificadores del paciente		cuando se informan los identificadores del paciente	entonces se mostrará el valor del NUHSA para el paciente seleccionado.										
9	Información relativa al género del paciente		cuando se informa el género del paciente	entonces se mostrará el valor <género> en base al <código de género> informado <table><tr><th>código género</th><th>género</th></tr><tr><td>male</td><td>Hombre</td></tr><tr><td>female</td><td>Mujer</td></tr><tr><td>other</td><td>Indeterminado</td></tr><tr><td>unknown</td><td>Desconocido</td></tr></table>	código género	género	male	Hombre	female	Mujer	other	Indeterminado	unknown	Desconocido
código género	género													
male	Hombre													
female	Mujer													
other	Indeterminado													
unknown	Desconocido													

10	Tooltip de información de contacto		cuando se posiciona el usuario sobre el icono relativo a la información de contacto	entonces se mostrará un tooltip conteniendo la información de contacto del paciente.												
11	Información de contacto del paciente		cuando se muestra la información de contacto del paciente	entonces se informarán los siguientes datos si existen en el registro <table><tr><th>Tipo de dato</th><th>Se informa</th></tr><tr><td>home</td><td>Si</td></tr><tr><td>mobile</td><td>Si</td></tr></table>	Tipo de dato	Se informa	home	Si	mobile	Si						
Tipo de dato	Se informa															
home	Si															
mobile	Si															
12	Dinámica del tooltip para la información de adscripción		cuando se posiciona el usuario sobre el icono relativo a adscripción	entonces se mostrará la información adicional de adscripción.												
13	Información de adscripción		cuando se informa la adscripción del paciente	entonces se informarán los siguientes datos en base a la existencia en el registro, en caso de no existir información de adscripción se informará con el valor "Sin datos" <table><tr><th>Tipo de dato</th><th>Se informa</th></tr><tr><td>Clave médica habitual</td><td>Si</td></tr><tr><td>Clave de enfermería</td><td>Si</td></tr><tr><td>Clave médica de desplazamiento</td><td>Si</td></tr><tr><td>Datos administrativos de la clave médica habitual</td><td>Si</td></tr><tr><td>Datos administrativos de la clave médica de enfermería</td><td>Si</td></tr></table>	Tipo de dato	Se informa	Clave médica habitual	Si	Clave de enfermería	Si	Clave médica de desplazamiento	Si	Datos administrativos de la clave médica habitual	Si	Datos administrativos de la clave médica de enfermería	Si
Tipo de dato	Se informa															
Clave médica habitual	Si															
Clave de enfermería	Si															
Clave médica de desplazamiento	Si															
Datos administrativos de la clave médica habitual	Si															
Datos administrativos de la clave médica de enfermería	Si															

14	Información de los datos administrativos de la adscripción del paciente		cuando se informan los datos administrativos de la clave médica y de enfermería actualmente adscritos al paciente	entonces se mostrarán el nombre y apellidos de las claves médicas adscritas						
15	Políticas aplicables a la información de adscripción en base a la adscripción actual del paciente		cuando se informan los datos de adscripción del paciente	entonces se mostrarán los datos de adscripción según las siguientes casuísticas <table><tr><td>Adscripción de desplazamiento activa</td><td>Datos de adscripción mostrados</td></tr><tr><td>Si</td><td>De la Clave médica de desplazamiento</td></tr><tr><td>No</td><td>De la Clave medica habitual De la Clave de equipo habitual De la Clave de enfermería habitual</td></tr></table>	Adscripción de desplazamiento activa	Datos de adscripción mostrados	Si	De la Clave médica de desplazamiento	No	De la Clave medica habitual De la Clave de equipo habitual De la Clave de enfermería habitual
Adscripción de desplazamiento activa	Datos de adscripción mostrados									
Si	De la Clave médica de desplazamiento									
No	De la Clave medica habitual De la Clave de equipo habitual De la Clave de enfermería habitual									
16	Identificación de clave médica activa		cuando un paciente dispone de una clave médica de desplazamiento	entonces se considera que es activa si la fecha actual <= fecha de adscripción de desplazamiento + número de meses desplazado						
17	Comportamiento cuando se produce error en la obtención de datos del paciente		cuando la información relativa a un paciente no ha podido ser obtenida	entonces se mostrará un mensaje de error adecuado y si es recuperable mostrará un botón de reintento para el usuario final						

Casos de prueba: precondiciones y escenarios

Para la formalización de los casos de pruebas y sus precondiciones, se usará lenguaje Gherkin. El proveedor los entregará en archivos de extensión .feature.

Los objetivos de esta entrega de feature son los siguientes:

- Gestión de pruebas y planes de pruebas.
- Visibilidad como *"live documentation"* del proceso de verificación funcional.
- Control de errores y extracción de métricas, cobertura funcional, contención, etc..

Este formato de entrega nos permitirá, por un lado, el alta de las pruebas de forma masiva en JIRA y, por otro, disponer de la redacción correcta para una posible automatización utilizando Cucumber según se recoge en la ["Normativa de automatización de pruebas"](#).

Algo a tener en cuenta desde el punto de vista de la organización de las pruebas es que éstas se agruparán, siempre que sea posible, en áreas funcionales que llamaremos Subsistemas. Cada uno de ellos tendrá asociado un código.

- El código del subsistema será numérico.
- La relación entre el subsistema y su código quedará reflejada en una tabla que se publicará en el espacio de Confluence del proyecto correspondiente. Para más información ver [ANEXO II](#)
- .

Archivo .feature

En las metodologías ágiles, los features (característica) es una unidad funcional de un sistema software que satisface una necesidad de las partes interesadas, representa una decisión de diseño y genera una opción de implementación.

Un archivo .feature también es un punto de entrada a la automatización de pruebas usando la herramienta cucumber. Este es un archivo donde se describirán las pruebas en lenguaje descriptivo. Es una parte esencial de Cucumber, ya que sirve como script de prueba de automatización y como documentación viva.

Estructura y definición

Como hemos comentado antes, éste será el formato de entrega de las pruebas. En esta sección vamos a centrarnos en su estructura y definición. Posteriormente hablaremos de su redacción, indicando un conjunto de buenas prácticas que nos ayuden a su correcta implementación.

De cara a la importación a JIRA, los archivos .feature deberán cumplir con una serie de características:

- Codificación: El fichero .feature debe estar codificado como "UTF-8 sin BOM" para que, al importarlo, se muestren los caracteres con tilde.
- El título: No puede llevar tildes ni contener espacios. Tampoco puede comenzar por "@", ni llevar "Ñ".

Como normal general, comenzará por el id de la historia de usuario, añadiendo una breve descripción. Esto nos ayudará a ubicarnos de forma rápida y sencilla. El formato sería el siguiente: **ID JIRA STIC_Breve Descripción.**

Por ejemplo:

El o los archivos .feature relacionados con la historia de usuario SIGLOCON-106, comenzará por "SIGLOCO N-106" seguido de una breve descripción (SIGLOCON-106_RecuentoDeFactura.feature).

- Caso de que se tenga que importar una nueva versión de un .feature, el nombre del archivo .feature, DEBERÁ COINCIDIR EXACTAMENTE CON EL ANTERIOR, a fin de que machaque los datos y no queden tests duplicados.

Más información relacionada con la importación en el artículo: <https://confluence.xpand-it.com/display/public/XRAY/Importing+Cucumber+Tests+-+REST>

A continuación, partiendo del siguiente ejemplo, se va a ir explicando cada una de las secciones que componen el archivo, así como las etiquetas que aparecen en el mismo.

PBT-342_BusquedaGoogle.feature

@REQ_PBT-342

Feature: Búsqueda en Google

Como usuario web, quiero buscar en Google para poder responder mis dudas.

Background:

#@PBT-144

Given que que una instancia de un navegador Chrome está accesible y operativa

@PROVEEDOR

Scenario: SUB001 - Búsqueda simple en Google

Given un navegador web en la página de Google

When se introduce la palabra de búsqueda "pingüino"

Then se muestra el resultado de "pingüino"

@AUTO_INDRA

Scenario Outline: SUB001 - Búsqueda simple en Google

Given un navegador web en la página de Google

When se introduce la palabra de búsqueda <frase>

Then se muestra el resultado de <frase>

And los resultados relacionados <relacionados>

Examples:

frase	relacionados	
pingüino	Pingüino emperador	
panda	Panda gigante	
elefante	Elefante Africano	

Lo primero que aparece es @REQ_PBT-342: éste es el id de la historia de usuario del JIRA de la STIC, con la que se relacionan las pruebas, junto con el prefijo REQ_, que es necesario para que se genere la relación historia-prueba en el JIRA una vez se carguen los ficheros .feature a la plataforma. Al cargar las pruebas en JIRA se establecerá un enlace entre cada prueba y su correspondiente historia de usuario. Esto nos permitirá obtener el porcentaje de cobertura de los requisitos.

Feature

El propósito de esta palabra clave es proporcionar una descripción a alto nivel de una funcionalidad, y agrupar las pruebas relacionadas. Se recomienda que dicha descripción coincida con el título de la historia.

Además, justo debajo de "Feature" y antes de "Background" se describirá de modo coloquial qué se prueba. Esto mejora la comprensión de los informes de resultados generados tanto para las pruebas automáticas como para las manuales, dando una visión más cercana a los lectores que no estén habituados a entender qué se ha probado.

Background

Bajo este epígrafe se crearán las precondiciones comunes a todos los escenarios. La intención es evitar repeticiones innecesarias y focalizar la prueba específica bajo el epígrafe de los Escenarios.

Al cargar las pruebas en JIRA, se creará una precondición por cada "Given" que aparezca en el apartado Background. Además, todos los escenarios del archivo, quedarán enlazados con sus precondiciones. Por otro lado, los escenarios se pueden enlazar también a precondiciones que ya estén dadas de alta en JIRA, con el objetivo de reutilizarlas evitando así su repetición. Para eso, en lugar de utilizar la palabra clave "Given", se usará la anotación #@id_precondicion, donde id_precondicion es el id del tique JIRA.

En nuestro ejemplo, las pruebas quedarán enlazadas a dos precondiciones: Una que se creará en el momento de la carga (ya que hay un único "Given"), y otra, a la precondición PBT-144. Es importante señalar que para enlazar a una precondición existente en JIRA, es necesario utilizar la anotación #@id_precondicion tal y como aparece en la imagen de arriba.

Scenario

Bajo esta palabra clave, estarán definidas la diferentes pruebas asociadas a una HU. Al cargar el archivo en JIRA, se creará una prueba por cada escenario.

Los títulos de los escenarios vendrán precedidos de la expresión **SUBXXX - título del criterio de aceptación que prueba**, donde **SUBXXX** es el subsistema sobre el que se prueba (**XXX** es el código del subsistema). Si no existiese una división funcional, estos "códigos de subsistemas" no aparecerían, por lo que no se incluirá el **SUBXXX**, sólo el título de la prueba.

EJEMPLO: El título de un escenario del proyecto DYECCWEB quedaría como sigue: *SUB002 - Pintar la sección de datos de paciente*, donde '002' es el código del subsistema "Identificación usuario" y 'Pintar la sección de datos de paciente' es el título del criterio de aceptación.

Etiquetado de los escenarios

Cada escenario llevará una serie de etiquetas que nos facilitarán su clasificación, y nos servirá para lanzar las pruebas por grupos, en el caso de que se automaticen. Las mínimas a utilizar son las que aparecen a continuación. En cualquier caso, si se estima oportuno, se pueden incluir otras, que previamente se hayan consensuado con la Oficina de Calidad.

- Se incluirá el código ALM del componente sobre en el que se ejecuta la prueba @CALMComponente.
- Si la prueba ha sido diseñada por el proveedor y no está automatizada, se colocará la etiqueta @PROVEEDOR.
- Si la prueba ha sido diseñada y automatizada por el proveedor, se colocará la etiqueta @AUTO_Nombre del proveedor de desarrollo (por ej.: @AUTO_INDRA o @AUTO_EVERIS).
- Si la prueba está prevista incluir como parte de un plan de pruebas de regresión , se colocará la etiqueta TESTING_REGRESION

Scenario Outline

Es posible la creación de una única prueba que se ejecutará para varios juegos de datos. En ese caso se utilizará el Scenario Outline que es un tipo de escenario donde se especifican datos de entrada. Son muy prácticos ya que gracias a esto no es necesario escribir un escenario por dato de entrada. Este tipo de escenarios tienen que llevar incluidos los datos en el formato que aparece en el ejemplo facilitado. En lo relacionado con el uso de las etiquetas, se seguirán las pautas indicadas anteriormente.

Ejemplos

A continuación se añaden algunos ejemplos dando continuidad a los utilizados en el apartado anterior.

US Simple: Acceso a PIT. Aunque esta historia es sencilla se podría dividir en dos .features.

ECU-33_AccesoPIT.feature

```
#language: en
@REQ_ECU-33
Feature: Acceso a la aplicación de PIT

    Background: profesional autenticado y un paciente identificado por su NUHSA
        Given Un profesional autenticado por un token, y con NUHSA, nombre y apellidos, género y fecha de nacimiento adecuados
            Given que el layout está cargado
            Given que el icono de "PIT" está pintado

    @AUTO_INDRA
    Scenario: SUB001 - Acceder al módulo PIT
        When se pulsa sobre el icono de "PIT"
        Then se podrá comprobar que se abre el módulo de "PIT" en una ventana modal según prototipo

    @AUTO_INDRA
    Scenario: SUB002 - Comportamiento modal de la ventana abierta
        When se abre el módulo de "PIT"
        Then se podrá comprobar que solo la ventana del módulo "PIT" está activa
        And no es posible la interacción fuera de la ventana
```

ECU-33_CierrePIT.feature

#language: en

@REQ_ECU-33

Feature: Cierre de la aplicación de PIT

Background: profesional autenticado y un paciente identificado por su NUHSA y una instancia de PIT abierta

Given Un profesional autenticado por un token, y con NUHSA, nombre y apellidos, género y fecha de nacimiento adecuados

Given que el layout está cargado

Given que hay una instancia de "PIT" abierta

@AUTO_INDRA

Scenario: SUB023 - Cierre de la ventana de PIT

When se cierra la ventana de "PIT"

Then se desactivará el comportamiento modal que impide las interacciones fuera del contenedor de PIT

@AUTO_INDRA

Scenario: SUB004 - Actualización información asociada al estado de incapacidad temporal del paciente en ECU

When se cierra la ventana de "PIT"

Then se refrescará de la información asociada a "situación incapacidad temporal"

US Compleja : Crear apartado de datos de identificación de usuario. Esta historia se podría dividir en los siguientes .
features dada la complejidad de la misma:

ECU-24_IdentificacionUsuario.feature

#language: en

@REQ_ECU-24

Feature: Comprobación de la funcionalidad que permite mostrar los datos de identificación del paciente

Background: Acceso a la información de un usuario con un profesional autenticado

Given Un profesional autenticado por un token

Given que el layout está cargado

@AUTO_INDRA

Scenario: SUB001 - Pintar la sección de datos de paciente

Given un paciente con un NUHSA, nombre y apellidos, género y fecha de nacimiento adecuados

When se muestra la información del paciente

Then se podrá comprobar que la información cargada es la correcta

And que el nombre primer apellido y segundo apellido se encuentran en la línea 1

And que el género, edad y nuhsa se encuentran en la línea 2

And que la información se muestra en el bloque denominado "Árbol"

@AUTO_INDRA

Scenario Outline: SUB002 - Desbordamiento de nombre y apellidos

Given que el layout está cargado

And un paciente <nuhsa> <nombre> <apellido1> <apellido_2> When se muestra la identificación del paciente

Then se podrá comprobar que <nombre> <apellido1> <apellido_2> se muestra en <numero_lineas> líneas

And el tamaño de letra se ajusta en función de la longitud de nombre y apellidos del paciente.

And está destacado con respecto al resto de la información

And sexo edad y nuhsa se muestra en la línea <num_linea_identificacion>

Examples:

	nuhsa	nombre	apellido_1	apellido_2
numero_lineas	num_linea_identificacion			
	AN1234567890	JOSE	PÉREZ	
RUIZ			1	2
	AN1234567891	JOSEFINA	FERNANDEZ	GARCIA DE LOS GRANDES
2		3		
	AN1234567892	LISANDRO JUAN	FERNANDEZ-CASASGRANDES	HERNANDEZ DE LA
RIVERA	2	3		

ECU-24_AdsquipcionPaciente.feature

#language: en

@REQ_ECU-24

Feature: Mostrar información de adscripción del paciente

Background: Acceso a la información de un usuario con un profesional autenticado

Given Un profesional autenticado por un token

Given que el layout está cargado

@AUTO_INDRA

Scenario Outline: SUB002 - Información de adscripción

Given que se han cargado los datos de identificación del paciente <nuhsa> When Se consulta en la información adicional del icono adscripción

Then se podrá comprobar que aparece un tooltip con la información de las claves de adscripción

<informar_de> And el tooltip no oculta ni total ni parcialmente el nombre y apellidos del paciente

Examples:

	nuhsa	
informar_de		
	AN1234567896	Medicina: JUANA GARCIA MORENO (12345678G) Enfermería: VANESA RECARTE LAZO (98765432E)
	AN1234567897	Medicina: DANIEL CENIZO SALVAGO (23456789G) Enfermería: Sin datos (87654321E)
	AN1234567100	Medicina: Sin datos (34567890G) Enfermería: RAFAEL RODRIGUEZ OSORIO

```

(76543210E) |
|AN1234567101|Medicina: Sin datos (45678901G) Enfermería: Sin datos
(65432109E) |
|AN1234567102|Medicina: SANDRA FERNANDEZ JIMENEZ (56789012G) Enfermería: Sin
datos |
|AN1234567103|Medicina: Sin datos (67890123G) Enfermería: Sin
datos |
|AN1234567104|Medicina: Sin datos Enfermería: ANTONIO RUIZ RODRIGUEZ
(54321098E) |
|AN1234567105|Medicina: Sin datos Enfermería: Sin datos
(43210987E) |
|AN1234567106|Sin
datos |

@AUTO_INDRA
Scenario Outline: SUB002 - Información de adscripción desplazado
    Given que se han cargado los datos de identificación del paciente <nuhsa> When Se
    consulta en la información adicional del icono adscripción
    Then se podrá comprobar que aparece un tooltip con la información de las claves de adscripción
    <informar_de> And el tooltip no oculta ni total ni parcialmente el nombre y apellidos del paciente

    Examples:
    |nuhsa
    |informar_de |
    |AN1234567109|Medicina: ADRIAN LORA GARCIA (76543210G) Enfermería: ESTELA MANZANO ROJO
    (23456789E)|
    |AN1234567110|Medicina: ISMAEL BANQUERI RUIZ (89012345G) Enfermería: Sin datos
    (34567890E) |
    |AN1234567111|Medicina: Sin datos (90123456G) Enfermería:
    (45678901E) |
    |AN1234567112|Medicina: Sin datos (14785236G) Enfermería: Sin datos
    (56789012E) |
    |AN1234567113|Medicina: NESTOR ROMERO ORTEGA (25896314G) Enfermería: Sin
    datos |
    |AN1234567114|Medicina: (36985214G) Enfermería: Sin
    datos |
    |AN1234567115|Medicina: Sin datos Enfermería: CARLOS RIVERO TORRES
    (67890123E) |
    |AN1234567116|Medicina: Sin datos Enfermería: Sin datos
    (78965432E) |
    |AN1234567117|Sin
    datos |

@AUTO_INDRA
Scenario Outline: SUB002 - Información de adscripción pediatría desplazado
    Given que se han cargado los datos de identificación del paciente <nuhsa> When Se consulta en la
    información adicional del icono adscripción
    Then se podrá comprobar que aparece un tooltip con la información de las claves de adscripción
    <informar_de> And el tooltip no oculta ni total ni parcialmente el nombre y apellidos del paciente

    Examples:
    |nuhsa |informar_de |
    |AN1234567118|Pediatría: LUIS MOLINA BERNALDEZ (32109876P) Enfermería: Sin datos(10987654E)|
    |AN1234567119|Pediatría: Sin datos (21098765P) Enfermería: Sin datos (12345678E)

```

ECU-24_ControlSeguridadPaciente.feature
<pre> #language: en @REQ_ECU-24 Feature: Control de la seguridad del paciente en base a datos identificativos Background: Acceso a la información de un usuario con un profesional autenticado Given Un profesional autenticado por un token Given un paciente con NUHSA, y datos personales adecuados Given que el layout está cargado @AUTO_INDRA Scenario: SUB002 - Política de no ocultación del nombre y apellidos Given que el layout está cargado And se muestra la información del paciente When se consulta en la información adicional de la edad del paciente Then se podrá comprobar que aparece un tooltip con la fecha de nacimiento con el formato dd-mm-yyyy And el tooltip no oculta ni total ni parcialmente el nombre y apellidos del paciente @AUTO_INDRA Scenario outline: SUB002 - Error en la obtención de datos del paciente Given que no se han obtenido los datos de identificación del paciente And el error es <tipo_error> When el sistema intenta pintar la información del paciente Then se podrá comprobar que se muestra el mensaje <mensaje_error> And un botón <boton_reintento> para intentar recuperar la información Examples: tipo_error boton_reintento mensaje_error recuperable Reintentar No ha sido posible obtener los datos de identificación del paciente. Inténtelo del nuevo no recuperable Error en la obtención de los datos de identificación del paciente. </pre>

Buenas prácticas

A continuación se indican una serie de buenas prácticas en Gherkin a tener en cuenta a la hora de escribir este tipo de escenarios. Están acompañadas de ejemplos para facilitar su comprensión:

#	Buena práctica
1	Escribe Gherkin para que las personas que no conocen la funcionalidad, la entiendan.

2	Un escenario, un comportamiento. Un escenario no puede estar representado por diversas acciones y resultados. Patrón incorrecto Given... When... Then... When... Then... Patrón correcto Given... When... Then...
3	Trata de evitar, en la medida de lo posible, referencias a los detalles de la interfaz de usuario. De esta forma, si la interfaz cambia, la prueba puede continuar siendo válida. Redacción incorrecta Given que se introduce un número entero And posteriormente se introduce otro When se pulsa el botón "SUMA" Then aparece el resultado en el recuadro de la esquina inferior izquierda Redacción correcta Given que se introduce un número entero And posteriormente se introduce otro When se calcula el resultado Then se comprueba que es correcto

4

Utiliza el "Scenario Outline" para cubrir distintas variaciones del mismo comportamiento.

Ejemplo incorrecto

Para probar la suma de dos números enteros, se diseñan tres pruebas. Una para sumar dos números enteros positivos, otra para sumar dos números enteros negativos y otra para sumar un número entero positivo, y un número entero negativo.

suma.feature

Scenario: Como usuario, quiero calcular la suma de dos números positivos

Given Se ha introducido "3" en la calculadora

And se ha introducido "1" en la calculadora

when se calcula el resultado

Then el número "4" aparece en la pantalla

Scenario: Como usuario, quiero calcular la suma de dos números negativos

Given Se ha introducido "-3" en la calculadora

And se ha introducido "-1" en la calculadora

when se calcula el resultado

Then el número "-4" aparece en la pantalla

Scenario: Como usuario, quiero calcular la suma de un número negativo y un número positivo

Given Se ha introducido "-3" en la calculadora

And se ha introducido "1" en la calculadora

when se calcula el resultado

Then el número "-2" aparece en la pantalla

Ejemplo correcto

Para probar la suma de dos números enteros, se diseña una única prueba. Se introducen como ejemplos la suma de dos números enteros positivos, la suma de dos números enteros negativos y la suma de un número entero positivo, y uno negativo.

suma.feature

Scenario Outline: Como usuario, quiero calcular la suma de dos números enteros

Given Se ha introducido <input_1> en la calculadora

And se ha introducido <input_2> en la calculadora

when se calcula el resultado

Then el número <output> aparece en la pantalla

Examples:

input_1	input_2	output
1	3	4
-1	-3	-4
-1	3	2
1	-3	-2

5	Redacción declarativa, no imperativa. Se redactará la prueba tratando que declarar cuál es el comportamiento esperado y no dando pautas o pasos a realizar para la prueba. Ejemplo incorrecto <pre> Given que un usuario se autentica con una tarjeta habilitada And el saldo disponible en su cuenta es positivo And el cajero tiene suficiente dinero And el cajero tiene papel suficiente para imprimir recibos When introduce la tarjeta en el cajero And escribe en el teclado el pin de la tarjeta And presiona el botón de confirmar pin And presiona el botón próximo a la opción extraer dinero And ingresa una cantidad menor o igual a mi saldo disponible And presiona el botón de confirmar extracción And presiona el botón de confirmar imprimir recibo ... </pre> Ejemplo correcto <pre> Given que un usuario se autentica con una tarjeta habilitada And el saldo disponible en su cuenta es positivo When selecciona la opción de extraer dinero And se introduce la cantidad de dinero menor al saldo disponible y disponible del cajero Then se obtiene el dinero And el dinero que obtuve se resta del saldo disponible de mi cuenta And el sistema devuelve la tarjeta automáticamente And el sistema muestra el mensaje de transacción finalizada </pre>
6	Hay que prestar especial atención a los datos que aparecen "quemados" en las pruebas. Es necesario garantizar que esos datos estarán siempre disponibles ya que, de no ser así, la prueba podría ofrecer un falso negativo La manera de hacerlo se consensuará con la STIC.
7	Si quieres añadir comentarios que puedan ser interesantes durante la ejecución de pruebas utiliza "#" y a continuación, incorpora el comentario. Por ejemplo: <pre> Feature: Búsqueda en Google # Autor:verificacion@oca.com </pre>
8	Incorpora siempre un título al Background del feature, ya que si no lo haces, cuando se importe el archivo a JIRA, la precondición se mostrará sin título y será complicado localizarla

9	Es conveniente que reutilices las precondiciones que ya están dadas de alta en JIRA. Para ello, basta con que indiques el id de la precondición ya creada con anterioridad acompañada de "@". Por ejemplo, en el caso de que ya contemos con la precondición "PBT-144" y vayamos a reutilizarla pondríamos: @REQ_PBT-342 Feature: Búsqueda en Google Como usuario web, quiero buscar en Google para poder responder mis dudas. Background: #@PBT-144 Given que que una instancia de un navegador Chrome está accesible y operativa
12	Incluye etiquetas identificativas a cada escenario, esto nos ayudará a realizar filtros a posteriori que nos ayude a localizar los test.
13	Recuerda esta máxima: • Given: es el punto de partida • When: acción que vamos a ejecutar • Then: resultado esperado
14	No escribas el mismo texto en las sentencias Given / When /Then, ya que si se automatizara la prueba, cucumber puede entender que son pasos duplicados Por ejemplo, una definición incorrecta sería: <i>Given there is money in my account</i> <i>Then there is money in my account</i>
15	Puedes usar "But" cuando quieras verificar que no se observa un resultado concreto (funciona igual que el Then), por ejemplo: <i>Given Cumplo una precondición When Ejecuto una acción</i> <i>Then Observo este resultado</i> <i>But No debería poder observar este otro resultado</i>

Entrega de los .feature

A continuación, se listan las características de la entrega de los archivos .feature.

- La redacción de las pruebas comenzará una vez la historia y sus criterios de aceptación hayan sido validados para su desarrollo. Conforme se vayan redactando los diferentes archivos .feature se irán adjuntando a la historia para su revisión por parte de la OCa. Una vez validados, se entregarán en Git, independientemente de si las pruebas van a ser automatizadas o no.
- Las pruebas se organizarán por subsistemas funcionales, dentro de la carpeta *"feature"*. Tendrán su propio repositorio, tal y como se indica en el [ANEXO II: Estructura del repositorio testproject](#) de la [Normativa de entrega](#).