# A.1: Desarrollo de los SPRINTS

## A.1.1 Sprint 1

Los Sprints anteriores, es decir, el 0a, 0b, y 0c consistieron en su mayoría en mitigaciones de riegos, de estas mayormente técnico, particularmente relacionados con conocimientos electrónicos, como también el conocimiento del protocolo de comunicaciones XMPP. Ya con mayor conocimiento electrónico, y sobre el protocolo XMPP podemos comenzar a desarrollar las modificaciones a «Openfire» y la codificación de la extensión para la librería «Smack».

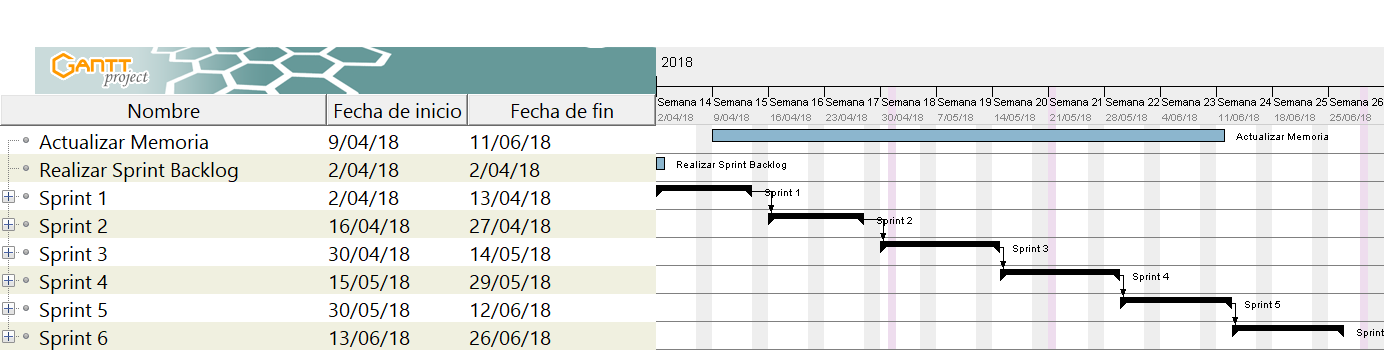
Durante este sprint, se comenzó la modificación de servidor XMPP Openfire.  
Los plazos del sprint tuvieron un leve retraso debido a que se gatillaron los riegos OF0348-RISK-04 y el OF0348-RISK-05, los cuales tienen que ver con estimaciones de tiempo, y disponibilidad de equipo de trabajo, por ende, se aplicaron las medidas de contingencia, es decir, se inyectaron mas horas hombre para este fin.

La reunión «Sprint Review» no se pudo realizar. Si bien se realizaron las pruebas de aceptación, pero no las realizo el «Product Owner». Esto se realizó, debido al profundo conocimiento del problema de equipo de testing.

### A.1.1.1 Planificación

La planificación será mediante una carta Gantt. Obedeciendo así la metodología de gestión del proyecto. Realizaremos tres Sprint para llegar a la culminación del proyecto, generando el producto de software esperado. Debemos mencionar que previo al desarrollo de los Sprint, se confeccionó un «Product Backlog» en donde se añaden todas las tareas a realizar en el presente proyecto, las cuales se agruparán en tres conjuntos, donde cada uno de estos se asigna a un sprint.

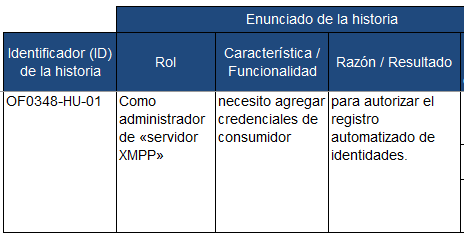
A continuación, en la imagen XXX se puede ver el grafico de la carta Gantt de planificación:

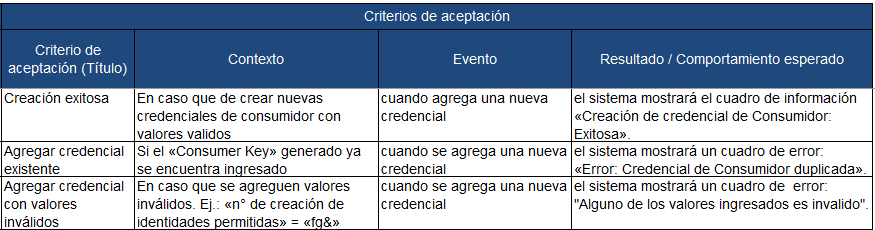


Como se ha mencionado anteriormente, se ha generado un «Product Backlog», el cual, contiene las Historias de Usuario a realizar durante el proyecto, este se puede ver en la imagen XXXX (Imagen Product Backlog). A continuación, veremos las Historias de Usuario seleccionadas para realizar durante el presente Sprint.

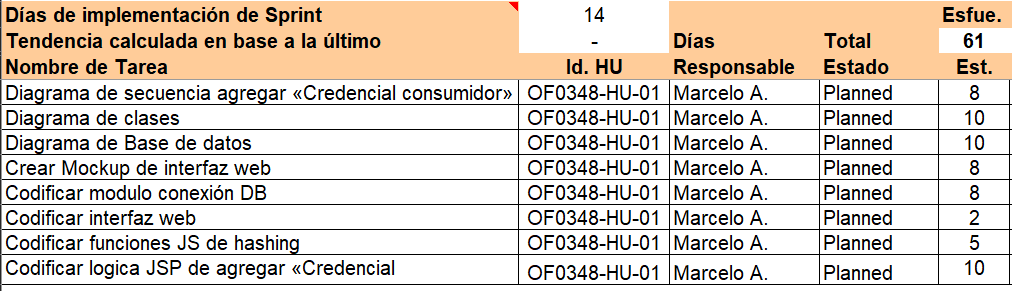






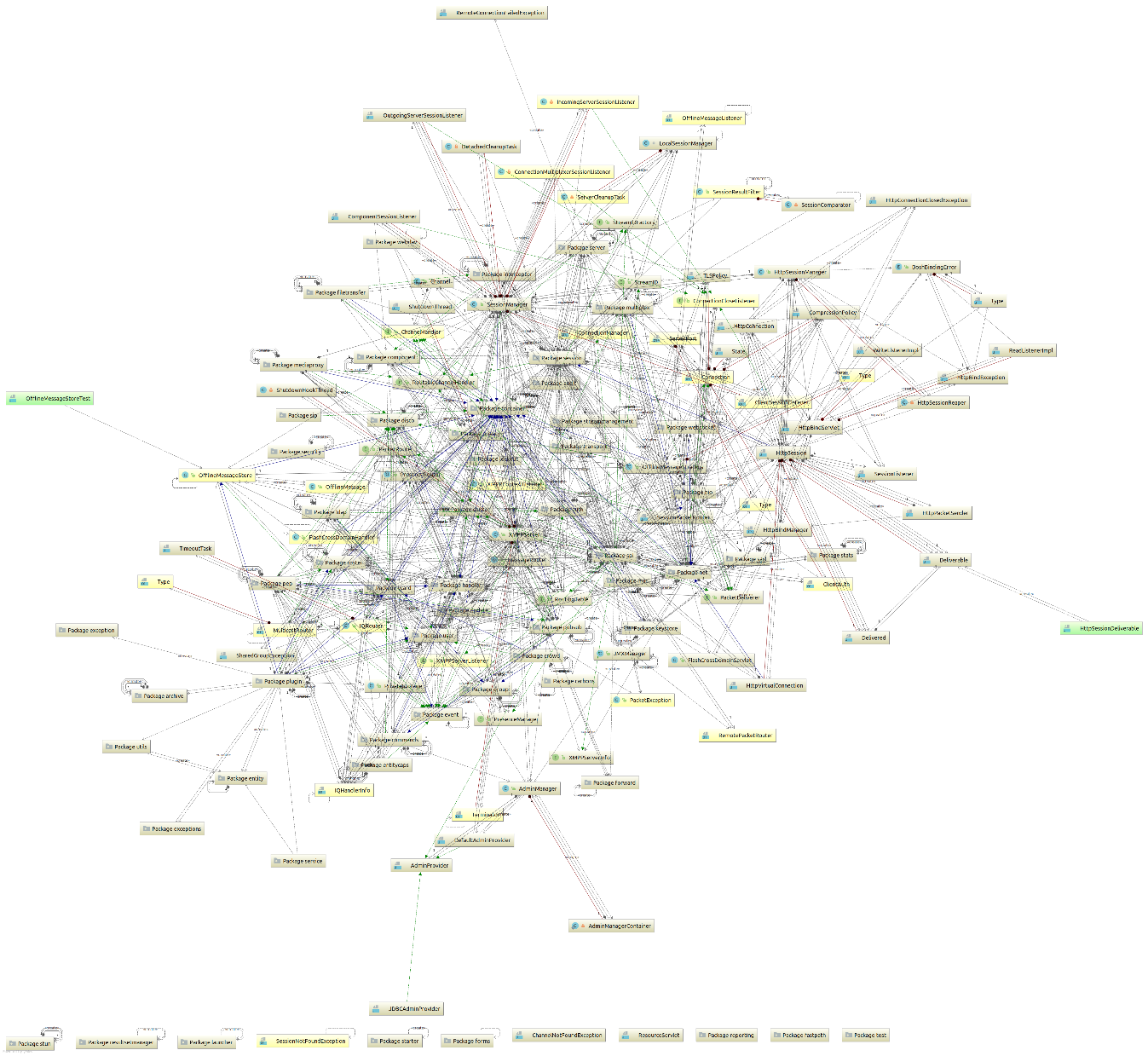


Ya seleccionadas las Historias de Usuario a desarrollar, las desglosamos en taras en la tabla XXXXX , a continuación.

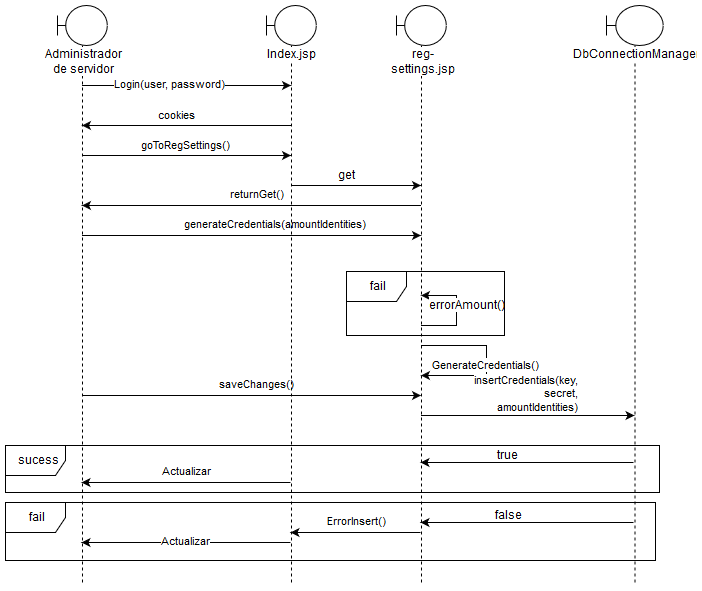


### A.1.1.2 Diseño

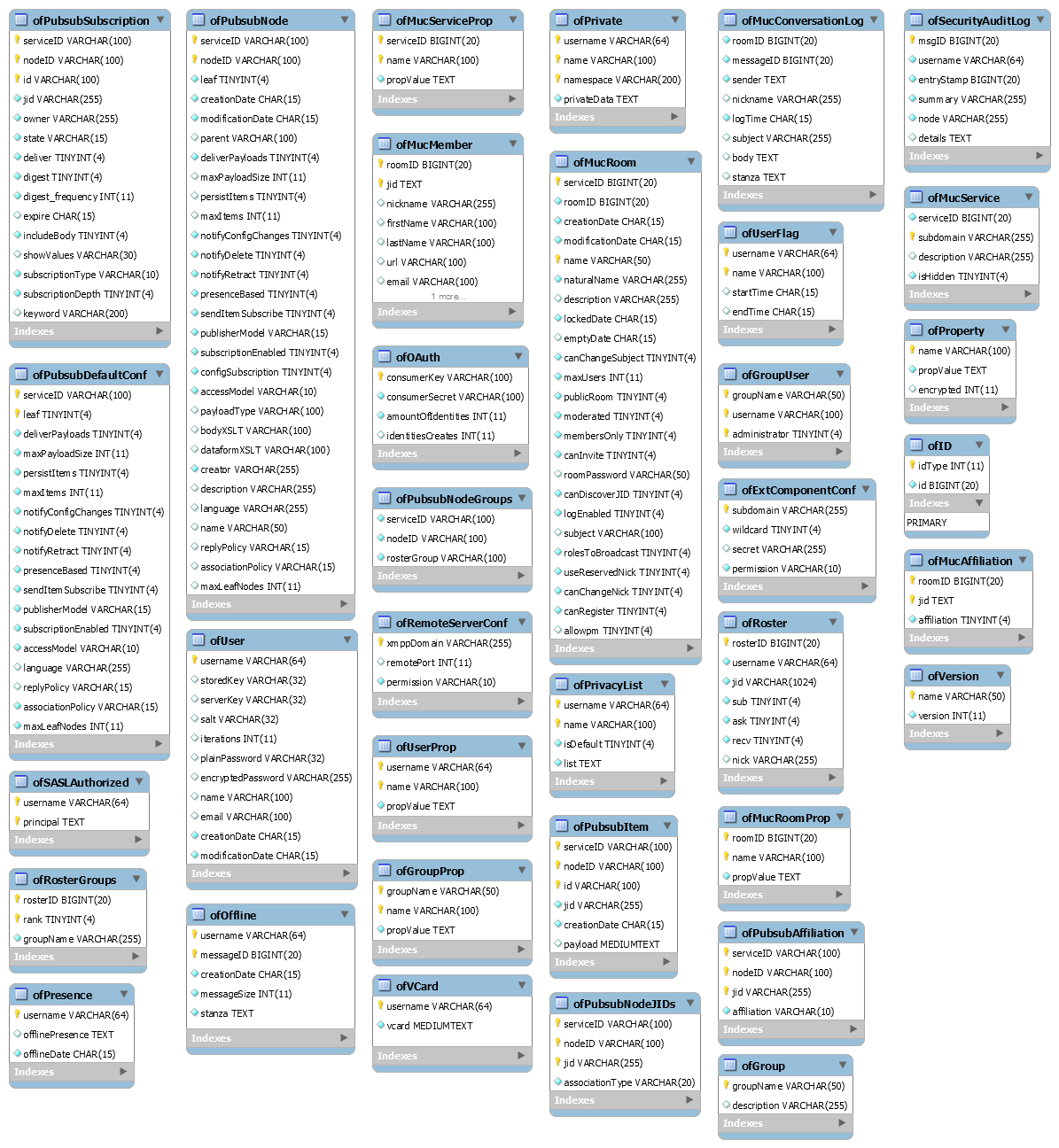
A partir de la «XEP-0348: Signing Forms» en conjunto con la «XEP-0077: Registration In-Band», se procedió a identificar las entidades que interactúan en el proceso de registro de nuevas cuentas en el servidor, las cuales se grafican mediante un diagrama de clases el cual está en la Imagen XXXX, remarcando las clases que serán intervenidas.



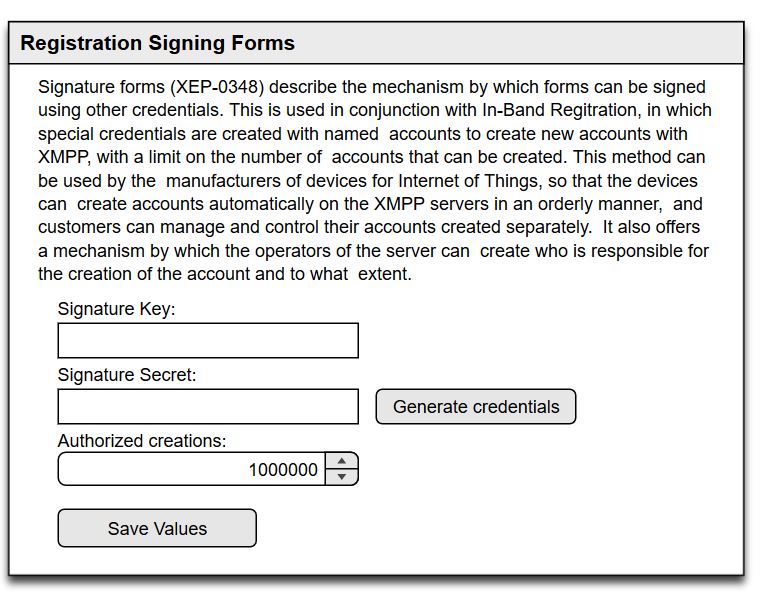
Ya identificadas las clases identificadas en el proceso, se realiza un diagrama de secuencia, que se puede apreciar en el diagrama XXXX



En cuanto al diseño de la base de datos, se explica mediante el siguiente diagrama de Entidad-Relación mostrado a continuación, en el grafico XXXXX

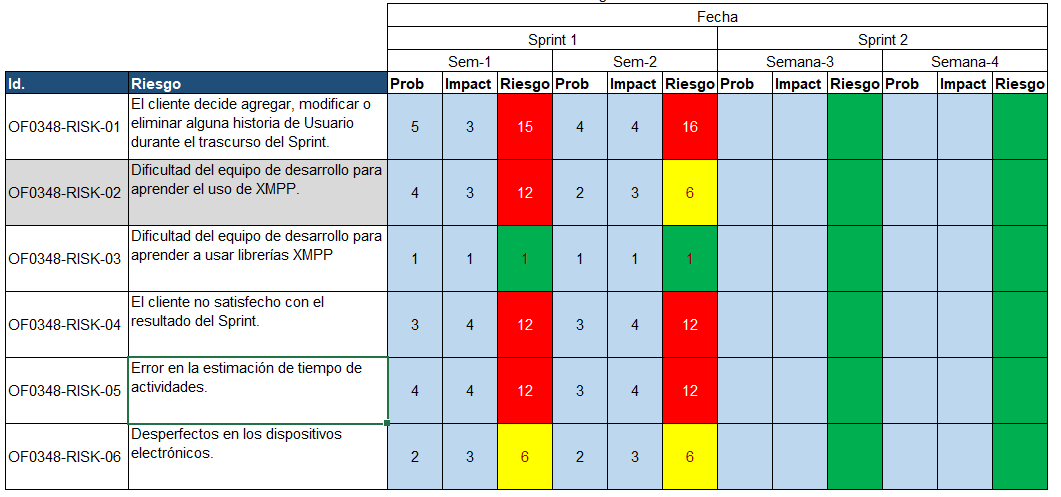


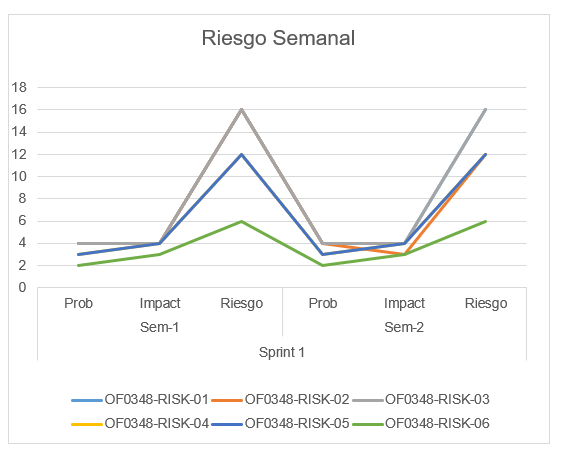
También se realizó un mockup de las modificaciones al front-end de «Openfire» para agregar la funcionalidad de agregar «Credenciales de Consumidor», a modo de prototipo no funcional, con el fin de que se puedan apreciar futuros cambios. El mockup se puede ver en el diagrama XXXXX

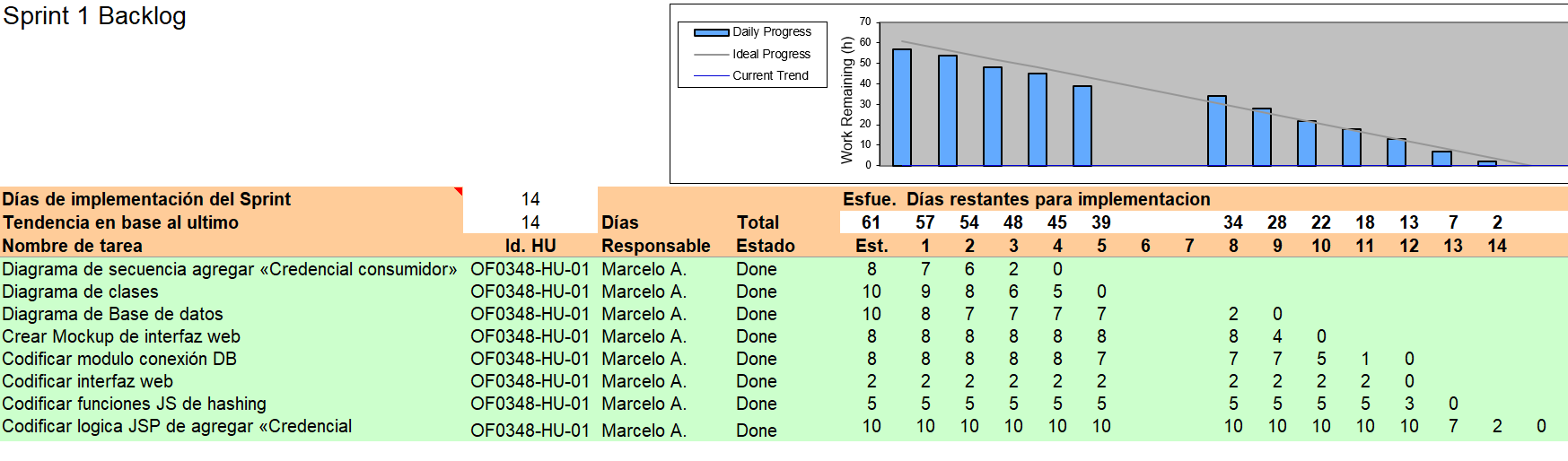


### A.1.1.3 Resultados

A continuación, en la Tabla XXXX, se detallan los riesgos identificados para el Sprint, y seguido el Diagrama XXXX con la evolución de estos.





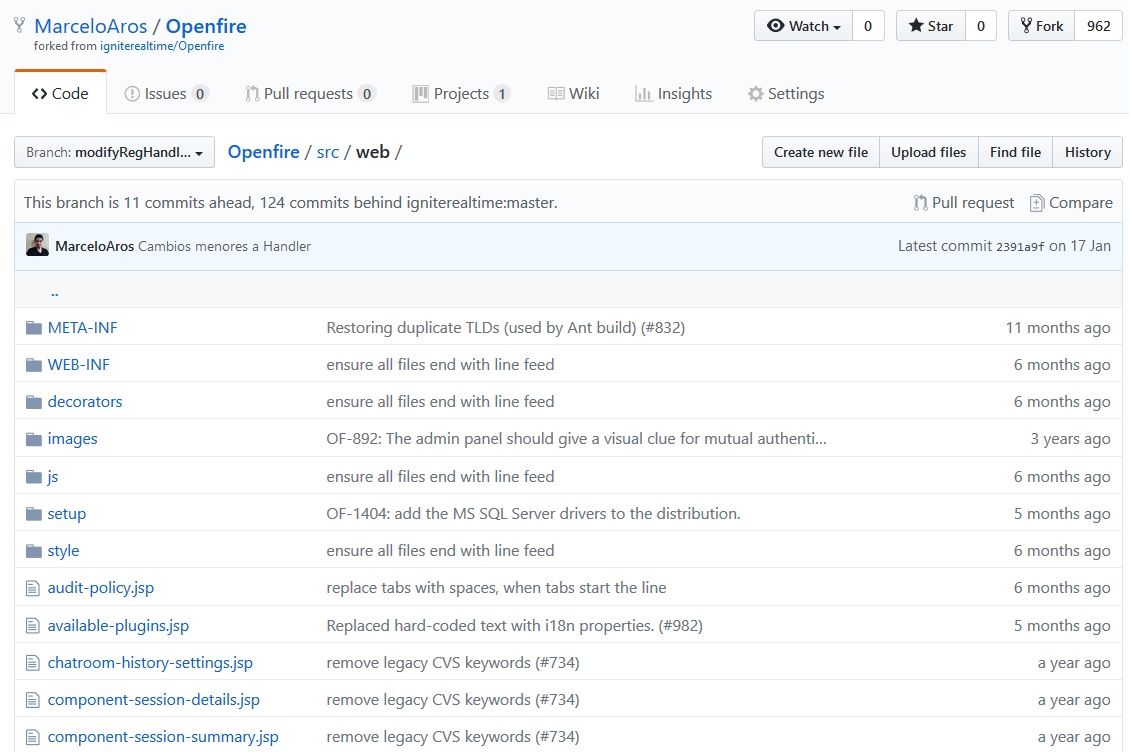


### A.1.1.4 Evidencia

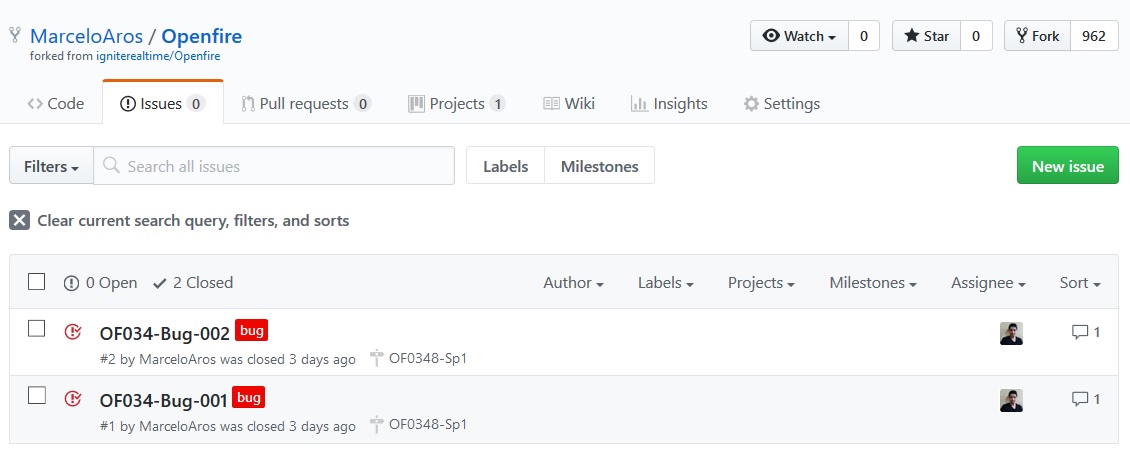
A continuación, se aprecia la evidencia en Github, tanto de la documentación del proyecto, como el proyecto mismo.

En el trascurso del desarrollo del Sprint 1 se generaron 2 incidencias, estas tienen relación con las validaciones. La primera obedece al fallo al validar una cantidad de identidades permitida para una «credencial de consumidor». La segunda al permitir ingreso de credenciales de consumidor, sin haber sido generadas con anterioridad.

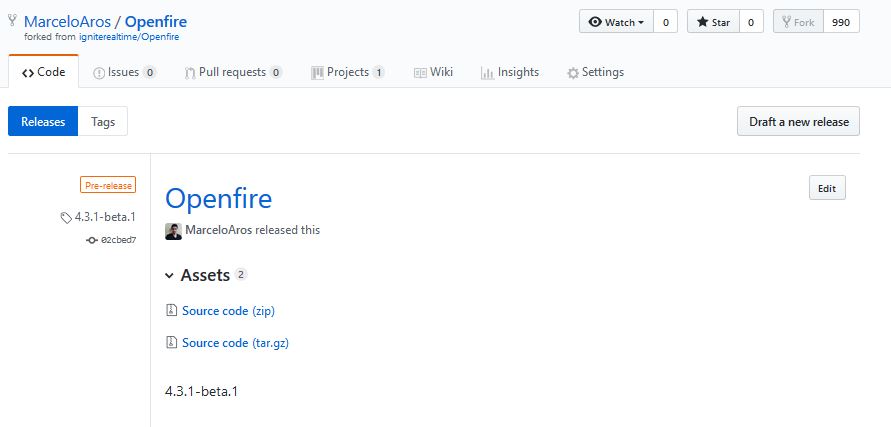




Los detalles del seguimiento de incidencias de sprint 1, se puede visualizar en la imagen XXXX a continuación.



Finalmente, se generó la versión preliminar 4.3.1-beta.1, la evidencia de esto, se puede apreciar en la imagen XXXXX



### A.1.1.5 Post-Mortem

**Problemas resueltos**

* El servidor «Openfire» es capaz de incorporar «Consumer Key», «Secret Key», cantidad de creación de identidades permitas, cantidad de creación de identidades utilizadas.

**Problemas futuros**

* Si bien las credenciales son agregadas correctamente en la base de datos del servidor «Openfire», mediante la consola de administración web, no es posible visualizar estos datos, ni llevar un registro y control sobre cuantas y que identidades ha creado un «Consumer Key». Esto se realiza en una de las historias de usuario asignadas al proyecto OF0348.

**Lecciones aprendidas**

* Con mejores estimaciones de tiempo y mejor manejo de los riesgos es mucho más factible cumplir con los plazos establecidos.

**Métricas**

* KLOC: Se finalizo este Sprint con 1.2 KLOC
* Cantidad de defectos: 2 defectos encontrados durante el desarrollo del sprint, todos solucionados, no quedó defecto pendiente de ser resuelto.
* Efectividad de pruebas: Se realizaron 3 pruebas de aceptación, las cuales pasaron.

## A.1.2 Sprint 2

### A.1.2.1 Planificación

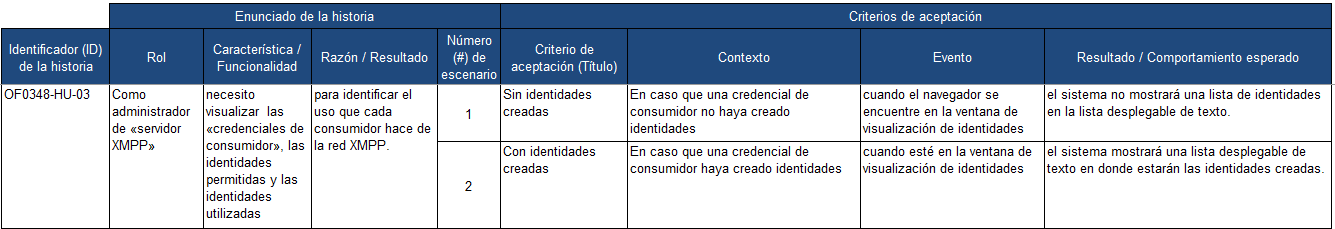
En el Sprint anterior se desarrollo el modulo de generación y almacenamiento de credenciales de consumidor con sus respectivas variables asociadas, es decir, cantidad de creación de identidades permitidas y cantidad de identidades actualmente creadas.

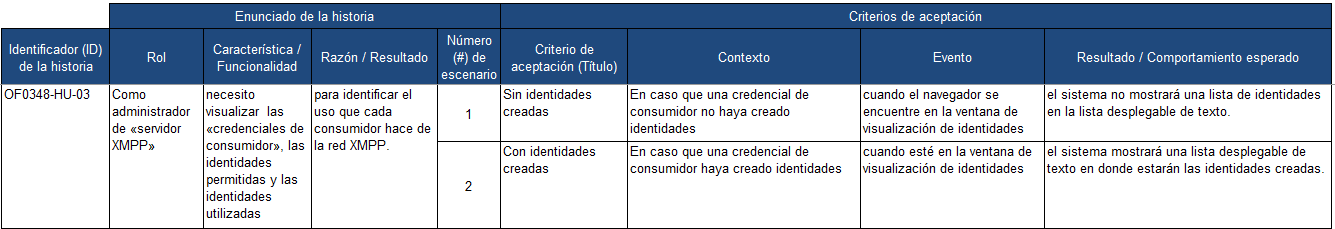
Siguiendo la planificación de carta Gantt entramos en plazo preestablecido al sprint 2.

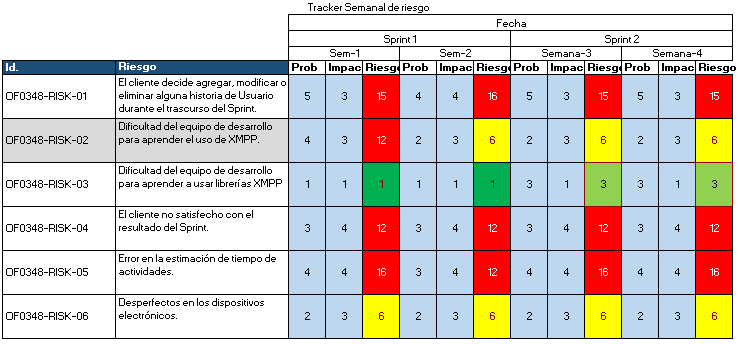
Durante el Sprint, se continuó añadiendo funcionalidades al servidor XMPP «Openfire», en particular el desarrollo de la historia de usuario OF0348-HU-03, la cual, hace uso de los datos almacenado en el desarrollo del sprint anterior, esto es, para finalmente visualizarlos de forma ordenada y coherente a lo almacenado, con la variable «oauth\_secret\_key» oculta. Se gatillaron los riegos OF0348-RISK-05, debido a la falta de experiencia con el desarrollo de «JavaServer Pages», por ende, se aplicó la medida de contingencia para dicho riesgo, la cual fue, inyectar más horas hombre.

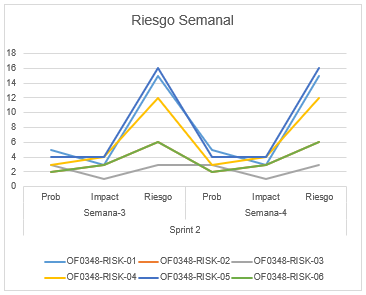


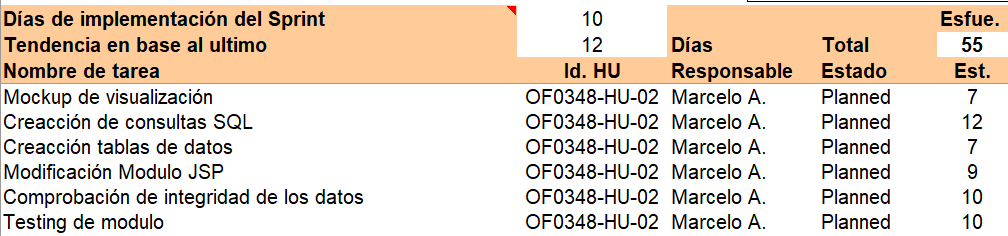








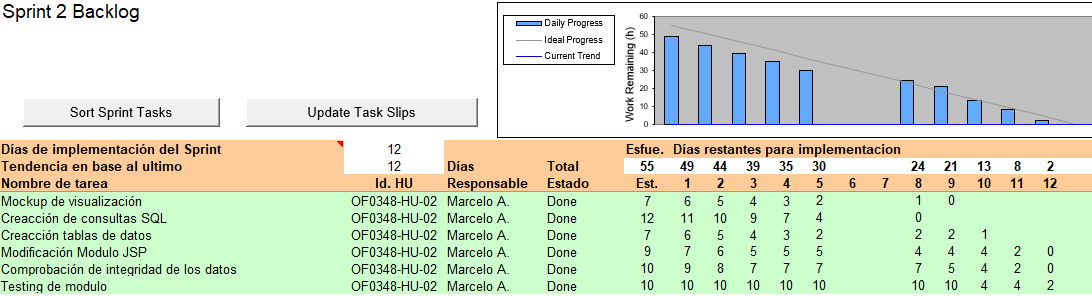




### A.1.2.2 Diseño

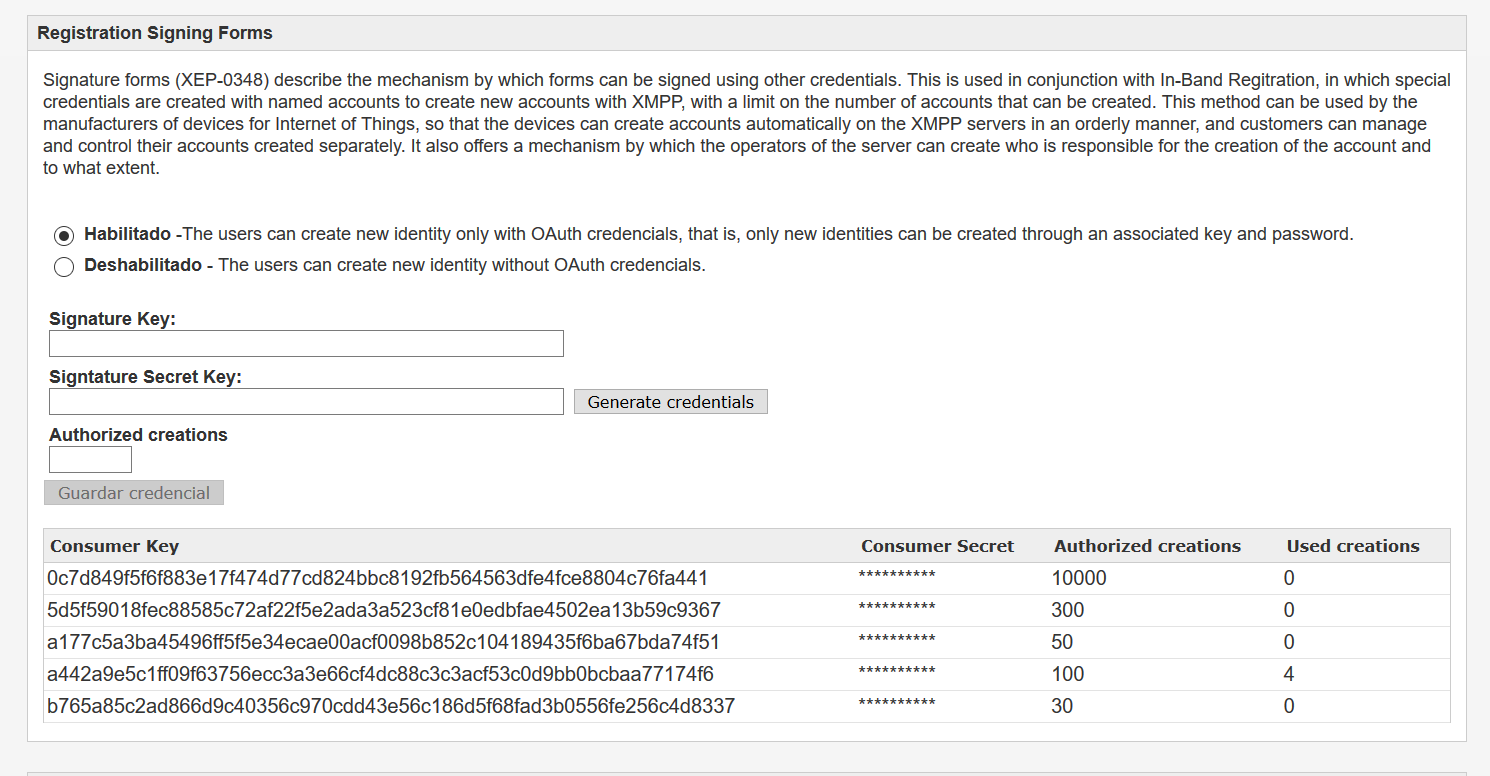
En el primer Sprint se logró identificar las clases involucradas en el desarrollo OP0348. En este Sprint se modificó la clase «IqRegisterHandler.class» y «reg-settings», como también se modificó los scripts iniciales de creación de las bases de datos.

### A.1.2.3 Resultados

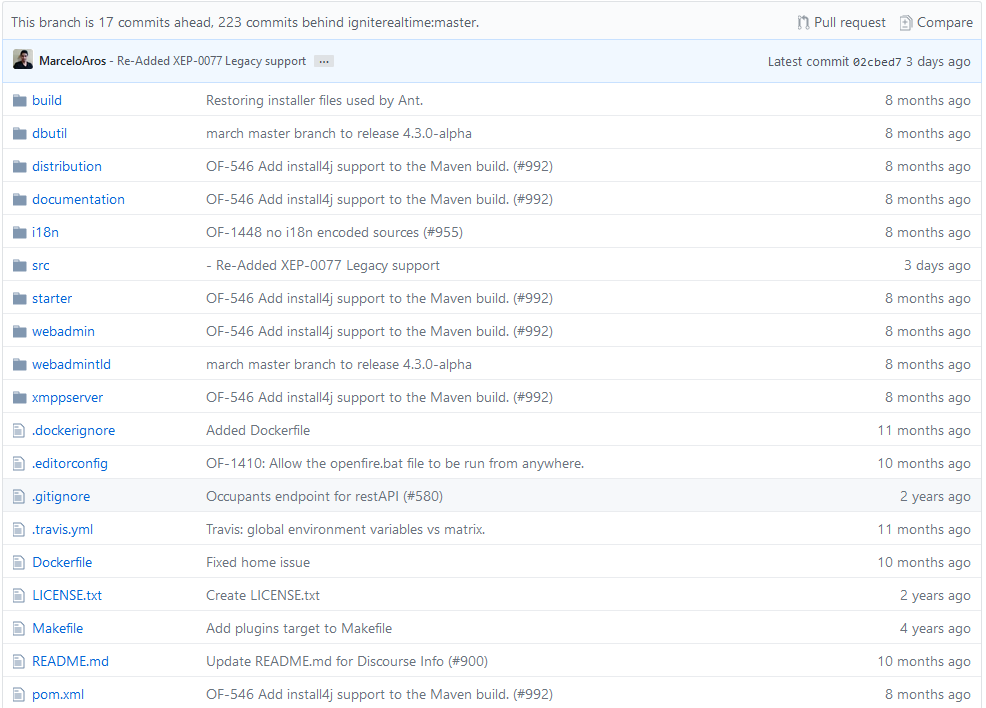


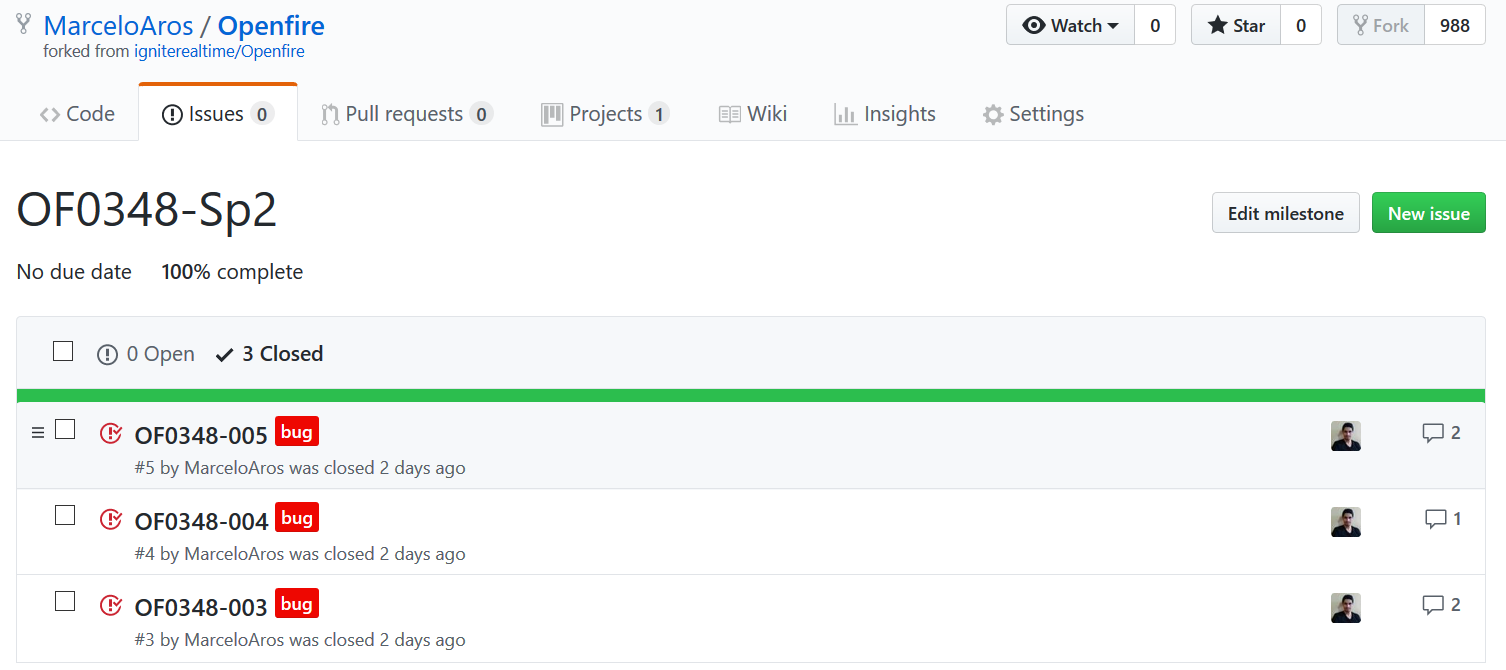
### A.1.2.4 Evidencia

El resultado del sprint se puede ver reflejado en los cambios en la consola de administración web, un pantallazo de estos cambios se puede apreciar en la imagen XXXX.

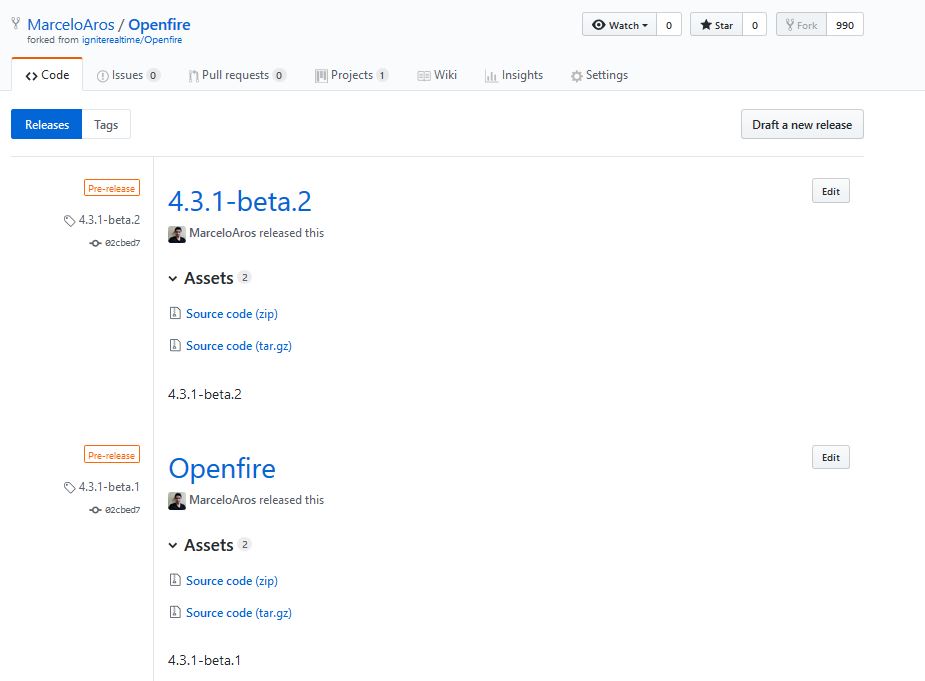


La evidencia del código fuente se puede revisar en Github con los todos los commits realizados a la fecha.





También se liberó una nueva versión preliminar de Openfire, con el tag de 4.3.1-beta.2



### A.1.2.5 Post-Mortem

**Problemas resueltos**

* El servidor XMPP «Openfire» es capaz de visualizar las credenciales de consumidor que posee registradas, de este modo, se sabe cuantas identidades potencialmente se podrían registrar en la red XMPP-IoT.

**Problemas futuros**

* Si bien, en este momento se pueden ingresar muchas credenciales de consumidor al ambiente del servidor Openfire, no estamos aportando un valor agregado, ya que falta el grueso de la implementación que es la lógica detrás del registro de nuevas identidades, la cual debiese que seleccionar el «Product Owner» en la próxima reunión «Sprint Planning».

**Lecciones aprendidas**

* Aún se continua con problemas de estimación de tiempo, en este caso, se supuso que, debido a la aparente simpleza de la HU, iba a ser un desarrollo simple, sin embargo, el equipo de desarrollo tuvo grandes inconvenientes con JavaServer Pages, ya que no se adelanto el trabajo con esta tecnología, pero se asume que es una de las consecuencias de trabajar con una metodología ágil.

**Métricas**

* KLOC: Se finalizo este Sprint con 1012 LOC.
* Cantidad de defectos: Se detectaron 3 bug en etapas tempranas de desarrollo.
* Efectividad en pruebas: Se realizaron dos pruebas de aceptación las cuales pasaron.

## A.1.3 Sprint 3

Luego de desarrollar la generación, almacenamiento y visualización de credenciales de consumidor.

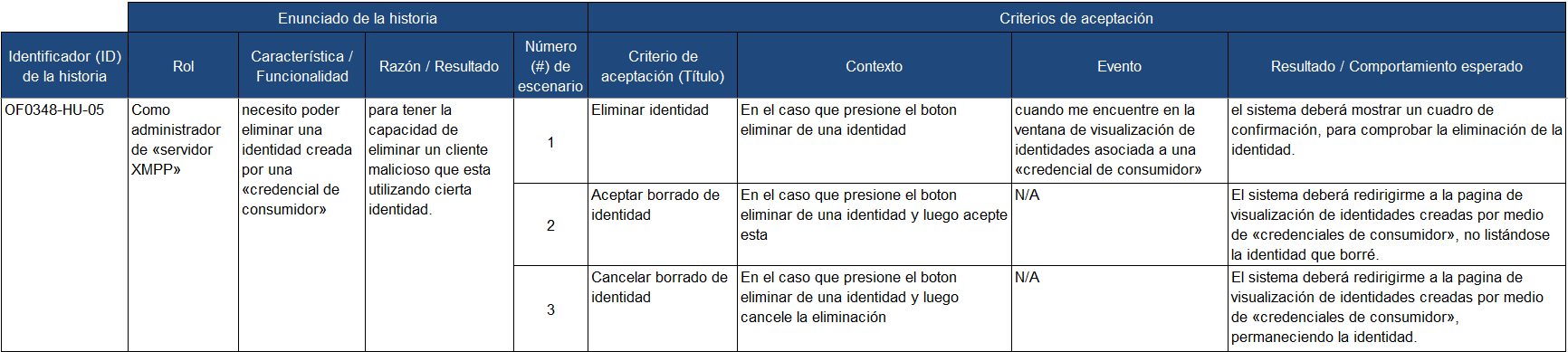
Durante este Sprint se procedió a dar inicio al desarrollo de la interacción lógica con clientes XMPP, es decir, el servidor Openfire crea conexiones a través de HTTPS para así enviar mediante «stanzas» XML, de acuerdo con lo establecido en el core de XMPP, datos de registro a los clientes.

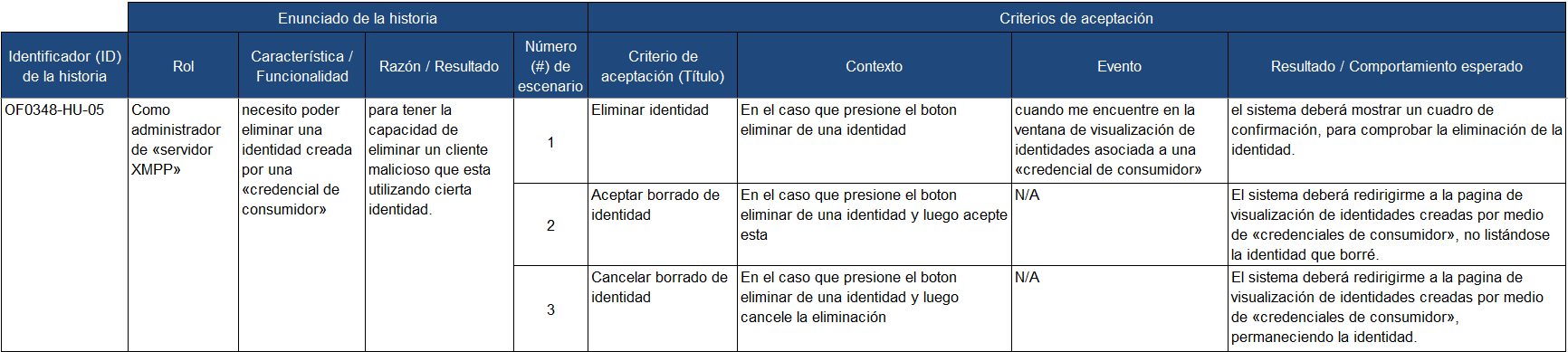
### A.1.3.1 Planificación

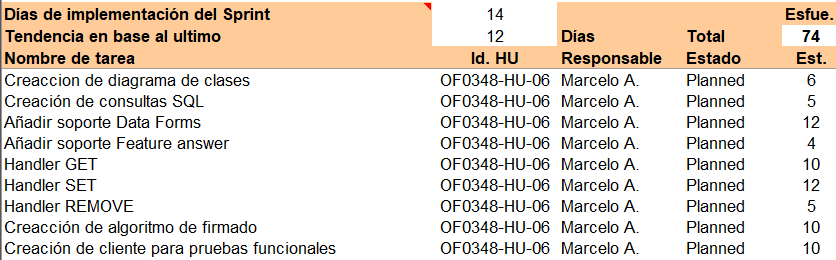
De acuerdo con la reunión de planificación «Sprint Planning» en la cual se seleccionó la OP0348-HU-06 para realizar con peso en números Fibonacci 89.











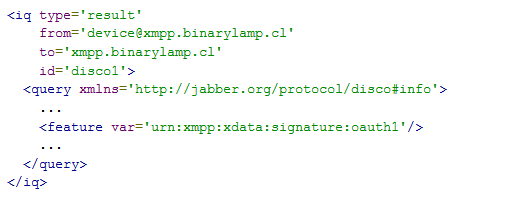
### A.1.3.2 Diseño

En este caso, todo el esfuerzo de desarrollo se centró en «IqRegisterHandler.class» que como el nombre dice es el encargado de capturar las solicitudes de creación de identidades de mediante el protocolo de extensión XMPP 0077. El gran problema al realizar el rediseño de esta clase fue su alto grado de complejidad, tanto así que es la segunda clase dentro de todas las que posee el servidor XMPP Openfire con mayor valor de complejidad ciclomática, con un , lo cual es sumamente alto, teniendo en cuenta para que una clase de muy alto riesgo el valor es de 50, y en este caso estamos, por poco, duplicando ese valor.

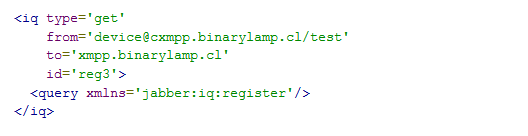
El servidor deberá preguntar por las «Features» por medio la consulta:



Y luego deberá capturar la respuesta del cliente, que estará conformada de acuerdo con la siguiente estructura:



Luego de que se valida que el cliente, tanto como el servidor soportan el XEP-0348, el cliente envía la «stanza» en la siguiente imagen:



Luego el host responde con la «stanza» con el formulario de registro en donde le solicita únicamente que el cliente complete cuatro valores, «username», «password», «oauth\_consumer\_key», y «oauth\_signature». También deja una serie de valores como opcionales, por ejemplo: «email», «first», «last», y el host envía envía «FORM\_TYPE», «oauth\_version», «oauth\_signature\_method», «oauth\_token», «oauth\_token\_secret», «oauth\_nonce» y «oauth\_timestamp». La «stanza» se puede ver a continuación:



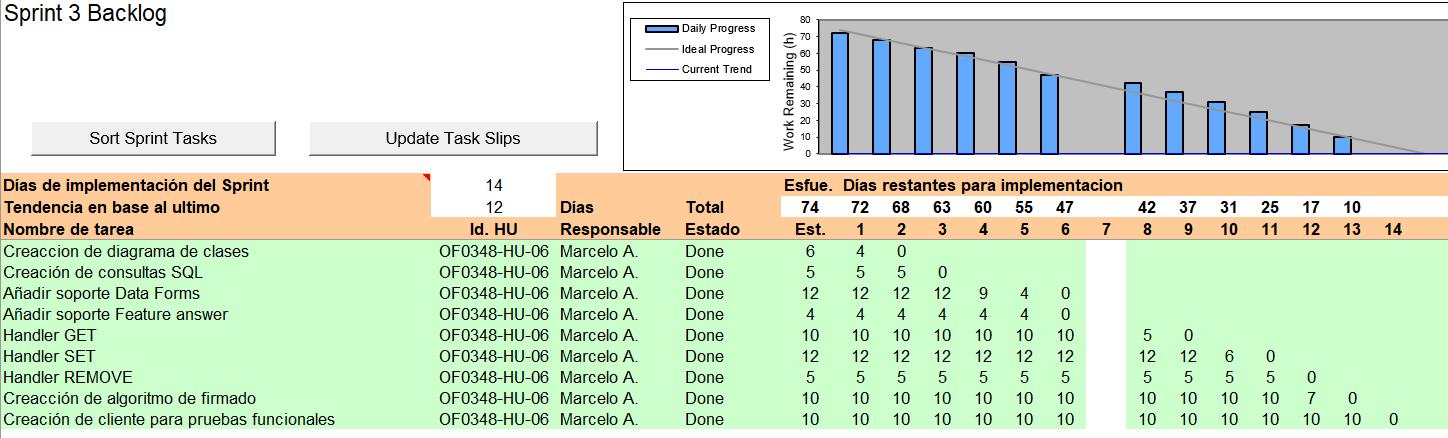
Luego el cliente recibe esta respuesta del servidor para luego comenzar la generación de «oauth\_signature» dependiendo del método firma elegido, será:

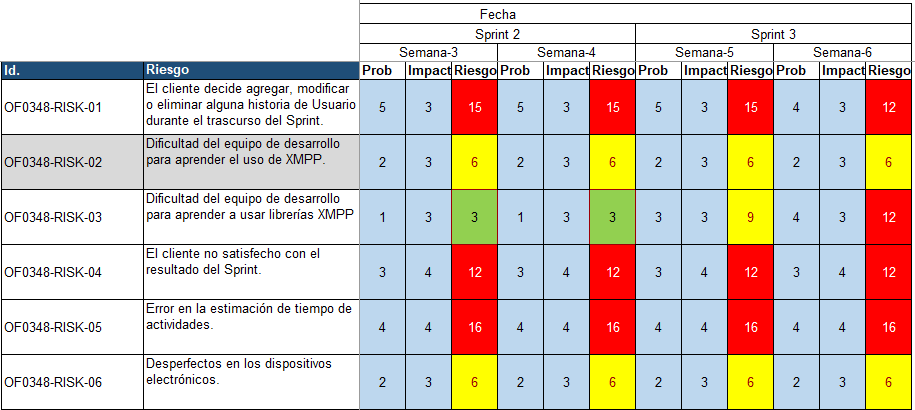
* HMAC-SHA1
* HMAC-SHA256
* RSA-SHA1
* PLAINTEXT

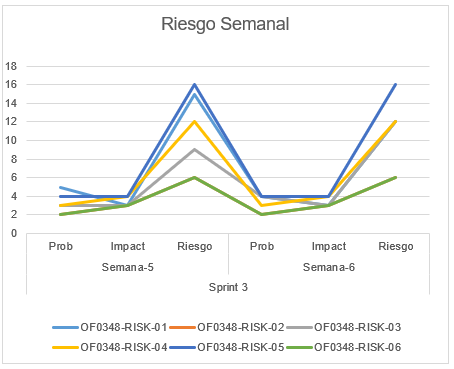
Por seguridad, se estableció como método de firmado único el HMAC-SHA256.

### A.1.3.3 Resultados

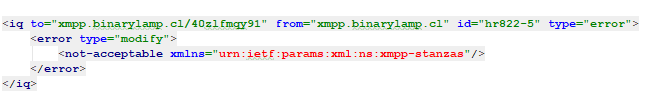
Luego de seguir de metodología de proyecto junto la de desarrollo, tenemos como la asignación de horas a cada tarea de la historia de usuario seleccionada para el desarrollo en este Sprint. Se puede apreciar el resultado en los gráficos e imagines a continuación:





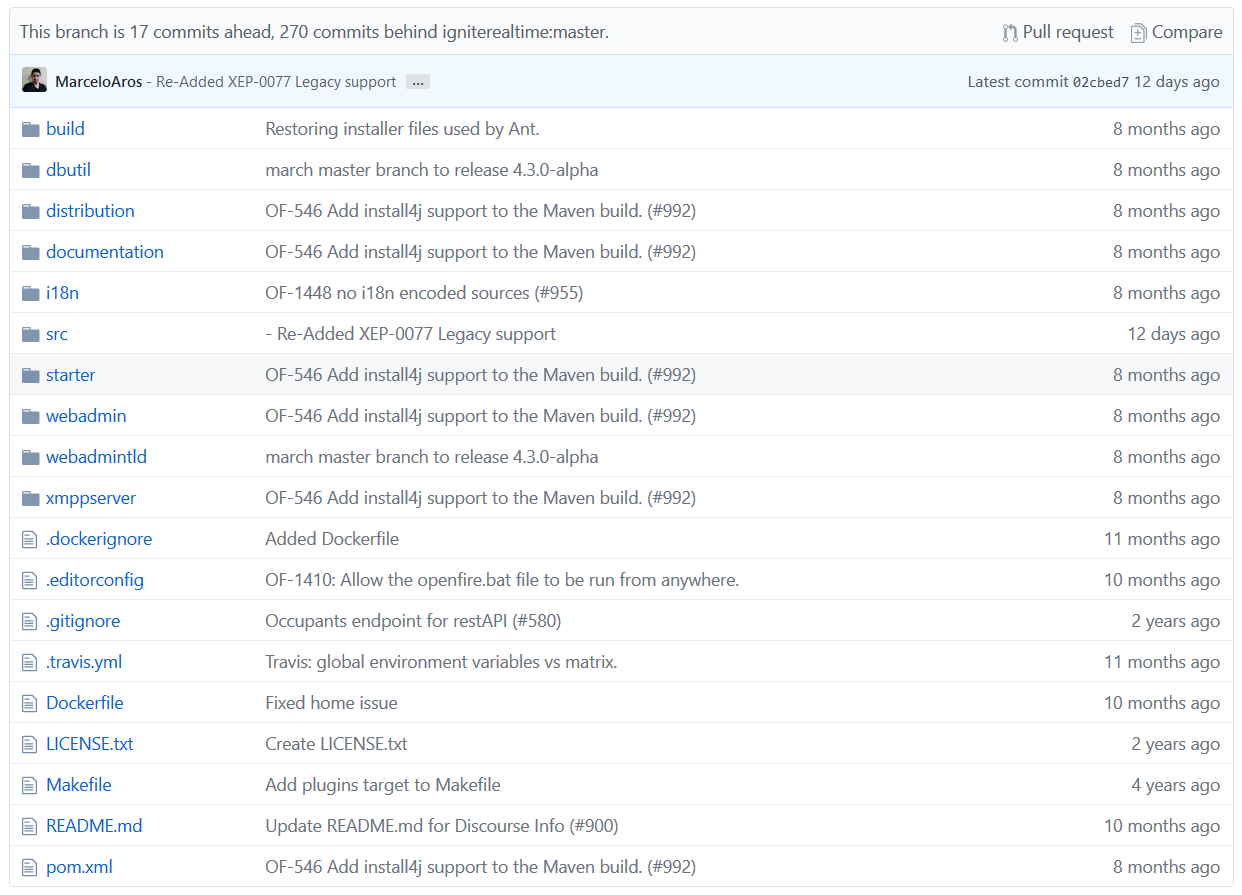




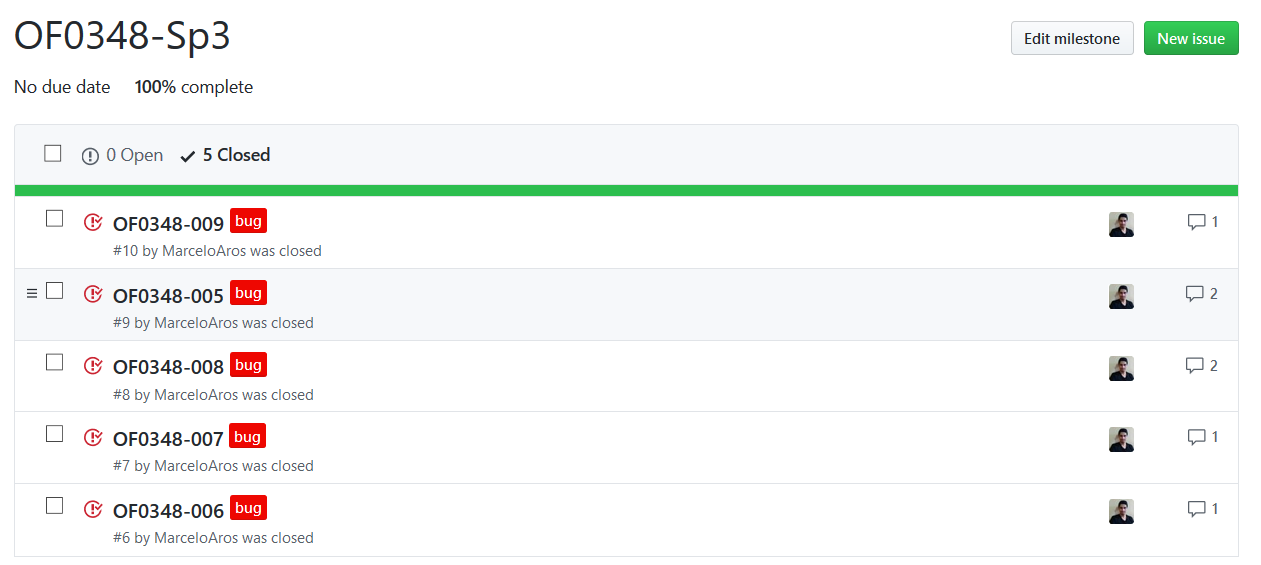


### A.1.3.4 Evidencia

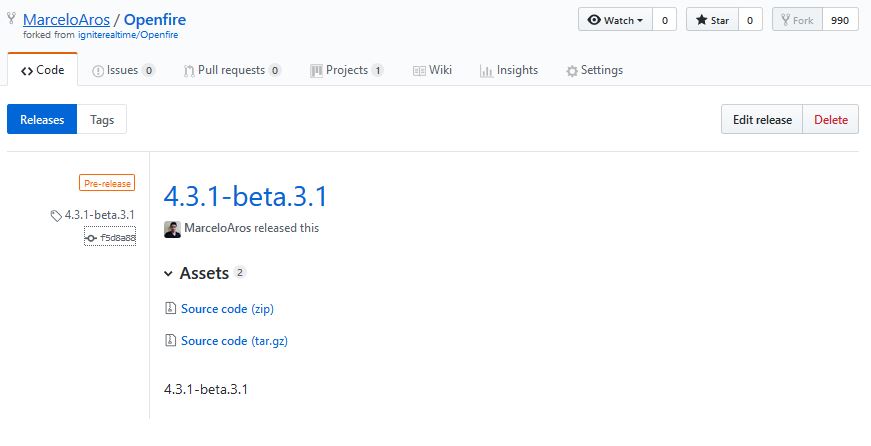
El código fuente fue almacenado en el repositorio «Openfire» de Github, en este lugar se realizaron los commits necesario para este fin, los detales se pueden visualizar en XXXXXX.



De igual modo, en Github se realizo el seguimiento de incidencias, los detalles se pueden apreciar en la imagen XXXX.



También al final del sprint se generó el reléase del producto preliminar de software con el título 4.3.1.-beta.3.1. la evidencia de esto, se puede ver en la imagen XXX, a continuación.



### A.1.3.5 Post-Mortem

**Problemas resueltos**

* Brindamos la solución al problema central del proyecto, el cual consistía en que el servidor XMPP «Openfire» pudiese responder a las solicitudes de registro de identidades mediante el uso del XEP-0348. De igual manera se brindó solución a los bugs asociados.

**Problemas futuros**

* Se ha completado el desarrollo asociado al servidor «Openfire». Pero aún queda el desarrollo correspondiente a librería «SMACK», el consiste en manera similar a la codificación de Servidor XMPP, es decir, agregar la funcionalidad de creación de identidades mediante la especificación XEP-0348.
* También tenemos las historias de usuario OF0348-HU-02 también la OF0348-HU-04 y la OF0348-HU-05.

**Lecciones aprendidas**

* Dentro de las consideraciones que debemos tener, esta la de evaluar mejor la gestión de riesgos al comenzar el proceso de un Sprint, pero investigando un poco más allá, se encontró que SCUM no determina de buena manera quien es el encargado de la gestión de riesgo, debido a que no puede gestionar ciertos riesgos intrínsecos al desarrollo de la metodología del proyecto, que en este caso es SCUM. Por ende, como los académicos no tienen consenso en este punto, se le asigno esta labor al director de proyecto.

**Métricas**

* KLOC: Se finalizo este Sprint con 2185 LOC
* Complejidad ciclomatica promedio de las clases generadas es de:
  + ev(G) = 101
  + antes de las modificaciones era de ev(G) = 56
* Cantidad de defectos: Se detectaron 5 bugs en etapas tempranas de desarrollo.
* Efectividad en las pruebas: Se realizaron 3 pruebas de aceptación, de las cuales inicialmente pasaron 2, luego de realizar las correcciones necesarias, se logró que pasaron los 3 caso de prueba de manera exitosa.
* Densidad de defectos = 2.29

## A.1.4 Sprint 4

Ya habiendo finalizado el desarrollo de las historias de usuario MUST correspondientes al servidor. Comenzamos la sección del proyecto referente a las historias de usuario MUST para la librería «SMACK», la cual consiste en adicionar la característica de registro de identidades en el servidor descrita en la XEP-0348. La historia de usuario seleccionada para este sprint es, como se puede ver la imagen en la sección A.1.5.1, la OF0348-HU-07. Si bien una cosa es importante reconocer a estas alturas del avance del proyecto, que, en las etapas tempranas de este, se tuvo un desliz en la selección del prefijo del proyecto, el cual es «OF0348» referenciando a «Openfire» con las letras «OF», sin embargo, el proyecto no solo contempla el desarrollo de características para el servidor «Openfire», sino que también para «SMACK», la librería desarrolla por Igniterealtime.

Frente a este inconveniente, se decidió mantener el prefijo, ya que modificarlo conllevaba un esfuerzo desproporcionado en respecto al impacto de este en el proyecto.

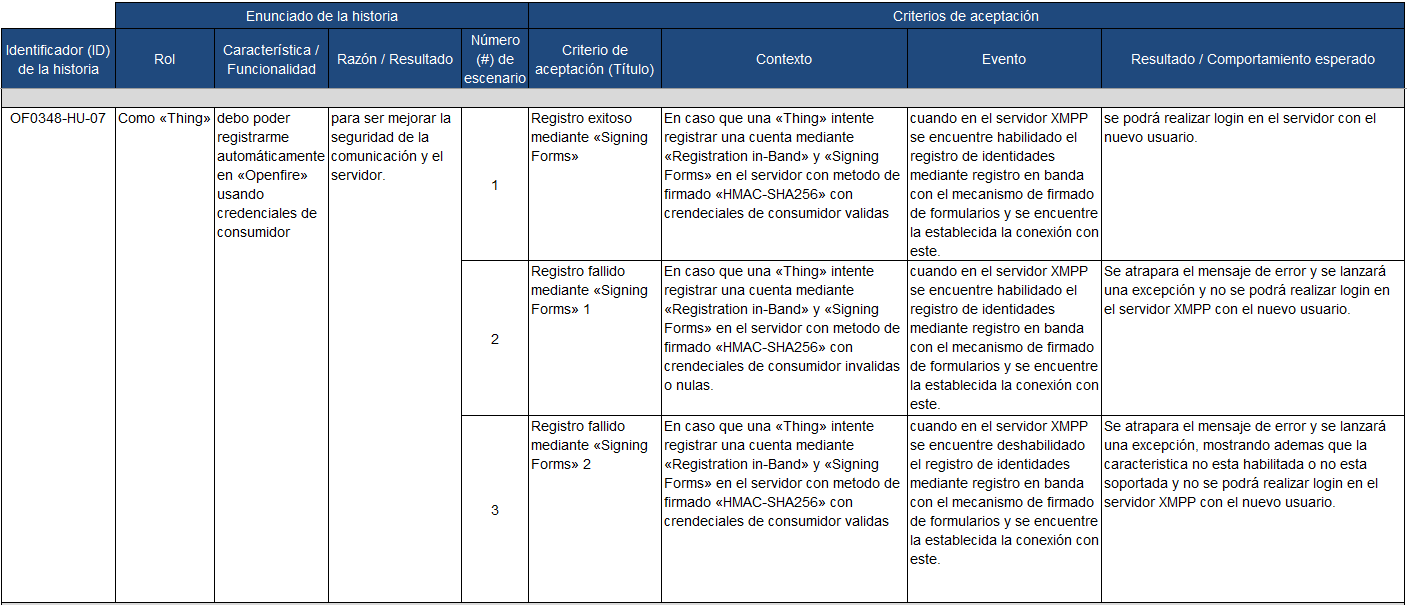
En otro punto, durante este desarrollo se debió tener sumo cuidado con la complejidad del código, ya que la librería puede ser usada por dispositivos con muy bajo poder de computo.

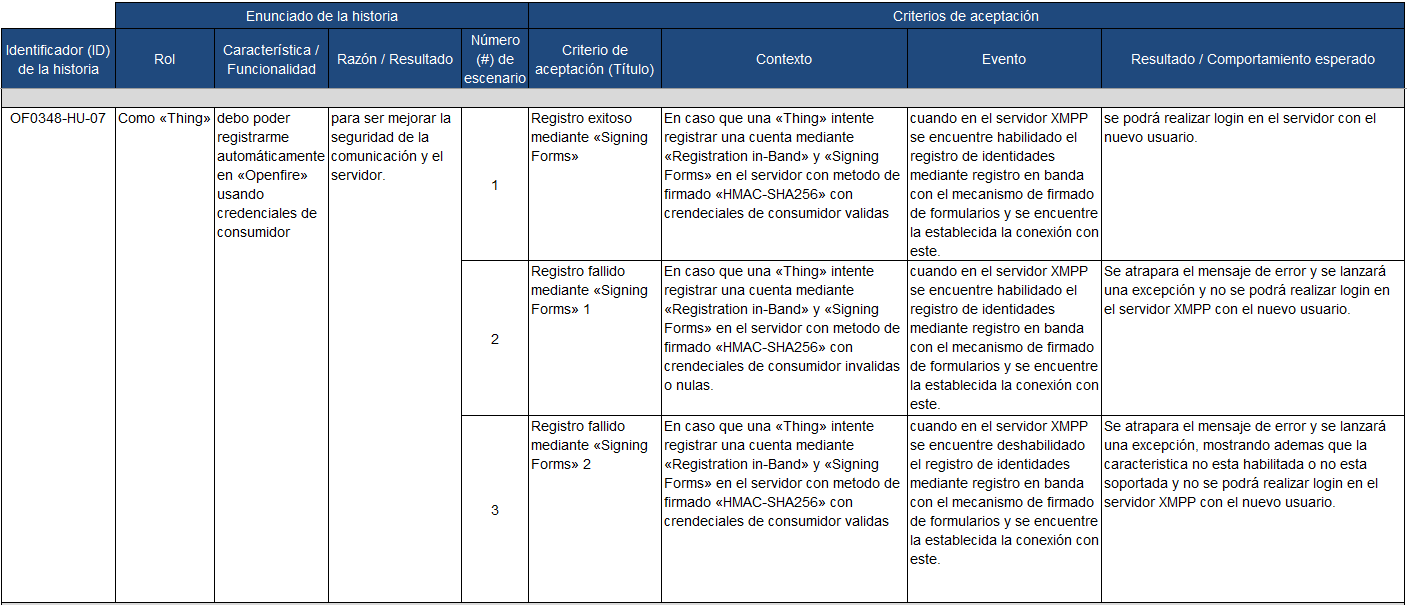
### A.1.4.1 Planificación

Como mencionamos anteriormente, se luego de Sprint Plan Ning se asignó la historia de usuario OF0348-HU-07 para el desamarrarlo de este Sprint.









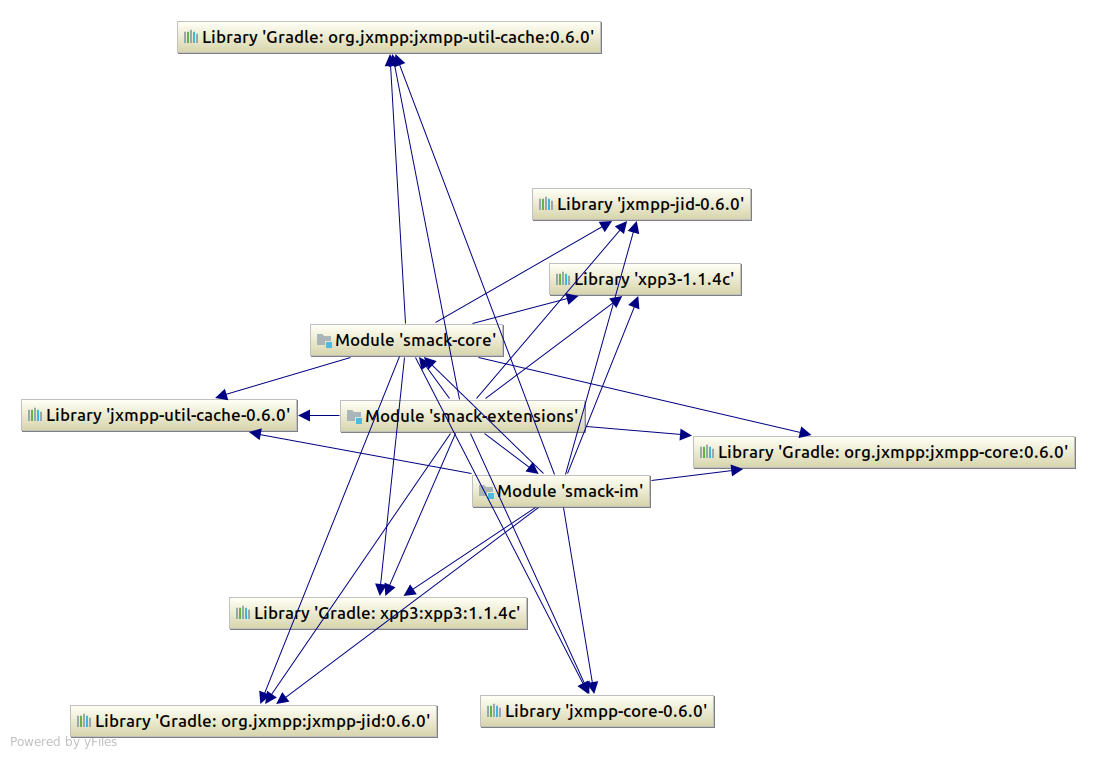


### A.1.4.2 Diseño

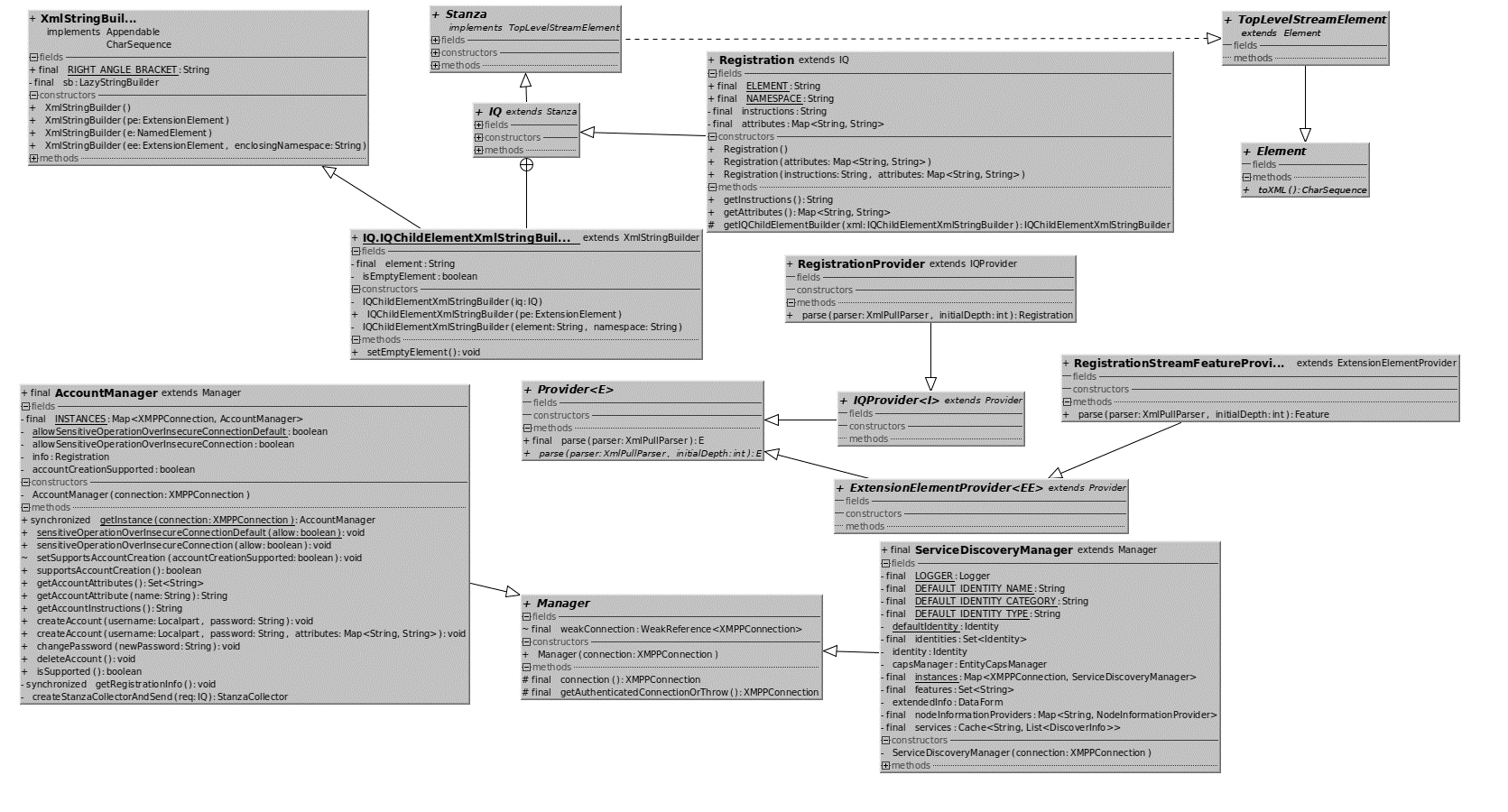
Del mismo modo como se inicio el diseño del servidor Openfire, se hizo con la librería «SMACK», es decir, un análisis de las entidades que interactúan en el proceso de registro en banda de nuestras identidades mediante del XEP-0077. El repositorio de SMACK, contiene los proyectos en la imagen XXXX que el subrayado, es decir «smack-extension» es el que se modificará.



El diagrama de paquetes del proyecto «smack-extension» se aprecia en la figura XXXX



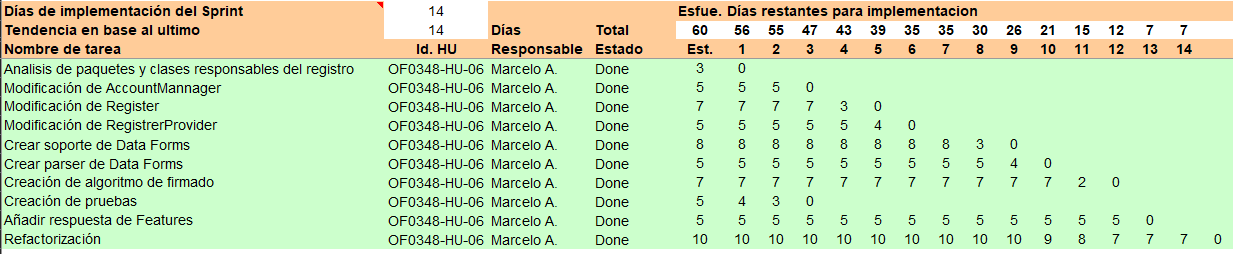
Para los efectos del desarrollo, se identifico las clases asociadas, las cuales se pueden ver en la imagen XXXX que esta a continuación

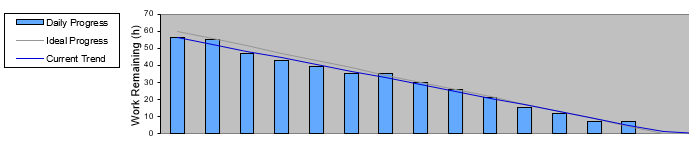


Luego de identificar las clases y como interactúan entre ellas, se comenzó la codificación de las taras del Sprint.

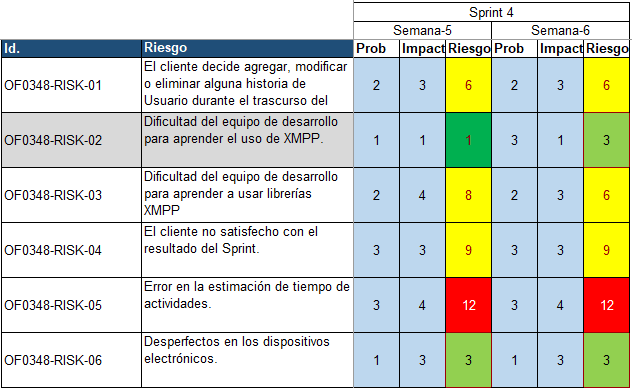
### A.1.4.3 Resultados

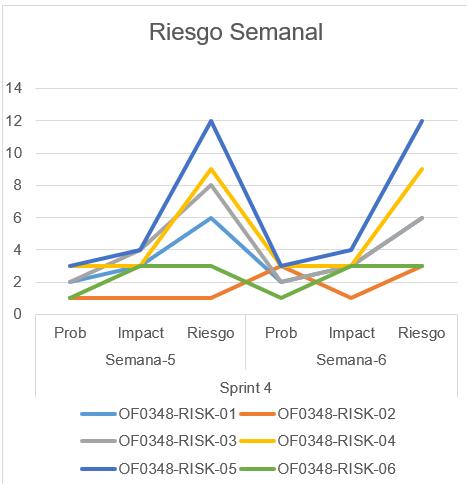
Cuando hablamos de los resultados del sprint 4, hay que poner especial énfasis a arquetipo del «XMLPullParser», ya que fue el que genero mayor inconveniente durante el desarrollo, por su estructura, conlleva una complejidad extra asociada a comprender su funcionamiento. Lo cual genero un pequeño retraso.





Por el lado de la gestión de los riesgos, durante este Sprint, se redujo su probabilidad tanto como su impacto del riesgo «OF0348-RISK-01», «OF0348-RISK-02» y «OF0348-RISK-06». Debido a que principalmente el sprint necesariamente no involucra el para el servidor Openfire, sin embargo, si algun bug tiene relación con lo desarrollado en codigo del servidor, se tendría que refactorizar ese codigo.



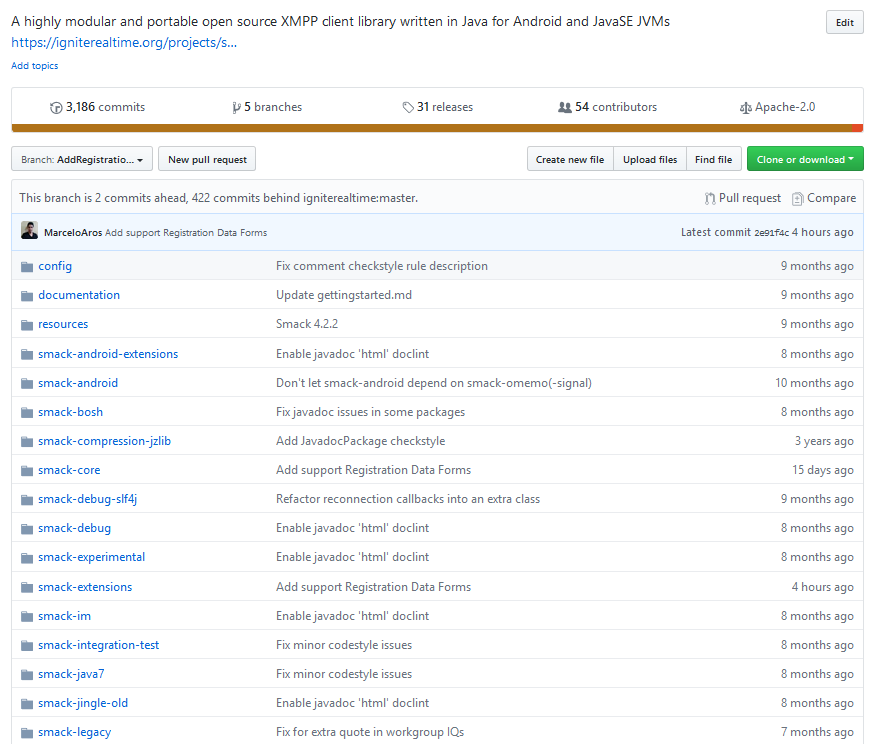


Luego, de finalizado el Sprint, las clases involucradas quedaron como muestra la figura XXXX, vemos claramente la adición de clases, como «DataFroms» con su correspondiente «FormField»

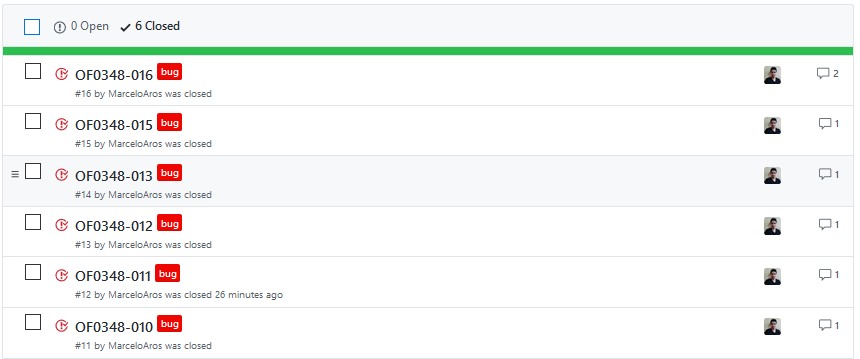


### A.1.4.4 Evidencia

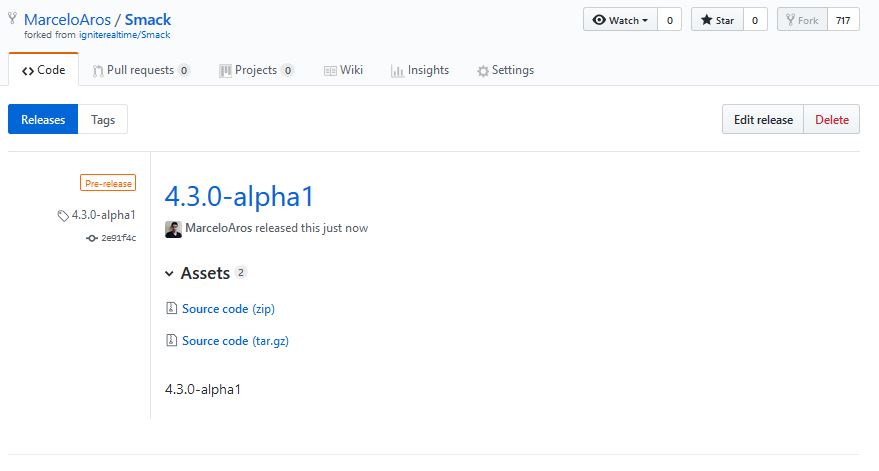
El código fuente se puede encontrar en el repositorio asignado para SMACK en el contexto del presente proyecto, las incidencias se pueden encontrar en el apartado de «Issues» de repositorio GitHub de Openfire.



En cuanto a la evidencia, de las incidencias detectadas, se pueden apreciar en la imagen XXXX



También se liberó nuevo release de la versión preliminar de la librería SMACK, con el tag de 4.3.0-alpha.1



### A.1.4.5 Post-Mortem

**Problemas Resueltos**

Durante el transcurso del presente sprint, se dio solución a la imperante necesidad del proyecto de contar con un cliente con la capacidad de usar el mecanismo registro en banda en conjunto con Formulario de Firmado «XEP-0348», todo esto para poder realizar las correspondientes pruebas de sistemas. También se dio solución a los bugs generados durante el proceso de desarrollo. De este modo, debido a al trabajo hasta la fecha se ha logrado generar un producto mínimo viable, ya que la librería SMACK cuenta con todas las adiciones de características descritas en la «XEP-0348».

**Problemas futuros**

* Registro de cuentas de usuarios y no cuentas de bots. Debido a que los tiempos de respuesta son reducidos, no hay tiempo para verificar si la identidad ha sido creada o no. Por lo tanto, se debe que proveer de una interfaz suficientemente flexible para este fin.

**Lecciones aprendidas**

* Se ha completado el desarrollo asociado a la librería SMACK. Reduciendo bastante la brecha en donde se gatillan los riegos.
* En lo que nos resta de estimación de tiempo, queda bastante que seguir aprendiendo. En cada iteración se ha mejorado.

**Métricas**

* KLOC: Se finalizo el sprint con 1209 LOC.
* Complejidad ciclomatica promedio de las clases generadas es de:
  + ev(G) = 14.6
* Cantidad de defectos: Se detectaron 7 bugs durante el transcurso del proceso de desarrollo.
* Efectividad de pruebas: Se realizaron 3 pruebas de aceptación, de las cuales inicialmente pasaron 2, luego de realizar la depuración con la correspondiente refactorización, se logró aprobar los 3 casos de pruebas.
* Densidad de defectos = 5.78

## A.1.5 Sprint 5

Debido a la imperante necesidad de testear el sistema en su conjunto, y depurar los errores existentes, que deben existir, porque no se ha testeado completamente, el servidor ni el cliente. Debido a que no hay clientes ni servidores que posean los mecanismos aquí desarrollados. Por ende, para evitar fallas sistémicas, y tener el producto solicitado por el Product Owner.

### A.1.5.1 Planificación

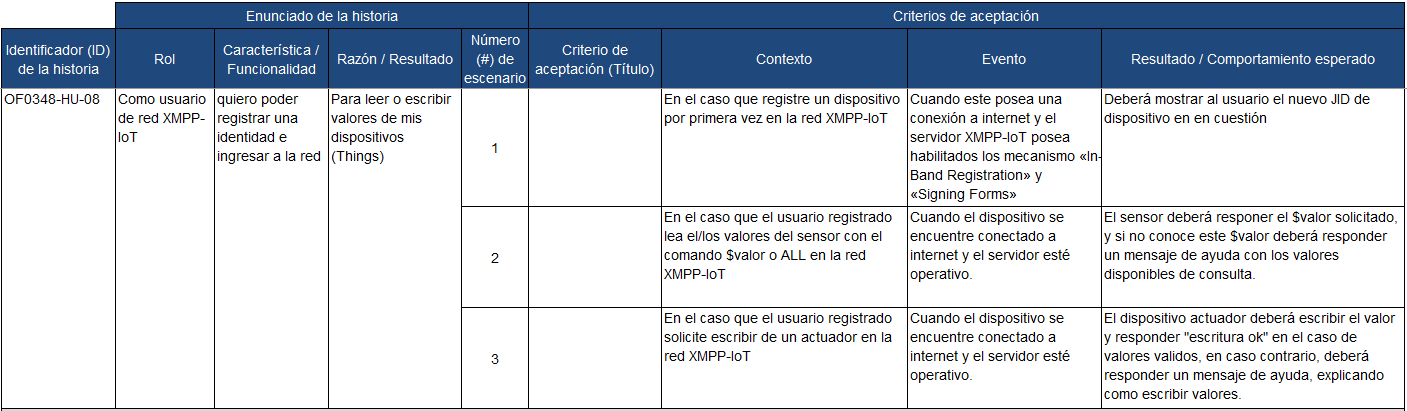
Ya ubicándonos en transcurso del ultimo sprint de proyecto, ya sea han desarrollado todas las historias de usuario MUST asociadas al servidor Openfire y la librería SMACK.

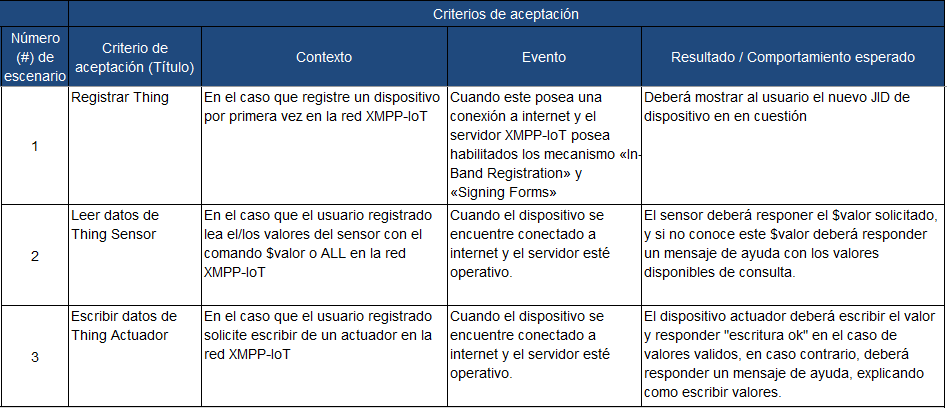
De acuerdo con la planificación del proyecto, se realizó la reunión Sprint Planning, en la cual, se asignó para el desarrollo de este sprint la historia de usuario OF0348-HU-08, como se puede apreciar en la tabla XXXX y XXXX





La historia de usario, se centra en la necesidad del usuario final del sistema, el cual, requiere obtener los valores leídos de sus sensores en este caso, temperatura y porcentaje de humedad, como también, establecer valores de sus actuadores, que en esta situación será un led, de manera similar funcionaría para una puerta, ventilación, calefacción, etc.  
Los detalles de la historia de usuario se pueden ver las tablas XXX y XXXX.



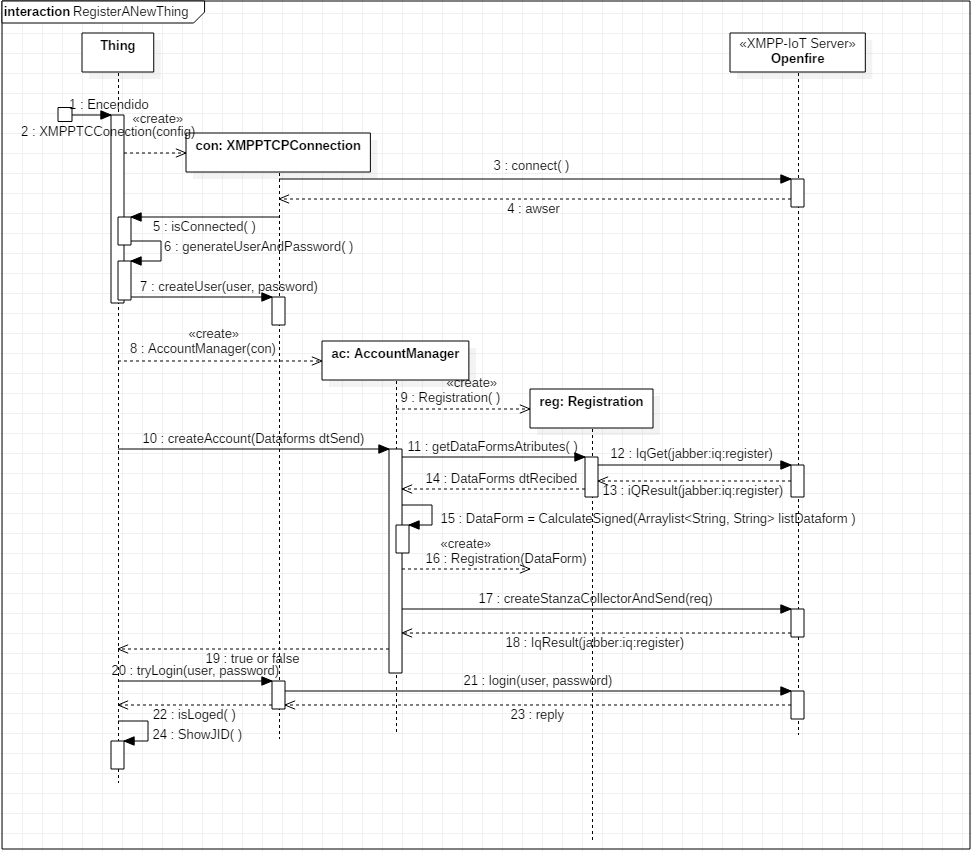


La historia de usuario seleccionada en la reunión sprint planning, se desgloso en la taras que se muestran en la tabla XXXX



### A.1.5.2 Diseño

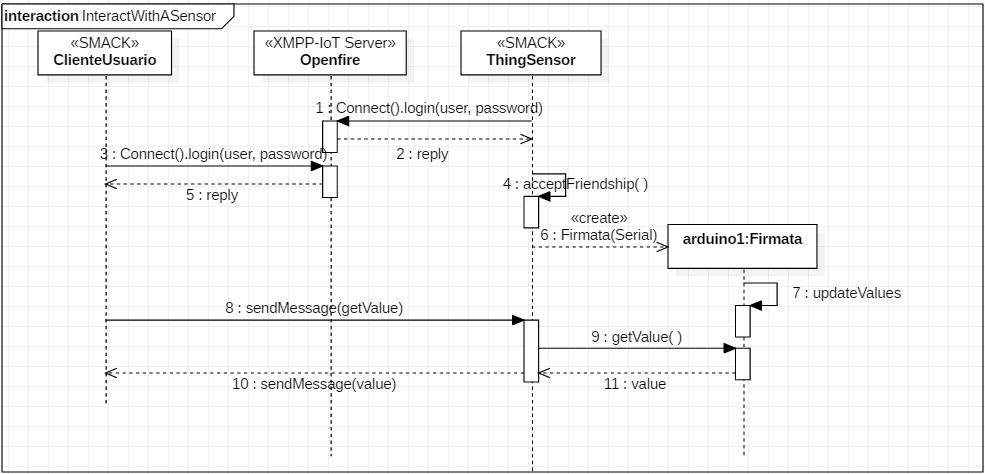
En este caso se utilizó lo desarrollado en la librería SMACK y el servidor Openfire, los cuales implementan los mecanismos «In-band Registration» en conjunto con «Signing Forms» para el registro de registro de identidades, debemos partid del supuesto que el vendedor de dispositivos de internet de las cosas, agregará las credenciales de consumidor al dispositivo, para que este realice el registro automatizado en el momento tenga una conexión a internet, para que el dispositivo muestre por alguna salida el nuevo «JID» asociado al dispositivo. En la imagen XXX se muestra el diagrama de secuencia de este caso.

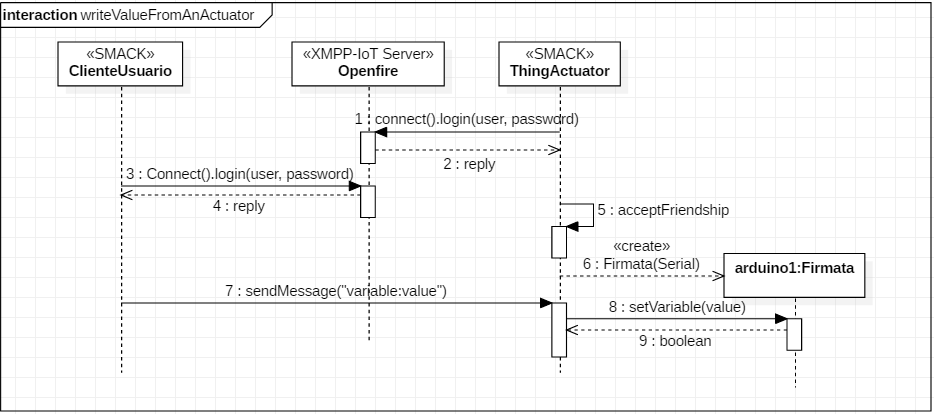


Por otro lado, tenemos el caso de cuando el usuario final, ya conectó el dispositivo a la red, además lo tiene emparejado con su «JID» quiere acceder a las lecturas de los sensores de este, en el caso que sea un dispositivo que tenga el funcionamiento de sensorizar variables, por ejemplo, ambientales.

En este caso para el sensor se han seleccionado las siguientes variables:

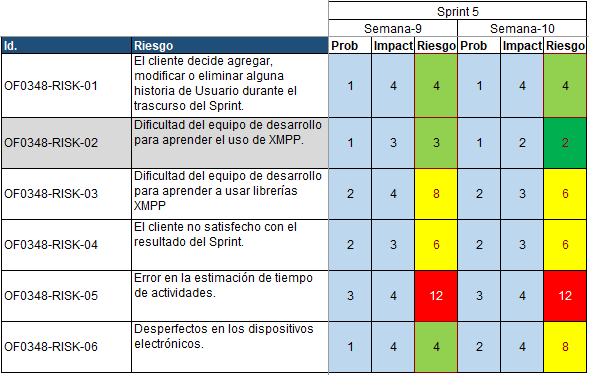
* Temperatura
  + Se medirá en grados centígrados, por ejemplo: 22° C.
* Humedad
  + Se medirá en porcentaje de humedad ambiental, por ejemplo 10% H.

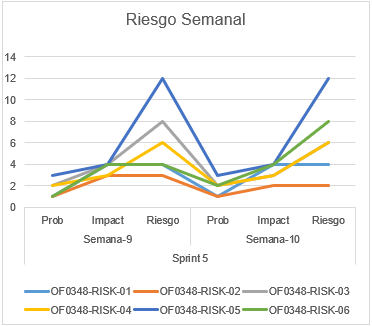




### A.1.5.3 Resultados

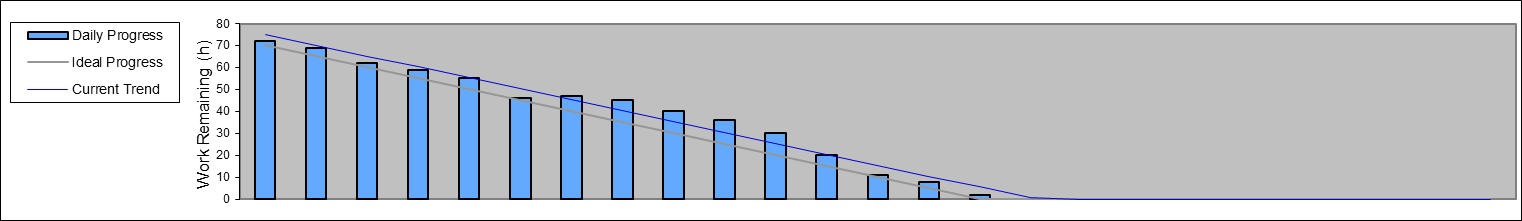
Durante el transcurso del sprint se gatilló nuevamente el riesgo habitual, relacionado con errar en la estimar las tareas. Por ende, como ya ha sido habitual se inyectó más horas al sprint. La evolución de los riesgos se puede ver a la tabla XXXX, a continuación, seguido de la imagen XXXX que contiene el grafico asociado a esta tabla.





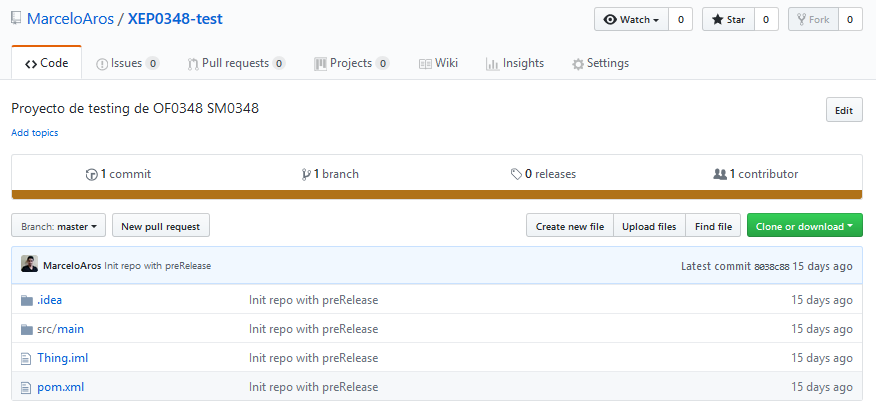
Como también se como es habitual se llevó registro del avance de las tareas, este se puede ver en la tabla XXXX y en el grafico XXXX ambos, a continuación.



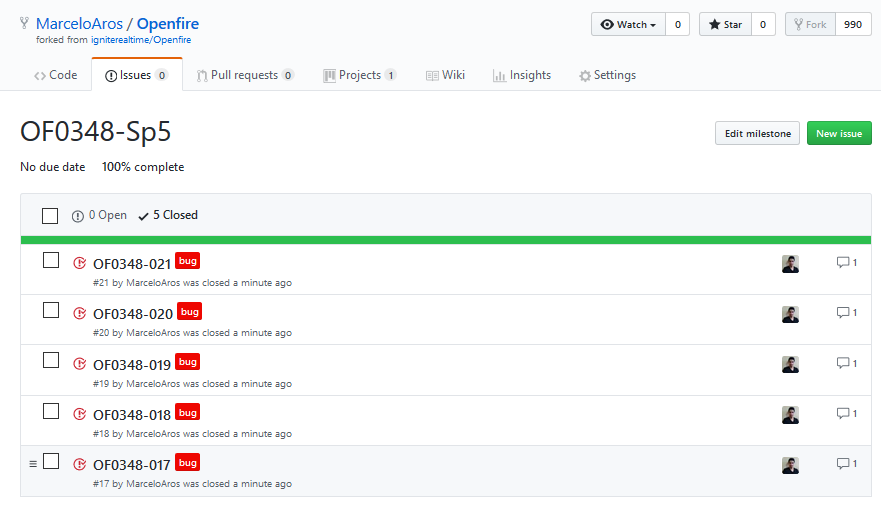


### A.1.5.4 Evidencia

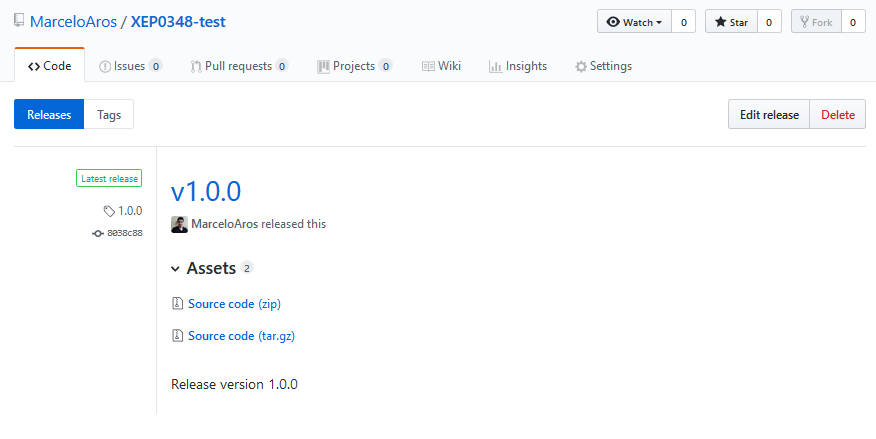
Toda la evidencia asociada a este sprint, esta ubicada en un cuarto repositorio GitHub, titulado XEP0348-test como se puede ver en la imagen XXX



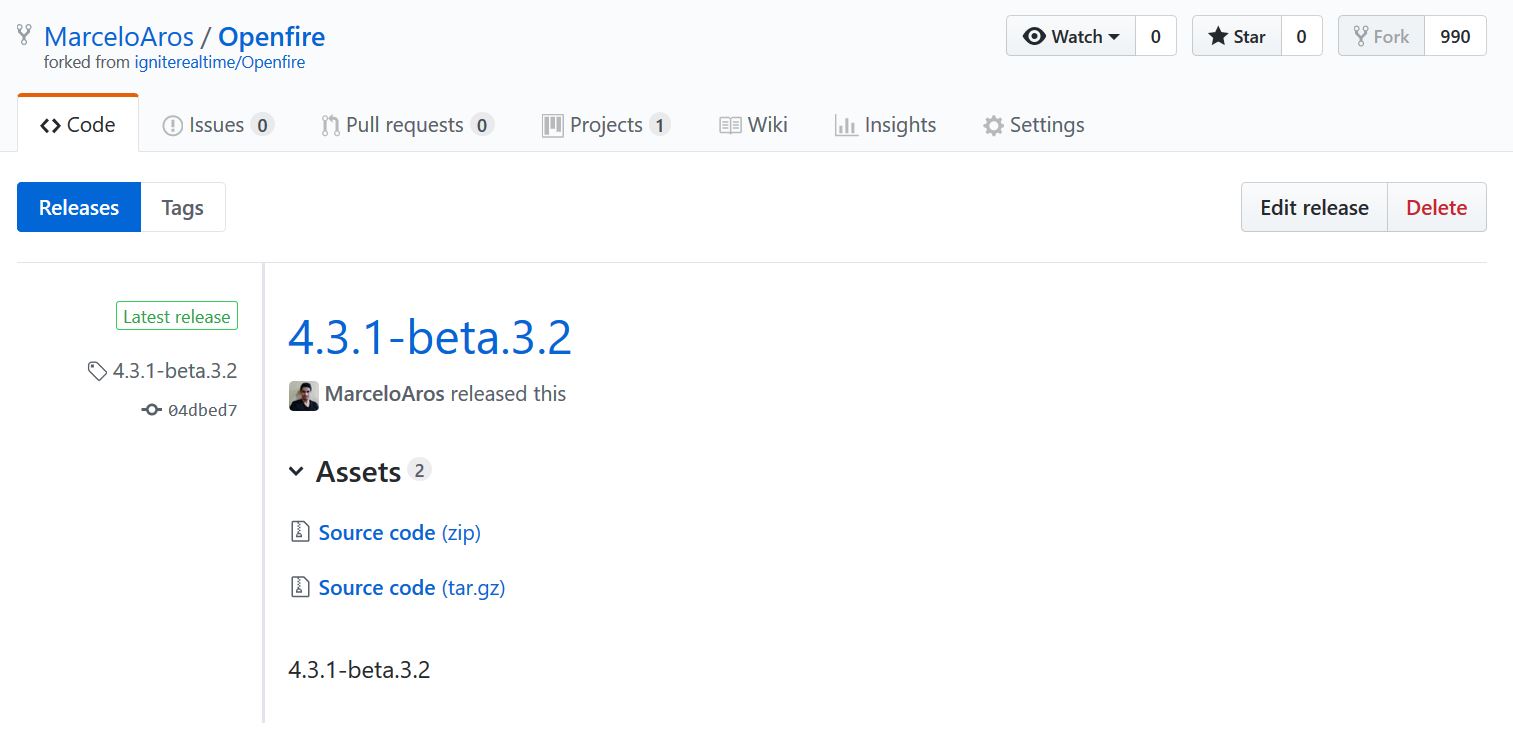
Dentro del sprint se detectaron incidencias, las cuales se reportaron mediante el sistema de reporte de estas en el repositorio GitHub Openfire, estas se pueden ver a continuación, en la imagen XXXX. Cabe mencionar que estas incidencias son tanto para el desarrollo de este sprint, como para Openfire y SMACK, realizando las correspondientes refactorizaciones.



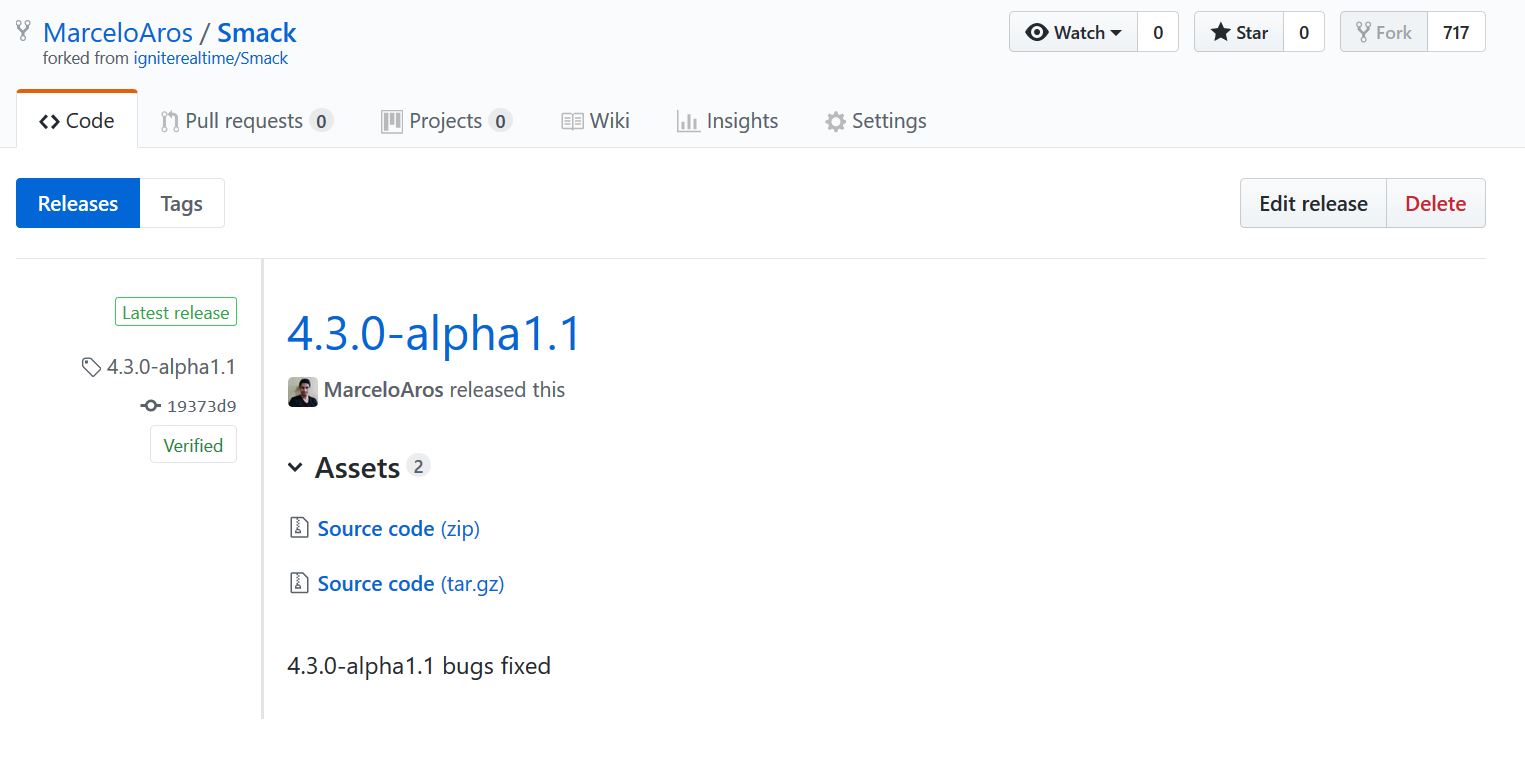
Al igual que en el caso del desarrollo de SMACK y de Openfire, se generó el reléase correspondiente, la cual es la versión 1.0.0 que contiene el «jar»



De igual modo, luego del proceso de refactorización a Openfire, se generó la versión 4.3.1-beta.3.2



Como también se refactorizó la librería SMACK, de este modo generando el reléase de la versión 4.3.0-alpha1.1



### A.1.5.5 Post-Mortem

**Problemas resueltos**

En este caso dimos solución a la necesidad de proveer un producto de software para el usuario final del sistema, como también testear el servidor XMPP-IoT Openfire modificado en conjunto con la librería SMACK, también modificada. En proceso, se identificaron diversos defectos originados en errores no detectados en etapas tempranas de desarrollo de SMACK y de Openfire. De este modo, según el alcance de este proyecto se logró generar las versiones definitivas de ambos, es decir Openfire 4.3.1-beta.3.2 y SMACK 4.3.0-alpha1.1

**Problemas futuros**

Quedaron tres historias de usuarios «Could» pendientes, debido al asignaciones temporales, las historias de usuario corresponden a eliminación de credenciales de consumidor junto con visualización y eliminación de identidades con una cierta credencial de consumidor.

Sin mencionar los XEPs de IoT que aún no se encuentran implementados a nivel que librería ni servidor.

**Lecciones aprendidas**

Cuando nos enfrentemos al desarrollo de un software que tiene un alto grado de incertidumbre, como el actual, el cual durante las reuniones Sprint Planning Poker todas las historias de usuario MUST tuvieron una valoración 55 o más. Debemos recurrir al criterio de un experto, buscar asesoría para reducir la brecha de incertidumbre.

**Métricas**

* LOC: Este sprint se finalizó con 1219 LOC.
* Complejidad ciclomatica promedio de las clases es de
  + ev(G) = 9
* Cantidad de defectos: Se detectaron cinco defectos. Se debe realizar la aclaración, que estos son transversales al proyecto, es decir, defectos en Openfire y en SMACK.
* Efectividad de pruebas: Se realizaron tres pruebas de aceptación, de las cuales todas fueron exitosas, es decir, todas encontraron defectos, un total de cinco defectos.
  + EP = 166.7%
* Densidad de defectos = 0.73