

Universidad ORT Uruguay
Facultad de Ingeniería

**GPRI: Gestión de preguntas y respuestas
inteligentes para Sagal**

Entregado como requisito para la obtención del título de Licenciado en Sistemas.

Nicolás González - 181592

Nicolás Caballero - 191100

Renzo Delfino - 196738

Tutor: Álvaro Ortas

2020

Declaración de autoría

Montevideo, Uruguay 13 de abril de 2020.

Nosotros, Nicolás González, Nicolás Caballero y Renzo Delfino, declaramos que el trabajo que se presenta en esa obra es de nuestra propia mano. Podemos asegurar que:

- La obra fue producida en su totalidad mientras realizábamos el Proyecto Final de Licenciatura en Sistemas;
- Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad;
- Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra;
- En la obra, hemos acusado recibo de las ayudas recibidas;
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuido por otros, y qué fue contribuido por nosotros;
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.



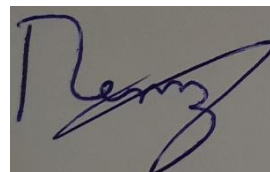
Nicolás González

13/04/2020



Nicolás Caballero

13/04/2020



Renzo Delfino

13/04/2020

Agradecimientos

Nos parece importante tomarnos un momento para agradecer a todas aquellas personas que nos acompañaron en este camino.

En primer lugar, a nuestras familias, por estar siempre apoyándonos durante toda la carrera universitaria. Fueron claramente un pilar fundamental, dando el soporte, contención y motivación para seguir siempre que fue necesario.

En segundo lugar, agradecer a la Universidad ORT Uruguay por ser nuestra guía en todo este proceso y a todos los docentes que nos acompañaron. En especial a nuestro tutor Álvaro Ortas por aceptar guiarnos en este camino, por sus consejos y su paciencia para ayudarnos.

En último lugar queremos agradecer a la empresa Sagal, por permitirnos desarrollar nuestro proyecto en su empresa y en especial a Alejandro Tocar, Simón Seitune y Alan Bukstein por siempre estar a disposición para resolver nuestras consultas y hacernos recomendaciones basándose en su experiencia.

Abstract

Dentro de la estructura del comercio electrónico, existen plataformas de integración entre grandes empresas de comercio y plataformas de venta como por ejemplo Mercado Libre.

Siendo Sagal una de estas plataformas, debe brindar diferentes servicios para facilitarle a las empresas el proceso de gestión de ventas electrónico a través de las diferentes plataformas.

La empresa Sagal, detectó que uno de sus subsistemas no estaba brindando suficiente valor a sus clientes que optaban por no usarlo. El subsistema se encarga de la comunicación entre las empresas y los usuarios finales que hacen preguntas a través de Mercado Libre y actualmente maneja 150.000 preguntas mensuales de los usuarios finales.

Es clave para las empresas que la comunicación con los clientes finales sea fluida ya que garantiza mayores niveles de ventas, si las preguntas son contestadas rápidamente.

Para resolver este problema nos propusimos reconstruir el subsistema utilizando nuevas tecnologías como Node.JS, React y MongoDB. A su vez, elegimos las metodologías SCRUM y BDD, las adaptamos a las necesidades y recursos del proyecto para así, poder cumplir con los objetivos propuestos. Dando como resultado, una pieza de software que agrega valor al servicio que brinda Sagal.

Palabras clave

Node.js, IBM Watson, Mercado Libre, integrador, ERP, SCRUM, BDD, React, MongoDB, Grails, Fastify, Cucumber, Machine Learning.

Glosario

- **Asp.net core:** *framework web* de Microsoft.
- **API:** *Application Programming Interface*, es una interfaz o protocolo de comunicación entre diferentes partes de un programa destinado a simplificar la implementación y el mantenimiento del software.
- **AWS:** Amazon Web Services, es un conjunto de diversos servicios en la nube brindando por la empresa Amazon.
- **AWS-CloudFront:** Amazon CloudFront, es un servicio de red de entrega de contenido que distribuye a globalmente datos, videos y APIs de modo seguro.
- **AWS-EC2:** Amazon Elastic Compute Cloud, es un servicio web que proporciona capacidad informática en la nube segura y de tamaño modificable.
- **AWS-S3:** Amazon Simple Storage Service, es un servicio de almacenamiento que ofrece escalabilidad, disponibilidad de datos y seguridad.
- **AirBnB:** es una empresa que ofrece alojamiento a particulares a través de una plataforma de software.
- **Atlassian:** es una empresa de *software* que crea productos para empresas y desarrolladores de *software*.
- **Backend:** es la parte del sistema que se encarga de toda la lógica interna, como puede ser la comunicación con la base de datos y con otros sistemas externos.
- **BDD:** *Behavior Driven Development*, es un proceso de desarrollo ágil que participa tanto los desarrolladores, como las personas del negocio utilizando un lenguaje en común.
- **Bitbucket:** servicio de alojamiento basado en la web, para proyectos que utilizan el sistema de control de versiones.
- **Branch:** termino empleado en programación que refiere a una rama de un repositorio, usado generalmente para trabajar.
- **Business to business (B2B):** empresa que brinda servicios a otras empresas.

- **Code review:** es la acción en el que uno o más desarrolladores verifican el código de otro en búsqueda de errores u oportunidades de mejora.
- **Confluence:** es un espacio de trabajo colaborativo entre equipos.
- **Cucumber:** herramienta de software que permite el desarrollo basado en el comportamiento (BDD).
- **Dashboard:** generalmente utilizado para describir vista de interfaz gráfica donde se muestran indicadores clave de rendimiento relevantes.
- **Docker:** sistema de gestión de contenedores.
- **Ecommerce:** sistema de compra y venta de productos y servicios que utiliza Internet como medio principal de intercambio.
- **Enzyme:** es una librería de JavaScript utilizada para realizar pruebas de una aplicación hecha en React.
- **Eslint:** es una herramienta de análisis de código estático para identificar patrones problemáticos encontrados en el código JavaScript.
- **Fastify:** *framework* web desarrollado para Node.js.
- **Feature:** funcionalidad o característica.
- **Framework:** marco de trabajo que contiene un conjunto estandarizado de prácticas y criterios.
- **Frontend:** es la parte de un programa o dispositivo a la que el usuario puede acceder directamente.
- **Grails:** es un *framework* para aplicaciones web.
- **Hotkeys:** atajos de teclado. Es una tecla o conjunto de teclas que efectúa una acción definida previamente.
- **IBM Watson:** inteligencia artificial que es capaz de responder a preguntas formuladas en lenguaje natural. Desarrollado por IBM.
- **IFrame:** elemento HTML que permite insertar un documento HTML dentro de un documento HTML principal.
- **Intercom:** compañía que brinda una plataforma de mensajería que permite a las empresas comunicarse con sus clientes.
- **Issues:** problemas que son representados como historia, bug o tarea.

- **Java servlet:** librería externa de Java que permite ampliar las capacidades de un *web service*.
- **Jest:** *framework* para pruebas de JavaScript mantenido por Facebook.
- **Linters:** herramienta para controlar los estándares del código.
- **MVP:** *Minimum Viable Product*, es un producto con suficientes características para satisfacer a los clientes iniciales y proporcionar retroalimentación para el desarrollo futuro.
- **Machine learning (ML):** aprendizaje automatizado de máquinas, es una rama de la inteligencia artificial.
- **Merge:** utilizado en GIT, es la acción de unir los cambios de una rama en otra.
- **Mockup:** es un prototipo de *frontend* que se utiliza para la demostración y evaluación del diseño.
- **Modal:** utilizado en *frontend*, es una ventana subordinada a la ventana principal. Crea un modo que deshabilita la ventana principal, pero la mantiene visible.
- **MongoDB Atlas:** es el servicio global de almacenamiento en la nube para base de datos.
- **MongoDB:** manejador de base de datos no relacionales, orientado a documentos.
- **Mongoose:** es un *Object Document Mapper*. Permite definir objetos con un esquema que se asigna a un documento de MongoDB.
- **Node.js:** entorno de ejecución para Javascript construido con el motor V8 de Google Chrome.
- **PWA:** aplicación web progresiva.
- **Pull request(PR):** son pedidos de aprobación de código para que otros integrantes del equipo puedan asegurar la calidad del mismo.
- **React:** librería de Javascript de código abierto diseñada para interfaces de usuario con el objetivo de facilitar el desarrollo.
- **Release:** entrega de software.
- **Render:** digitalización de imagen, que se presenta en la pantalla.
- **Request:** solicitud de información a un servicio.

- **Responsive:** diseño adaptativo utilizada en diseño web para visualizar correctamente una misma página en distintos dispositivos.
- **Rest:** Arquitectura de software que utiliza protocolos HTTP y JSON.
- **Scrum:** es un marco de trabajo para el desarrollo ágil de software.
- **Service worker:** es una secuencia de comandos que el navegador ejecuta en segundo plano.
- **Styled components:** librería de Javascript para crear componentes visuales mediante CSS en Javascript.
- **Swagger:** es un *framework* que permite a los desarrolladores documentar y consumir servicios web REST.
- **Tags:** etiqueta utilizada para catalogar preguntas.
- **Template:** plantillas utilizadas para contestar preguntas.
- **Token:** actúa como llave electrónica para acceder a un recurso mediante un proceso de sustitución de datos.
- **UX:** experiencia de usuario es el conjunto de factores y elementos relativos a la interacción del usuario con un entorno o dispositivo concreto.
- **Web service:** es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

Índice

1.	Introducción	16
1.1.	Conformación del equipo	16
1.2.	Objetivos	16
1.3.	Estructura del documento	18
2.	Cliente y descripción del problema	20
2.1.	Descripción del cliente	20
2.2.	Descripción del problema	21
3.	Solución planteada	24
3.1.	Objetivo de la solución	24
3.2.	Requerimientos funcionales (RF)	26
3.3.	Requerimientos no funcionales (RNF).....	27
3.4.	Tecnologías aplicadas.....	29
3.5.	Arquitectura	33
3.6.	Decisiones de diseño.....	39
4.	Metodología	41
4.1.	Metodologías utilizadas	42
4.2.	Personalización de SCRUM y BDD	44
5.	Proyecto.....	48
5.1.	Planificación	48
5.2.	Ejecución del proceso de desarrollo	51
5.3.	Ejecución del proceso de gestión	60
5.4.	Gestión de riesgo	67
5.5.	Métricas.....	69

5.6.	Entrega al cliente	75
5.7.	Resumen	77
6.	Aseguramiento de la calidad	78
6.1.	Calidad de los requerimientos	78
6.2.	Calidad del producto final	78
6.3.	Calidad del modelo de <i>machine learning</i>	80
7.	Gestión de la configuración.....	81
7.1.	Control de versionado	81
7.2.	Documentación	83
7.3.	Gestión de proyecto.....	86
8.	Conclusiones.....	88
8.1.	Conclusiones generales.....	88
8.2.	Cumplimiento de objetivos.....	88
9.	Lecciones aprendidas.....	91
10.	Continuidad del proyecto	93
11.	Referencias.....	95
12.	Anexos.....	100
12.1.	<i>Backlog</i>	100
12.2.	<i>Issues</i>	102
12.3.	<i>User Stories</i>	103
12.4.	<i>Burndown</i>	161
12.5.	<i>Sprint reports</i>	165
12.6.	<i>Sprint retrospectives</i>	172
12.7.	Reuniones con el cliente.....	180
12.8.	Reuniones del equipo	183
12.9.	<i>Endpoints de backend</i> con Swagger.....	191

12.10.	<i>Mockups</i>	192
12.11.	Reporte <i>tests</i>	200
12.12.	Reporte pruebas de concurrencia y rendimiento	203
12.13.	Estadísticas de las preguntas actuales de Sagal	204
12.14.	Interfaces requeridas para salida a producción.....	206
12.15.	Manual de usuario	210
12.16.	Reporte de riesgos	213
12.17.	Encuestas de satisfacción.....	217
12.18.	Tabla con modelo de distribución de esfuerzo.....	221

Índice de tablas

Tabla 3-1: comparativa entre diferentes <i>frameworks web</i>	30
---	----

Índice de ilustraciones

Ilustración 2.1: interacción actual de Sagal con los clientes.	21
Ilustración 3.1: interacción propuesta del equipo.	25
Ilustración 3.2: comparativa entre diferentes <i>frameworks web</i> de Node.js.	31
Ilustración 3.3: diagrama de alto nivel.	33
Ilustración 3.4: diagrama de componentes.	35
Ilustración 3.5: diagrama de comunicación.	37
Ilustración 3.6: diagrama de despliegue.	38
Ilustración 4.1: ciclo estándar de SCRUM + BDD.	43
Ilustración 4.2: ciclo personalizado de SCRUM + BDD.	45
Ilustración 5.1: planificación.	49
Ilustración 5.2: proceso personalizado de SCRUM + BDD.	51
Ilustración 5.3: <i>mockup</i> de <i>frontend</i> para web.	52
Ilustración 5.4: <i>mockups</i> versión móvil.	53
Ilustración 5.5: ejemplo de Gherkin.	54
Ilustración 5.6: ejemplo de caso de prueba.	54
Ilustración 5.7: ejemplo de corrida de <i>tests</i> en <i>backend</i>	55
Ilustración 5.8: resultado de prueba de concurrencia.	56
Ilustración 5.9: Gitflow visualización en Sourcetree.	58
Ilustración 5.10: ejemplo de <i>pull request</i>	59
Ilustración 5.11: ejemplo de <i>user story</i>	62
Ilustración 5.12: <i>sprint 2 report</i>	65
Ilustración 5.13: ejemplo de <i>retrospective</i> en <i>sprint 5</i>	67
Ilustración 5.14: tarea para mitigar el riesgo de no puesta en producción.	69
Ilustración 5.15: productividad del equipo.	70
Ilustración 5.16: <i>burndown</i> MVP 1.	71
Ilustración 5.17: <i>burndown</i> MVP 2.	72
Ilustración 5.18: distribución de esfuerzo de equipo.	73
Ilustración 5.19: modelo de distribución de esfuerzo.	74
Ilustración 7.1: repositorios del equipo.	81
Ilustración 7.2: reglas de los repositorios en Bitbucket.	83

Ilustración 7.3: página de Confluence.....	84
Ilustración 7.4: OneDrive online con documentos de Word	85
Ilustración 7.5: Google drive	86
Ilustración 7.6: <i>sprint</i> documentación - <i>task board</i>	87
Ilustración 10.1: reunión con Sagal para definir pasos a seguir.....	93

1. Introducción

El presente documento tiene como objetivo describir el proyecto final de carrera “GPRI: Gestión de preguntas y respuestas inteligentes para Sagal” que es requisito para poder finalizar la carrera de Licenciatura en Sistemas. Este proyecto fue realizado entre agosto 2019 y abril 2020.

1.1. Conformación del equipo

Los integrantes del equipo cursamos la carrera en conjunto. Por lo que se dio naturalmente la conformación para este proyecto, ya que conocíamos nuestro nivel de compromiso y profesionalismo.

El equipo está conformado por:

- **Nicolás González:** estudiante de Licenciatura en Sistemas, Team Lead en Qualabs.
- **Nicolás Caballero:** estudiante de Licenciatura en Sistemas, Implementation Lead en Sagal.
- **Renzo Delfino:** estudiante de Licenciatura en Sistemas, Desarrollador en Sagal.

1.2. Objetivos

En esta sección describiremos los objetivos planteados por el equipo que nos orientaron para determinar el éxito del proyecto.

Estos objetivos se pueden dividir en tres grandes tipos:

- Generar valor genuino al cliente.
- Mejorar en procesos de gestión de proyectos ágiles.
- Profesionalización del trabajo en equipo.

Generar valor genuino al cliente

Dentro de los pilares de la carrera de Licenciatura en Sistemas y los procesos ágiles, la generación de valor al cliente es un objetivo central. Por lo cual lo consideramos como algo fundamental para poder medir el impacto de nuestro proyecto. Decidimos medir este objetivo considerando:

1. Poder entregar dos MVP. En los capítulos siguientes profundizaremos en el alcance de estos MVPs, pero consideramos que era necesario poder tener una primera entrega para validar varias acciones de nuestros clientes.
2. Obtener un 4 sobre 5 de promedio en la encuesta realizada a empresa X y a Sagal luego de cada MVP. Dentro de nuestra estructura de *feedback* sobre el proyecto planeamos medir la satisfacción directamente.
3. Lograr que el modelo de etiquetas automáticas por *machine learning* tenga una confianza de 70%.
4. Luego del primer MVP, lograr que se aumente el uso del sistema. Dentro de la descripción del problema profundizaremos en este punto. Si bien al finalizar el proyecto no tendremos datos suficientes para medir este objetivo, esta métrica queda definida para la continuidad futura del proyecto.

Mejorar en procesos de gestión de proyectos ágiles

Como equipo nuestro objetivo fue poder continuar trabajando en proyectos juntos, para esto detectamos que necesitábamos mejorar nuestra gestión de proyectos ágiles, por lo que para medirlo consideramos:

1. Iterar sobre un proceso definido de gestión que permita mejorar la productividad del equipo en comparación con los *sprints* anteriores. Una vez definido un proceso de gestión personalizado, midiendo nuestra productividad, debe ir aumentando a medida que pasen los *sprints*.
2. Tener cero fallas críticas en producción. La clasificación de fallas la expondremos más adelante.

3. Tener un retrabajo que no supere un 20% del total del esfuerzo. Según CMMI[1] nivel 5, el retrabajo no debería de superar el 20%. Si bien CMMI está pensado para proyectos de tipo cascada, creemos que puede ser un buen marco de referencia para nuestro proyecto.

Profesionalización del trabajo en equipo

Como mencionamos anteriormente, nuestro objetivo como equipo era poder continuar trabajando juntos en diferentes proyectos. Para esto, si cumplíamos con los objetivos anteriormente definidos, considerábamos que nuestro equipo logro el nivel de profesionalización establecido. Además, en específico para este objetivo, consideramos que aprobar la instancia académica es clave para consolidarnos como equipo.

1.3. Estructura del documento

Se presentará un breve resumen de los capítulos presentados como forma de guiar al lector en cómo decidimos organizar el contenido del proyecto.

A su vez, optamos por agregar un glosario con términos que permitan al lector ponerse en contexto del dominio y el problema planteado.

Cliente y descripción del problema

Este capítulo cuenta con una descripción del entorno y cuáles son los actores relevantes que operan en él. Además, detallamos cómo surge el problema y como, en conjunto con Sagal[2], llegamos a una solución.

Solución y arquitectura propuesta

Dentro de este capítulo profundizaremos sobre la solución alcanzada, no solo los requerimientos que cubre a nivel de negocio sino también como está diseñada internamente.

Metodología utilizada

En este capítulo detallaremos cuáles fueron las metodologías utilizadas en las diferentes etapas del proyecto y que personalizaciones realizamos a las diferentes metodologías, para adecuarlas a nuestro equipo y al proyecto.

Ejecución del proyecto

Este capítulo cuenta con el detalle de la instanciación del proceso que se describió en el capítulo anterior. Detallaremos cada etapa de la ejecución del proyecto, cómo nuestras personalizaciones se materializaron durante la ejecución y qué resultados obtuvimos de cada etapa.

Conclusiones y pasos siguientes

Detallamos cuáles son las conclusiones de todo el proceso y la ejecución del proyecto, como esto se relaciona con nuestros objetivos planteados al principio y cuáles son los pasos futuros del proyecto.

Lecciones aprendidas

Dentro de este capítulo documentamos todo lo que fuimos aprendiendo a lo largo del proyecto.

2. Cliente y descripción del problema

Para adentrarnos en cómo se identificó el problema vemos necesario empezar describiendo en qué rubro se desarrolla nuestro cliente y que soluciones brinda en el mercado. También dejamos algunas puntualizaciones de lenguaje que definimos para facilitarle al lector el entendimiento del problema.

2.1. Descripción del cliente

Sagal[2], nuestro cliente, surge como proyecto de tesis de la Universidad ORT Uruguay en el año 2016.

Es una aplicación que brinda servicios *business to business* y se encarga de integrar los ERP[3] de comercios de venta de productos con los *ecommerce*. De esta manera, les permite a las empresas acceder a los siguientes servicios:

- **Publicación Masiva:** la posibilidad de publicar todos los artículos del ERP de forma masiva.
- **Sincronización de stock:** el stock del producto publicado en los canales digitales está sincronizado con el reportado por el ERP.
- **Gestión de pedidos:** verificación del estado de los pedidos en los diferentes canales.
- **Facturación automática:** crear facturas de forma automática en los ERP.
- **Múltiples canales:** la posibilidad de gestionar múltiples canales de venta online.

En esta estructura, es común que una empresa utilice muchas cuentas del mismo *ecommerce* para facilitar la gestión de sus diferentes productos publicados. Esto hace más atractivo el uso de Sagal, que permite gestionar las diferentes cuentas dentro de un mismo *ecommerce*, centralizadas como un mismo usuario.

Actualmente Sagal cuenta con más de 80 clientes en Latinoamérica, publicando más de 300.000 mil artículos en diferentes plataformas electrónicas de comercio.

2.2. Descripción del problema

Dentro de los servicios de integración mencionados, Sagal cuenta con un módulo que recibe las preguntas realizadas por los usuarios finales dentro de los diferentes *ecommerce* y les permite contestarlas desde ahí, sin tener que acceder a cada cuenta en particular.

El siguiente diagrama muestra la forma de interrelación de los distintos componentes del problema.



Ilustración 2.1: interacción actual de Sagal con los clientes.

Los pasos serían de la siguiente forma:

1. Empresa X, empresa cliente de Sagal, publica 50.000 productos en Mercado Libre[4] a través de Sagal.
2. El usuario final, cliente de Empresa X, interesado en comprar esos productos, hace preguntas sobre los mismos en Mercado Libre.
3. En el flujo actual, la Empresa X opta por contestar esas preguntas directamente en Mercado Libre y no mediante Sagal.

Actualmente Sagal maneja un promedio mensual superior a las 158.000 preguntas (véase anexo **12.13: Estadísticas de las preguntas actuales de Sagal**) que realizan los usuarios finales a los diferentes clientes de Sagal, esto es, Empresas X. Dentro de

este volumen de preguntas, Sagal detectó que las Empresas X no estaban haciendo uso de su plataforma para contestar las preguntas y lo estaban haciendo directamente en los *ecommerce*. Esto fue identificado como un punto que requería mejora dentro de Sagal ya que algunos de sus competidores (Producteca[5], Astroselling[6]) cuentan funcionalidades dentro del subsistema de preguntas que Sagal no ofrece, por lo que debe mantenerse actualizada para poder ser competitivo en el mercado.

Sagal, luego de un relevamiento con sus clientes, también detecto los siguientes problemas:

- El módulo actual de respuesta no contiene información detallada respecto al artículo sobre el que se está preguntando. Esto genera ineficiencias a la hora de responder, ya que se debe buscar la información requerida para contestar la pregunta en algún otro sistema, ya sea el ERP o directamente en MercadoLibre.
Por ejemplo: los artículos pueden tener diferentes variantes como color, modelo o talla.
- Para poder llegar a la pregunta se deben dar tres pasos; cada paso lleva a una nueva vista, lo cual requiere pedir nuevos datos al servidor y eso provoca demoras a la hora de traer los datos de las preguntas.
Por ejemplo: el sistema actual cuenta con una lista de preguntas, desde la cual se hace *click* en una de ellas, lo que recarga la página y nos lleva a otra vista.
- Una vez que se accede a una pregunta, se debe volver a recargar la vista donde se encuentra el listado de preguntas, lo que también agrega demoras si se necesita navegar entre ellas o se hace *click* por error.

Los puntos anteriores, surgidos del relevamiento antes mencionado, solicitados a Sagal expresamente por sus clientes, los tomamos como puntos focales durante el desarrollo de la solución.

Dentro de este contexto, tuvimos varias reuniones con Sagal y con ORTs, laboratorio de ingeniería de software de nuestra facultad, para refinar los posibles

acercamientos a una solución para este problema que le permitiera a Sagal generar una ventaja competitiva y facilitarles el trabajo operativo a sus clientes. En los capítulos siguientes nos dedicamos a explicar en detalle qué solución alcanzamos.

3. Solución planteada

Dentro de este capítulo vamos a ver nuestra solución propuesta, cómo definimos la arquitectura y cuáles fueron los requerimientos principales en los que debimos trabajar. Incluimos en este capítulo cuales fueron las tecnologías usadas dentro de la solución y que criterios usamos para la elección de éstas. Para cerrar el capítulo, veremos los requerimientos no funcionales y como estos afectaron nuestras decisiones de diseño de la solución.

3.1. Objetivo de la solución

Luego de tener varias conversaciones con el cliente y determinar los problemas planteados anteriormente, sugerimos tres grandes cambios para agregar valor a Sagal. El primero fue separar el servicio de preguntas, es decir, aislarlo como un servicio independiente, pero mantenerlo dentro de la plataforma de Sagal, de forma de optimizar el rendimiento, tener un mejor control de llamadas y permitir cierta portabilidad y modificabilidad deseada a futuro. A su vez, determinamos que debíamos desarrollar una nueva *UX (user experience)*^[7] para adaptarnos a los requerimientos de la usabilidad. Por último, incorporar el uso de *machine learning* para el etiquetado automático de las preguntas, de forma tal de mejorar la usabilidad y rapidez de uso del sistema. Puede sonar extraño, en una primera instancia, suponer que *machine learning* fuese a mejorar la usabilidad, pero esto apoya al usuario, reduciendo su trabajo al crear las respuestas automáticas.

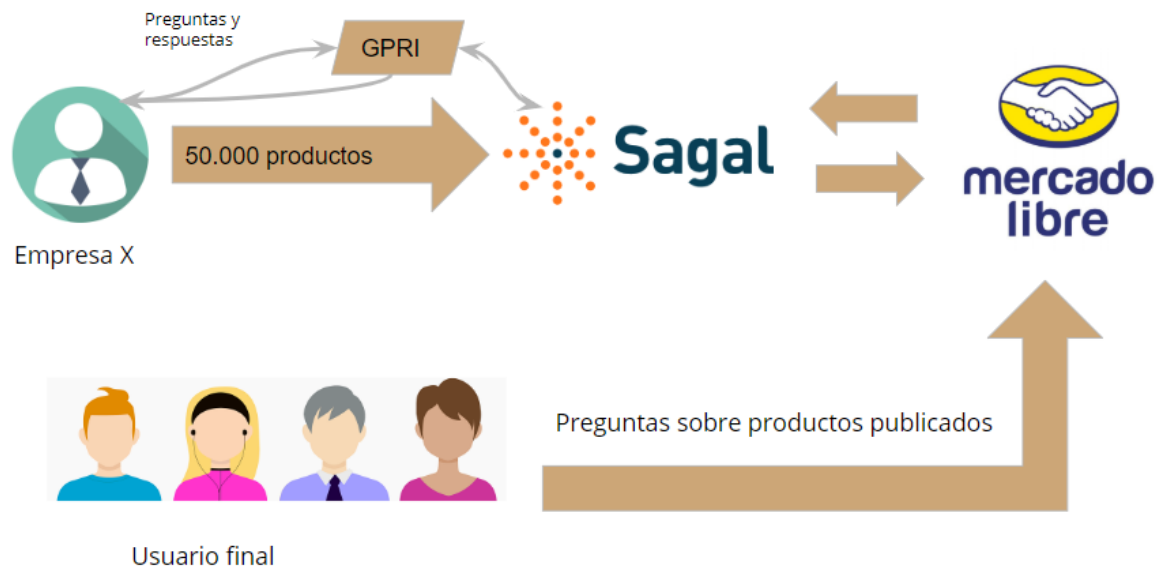


Ilustración 3.1: interacción propuesta del equipo.

En la Ilustración 3.1, podemos ver el cambio en el flujo, asegurando que Empresa X se mantenga dentro de la plataforma de Sagal.

El funcionamiento de GPRI, a grandes rasgos, es el siguiente:

- **Notificación sobre preguntas:** el usuario final hace una pregunta sobre un artículo en Mercado Libre y esa pregunta pasa por los sistemas internos de Sagal y, por último, notifica a nuestro sistema, GPRI, que una nueva pregunta fue creada o actualizada.
- **Etiquetado automático de *machine learning*:** una vez que Sagal notifica sobre las preguntas a responder, las toma el módulo de *machine learning* de GPRI para que sean catalogadas y etiquetadas y devuelve a que tópicos corresponde cada pregunta.
- **Notificación sobre artículos:** para poder brindar la información actualizada de los artículos a la hora de responder una pregunta y ver sus detalles, fue necesario contar con un servicio que permite a Sagal notificar a GPRI de cualquier cambio que haya habido sobre los artículos (precio, tipo, color, inventario, etc.).
- **Responder preguntas:** una vez llegan preguntas a responder, se muestran en una sección para poder responderlas rápidamente. Se despliega la

primera pregunta, se responde y se despliega la siguiente para responder. Esta funcionalidad provee diversas herramientas que ayudan al usuario de Empresa X a ser más eficiente en la respuesta, tales como:

- Etiquetado automático de preguntas para saber cuál es el tópico y tener una idea de que están preguntando sin necesidad de leerla.
- Creación y uso de plantillas para responder sin tener que escribir texto.
- Si el usuario tiene configurado el módulo respuesta automática, asociando *templates* a las diferentes etiquetas de las preguntas, puede generar respuestas automáticas para responder con una única acción.
- **Historial de preguntas:** todas las preguntas respondidas, eliminadas o cerradas, se pueden visualizar en una sección de historial. Si la pregunta fue respondida en GPRI, se puede obtener datos como quien la respondió y a qué hora fue respondida, entre otros.

Usuarios

Los usuarios que usarán GPRI son los que se identifican como Empresa X en la Ilustración 3.1. Actualmente, estos usuarios usan la aplicación desarrollada por Sagal sin GPRI. Su función es responder las preguntas que hacen los usuarios finales, como se muestra en el diagrama mencionado.

3.2. Requerimientos funcionales (RF)

Basándonos en la clasificación de IEEE[8], optamos por dividir los requerimientos en funcionales y no funcionales. Esta clasificación nos resulta más sencilla para poder explicar claramente los requerimientos y el problema.

Los requerimientos más relevantes fueron los siguientes:

- **Al visualizar las preguntas, la Empresa X debe poder responderlas una tras otra:** al aparecer el listado de preguntas, debemos poder contestar una tras otra, sin mayores cambios de pantallas o vistas entre preguntas.
- **Se deben poder ver los detalles del artículo al responder una pregunta:** Dentro de la vista de detalles de preguntas, se debe incluir detalles del artículo. Algunos de estos serían: Título, Precio, Stock, Garantía y Variantes del producto como talle y color.
- **Poder responder preguntas con recomendaciones de respuestas:** se le debe permitir al usuario de Empresa X crear y usar respuestas predefinidas según las etiquetas que *machine learning* asigne a las preguntas.
- **Etiquetado automático de preguntas mediante IBM Watson:** GPRI debe etiquetar las preguntas, en forma automática, en el momento que las recibe.
- **Permitir guardar textos predefinidos para las respuestas:** se debe permitir la posibilidad de que los usuarios de Empresa X guarden textos predefinidos a la hora de responder, haciendo uso de los datos que están en el sistema.
- **Dashboard con métricas de uso y rendimiento:** los usuarios de Empresa X deben poder ver estadísticas sobre el uso del sistema.
- **Comunicación con Sagal:** GPRI debe poder recibir notificaciones de Sagal y enviar correctamente las respuestas a las preguntas generadas por Empresa X. Estas notificaciones incluyen actualizaciones de los artículos y actualizaciones de las preguntas.

Los requerimientos en detalle se pueden ver en el anexo **12.1: Backlog** y en anexo **12.3: User Stories**.

3.3. Requerimientos no funcionales (RNF)

Entre los requerimientos no funcionales relevados detectamos como más importantes:

- **Usabilidad:** fue el RNF más importante definido por el cliente, ya que lo identifiqué como el problema que presenta el sistema actual de Sagal en su sección de preguntas. GPRI debe ser *responsive*, que las acciones para responder preguntas no requieran más de 2 pasos y que la información desplegada al usuario para poder responder la pregunta no lo sature.
- **Modificabilidad:** la aplicación fue construida pensando en ser los cimientos para una reestructura de la web original de Sagal en el futuro. Debido a eso el cliente nos proporcionó un experto en *frontend* para que hiciera *code review* de las *pull requests* en dicho componente. Por esto mismo se nos solicitó el uso de *ESLint*[\[9\]](#) para seguir un estándar de codificación, siguiendo los lineamientos de *AirBnB*[\[10\]](#).
- **Performance:** la carga de las preguntas debe ser menor de dos segundos para un máximo de cincuenta usuarios concurrentes.
- **Portabilidad:** en ambiente web se debe poder usar para *Edge, Safari, Chrome* y *Firefox*. En dispositivos móviles debe ser portable en *iOS* y *Android*.
- **Confiabilidad:** el sistema debe tener en producción, como máximo, la siguiente cantidad de fallas:
 - 0 fallas críticas.
 - 5 fallas medias.
 - Sin limitante de fallas leves.

Por la descripción de la clasificación de fallas véase el capítulo 5.5: Métricas.

- **Modelado de *machine learning*:** el modelo debe tener una confianza de, como mínimo, 70% y brindar herramientas para que Sagal pueda reentrenar el modelo.
- **Interoperabilidad con las bases de datos:** la aplicación debe persistir la información generada por su uso, como son las respuestas de los usuarios, etiquetas y toda la información que pueda ser usada para generar métricas.
- **Documentación:** es importante para Sagal poder acceder a la documentación generada dentro del proyecto, así como una documentación

en código que permita fácilmente ver qué puntos de contacto tiene la aplicación.

- **Software:** el *backend* debe ser desarrollado en Node.js[11], el mismo generado en un container de Docker[12], y el *frontend* debe ser realizado con React[13]. La comunicación entre los diferentes componentes debe ser mediante *REST*[14]. Para el etiquetado de las preguntas por *machine learning* se debe usar IBM Watson[15].
- **Hardware:** el servicio de *backend* debe ejecutarse en servidores de AWS (*Amazon Web Services*[16]), el servicio *frontend* debe poder alojarse en AWS-S3[17].

3.4. Tecnologías aplicadas

Debido a los requerimientos solicitados por el cliente en cuanto a tecnologías para este proyecto, optamos por dividir este subcapítulo en dos partes, una para las requeridas por el cliente y otra para las elegidas por el equipo.

Para elegir tecnologías definimos ciertos criterios para evaluar las tecnologías y quedarnos con la que cumpliera la mayor cantidad de criterios.

Requeridas por el cliente

Como comentamos en los RNF, el cliente tiene como objetivo usar la aplicación como base para desarrollos futuros de Sagal a nivel de *frontend*. Para esto, se nos solicitó usar las siguientes tecnologías en este ámbito:

- React[13]
- Styled components[18]
- ESLint[9]

Para las funcionalidades de *machine learning* se nos solicitó usar *IBM Watson*.

Elegidas por el equipo

Para definir la tecnología del *backend* tomamos en cuenta los siguientes criterios:

- Rendimiento del *framework* y la plataforma. Medida tomada de Web Framework Benchmarks[19].
- Preferencias del cliente (Alta, Media, Baja).
- Experiencia profesional del equipo (Alta, Media, Baja).
- Comunidad. Tomamos la evaluación de “*Other Frameworks, Libraries and tools*”[20].

Evaluamos tres opciones:

- Node.js (*Framework*)[11] y Fastify (Plataforma)[21]
- Java Servlet (*Framework*)[22] y Grails (Plataforma)[23]. Sagal está desarrollado con esta tecnología.
- .NET (*Framework*)[24] y Aspcore (Plataforma)[25]

Criterios/Herramientas	Fastify (Node.js)	Grails (JavaServlet)	Aspcore (.NET)
Preferencia del cliente	Alta	Media	Baja
Performance	Ranking: 99	Ranking: 215	Ranking: 10
Experiencia profesional del equipo	Media	Alta	Media
Comunidad	49.9%	Sin considerar	23.7%

Tabla 3-1: comparativa entre diferentes *frameworks web*.

El criterio primordial fue la preferencia del cliente, debido a que estamos agregando un nuevo conjunto de tecnologías a su solución actual, ellos conocen las limitaciones que tiene Grails por lo que tuvo un mayor peso sobre la decisión final. Sumado a esto

el contar con una mejor *performance* de rendimiento y una experiencia profesional del equipo media, decidimos usar Node.js y Fastify.

A su vez, para el gestor de *web service*, hicimos comparaciones extra para asegurarnos que dentro del *Framework* elegido teníamos una plataforma con buen *performance*. Como resultado de la comparativa reafirmamos la elección de Fastify ya que brinda una mayor *performance* entre el resto.

Framework	Version	Router?	Requests/sec
hapi	18.1.0	✓	29,998
Express	4.16.4	✓	38,510
Restify	8.0.0	✓	39,331
Koa	2.7.0	X	50,933
Fastify	2.0.0	✓	76,835
-			
http.Server	10.15.2	X	71,768

Ilustración 3.2: comparativa entre diferentes *frameworks web* de Node.js

También definimos Swagger[26] como herramienta para documentar la API resultante del desarrollo. Analizamos varias opciones, pero la integración con Swagger requería una inversión en tiempo mucho menor que las otras soluciones.

Para la base de datos optamos por el uso de MongoDB[27] por las siguientes razones:

- **Rápida escritura.** GPRI va a recibir en torno a las 158.000 preguntas mensuales (véase anexo **12.13: Estadísticas de las preguntas actuales de Sagal**), a su vez, recibe todas las notificaciones de los cambios en los artículos, por lo que decidimos darle prioridad a buscar una base de datos que sea óptima en las inserciones para no generar un cuello de botella.
- **Desnormalización de la información.** Esto quiere decir que toda la información necesaria para las preguntas y artículos va a estar contenida dentro de un mismo documento de MongoDB y, por tanto, nos evitamos el uso de uniones entre diferentes documentos. Esto nos permite que las

consultas a la base de datos sean rápidas y sencillas de realizar puesto que toda la información esta centralizada.

- **Escalable.** Es posible el uso de nodos para replicar la información y asegurar la confiabilidad de la información.

Para la interacción entre la base de datos y el *backend* nos apoyamos en Mongoose[28], ya que se basa en una solución de esquemas, lo cual nos permite manejar los documentos de MongoDB como modelos predefinidos.

Con respecto al *frontend*, ya que uno de los requerimientos no funcionales fue la portabilidad, decidimos investigar las herramientas que nos ayudaran a mejorar la experiencia de usuario. En esta investigación surgió una tecnología desconocida por todos los miembros del equipo: *Progressive Web Application (PWA)*[29].

PWA son básicamente aplicaciones web, que mediante el uso de *service workers*, permite seguir ejecutándolas en segundo plano sin tener que vivir dentro del navegador. Esto se traduce en tener un acceso directo en la pantalla del celular a nuestra aplicación, sin necesidad de instalarse mediante una tienda de aplicaciones y sin tener que estar ejecutando un navegador, como por ejemplo Google Chrome. A su vez, si se quisiera, ésta se podría instalar y ejecutar dentro de *Windows* como una aplicación de escritorio.

En términos generales para *testing* decidimos utilizar Cucumber[30] junto con Jest[31], ya que se adaptan perfectamente a las tecnologías utilizadas. A su vez, dado que tanto el *backend* como el *frontend* son en Javascript, pudimos utilizar las mismas librerías y el mismo dialecto para gestionar el *testing*. Estas herramientas son para realizar BDD, como se explicará en el siguiente capítulo. A su vez, para *frontend* decidimos utilizar Enzyme[32] para simular el *render* de componentes gráficos, ya que varios *tests* nos requerían comportamiento de integración entre distintos componentes.

3.5. Arquitectura

Diagrama de alto nivel

En el siguiente diagrama podemos ver a alto nivel como se relacionan los diferentes componentes de GPRI y los servicios externos.

Todo lo que se encuentra en verde es lo contemplado por nuestra solución, lo que está en amarillo son los servicios externos y la plataforma de Sagal en naranja.

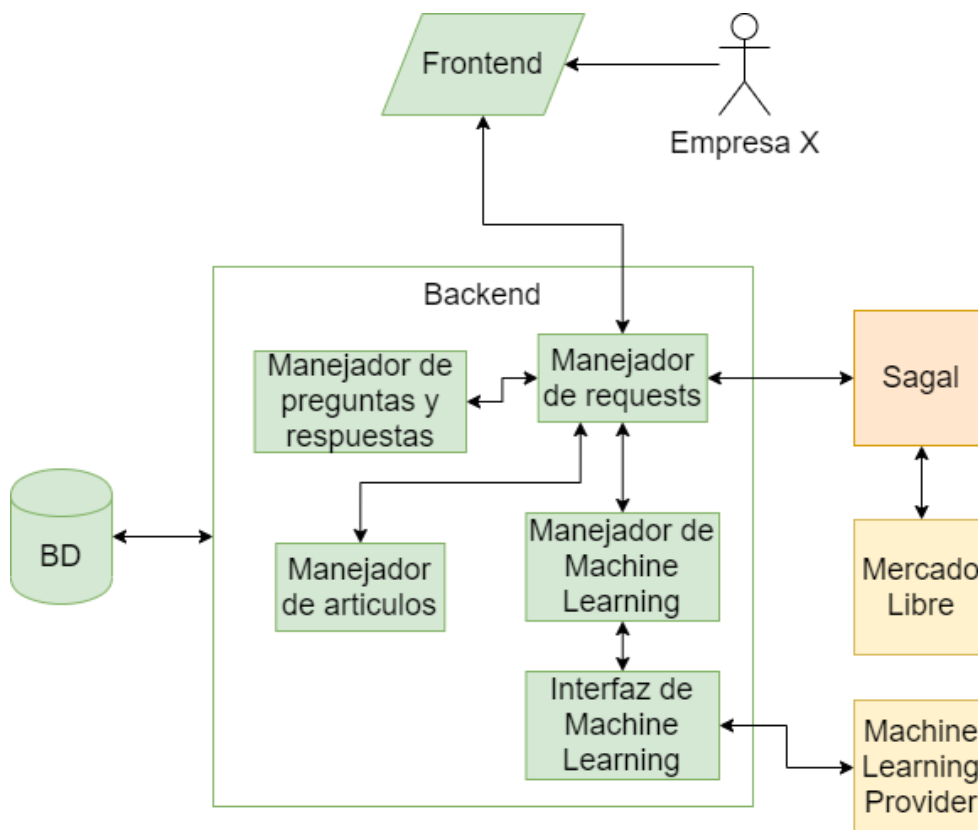


Ilustración 3.3: diagrama de alto nivel

- **Frontend:** es el responsable de conversar con el usuario final, Empresa X, y enviar los datos al *backend* y de devolverle la información que necesita.
- **Backend:**
 - **Manejador de requests:** encargado de obtener las diferentes *requests* y procesarlos en los diferentes manejadores.

- **Manejador de preguntas y respuestas:** encargado de procesar preguntas y respuestas de los diferentes usuarios.
- **Manejador de artículos:** encargado del manejo de datos de los artículos asociados a las diferentes preguntas.
- **Manejador de *machine learning*:** se encarga del procesamiento de la configuración del usuario y el reentrenamiento.
- **Interfaz de *machine learning*:** estructura genérica que permita que se pueda establecer otro motor de *machine learning* si se desea, siempre respetando la misma interfaz.
- **Base de datos:** base de datos central de la aplicación, la misma tiene comunicación directa con el *backend*.
- **Sagal:** componente externo que se encarga de las diferentes prestaciones, así como la comunicación entre nuestro *backend* y los diferentes *ecommerces* en especial Mercado Libre.

Diagrama de componentes

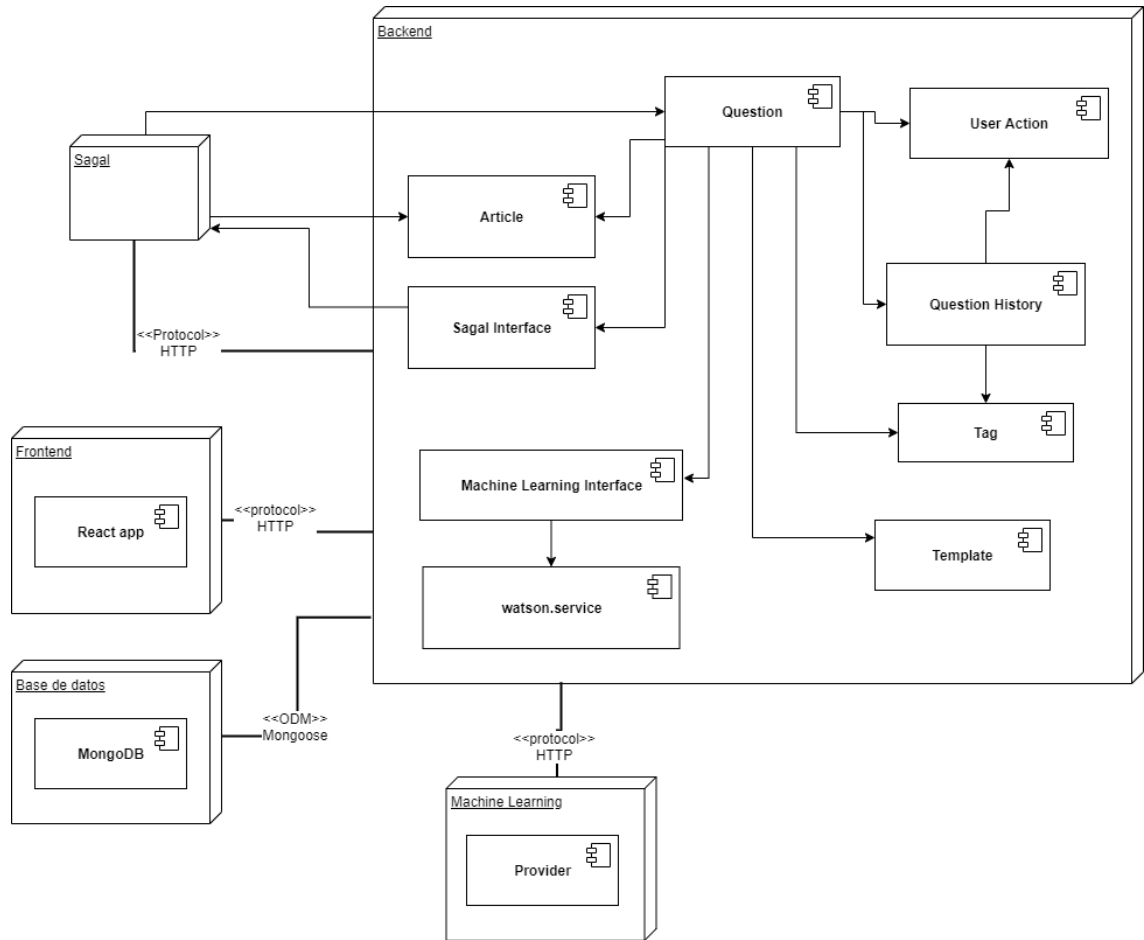


Ilustración 3.4: diagrama de componentes.

Con el diagrama en la Ilustración 3.4 tenemos una vista en detalle de los servicios dentro del *backend*. Pasaremos a detallar las funciones de cada uno de los diferentes servicios.

- **Backend:**
 - **Question:** servicio responsable de procesar las preguntas enviadas por Sagal y llamar a los otros servicios involucrados. Cuando se contesta una pregunta es el responsable de llamar a la interfaz de Sagal.
 - **Sagal interface:** encargado del envío de las respuestas a Sagal.

- **User Action:** responsable del control de las diferentes acciones de los usuarios, por ejemplo, contestar una pregunta y asignar o eliminar *tags*.
- **Question history:** encargado del control del historial de preguntas de los diferentes usuarios.
- **Tag:** responsable de guardar las etiquetas ingresadas por Empresa X.
- **Template:** encargado de guardar y devolver los *templates*.
- **Article:** responsable de recibir las actualizaciones de los artículos de parte de Sagal y actualizarlas en la base de datos.
- **Machine learning interface y Watson Service:** responsables de la comunicación con el proveedor de *machine learning*.

Diagrama de comunicación

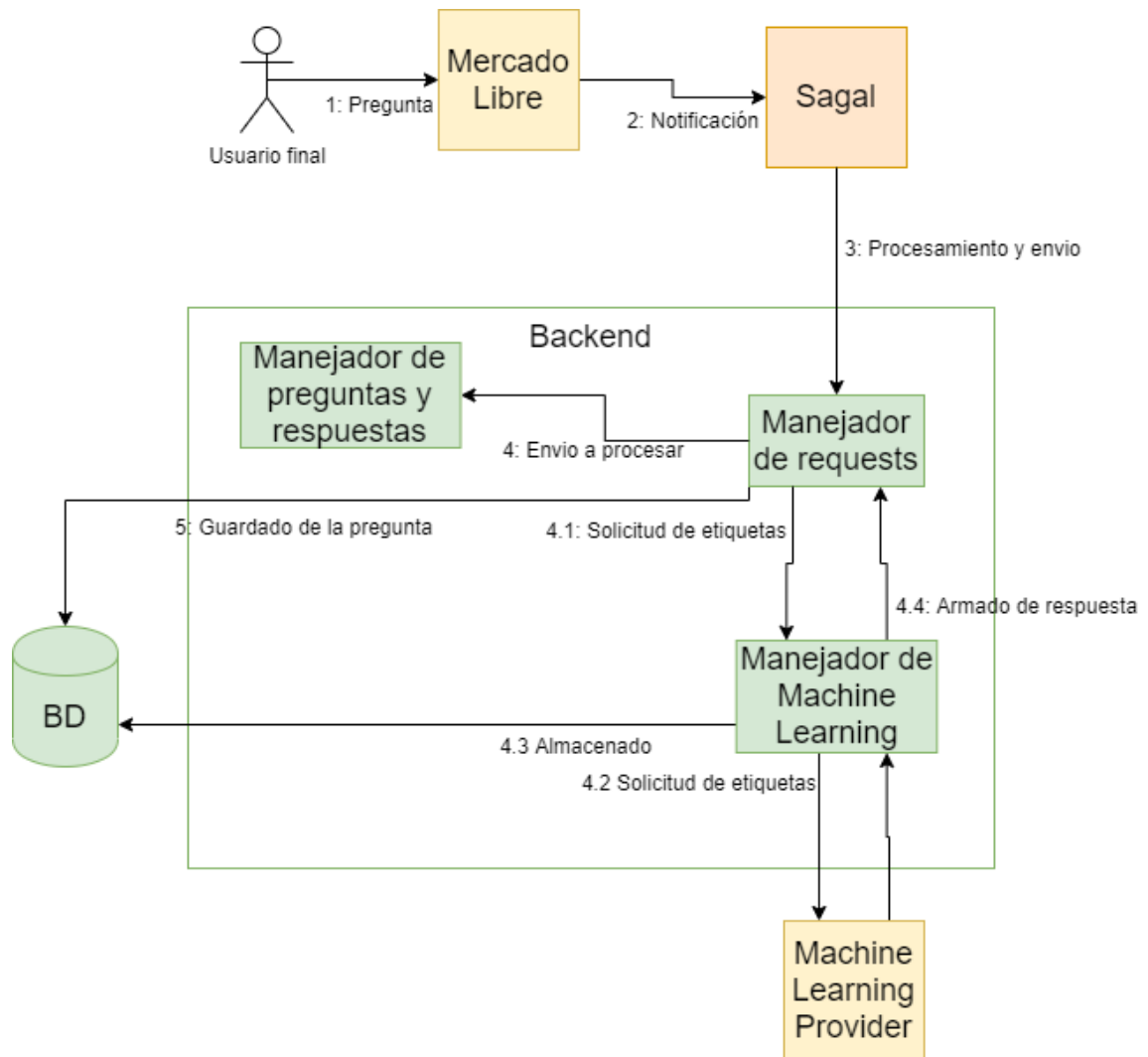


Ilustración 3.5: diagrama de comunicación

A continuación, detallamos los pasos cuando llega una pregunta nueva sin responder desde Sagal:

1. Usuario final hace una consulta sobre un artículo publicado en Mercado Libre de Empresa X.
2. Mercado Libre envía una notificación sobre esa pregunta a Sagal.
3. Sagal realiza su procesamiento interno para procesar esa notificación a una estructura entendible y manda una notificación a GPRI con esa información.

4. GPRI recibe la notificación en su manejador de *requests* que lo deriva al manejador de preguntas y respuestas. Se evalúa la pregunta recibida de Sagal ya que la misma puede estar respondida, eliminada o no contestada y el flujo puede variar dependiendo del estado.
 - 4.1. Si se detecta que es una pregunta nueva para ser respondida, se envía la pregunta al manejador de *machine learning*.
 - 4.2. El manejador hace la solicitud al *machine learning provider*.
 - 4.3. Se almacena el resultado de las etiquetas en la base de datos y se almacenan las etiquetas en la pregunta.
 - 4.4. El manejador de *machine learning* arma la respuesta automática basándose en los *templates*.
5. La pregunta con toda la información provista por Sagal, más los tags de *machine learning* y su respectiva respuesta automática, es almacenada en la base de datos.

Vista de despliegue

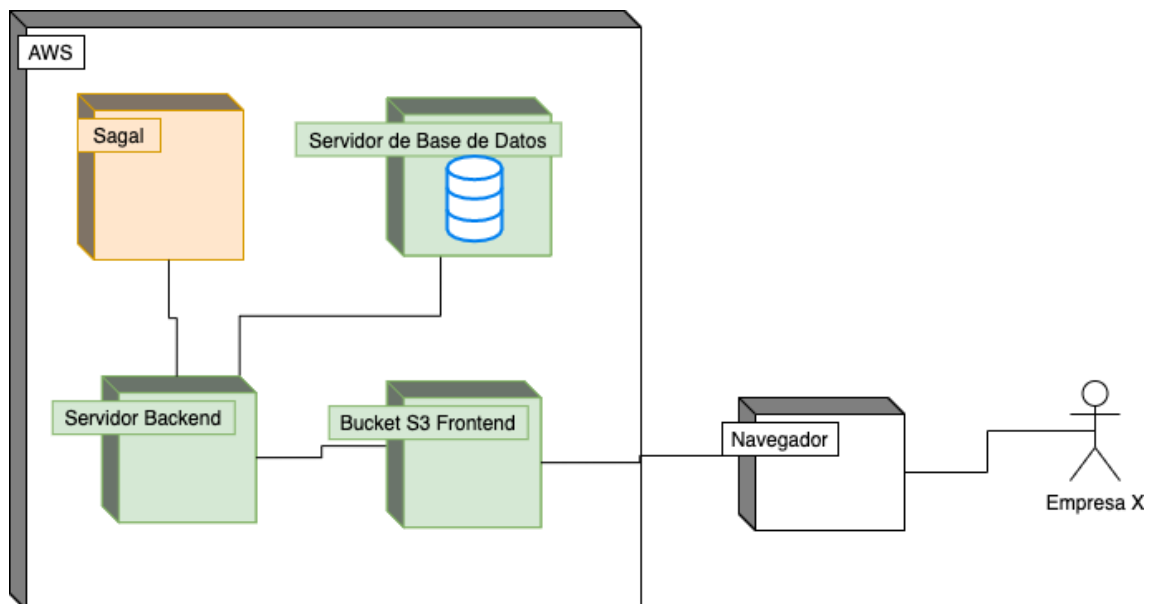


Ilustración 3.6: diagrama de despliegue

Contamos con los siguientes componentes de despliegue contenidos dentro de AWS, en diferentes instancias, que son las siguientes:

- **Sagal:** componente del cliente que corre dentro de una instancia de AWS-EC2.
- **Base de datos:** componente que corre dentro de una instancia de AWS por parte de MongoDB Atlas.
- **Frontend:** el mismo se encuentra alojado en una instancia de AWS-S3[17] como página estática brindado mediante AWS-CloudFront[33].
- **Backend:** el mismo corre dentro de una instancia de AWS-EC2[34].

3.6. Decisiones de diseño

Usabilidad

Empezaremos viendo la usabilidad, ya que fue identificado por Sagal como clave en la resolución del problema. Para tomar esto en cuenta en el diseño de la solución, fue clave la elección de React para desarrollar el *frontend*, ya que la flexibilidad para incorporar diferentes elementos visuales y agregar funcionalidades es fundamental. Como veremos en los capítulos siguientes, la usabilidad impactó en nuestras definiciones del proceso de desarrollo, teniendo que agregar validaciones con Empresa X y Sagal.

Modificabilidad

Para cubrir este punto, usamos un *linter*[35] y la configuración propuestos por el cliente. Esto también afectó nuestro proceso de desarrollo, teniendo que incorporar revisiones de código de parte de Sagal, así como también la posibilidad de intercambiar la interfaz de *machine learning* por si luego se decide cambiar IBM Watson.

Performance

La elección de la tecnología usada en el *backend* se vio directamente influida por este punto, como mencionamos anteriormente. A su vez, los diferentes manejadores se

crearon pensando en los impactos del rendimiento que podían tener los diferentes pedidos.

La elección de la base de datos también tuvo esto en cuenta.

Portabilidad

Para tomar en cuenta este requerimiento, y buscando mantenernos dentro del alcance del proyecto, optamos por hacer que el sistema sea una PWA (*Progressive Web Application*) para poder ser usado así, de forma “nativa” dentro de los diferentes dispositivos móviles requeridos como se comentó anteriormente.

Interoperabilidad con las bases de datos

Para este punto el uso de Mongoose[28] como capa que separa el dominio de la gestión de la base de datos fue una decisión clave.

Confiabilidad

Este requerimiento se ve reflejado en la elección de diferentes herramientas para las pruebas automáticas y de carga sobre el sistema, que nos garantizaron la confiabilidad. A su vez, tener separado el *backend* y el *frontend* nos permitió hacer pruebas individuales sobre los diferentes componentes.

4. Metodología

En este capítulo profundizaremos sobre las características de este proyecto y como eso impactó en la elección de la metodología. También veremos qué personalizaciones le agregamos a las metodologías elegidas y su fundamento. Nuestra elección de metodologías se dio en dos niveles, una para el proceso de gestión y otra para el proceso de desarrollo.

Las características consideradas fueron:

Poca incertidumbre en los requerimientos

Empezando por la definición de los requerimientos, es importante destacar que esta solución ya existía en el sistema de Sagal. Por lo tanto, si bien el cliente agregó algunos requerimientos como mejora dentro de la solución, no había incertidumbre en los mismos. Además, esta solución ya existe en otros competidores y en otras plataformas a nivel mundial[36].

Conocimiento del equipo

Nos conocíamos como equipo, pero nunca trabajamos juntos en un proyecto de nivel profesional como este. Por lo que nos conocíamos a nivel personal y sabíamos el compromiso profesional que tiene cada uno.

Si bien como equipo trabajamos en algunos proyectos académicos juntos, no conocíamos la productividad para un desarrollo de este alcance. Para apalancarnos en esto, buscamos potenciar las habilidades de cada miembro y generar especificidad en las tareas a realizar.

Distribución de esfuerzo no uniforme

Como los tres miembros trabajamos, nuestro esfuerzo iba a ser variable en el transcurso del proyecto, por lo que las metodologías deben ser capaces de considerarlo. De cualquier manera, nuestros horarios para trabajar en el proyecto coincidieron en su gran mayoría.

Alcance del producto

El cliente quería usar GPRI como base para las mejoras a futuro de su sistema central y su interfaz con los usuarios. Esto implicó una buena gestión de la documentación del proyecto y seguimiento de determinados estándares de codificación.

Niveles de calidad y confiabilidad altos

Este producto fue puesto en producción sustituyendo al sistema actual, por lo que requería altos niveles de calidad y confiabilidad.

Entregas incrementales

El cliente quería entregas incrementales del producto funcionando para poder evaluar cómo se iban desarrollando los cambios.

Poco conocimiento de las tecnologías

No teníamos conocimiento de algunas de las tecnologías a implementar para la solución, por lo que también podía verse impactado el diseño final de la solución.

Poco conocimiento de las metodologías

En nuestros trabajos actuales, ninguno de los tres miembros del equipo trabajó metodologías ágiles a nivel profesional, por lo que sería la primera vez que las aplicamos en un proyecto con este alcance, por eso forma parte de nuestros objetivos.

4.1. Metodologías utilizadas

Basándonos en las características del proyecto mencionadas anteriormente elegimos SCRUM[37] como metodología para la gestión y BDD (*Behavior driven development*)[38] para el desarrollo.

Para la elección de estas metodologías consideramos las siguientes características:

- SCRUM
 - Entregas incrementales.
 - Poco conocimiento de las tecnologías.
 - Poco conocimiento de las metodologías.
 - Conocimiento del equipo.
- BDD
 - Niveles de calidad y confiabilidad altos.
 - Poco conocimiento de las tecnologías.
 - Alcance del producto.

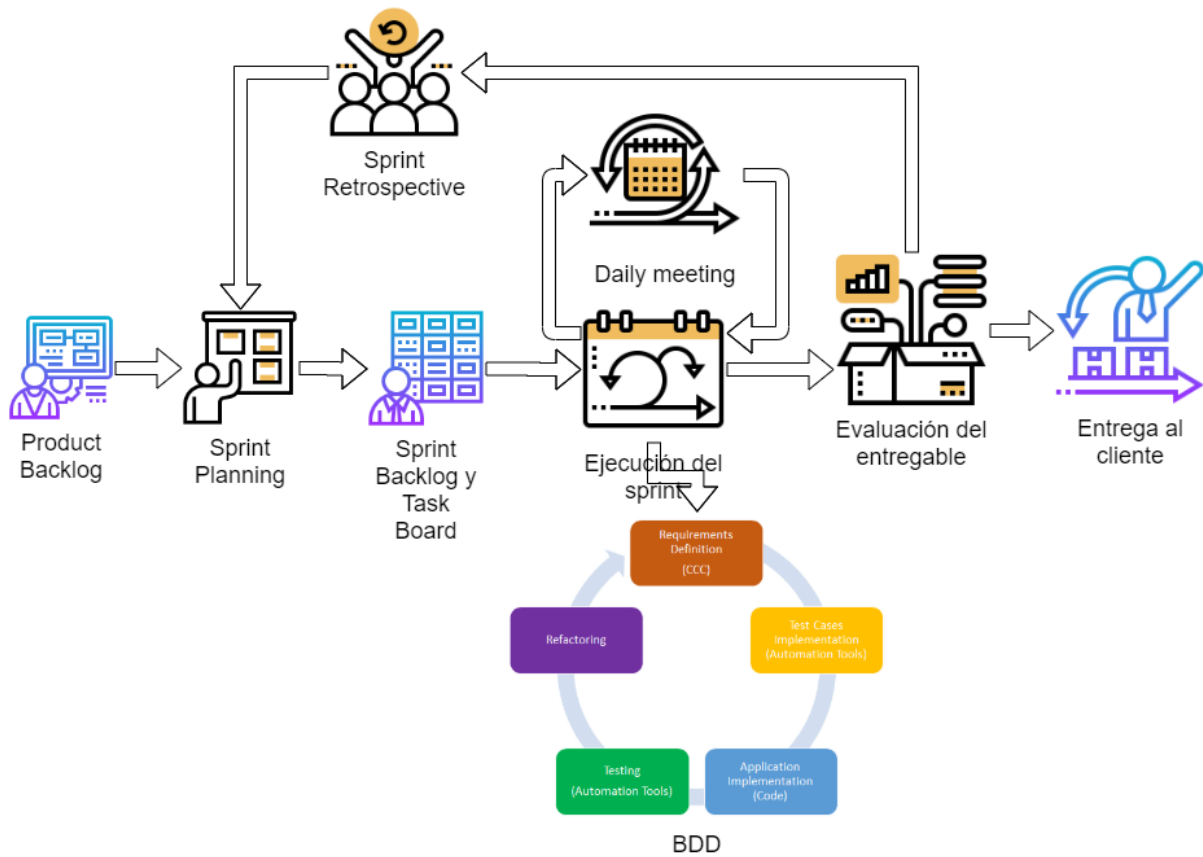


Ilustración 4.1: ciclo estándar de SCRUM + BDD.

BDD

A continuación, detallamos el ciclo de BDD que se muestran en la figura anterior:

- **Requirements Definition:** se definen las historias de usuario que siguen una narrativa y son especificadas por escenarios. Seguidamente mostramos la sintaxis para la narrativa y escenarios que sugiere BDD.

Narrativa:

Como [tipo de usuario]

Quiero [funcionalidad]

Para [beneficio].

Escenario [nombre del escenario]:

Dado [un contexto].

Cuando [evento o acción que se ejecuta].

Entonces [resultado].

- **Test cases implementation:** tomando como base los escenarios definidos se codifican los casos de prueba para el requerimiento.
- **Application implementation:** el proceso de desarrollo de la funcionalidad.
- **Testing:** ejecución de las pruebas automáticas diseñadas en el paso **test cases implementation**.
- **Refactoring:** se refactoriza el código para asegurar que sea más eficiente.

4.2. Personalización de SCRUM y BDD

Una vez que definimos las metodologías a usar para el proyecto identificamos algunas de las características de éste que nos impulsaron a hacer personalizaciones.

- Poca incertidumbre en los requerimientos. (SCRUM y BDD)
- Distribución de esfuerzo no uniforme. (SCRUM)

- Poco conocimiento de las tecnologías. (SCRUM, BDD)
- Poco conocimiento de las metodologías. (SCRUM)

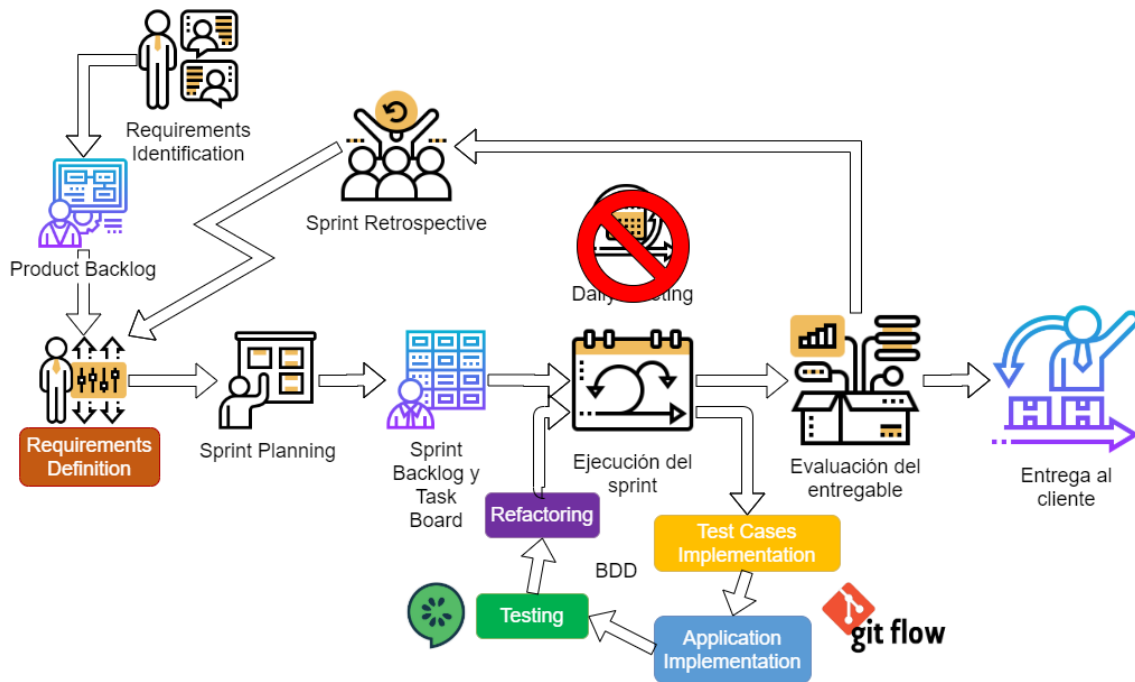


Ilustración 4.2: ciclo personalizado de SCRUM + BDD.

En la Ilustración 4.2 podemos ver como cambiamos la estructura para que se adaptara a nuestras necesidades en el proyecto.

Identificación de Requerimientos (*Requirements Identification*)

Como una de las características era que los requerimientos presentaban poca incertidumbre, esto nos permitió realizar la definición de requerimientos antes de entrar al ciclo de SCRUM, permitiéndonos aplicar un ciclo de vida incremental. Optamos, a su vez, por dividir la definición de requerimientos de BDD en dos partes. La primera, denominada identificación de requerimientos (véase la actividad de *Requirements Identificacion* en la Ilustración 4.2), consistió en crear una lista de historias de usuario que representaban las funcionalidades que debían ser implementadas. De esta manera creamos una primera versión del *product backlog*.

Definición de Requerimientos (*Requirements Definition*)

La segunda etapa fue la definición de requerimientos (Véase la actividad de *Requirements Definition* en la Ilustración 4.2), que consistía en especificar los escenarios de cada historia de usuario, según sugiere BDD. Esto nos permitió crear la segunda versión del *product backlog*.

Daily meeting

Otra de las personalizaciones realizadas fue no llevar a cabo la *Daily meeting*, ya que, en la mayoría de los días, estábamos trabajando sobre la misma hora en el proyecto estando conectados mediante una llamada en conferencia, por lo que no veíamos ninguna utilidad a esta ceremonia. Las características que mencionamos anteriormente que nos llevaron a tomar esta decisión fueron:

- Conocimiento del equipo
- Distribución del esfuerzo no uniforme

Roles

Tomando en cuenta las características de “conocimiento del equipo” y “poco conocimiento de las metodologías”, optamos por, sumados a los roles de SCRUM, agregar algunos, buscando generar un nivel de especialización dentro del proyecto, de manera de poder centralizar la toma de decisiones en los diferentes enfoques del mismo.

- **Gerente del proyecto:** responsable de mantener toda la información sobre la gestión del proyecto actualizada y asegurarse de tener los materiales necesarios para cumplir con las entregas de evaluación académicas. También es el responsable de controlar la aplicación de la metodología.
- **Backend engineer:** responsable de la arquitectura del *backend*, así como asegurarse que se respete y se sigan los estándares utilizados. Es el referente a la hora de medir el impacto de los cambios en este componente de la solución.

- **Frontend engineer:** responsable de la arquitectura del *frontend* y de asegurarse que se siguen los estándares de codificación propuestos por el cliente. Referente de medición de impacto en los cambios de este componente.

5. Proyecto

En este capítulo mostraremos como se instancio el proceso que describimos en el capítulo anterior. También incluiremos en más detalle cómo se materializaron las personalizaciones a las diferentes metodologías elegidas.

Entraremos en detalle de la planificación del proyecto, el alcance propuesto para los diferentes entregables, como hicimos el seguimiento de los diferentes *sprints* y qué resultados obtuvimos de cada uno de ellos.

5.1. Planificación

En este proyecto trabajamos con *releases feature driven*[\[39\]](#) y como mencionamos en el capítulo 1.2: Objetivos, debíamos entregar dos MVP. Sobre el final del proyecto, se dieron inconvenientes que demoraron la salida del primer MVP, por lo que eso recortó el alcance de algunos requerimientos quedando planificados para luego de la entrega y evaluación académica. Esto se verá en detalle en el capítulo 10: Continuidad del proyecto.

Como punto central de la planificación, están la entrega de los MVP, que debían como mínimo cumplir con las siguientes funcionalidades:

MVP 1: el objetivo de esta entrega era cubrir las funcionalidades que tiene el sistema de Sagal actualmente. Esto implica, visualización y respuesta de las preguntas pendientes, historial de las mismas y la posibilidad de buscar dentro de las preguntas aplicando filtros. Esto permite sustituir un sistema por otro.

MVP 2: en este entregable se agregarían el etiquetado por *machine learning*, que la aplicación sea *responsive* y que sea una PWA (*Progressive Web Application*)[\[29\]](#).

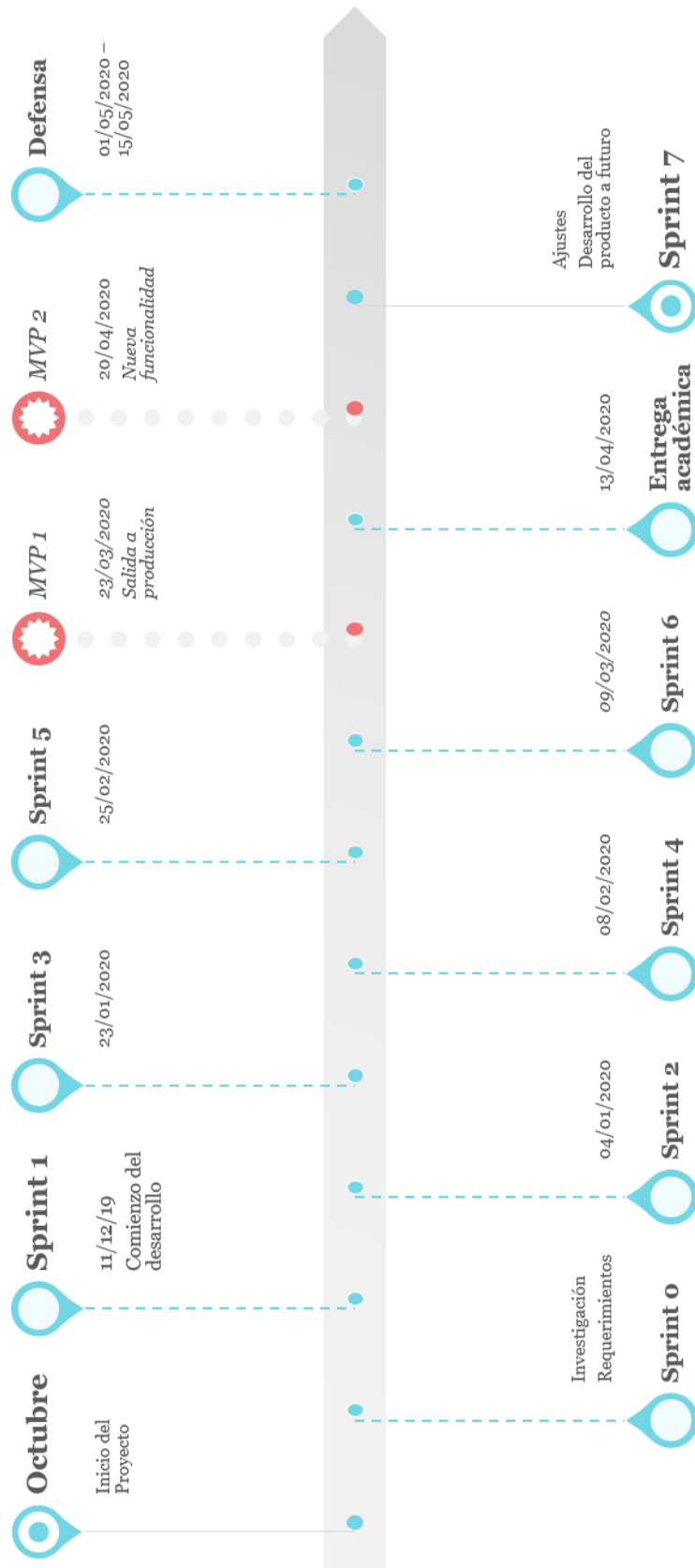


Ilustración 5.1: planificación.

Sprint 0

Dentro de nuestro proceso de metodología previsto, el relevamiento de requerimientos se dio por fuera de los ciclos de desarrollo. Todo el proceso de relevamiento, así como la definición de las tecnologías, se dio en un período de tiempo previo a la ejecución de los *sprints*, al que llamamos *Sprint 0*. En este período también incluimos todas las tareas de investigación de las tecnologías a utilizar, así como breves pruebas de concepto en la utilización conjunta de las mismas, para asegurar la viabilidad de las tecnologías que se habían planteado. A su vez, manejamos a alto nivel el tipo de arquitectura que pensábamos utilizar para la solución.

Definición de roles

Dentro del proyecto, definimos ciertos roles buscando potenciar las habilidades de los miembros del equipo y generar especificidad en ciertas tareas. SCRUM no prevé roles para dividir las tareas, pero los dividimos de la siguiente forma:

- **Gerente del proyecto:** Nicolás González
- **Backend Engeneer:** Renzo Delfino
- **Frontend Engeneer:** Nicolas Caballero

Dentro de los roles de SCRUM los asignamos de la siguiente forma:

- **Scrum Master:** Nicolás González
- **Product Owner:** Alan Bukstein
- **Development team:** Nicolás González, Renzo Delfino, Nicolás Caballero.

5.2. Ejecución del proceso de desarrollo

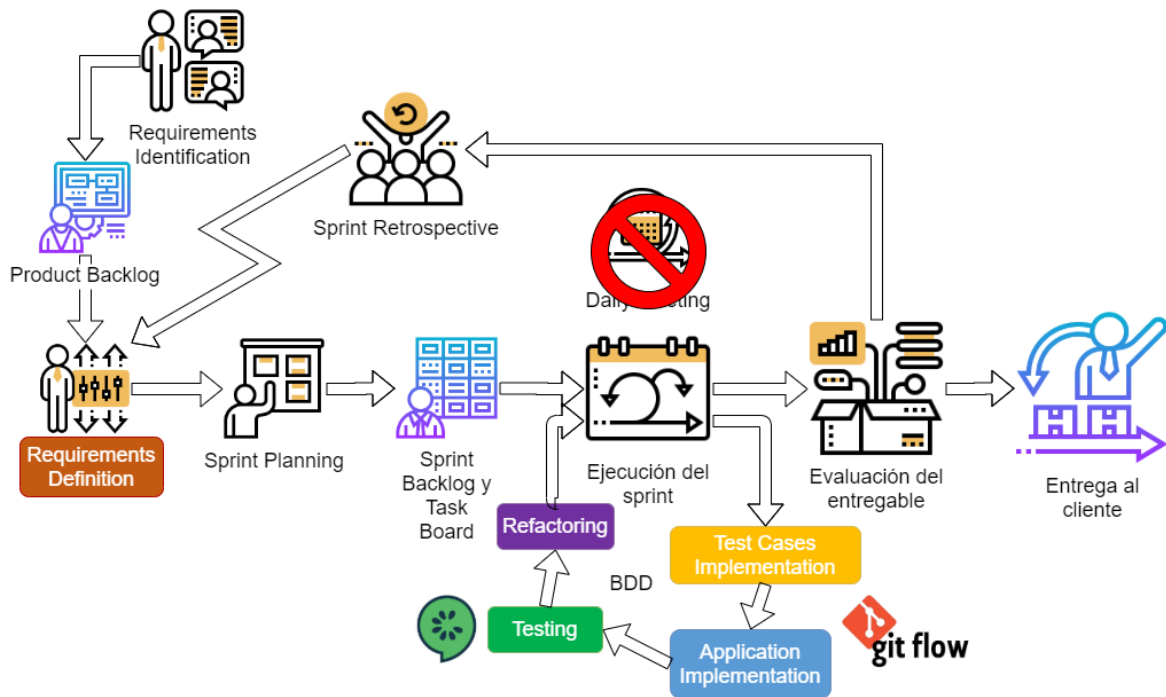


Ilustración 5.2: proceso personalizado de SCRUM + BDD.

Agregamos nuevamente la imagen de nuestra metodología definida para que sea fácil la identificación de las diferentes ceremonias y procesos que detallaremos.

Definición de requerimientos (*Requirements Identification*)

Dentro del *Sprint 0* que mencionamos anteriormente, empezamos a tener reuniones semanales con Sagal para discutir los requerimientos necesarios e ir afinando el alcance del proyecto. Estas reuniones se dieron entre los miembros del equipo y el *Product Owner*. Definimos cuales serían las responsabilidades y tareas pendientes desde el lado de Sagal para resolver la comunicación entre GPRI y la actual aplicación de Sagal.

A partir de estas reuniones fuimos generando el *Product Backlog*, escribiendo a alto nivel las historias de usuario y validándolas con el *Product Owner*. Estas historias se escribieron siguiendo la metodología explicada en el capítulo anterior, con la

particularidad de que, en este punto, solo consistieron en la narrativa propuesta por BDD.

Mockups

Dentro de este proceso, generamos *mockups* (véase anexo **12.10: Mockups**) para validar cómo veíamos implementadas las diferentes funcionalidades del sistema y como se vería eso en el uso por Empresa X. Una vez validados los diferentes *mockups* por Sagal se agendó una reunión con una de las Empresas X que van a operar con el sistema. Dentro de esta muestra se validaron los *mockups* y las diferentes historias de usuario trabajadas hasta el momento. Estos mockups se pueden ver en las Ilustraciones 5.3 y 5.4.

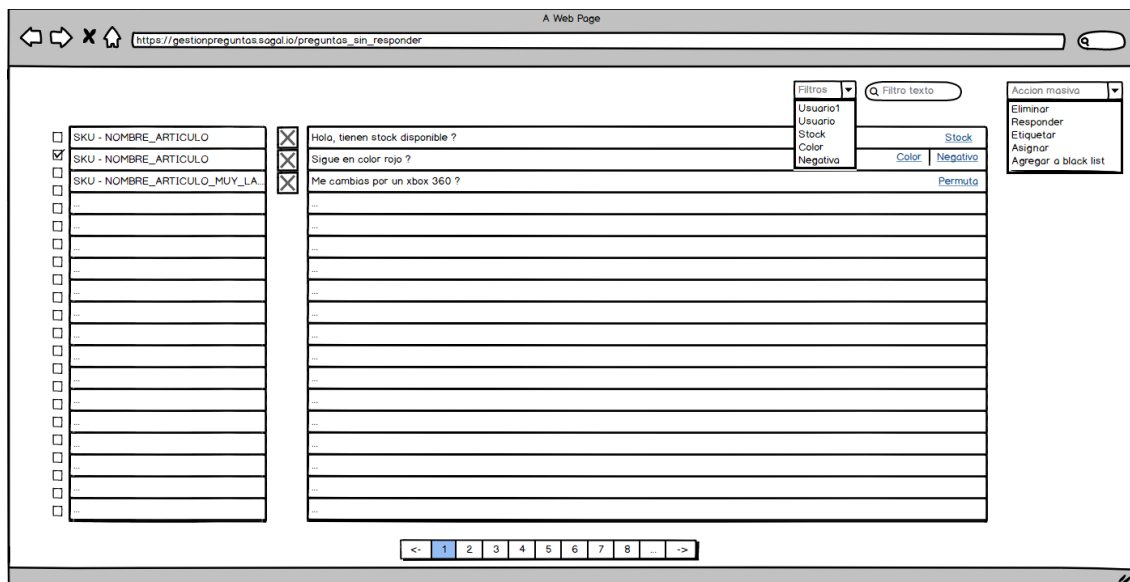


Ilustración 5.3: *mockup* de *frontend* para web.

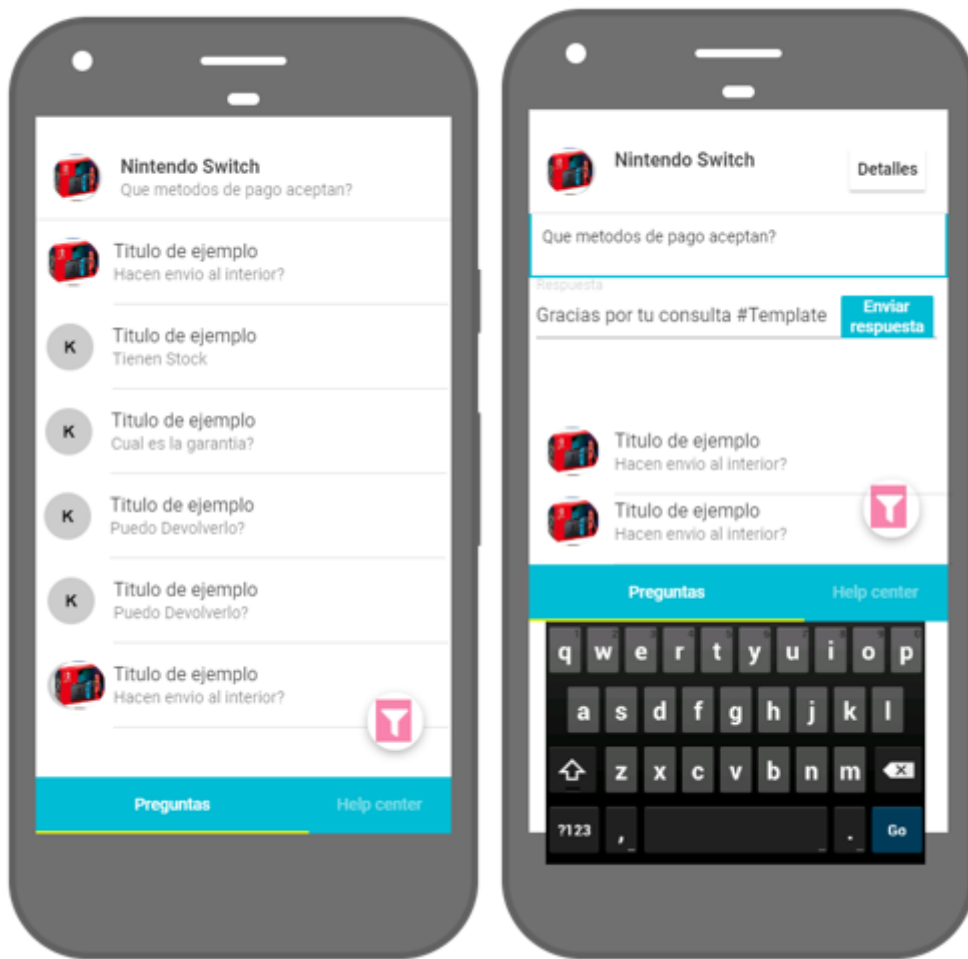


Ilustración 5.4: *mockups* versión móvil.

Definición de requerimientos (*Requirements Definition*)

En este punto, tomábamos los escenarios definidos en el paso anterior y los traducíamos a Gherkin[40], un lenguaje diseñado para manejar escenarios, interpretarlos y permitir ejecutarlos desde el código.

Dejamos a modo de ejemplo algunos escenarios:

```

Feature: As a user i want to filter questions so i find the most relevance

Scenario: Filter by article code
  Given i have 10 questions with article code 123 and 10 questions with article code 456
  When i get questions filter by article code 123
  Then i see the 10 questions with article code 123

Scenario: Filter by article title
  Given i have 10 questions with article title Artículo Test and 10 questions with article title Otro
  When i get questions filter by article title Artículo Test
  Then i see the 10 questions with article title Artículo Test

Scenario: Filter by question customer
  Given i have 10 questions with customer name Pepe and 10 questions with customer name Jose
  When i get questions filter by customer name Pepe
  Then i see the 10 questions with customer name Pepe

```

Ilustración 5.5: ejemplo de Gherkin

Para más detalle de los escenarios utilizados se pueden ver en **12.3: User Stories**.

Implementación de casos de prueba (*Test cases implementation*)

Tomando como entrada el código de Gherkin, se implementaban los casos de prueba que referencian a dicho escenario, de manera de que, más adelante en el proceso, se puedan ejecutar automáticamente dichos casos.

```

test('Create existing template', ({given, when, then}) =>{
  let template;

  given('i have one template with stock name', async () => {
    template = generateTemplate();
    response = await request(fastify.server)
      .post('/api/templates')
      .send(template)
      .set('Authorization', 'bearer ' + validToken);
  });

  when('i create one template with stock name', async ()=> {
    response = await request(fastify.server)
      .post('/api/templates')
      .send(template)
      .set('Authorization', 'bearer ' + validToken);
  });

  then('returns template already exists error and template is not created', async ()=>{
    expect(response.statusCode).toEqual(400);
    expect(response.body.message).toBe("Template with name stock already exists");
  });
});

```

Ilustración 5.6: ejemplo de caso de prueba.

Desarrollo (*Application implementation*)

Una vez escritos en Gherkin, pasábamos a la etapa del desarrollo de la(s) historias de usuario(s) correspondientes.

Testing

Una vez terminada la etapa de *Application Implementation*, se ejecutaban los casos de prueba automatizados en los pasos anteriores. Esto nos sirvió como una base de confianza fuerte para poder continuar con el desarrollo de las funcionalidades. Podemos ver una ejecución de los *tests* en la Ilustración 5.7.

```
PS D:\Repositorios\backend> npm test
> sagal-questions@1.0.0 test D:\Repositorios\backend
> jest step_definitions

PASS src/__tests__/step_definitions/template.creation.steps.js (15.51s)
PASS src/__tests__/step_definitions/question.visualization.steps.js (15.403s)
PASS src/__tests__/step_definitions/template.update.steps.js (15.574s)
PASS src/__tests__/step_definitions/question.answer.steps.js (15.524s)
PASS src/__tests__/step_definitions/question.history.steps.js (15.365s)
PASS src/__tests__/step_definitions/question.pagination.steps.js (16.759s)
PASS src/__tests__/step_definitions/question.filter.steps.js (17.337s)
PASS src/__tests__/step_definitions/question.details.steps.js
PASS src/__tests__/step_definitions/template.visualization.steps.js
PASS src/__tests__/step_definitions/template.delete.steps.js
PASS src/__tests__/step_definitions/template.pagination.steps.js

Test Suites: 11 passed, 11 total
Tests:       39 passed, 39 total
Snapshots:  0 total
Time:        20.72s
Ran all test suites matching /step_definitions/i.
PS D:\Repositorios\backend>
```

Ilustración 5.7: ejemplo de corrida de *tests* en *backend*.

Como parte del proceso de *testing*, sumamos las siguientes actividades *pre-release* que nos permitían medir la respuesta del sistema para ver como nuestros cambios impactaban en el rendimiento del sistema. Véase anexo **12.11: Reporte *tests*** para visualizar el resultado final de los *tests*, tanto en *frontend* como *backend*.

Pruebas de concurrencia/sobrecarga: decidimos realizar una prueba de concurrencia contra el *backend* para ver cómo responde cuando está recibiendo *requests* complejas desde 50 usuarios concurrentes. En particular, obtener las preguntas de la página 1 del historial de preguntas. Como se puede ver en la Ilustración 5.8 y en el anexo **12.12: Reporte pruebas de concurrencia y rendimiento**.

```
{
  "totalRequests": 5000,
  "totalErrors": 0,
  "totalTimeSeconds": 117.99262780000001,
  "rps": 42,
  "meanLatencyMs": 1174.9,
  "maxLatencyMs": 1968,
  "minLatencyMs": 440,
  "percentilesMs": {
    "50": 1166,
    "90": 1451,
    "95": 1527,
    "99": 1693
  }
}
```

Ilustración 5.8: resultado de prueba de concurrencia.

Esta prueba se ejecutaba al final del *sprint*, de manera de asegurarnos que no se viera impactado el rendimiento de la aplicación. En caso de que hubiera algún impacto se priorizaba para el siguiente *sprint*.

Refactoring

En caso de que los *tests* no pasaran, se hacía un *refactor* en el código fuente de forma que se pueda mejorar la calidad y la eficiencia de éste.

Consideramos como *refactor* a todos los cambios o ajustes de las funciones o métodos que tuvieran como objetivo hacer más eficiente la funcionalidad existente.

Nuestro objetivo en este caso era asegurarnos de no hacer *over-engineer* y evitar que el *refactor* incluyera grandes niveles de rediseño de la solución.

Como mencionamos previamente, las historias de usuario dividían sus tareas en *Front y Back*, por lo cual ese desarrollo se hacía en simultáneo, lo que generó muy poco trabajo de integración entre ambos componentes. En caso de que existiera se realizó en esta etapa.

Pull requests

Una vez que la historia de usuario se movía a *done*, el desarrollador que había terminado con la funcionalidad levantaba una *pull request*. Una *pull request*, es la manera de subir los nuevos cambios al código y unificarlos con la versión actual del sistema. Una vez que la *pull request* estaba subida, se agregaba a los miembros del equipo para que hicieran una revisión del código y ejecutaran los *tests* automáticos sobre el mismo. En caso de que pasaran correctamente se unía el código a la rama principal. Se puede ver un ejemplo en la Ilustración 5.9.

En uno de los requerimientos mencionados en el capítulo anterior, el cliente quiere utilizar nuestro proyecto como base de crecimiento futuro, por lo tanto, designó a un responsable de su lado para hacer *code review* de las *pull requests* sobre el *frontend*. De esta manera, el cliente se aseguraba que estuviéramos cumpliendo con los estándares establecidos. Acordamos con él que, en caso de tener comentarios o cambios sugeridos, se incorporarían para la próxima *pull request* y no se harían cambios sobre la *pull request* actual. De esta manera no se frenaba el desarrollo y no le exigíamos al cliente mantener el ritmo de aprobación de las *mismas*.

Una vez que la *pull request* estuviera aprobada y unificada en la rama correspondiente se daba por terminada la historia de usuario. Agregamos un ejemplo de este proceso de unificación de las ramas, basado en GitFlow[41] en la Ilustración 5.9 y un ejemplo de la *pull request* aprobada y *mergeada* en la Ilustración 5.10.

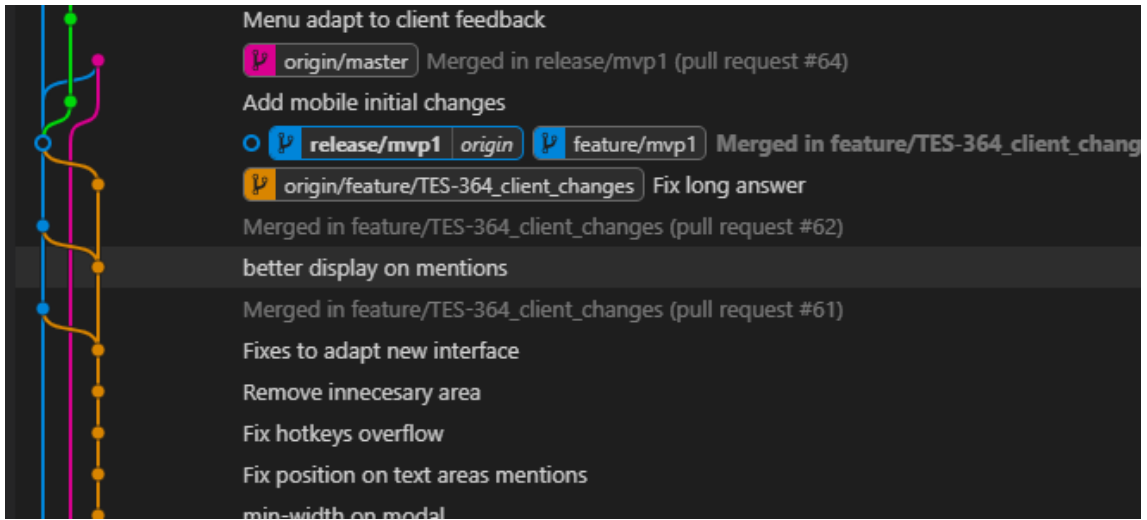


Ilustración 5.9: Gitflow visualización en Sourcetree

Feature/TES-13 filters

feature/tes-13-filters → develop **MERGED**
Created 2020-01-30 · Last updated 2020-02-02

Unapprove ...

Merged pull request
Merged in feature/tes-13-filters (pull request #38)
a559091 · Nicolás Caballero · 2020-01-30

- added filter tests
- Refactor api mocks
- Fixes
- add texts
- move to constants.js

16 files

src/	__tests__		
filter.history.feature		+4	-24
filter.question.feature		+12	-12
mocks			
api			
constants.js		+7	
helpers.js		+38	
index.js		+57	
axios.js		-48	
constants.js		-3	
questions.js		+32	-18
step_definitions			
filter.history.js		+49	
filter.question.js		+128	
history.pagination.js		+2	-2
history.visualization.js		+2	-2
question.answer.js		+2	-2
question.details.js		+7	-8
question.pagination.js		+2	-2
question.visualizatio...		+2	-2

Ilustración 5.10: ejemplo de *pull request*.

5.3. Ejecución del proceso de gestión

En esta sección detallaremos todas las ceremonias usadas para la gestión del proyecto, junto con sus salidas.

Sprint planning meeting

Como puntapié inicial de la ceremonia, estimábamos las métricas definidas para nuestra gestión del proyecto, se profundizarán sobre éstas en el subcapítulo 5.5: Métricas.

Dentro de esta ceremonia, tomábamos las historias de usuario generadas en el paso *Requirements Definition*, y usando la técnica de *affinity estimation*[\[42\]](#), estimábamos el tamaño de cada historia en *story points*. Luego creábamos las tareas asociadas a esa historia y estimábamos las horas para cada tarea. En caso de que una historia de usuario tuviera impacto sobre *frontend* y *backend*, lo tomábamos en cuenta dentro de la creación de las tareas, diferenciándolas como Front y Back en su nombre. Esta separación nos facilitó la asignación de las mismas y en caso de tener bloqueos podíamos identificar rápidamente en qué componente teníamos cuellos de botella. Esto también se asociaba a los roles que teníamos definidos previamente.

El cliente nos indicó que no tenía interés en participar de los *sprint plannings* para la selección de estas historias, por lo que sólo hacía una priorización de las historias a nivel de *release*. Antes de dar comienzo a cada *sprint*, seleccionábamos qué historias serían parte de nuestro *sprint backlog*, elegíamos las historias basándonos en las prioridades de *release* propuestas por el cliente y las estimaciones en *story points*.

Como revisión final nos asegurábamos de que las historias elegidas, sus *story points* y las horas de trabajo estimadas fueran razonables para la productividad que se tenía en el equipo. Los primeros *sprints* fueron de medición y conocimiento de la velocidad del equipo por lo que fue esperado no tener un estimado correcto para las historias de usuario. Además, hacíamos un análisis de los posibles riesgos involucrados en cada *sprint*, los documentábamos y pensábamos posibles

mitigaciones para cada uno. Revisábamos si los riesgos de los *sprints* anteriores se habían materializado o habían sido mitigados correctamente, sobre estos puntos profundizaremos en el subcapítulo siguiente 5.4: Gestión de riesgo.

A continuación, mostramos la información que registramos para la definición de las tareas de usuario:

Título: seguimos el formato propuesto por BDD para las historias de usuario, Como [rol], quiero poder [funcionalidad] para [beneficio]. En caso de ser una tarea o un bug, era una breve descripción del mismo.

Tipo: determina el tipo de evento (historia, tarea, bug).

Estado: el estado actual (*to do, in progress, done*).

Tags: etiqueta para definir el tipo de actividad y poder obtener análisis basándose en esto (Gestión, Desarrollo, *Testing*, Investigación, Retrabajo).

Versión afectada: Indica a que versión está asociada esta funcionalidad o tarea.

Story points: La cantidad de puntos de historia definidos por el equipo.

En la Ilustración 5.11 podemos ver como ejemplo una historia de usuario con sus subtareas (véase el anexo **12.3: *User Stories*** para profundizar sobre las mismas)

Como usuario quiero poder filtrar las preguntas para poder buscar las que son mas relevantes

Attach Create subtask Link issue ...

Description

Filtrado por SKU:

- Dado que quiero filtrar las preguntas.
- Cuando elijo el filtro de SKU.
- Entonces tengo ver solo las preguntas con el SKU elegido.

Filtrado por Titulo:

- Dado que quiero filtrar las preguntas.
- Cuando elijo el filtro de titulo.
- Entonces tengo ver solo las preguntas con el titulo elegido.

Subtasks

... +
100% Done

TES-131	Front - Escribir escenarios	↑		DONE
TES-132	Front - Probar escenarios	↑	NC	DONE
TES-133	Back - Escribir escenarios	↑	RD	DONE
TES-134	Back - Probar escenarios	↑	RD	DONE
TES-135	Front - Componente de combo box para seleccion...	↑	NC	DONE
TES-136	Front - input para escribir el filtro que se quiere	↑	NC	DONE

Ilustración 5.11: ejemplo de *user story*.

Sprint execution

En la ejecución del *sprint*, recorrimos el proceso de desarrollo definido en el subcapítulo anterior, a modo de repaso consistía en:

- *Test Cases Implementation*
- *Application Implementantion*
- *Testing*
- *Refactoring*

Para la asignación de tareas, seguimos como criterio tratar de mantener un miembro del equipo como “experto” en *backend* y otro en *frontend*, basándonos en los roles definidos previamente. De esta manera logramos cierto nivel de especialización que nos permitió avanzar rápidamente en el desarrollo de la solución. Preferimos realizarlo de esta manera, ya que siendo tres integrantes del equipo nos evitaba perder tiempo por el cambio de contexto.

El registro de horas de trabajo se hizo mediante JIRA[43], lo que nos permitió llevar un mejor control de las mismas y poder sacar reportes personalizados sobre la dedicación del trabajo.

En la Ilustración 5.12 mostramos el resultado de la ejecución del sprint 2 que nos brinda JIRA[43] y en el anexo **12.5: *Sprint reports***

Date	4 ene. 2020
Team	Tesis
Participants	@Nicolas Gonzalez @Nicolás Caballero @Renzo Delfino

Resultados y Productividad

	Sprint 1	Acumulado
Story Points	21	21
Horas Trabajadas	115	125
Productividad	0,18	0,168

Background

- Primer sprint
- Primera planning
- Cerrar el sprint, ver velocidad
- Redefinir testing, planning, scope y manejo de comentarios del cliente
- Reglas de trabajo y aseguramiento de la calidad

se encuentran todos

los reportes de los *sprints* ejecutados dentro del proyecto. Sus respectivos *burndowns charts* se encuentran en el anexo **12.4: Burndown**.

Sprint Report

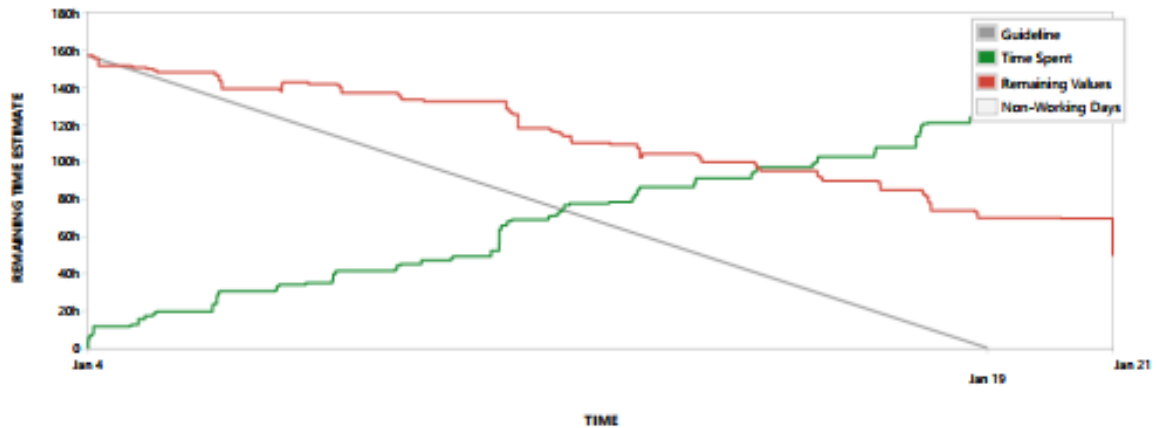


Sprint 2



Closed sprint, ended by Nicolas Gonzalez 04/Jan/20 6:00 PM - 21/Jan/20 8:01 PM [View linked pages](#)

Entregar el release



Status Report

Completed Issues

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (71)
TES-4	Como usuario al visualizar una pregunta quiero poder visualizar todos los datos relacionados para poder contestar fácilmente	Story	Highest	DONE	8
TES-8	Como usuario quiero poder tener un paginado de preguntas a responder para poder navegar entre ellas	Story	Medium	DONE	21
TES-9	Como usuario quiero poder visualizar todas las preguntas ya respondidas para tener un historial de las mismas ademas de un seguimiento	Story	Medium	DONE	8
TES-11	Como usuario quiero poder visualizar las preguntas mas antiguas sin contestar primero para responderlas lo antes posible	Story	Medium	DONE	2
TES-12	Como usuario quiero al contestar una pregunta que desaparezca para visualizar solo las no respondidas	Story	Medium	DONE	3
TES-14	Como usuario quiero poder contestar una pregunta para responderle a los clientes finales	Story	Highest	DONE	8
TES-B1	Preparar el primer release	Story	High	DONE	21

Issues Not Completed

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (16)
TES-13	Como usuario quiero poder filtrar las preguntas para poder buscar las que son mas relevantes	Story	Highest	TO DO	13
TES-87	Como usuario quiero poder comunicarme con el soporte tecnico	Story	Highest	TO DO	3

Issues Removed From Sprint

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (- - 8)
TES-42	Conexión e interfaz con Sagal	Story	High	TO DO	- - 8

Ilustración 5.12: *sprint 2 report*.

Sprint retrospective

En esta ceremonia evaluamos el sprint que se finalizó. Consistía en aplicar algunas técnicas que nos permitieron analizar como estábamos llevando adelante el proyecto como equipo, analizar cómo nos sentíamos y qué debíamos mejorar. De esta manera buscamos cómo podíamos ir mejorando el proceso a medida que transcurría el proyecto. Para esto, analizábamos las diferentes métricas que surgían de la aplicación de esta metodología, hacíamos un análisis de los riesgos involucrados en cada sprint, analizábamos como nos sentíamos cada uno y qué podíamos mejorar para mejorar el proyecto.

Usamos las siguientes técnicas para evaluar el equipo:

- ***Start, stop, keep***: Consiste en charlar como grupo qué actividades deberíamos dejar de hacer, cuáles deberíamos de comenzar a hacer y cuáles tenemos que continuar haciendo. Nos sirvió para identificar algunos puntos de mejora como equipo y cómo deberíamos encarar determinados problemas. (véase anexo **12.6: *Sprint retrospectives***)
- ***Positive & true***: Consiste en preguntarse en ronda, alguna pregunta que este pensada para tener una respuesta positiva y verdadera sobre la experiencia. Por ejemplo, ¿Qué hiciste bien en la última iteración? De esta forma se logra mantener al equipo positivo.

Nos basamos en las técnicas propuestas por *Retromat Blog*[\[44\]](#).

De este proceso pudimos obtener varias mejoras en nuestro proceso de trabajo, así como ver que inconvenientes afectaban al trabajo en equipo, como ejemplo podemos ver la Ilustración 5.13.

Retrospective

Start doing	Stop doing	Keep doing
<ul style="list-style-type: none">• Monitorear al beta tester	<ul style="list-style-type: none">• Esperar que el cliente cumpla con el compromiso	<ul style="list-style-type: none">• Mantener el nivel de compromiso•

Positive & True

Nicolas C

- Mitigar la puesta en producción
- Contemplar las horas del examen
- Ayudar al equipo

Renzo

- Organizar las reuniones con el cliente para resolver problemas
- Dedicación de horas

Nicolas G

- Dedicarle horas a la presentación
- Segunda revisión
- Presión al cliente

Ilustración 5.13: ejemplo de *retrospective* en *sprint 5*.

Sprint review

En esta ceremonia el objetivo es evaluar el estado del producto a entregar basado en los requerimientos que se establecieron en el *sprint backlog*. Debía realizarse con el cliente, pero a pedido del mismo la realizábamos cada dos *sprints*. Esto consistía en tomar los escenarios generados en la etapa de *Requirements Definition* y que se siguieran para confirmar que el producto los cumple. En los *sprints* que no se realizaron las *reviews* con el cliente, la hicimos nosotros mismos con algún miembro del equipo que no participó mayoritariamente en el desarrollo de ese requerimiento. En caso de encontrar algún bug o comportamiento no esperado, lo registrábamos para solucionarlo en el sprint siguiente.

5.4. Gestión de riesgo

Dentro de la gestión de riesgo tuvimos varios puntos que nos parecen importantes resaltar. Cuando comenzamos con el proyecto no le dimos importancia, realizando

una gestión “*ad hoc*” sin detallar ni evaluar los riesgos de ninguna manera en particular. Se generaban de forma intuitiva tareas de mitigación basándonos en sobreestimación o tareas de investigación para determinadas tecnologías. Esto ocurrió en los *sprints* del 1 al 3.

Dentro de este manejo de riesgos podemos detallar algunas de las tareas que se generaron y en retrospectiva cuáles consideramos como riesgos:

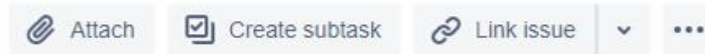
- Implementación de PWA (*Progressive Web Application*) para la aplicación
 - Se realizó como parte de la investigación del sprint 0
- Uso de Watson como plataforma de *machine learning*
 - Sobreestimación de la tarea de implementación
- Entrenamiento del modelo predictivo
 - Sobreestimación para la categorización de preguntas

A partir del sprint 4, viendo que se estaban generando dificultades en el avance del proyecto que podían constituirse en problemas, agregamos como parte de nuestra *Sprint Planning* la identificación de estos riesgos. Nuestra estrategia de gestión de riesgos fue evaluarlos por sprint, lo que nos daba mayor flexibilidad, debido a que siempre eran como máximo tres o cuatro riesgos por lo que no había necesidad de priorizarlos, esto es, conocer el impacto y la probabilidad. Además, como ventaja, podíamos planificar la mitigación del riesgo teniendo toda la información necesaria para hacerlo.

Si lo contrastamos con las metodologías tradicionales, en las cuales se planifican todos los riesgos posibles del proyecto y se miden los impactos y probabilidades de cada uno para definir la estrategia de mitigación, nos hubiera insumido un tiempo considerable del proyecto, y en caso de que cambiaran las condiciones iniciales en las que se evaluaron dichos riesgos habría que volver a replanificarlos. Como estábamos aplicando una metodología ágil donde ejecutamos todo el proceso de desarrollo en cada *sprint*, nos permitió ahorrarnos tiempo en la gestión de estos riesgos y asegurarnos de tener toda la información necesaria a la hora de planificar y mitigar los mismos.

Para resaltar, tuvimos un riesgo que se convirtió en problema, que nos insumió horas de trabajo de parte del equipo para compensar por la mala gestión del mismo. Esto terminó impactando en el alcance de los *sprints* finales. En la Ilustración 5.14 podemos ver la tarea y el tiempo que le llevó al equipo.

Generar interfaces para conexión con Sagal



Description

Dado que el riesgo se convirtió en un problema, estimamos hacer nosotros mismos las interfaces para poder hacer funcionar el MVP y que se comience a usar.

Time tracking



Ilustración 5.14: tarea para mitigar el riesgo de no puesta en producción.

El detalle del documento con los riesgos se puede ver en el anexo **12.16: Reporte de riesgos**.

Finalmente, los riesgos eran evaluados en la *sprint retrospective*, para determinar si la mitigación había sido adecuada o si necesitaba adecuación, o si el riesgo se había eliminado.

5.5. Métricas

Dentro de este subcapítulo explicaremos las métricas usadas en el seguimiento del proyecto, para eso detallaremos el significado, qué utilidad le dimos y qué resultados obtuvimos en el proyecto.

Nótese que estas métricas eran estimadas en el *sprint planning*, en función de los valores de los *sprints* anteriores, y eran evaluadas en el *sprint retrospective*.

Productividad

Esta métrica se calcula tomando el promedio de *story points* realizados en los últimos tres *sprints* y dividiéndolos por las horas de desarrollo que registraron los integrantes dentro de la plataforma. Esto nos sirvió para evaluar nuestra productividad y mejorarla, como parte de nuestros objetivos. La usábamos como elemento de estimación en la *sprint planning meeting*. Para medir esto, no se tomaron en cuenta las horas dedicadas a gestión. En caso de que nuestra productividad bajara de 0.2, debíamos analizar las causas para tomar acciones sobre cómo mejorarla.

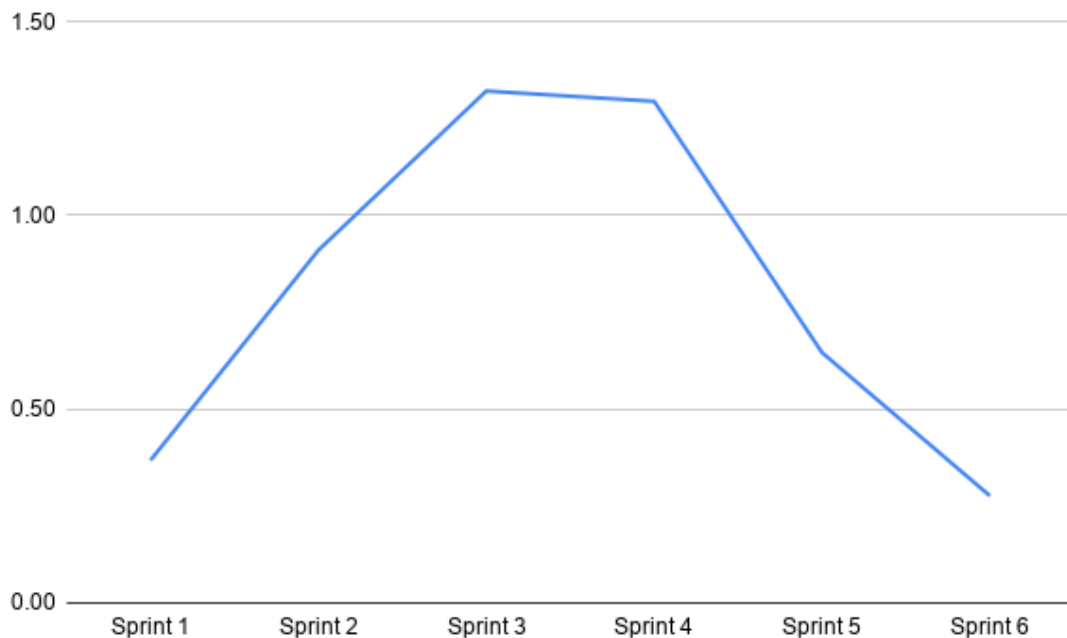


Ilustración 5.15: productividad del equipo

Tomando en cuenta los datos de los *sprints* 2, 3 y 4 de la Ilustración 5.15 podemos decir que la productividad del equipo fue de 1.10 puntos de historia por hora. En el sprint 1 se da un valor mucho menor debido a dificultades con la estimación de las tareas y asignación de puntos de historia. Luego de las reuniones de tutoría pudimos mejorar este proceso y unificar criterios dentro del equipo. La productividad del equipo se vio disminuida en los *sprints* 5 y 6 debido a la salida en producción, que

tuvo algunos inconvenientes que enfocaron el esfuerzo del equipo en tareas que no estaban directamente vinculadas con las historias de usuario estimadas.

Release Burndown

Esta métrica se calcula en basándose en la cantidad de *story points* consumidos por *sprint* y la usamos para ver que tanto teníamos que variar la inversión en horas en el proyecto para poder cumplir con las historias de usuario pendientes. En la Ilustración 5.16 podemos ver el *burndown* del MVP 1 y en la Ilustración 5.17 el *burndown* del MVP 2 que aún está en progreso.

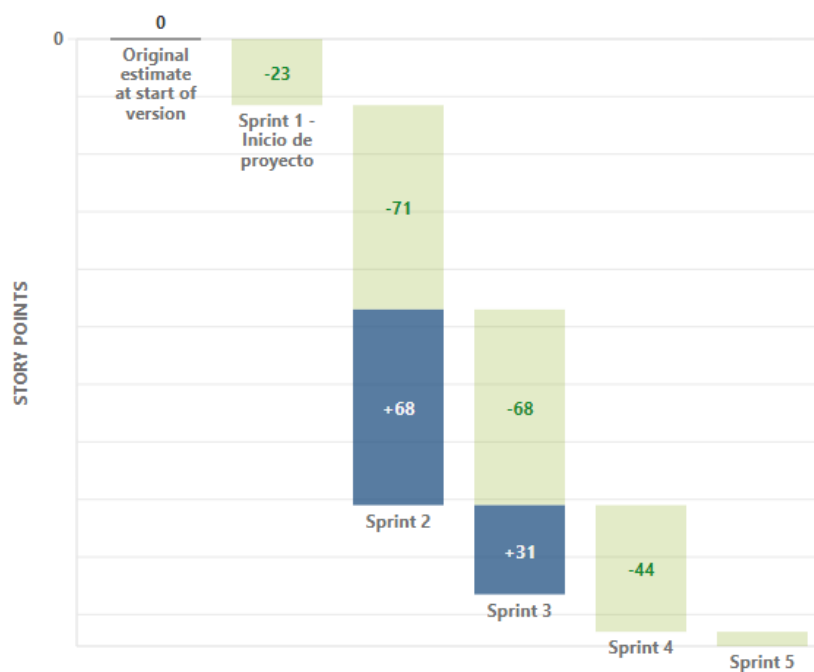


Ilustración 5.16: *burndown* MVP 1.

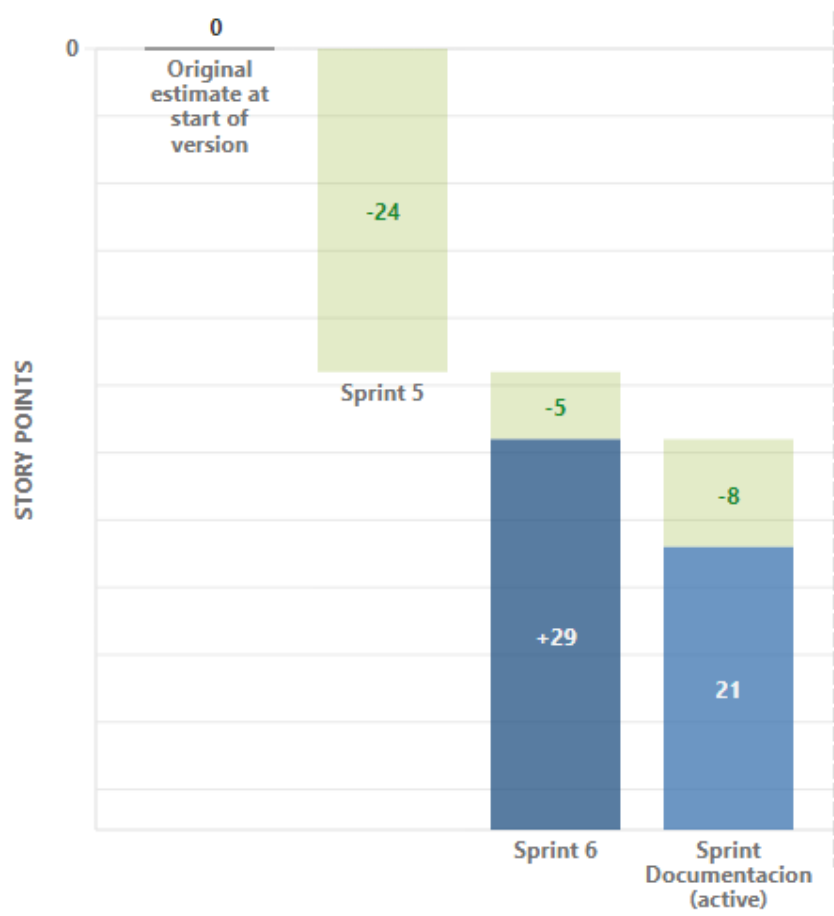


Ilustración 5.17: *burndown* MVP 2.

Distribución del esfuerzo

Esta métrica nos sirvió para evaluar si estábamos adjudicando correctamente nuestro esfuerzo dentro del proyecto. Para esto, optamos por contrastarlo con un modelo presentado más adelante.

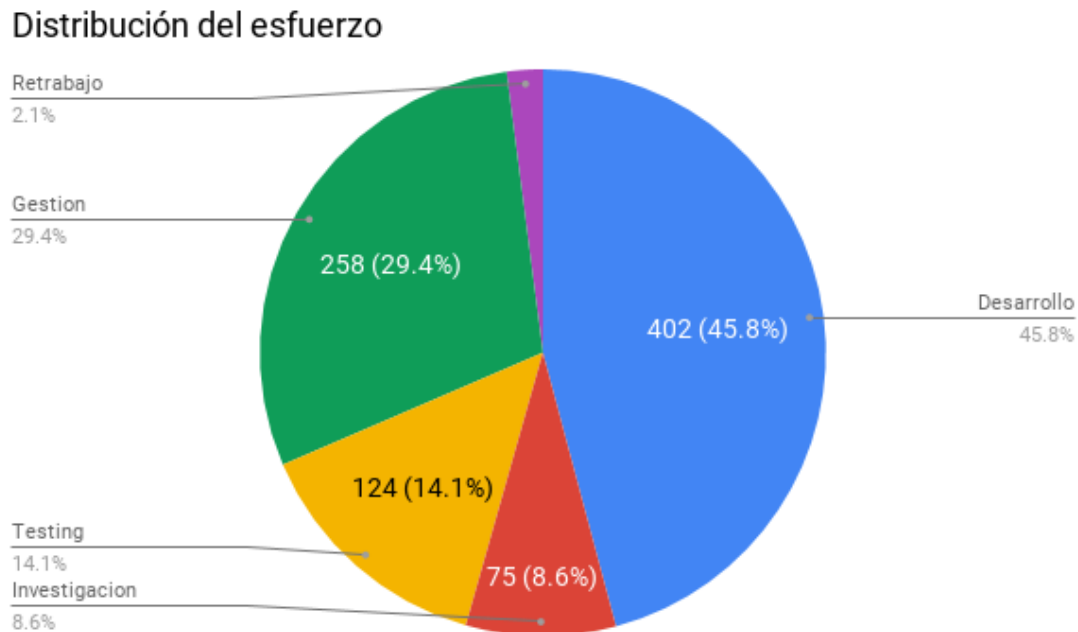


Ilustración 5.18: distribución de esfuerzo de equipo.

El proyecto nos consumió un total de 877 horas de trabajo, como se puede ver en la Ilustración 5.18, la tarea que más trabajo nos insumió fue el desarrollo (402 horas) con un 45.8% del total. Luego le sigue gestión con un 29.4% esto incluye todas las tareas de gestión de la metodología como las *planning meetings*, *requirement definition*, *retrospectives*, reuniones con el cliente, reuniones de tutoría y la confección del presente documento.

En la Ilustración 5.19 podemos ver el modelo de distribución de esfuerzo por tareas propuesto por “Primandari A”[\[45\]](#) para estimar el costo en proyectos de *software* chicos y medianos. En el gráfico tomamos las agrupaciones generales, las agrupaciones específicas se pueden encontrar en el anexo **12.18: Tabla con modelo de distribución de esfuerzo**. En comparación, teniendo en cuenta que este

proyecto tiene un énfasis superior en la gestión debido a los requerimientos académicos, creemos que nuestra distribución se acerca a lo identificado por el modelo, lo que nos demuestra que no hubo grandes desviaciones en la implementación del proyecto.

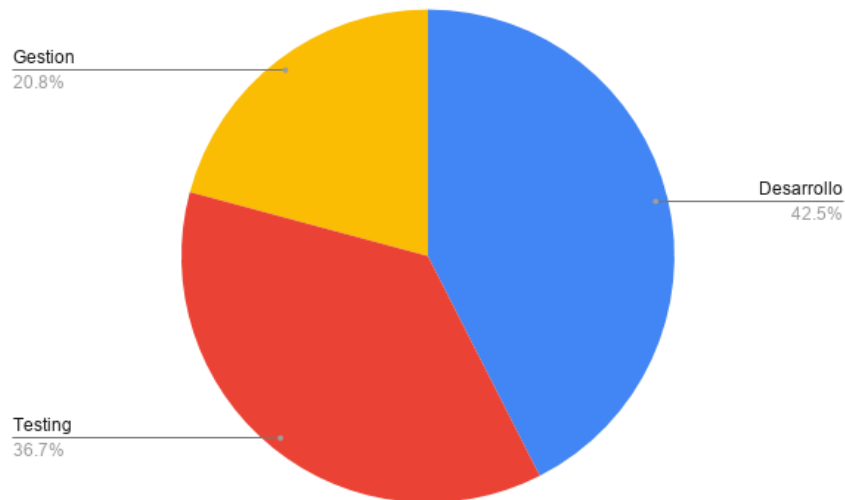


Ilustración 5.19: modelo de distribución de esfuerzo.

Fallas por story point

Definimos tres niveles para categorizar estas fallas:

- Críticas: falla que impide continuar con el uso del sistema o implica pérdida de datos.
- Medias: falla que no impide continuar con el uso del sistema, pero lo dificulta, no implica pérdida de datos.
- Leves: falla no relacionada con la funcionalidad del sistema, sino que se limita a ajustes estéticos.

Para contabilizar los *bugs* los medimos *post-release*. Esto nos servía para ver qué tan confiable había resultado nuestra solución, basándonos en los criterios definidos en las fallas. El cálculo es la cantidad de bugs por *story points* del *release*.

Actualmente no tenemos información suficiente, ya que hubo ciertas demoras de la salida en producción y el control del uso del cliente. Hasta el momento tenemos 25 bugs reportados para el MVP 1, que insumieron un 2.1% de trabajo en el total y ninguno de estos *bugs* fueron catalogados como Críticos o Medios.

Valor actual: $25(\text{bugs})/211(\text{story points release 1}) = 0.118$

Para ver la descripción de las fallas se puede visualizar en anexo **12.2: Issues**.

Retrabajo

Nuestro objetivo planteaba que el retrabajo estuviera por debajo del 20% del total del esfuerzo, si bien el retrabajo está dentro de nuestros parámetros esperados, entendemos que debido a las demoras en salir a producción el valor aun no es representativo ya que consideramos que a la fecha de la escritura de este documento no hay datos suficientes para considerar ese valor como final.

Actualmente tenemos un 2.1% de retrabajo en el total del proyecto.

5.6. Entrega al cliente

Dividimos las entregas al cliente en tres categorías diferentes ya que se manejaron diferentes criterios sobre las diferentes piezas a entregar.

Documentación técnica

Con respecto a la documentación técnica generada con el proyecto, el cliente tenía acceso a ciertos documentos generados en Confluence con decisiones de diseño, tecnologías elegidas y la documentación con las interfaces necesarias para que nuestro *backend* se comunique con Sagal. El documento con las interfaces puede observarse con detalle en el Anexo **12.14: Interfaces requeridas para salida a producción**. Junto con la entrega del MVP 2, se entregarán todos los documentos técnicos generados durante la ejecución del proyecto, incluido este documento

como referencia. Además, incluimos la documentación técnica del *backend* generada por Swagger (véase anexo **12.9: Endpoints de backend con Swagger**).

Manual de usuario

En el *sprint 5* se generó un manual de usuario siguiendo los estilos que mantiene Sagal en su página de ayuda[46] y le fue entregado al responsable de atención al usuario como referencia.

Software

Como productos de software a entregar tenemos cuatro categorías:

- *Backend*
- *Frontend*
- Base de datos
- *Machine learning*

Respecto a la entrega del *backend*, se generó una imagen de Docker[12] para que Sagal la descargue y la instancie en un AWS-EC2[34] dentro de su estructura existente.

Para el *frontend* se entregó un compilado del código con las credenciales necesarias para acceder al AWS-S3 de Sagal y se subió automáticamente mediante un comando provisto por AWS.

Por último, Sagal se encargó de instanciar una base de datos MongoDB en MongoDB Atlas para la persistencia.

Para la entrega del MVP 2, es necesario que Sagal cree una cuenta en *IBM Watson* para que interactúe con el módulo de *machine learning*. Esta información en detalle puede observarse en el capítulo 10: Continuidad del proyecto.

Una vez finalizado el proyecto, Sagal tendrá acceso de escritura de los repositorios para que puedan continuar con el mantenimiento y agregar nuevas funcionalidades si así lo desean.

5.7. Resumen

En suma, logramos definir una metodología personalizada que nos dio confianza y flexibilidad para trabajar ante los requerimientos del cliente. Si bien al principio teníamos dudas de la aplicación de BDD en el proyecto, evaluamos su uso como extremadamente positivo ya que nos brindó confianza a la hora de presentar y entregar el producto, logrando no tener bugs críticos en producción.

Nos hubiera gustado profundizar en las técnicas de *sprint retrospective* pero como equipo funcionamos muy unificados y con comunicación constante lo que no nos permitió sacarle el máximo jugo a las técnicas aplicadas.

6. Aseguramiento de la calidad

En este capítulo resumiremos las distintas actividades y prácticas realizadas, así como los roles y herramientas que contribuyeron al aseguramiento de la calidad durante nuestro proyecto.

6.1. Calidad de los requerimientos

- **Reuniones con Sagal:** tuvimos varias reuniones para determinar los requerimientos con diferentes involucrados, esto es, *Product Owner*, usuarios finales y equipo de desarrollo de Sagal. Esto nos permitió tener clara la perspectiva del cliente en cuanto a su problema con el sistema anterior.
- **Mockups:** una de las primeras acciones para asegurar los requerimientos relevados fue realizar Mockups para luego validarlos con los involucrados, como se comentó anteriormente.
- **Reuniones con usuarios finales:** una vez tuvimos el primer prototipo del sistema, buscamos validarlo con los usuarios finales, esto es, integrantes de la empresa X que son, a su vez, clientes de Sagal. Coordinamos una reunión con la encargada de atención al cliente de una Empresa X que es también la encargada de supervisar a los que responden las preguntas que reciben de Mercado Libre. Tuvo la oportunidad de probar el sistema, priorizar los requerimientos que consideraba más importantes, validar si los requerimientos eran adecuados y si iban a ser útiles para responder preguntas eficaz y eficientemente.

6.2. Calidad del producto final

- **Definición de un plan de *releases*:** junto al *product owner* determinamos dos *releases* del producto. Esto permitió que el cliente pudiera usar el producto en forma temprana y nos diera *feedback* operativo rápidamente.

- **Aplicación del ciclo de BDD:** el uso y aplicación de BDD nos permitió definir requerimientos de calidad y que, a partir de éstos, se pudieran diseñar directamente los casos de prueba y ejecutarlos de forma automática. Este diseño y automatización de la prueba conformaban el eje fundamental de las pruebas *pre-release*.
- **Sprint review:** la revisión del producto final de cada sprint, por parte del *Product Owner*, contribuía a que nos aseguráramos de que estábamos construyendo el producto correcto.
- **Sprint retrospective:** para asegurar la mejora del proceso del equipo, realizábamos los *sprint retrospective*. El aplicar una forma de trabajo de calidad, ayudaba, de manera indirecta, a la calidad del producto.
- **Realease review:** además de las revisiones del producto durante los *sprint review*, el *product owner* revisaba, en forma integrada, la salida de un reléase. Esto facilitaba y preparaba la prueba *post-release*, que deberían hacer el cliente y los usuarios finales (véase anexo **12.7: Reuniones con el cliente**).
- **Testing post-release:** para asegurar aún más la calidad del producto, luego de realizar la *release review*, se liberaba el producto al cliente y usuarios finales, junto a los escenarios de las historias de usuario especificadas, para que éstos probaran el producto y revisaran los demás entregables, de manera de obtener *feedback*.
- **Product owner:** al tener un *product owner* pudimos tener un referente del lado del cliente para priorizar las *user stories*.
- **Code reviewer:** para asegurar la modificabilidad y que el cliente pueda ver el avance de todo el *frontend*, se dispuso por parte de nuestro cliente, un reviewer, en efecto Simón Seitune. El mismo tenía acceso de lectura al repositorio y podía comentar los *pull request* con recomendaciones o posibles mejoras que luego terminaban afectando la modificabilidad.
- **Uso de standard de linter:** otro de los puntos clave fue el uso de *linter*, el mismo fue utilizado a nivel de *frontend* a pedido del cliente. Esto permite asegurar un código *standard* y aporte a la modificabilidad.

- **Standard API REST:** el uso de un protocolo estándar para la comunicación entre Apis en cierta medida también apoya nuestra calidad en el código.
- **Gitflow:** seguir con los estándares de Gitflow nos permitió tener una mejor gestión en las *features* que realizamos y en definir los *releases*. También nos permitió evitar tener pérdidas de códigos y reducir los conflictos ya que cada integrante trabajaba en *branches* distintas separadas por *feature*.

6.3. Calidad del modelo de *machine learning*

- **Entrenamiento:** para entrenar el modelo, optamos por la creación de un *script* que permita tomar preguntas y que las catalogue automáticamente según ciertos patrones en las preguntas, como pueden ser “stock”, “permutas”, entre otros. Hacerlo de esta manera nos permitió entrenar el modelo de una manera más rápida y precisa.
- **Validación:** para validar el modelo solicitamos la ayuda del equipo de Sagal, para catalogar 1000 preguntas en las cinco diferentes categorías mencionadas en capítulos anteriores. Luego de tener el modelo entrenado, corrimos un *script* para tomar los datos de los archivos *Excel* entregados por el cliente y generar diferentes archivos JSON, los cuales eran enviados a *Watson* para obtener su categorización y de esta forma evaluar la confianza del modelo, llegando al 70% como fue planificado.
- **Respaldo de datos:** una vez que tuvimos listo el archivo para entrenar el modelo, creamos un *script* que nos permitió insertarlo en nuestra base de datos de MongoDB, en un formato entendible por el sistema. Esto se realizó para que, si se desea en un futuro reentrenar el modelo, se pueda realizar utilizando las preguntas catalogadas por nosotros más las preguntas que cataloguen los usuarios que usen el sistema.

7. Gestión de la configuración

En este capítulo detallaremos como realizamos la gestión de la configuración (SCM)[47], esto es, los repositorios utilizados, control y seguimiento de cambios, así como las herramientas utilizadas para el versionado de documentos.

7.1. Control de versionado

Para el control de versionado utilizamos diversas herramientas dependiendo el tipo de elemento.

Código fuente

Para el versionado del código utilizamos repositorios en Bitbucket[48] como se puede observar en la Ilustración 7.1.



Ilustración 7.1: repositorios del equipo

Decidimos dividirlo en 2 repositorios:

- **Backend:** contiene el código fuente y *testing* respecto a la lógica interna del sistema como mostrar las preguntas, responder, etiquetado automático, la interacción que realiza con Sagal, entre otros.

- **Frontend:** contiene el código fuente y *testing* respecto a la visualización de la información del usuario y la interacción que realiza con el *backend*.

Se puede observar que hay un tercer repositorio "*machinelearnig*". En un principio pensamos utilizarlo como un proyecto independiente del *backend*, pero lo descartamos ya que la implementación fue muy pequeña como para tenerlo separado en otro proyecto y se realizó como parte del *backend*. Además, en el *backend* se implementó una interfaz genérica para soportar otros modelos de *machine learning* como describimos en capítulos anteriores.

La razón de porque lo dividimos de esa forma fue porque nos permitió trabajar de forma más eficiente y organizada, ya que el impacto de los cambios repercute sólo en el área en donde se estén realizando los mismos, a la vez que la distribución de las *branches* es más prolija al separar *backend* y *frontend*.

Los repositorios son *Git* ya que el equipo posee experiencia en este tipo de herramientas de control de versiones y utilizamos *GitFlow*[\[41\]](#) como forma de organizar las *branches* de los repositorios.

Para los repositorios, definimos niveles de acceso, tanto para nosotros como desarrolladores, así como para nuestro cliente y *code reviewer*.

- **Desarrolladores:** Acceso de Administrador (todos los permisos).
- **Cliente:** Acceso de lectura.
- **Code Reviewer:** Acceso de lectura.

Por último, definimos ciertas restricciones sobre los repositorios para que cada integrante respete los estándares de *GitFlow*. Las restricciones que aplicamos fueron las siguientes:

- **Branch develop**
 - No se permite a nadie eliminar la *branch*.
 - Realizar un *merge* a esta *branch* sólo puede realizarse si se tiene una aprobación previa por alguno de los integrantes del proyecto.
- **Branch master**

- No se permite a nadie eliminar esta *branch*.
- Realizar un *merge* a esta *branch* sólo puede realizarse si se tiene la aprobación de 2 integrantes del proyecto. Además, sólo Nicolás Caballero puede realizar el *merge* ya que es la persona a cargo de realizar esta acción por su experiencia previa en este tipo de tareas.

En la ilustración 7.2, se pueden observar las reglas definidas en Bitbucket.




Branch	Access type	Users and groups	Custom settings
 develop (Development) <small>From branching model</small>	Write access	None	Deleting this branch is not allowed Rewriting branch history is not allowed
	Merge via pull request	Everybody	Check for at least 1 approval
 master (Production) <small>From branching model</small>	Write access	None	Deleting this branch is not allowed Rewriting branch history is not allowed
	Merge via pull request	 Nicolás Caballero	Check for at least 2 approvals

Ilustración 7.2: reglas de los repositorios en Bitbucket

7.2. Documentación

Confluence

Decidimos utilizar Confluence[49] para gestionar los documentos más relevantes asociados a la gestión del proyecto, como utilizamos casi todo el *conjunto de Atlassian*[50], podíamos asociar los documentos de Confluence a las *user stories* en Jira[43] e incluso vincularlas a la *branch* de SourceTree[51].

Generamos varios documentos que detallaron nuestros procesos de gestión, diseño y comunicación con el cliente.

Esta herramienta fue utilizada como nuestra referencia ante cualquier duda. En la misma almacenamos documentos del tipo: *sprint retrospectives* (véase anexo 12.6: **Sprint retrospectives**), decisiones de diseño(véase anexo 12.8: **Reuniones del equipo**), reuniones con los clientes, *feedback* de cliente, compromiso de horas, manual de usuario del *reléase* (véase anexo 12.15: **Manual de usuario**), gestión de

riesgos (véase anexo **12.16: Reporte de riesgos**), entre otros, como se puede apreciar en la Ilustración 7.3.

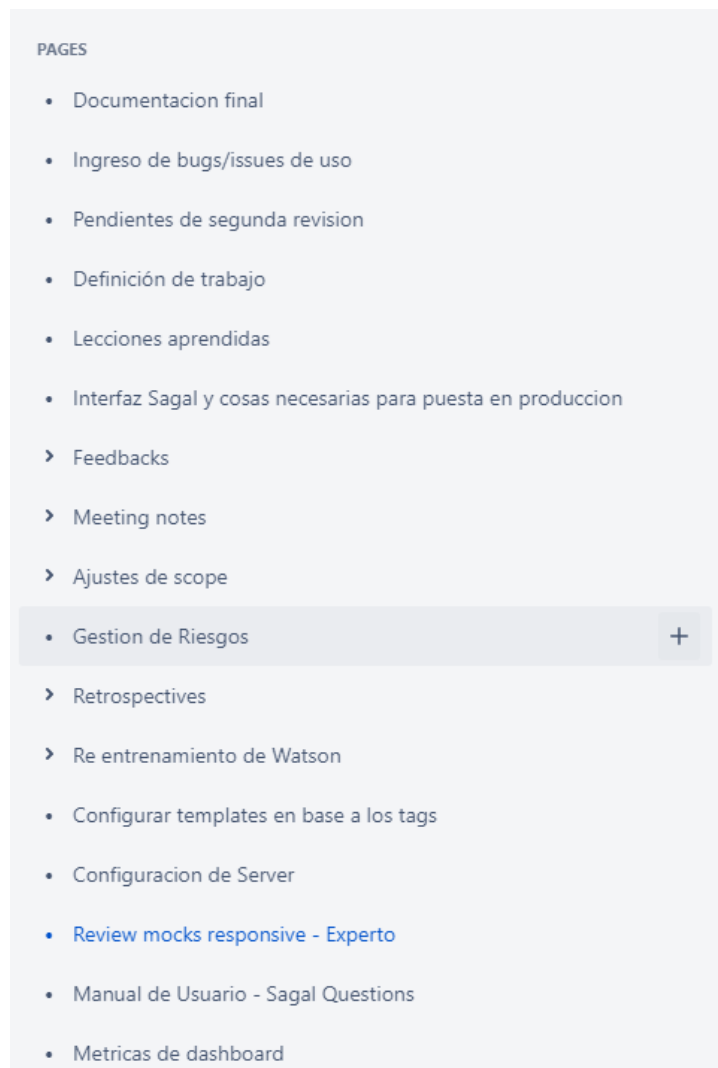


Ilustración 7.3: página de Confluence

Swagger

Utilizamos Swagger para documentar los *endpoints* del *backend* y entregárselos al cliente. (Véase anexo **12.9: Endpoints de backend con Swagger**)

One Drive - Word Online

En el desarrollo de este documento utilizamos Word Online[52] para trabajar en conjunto, y mantener los estilos solicitados, ya que éstos vienen en el *template*

realizado para Word. Nos distribuimos en diferentes archivos para ser más productivos, como se puede ver en la Ilustración 7.4.

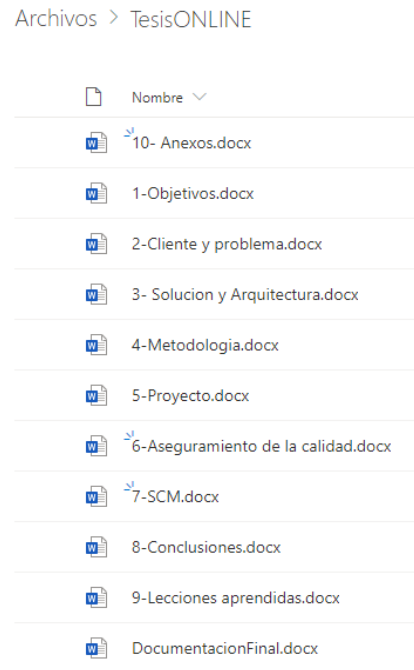


Ilustración 7.4: OneDrive online con documentos de Word

Google Drive

Para realizar las diferentes versiones de la presentación de la defensa, almacenar ciertos *scripts* utilizados en el proyecto, guardar los *mocks* realizados, *sprint reports* y diagramas decidimos utilizar Google drive[53]. En la Ilustración 7.5 se puede visualizar las diferentes carpetas utilizadas.

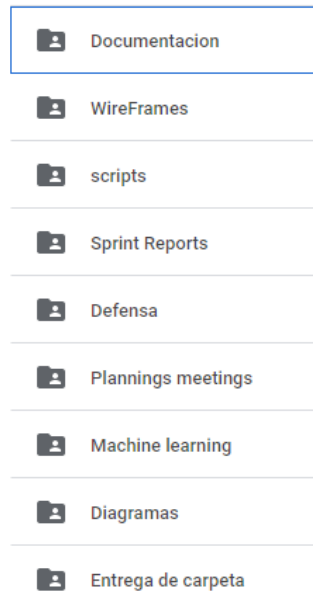


Ilustración 7.5: Google drive

7.3. Gestión de proyecto

Jira

Para la gestión general del proyecto en lo que respecta a registro de *user stories*, tareas, escenarios, control de *issues* y medición de la velocidad utilizamos Jira, ya que el equipo posee un amplio conocimiento de esta herramienta.

Además, nos permitió llevar un registro de las horas realizadas por cada integrante, pudimos tener una imagen en cada momento del estado del trabajo de cada uno en el *task board* del *sprint*, así como tener un *backlog* detallado con diferentes tareas con sus respectivas prioridades. Esto simplificó el trabajo de gestión interpersonal. A partir del cuarto *sprint* decidimos compartir el *backlog* con el *Product Owner* para que pudiera ver el estado y el avance del proyecto

En la Ilustración 7.6 se puede visualizar las tareas en un momento dado del “*sprint* documentación”.

Sprint 7

Documentar todo el proyecto

Q NC RD Only My Issues Recently Updated

TO DO IN PROGRESS DONE

TES-417 TO DO 11 sub-tasks Documentación Final

Proyecto TES-422	Objetivos TES-418
Revision de anexos TES-425	Cliente y problema TES-419
Revision final TES-426	Solucion y Arquitectura TES-420
SCM TES-435	Metodologia TES-421
	Conclusiones TES-423
	Lecciones aprendidas TES-424
	Aseguramiento de la calidad TES-434

Ilustración 7.6: *sprint* documentación - *task board*

8. Conclusiones

En este capítulo mostraremos las conclusiones generales del proyecto y haremos una evaluación de los objetivos que establecimos al inicio, de manera de verificar su grado de cumplimiento.

8.1. Conclusiones generales

Luego de varios meses de trabajo, nos sentimos muy conformes con el alcance logrado. A pesar de que, al momento de escribir este documento, aún no hemos podido poner en producción el segundo MVP, el cliente está satisfecho con el producto alcanzado, así como las empresas X, que está esperando con ansias el lanzamiento del segundo MVP para obtener las nuevas funcionalidades.

Destacamos también el aprendizaje y trabajo en equipo que logramos, ya que este proyecto nos permitió reafirmar la dedicación, profesionalidad y confianza de cada uno de nosotros para proyectos futuros en conjunto.

Tenemos planificado seguir trabajando con Sagal para finalizar las funcionalidades que quedaron pendientes y solucionar errores que puedan ocurrir en el lanzamiento del MVP 2.

8.2. Cumplimiento de objetivos

Generar valor genuino al cliente

Este objetivo optamos por medirlo con los siguientes puntos:

- **Poder entregar dos MVP:** hasta la confección de este documento se entregó el primer MVP y se terminó el desarrollo del segundo MVP, que tiene una fecha tentativa de salida a producción del 20 de abril del corriente.

Consideramos este objetivo como cumplido ya que el desarrollo está listo y solo falta ajustar el pasaje a producción.

- **Obtener 4 sobre 5 de promedio en la encuesta realizada a Empresa X y a Sagal:** realizamos la primera encuesta con una de las empresas que probó el sistema y con Sagal, de la cual obtuvimos un promedio de 4 después de la entrega del primer MVP (véase anexo **12.17: Encuestas de satisfacción**). Una vez se libere el segundo, haremos la segunda encuesta. Por lo tanto, este objetivo está pendiente de la salida a producción del segundo MVP.
- **Lograr que el modelo de etiquetas automáticas tenga una confianza del 70%:** como mencionamos en los capítulos anteriores se logró entrenar un modelo con este nivel de confianza, por lo que podemos dar este objetivo como cumplido.
- **Aumentar el uso del sistema:** como mencionamos en el capítulo de objetivos, no tendremos suficiente tiempo para evaluar efectivamente este punto.

Mejorar en procesos de gestión de proyectos ágiles

- **Mejorar la productividad del equipo:** como vimos en las métricas del proyecto, nuestra productividad fue creciendo en los *sprints*, por lo que podemos considerar este objetivo como cumplido.
- **Cero fallas críticas en producción:** si bien tenemos la aplicación en producción, consideramos que no pasó suficiente tiempo de uso como para dar este objetivo como cumplido.
- **Tener un retrabajo mejor al 20% del total del esfuerzo:** actualmente nuestro nivel de retrabajo está muy por debajo del objetivo planteado, pero igual que en el objetivo anterior, no consideramos que pasó suficiente tiempo para que la métrica sea la definitiva, por lo que por ahora no lo consideramos como cumplido.

Profesionalización del trabajo en equipo

Logramos cumplir varios de los objetivos propuestos anteriormente por lo que consideramos que vamos por buen camino, pero hasta la aprobación de la instancia académica consideramos que este objetivo aún está pendiente.

9. Lecciones aprendidas

Dentro de este capítulo veremos los aprendizajes que tuvimos como parte de la ejecución del proyecto y las recomendaciones podríamos hacer a proyectos futuros.

- **No invertir demasiado tiempo, al inicio del proyecto, en el relevamiento y decisiones de arquitectura:** al comienzo del proyecto dedicamos un mes y medio al relevamiento y a la toma de decisiones respecto de la arquitectura. Viéndolo en retrospectiva, y luego de mejorar nuestro proceso de gestión del proyecto, entendemos que la inversión de tiempo en esas decisiones no se tradujo en resultados tangibles. Conviene distribuir estas tareas a lo largo de todo el proyecto y no tratar de realizarlas solo al inicio.
- **Cuando trabajamos con una aplicación para salir en producción, es importante tener confiabilidad en el sistema:** en un principio sentimos que la implementación de BDD nos insumía un tiempo que no traía resultados. A medida que fue progresando el proyecto, empezamos a ver el valor de identificar y solucionar los errores antes de entregar el producto, esto fue gracias a nuestra inversión de tiempo en pruebas automáticas. Esto nos permitió generar una mayor confiabilidad en el sistema. Es importante invertir en *testing* en el proceso de desarrollo para que la inversión en reparación de los errores baje.
- **Siempre hay riesgos, no gestionarlos correctamente es un error. No hay nada que suplante un plan de riesgos:** al inicio del proyecto no tuvimos en cuenta la gestión de los riesgos y esto nos generó dificultades para avanzar con el proyecto y salir a producción con el MVP 1. La experiencia en este proyecto nos demostró que es clave poder anticiparse a los problemas discutiendo estrategias de mitigación y su implementación.
- **La comunicación con el cliente es clave, es necesario agendar reuniones periódicas:** la comunicación con el cliente fue muy buena en general, pero fue necesario realizarles seguimiento en las etapas finales para que mantengan su compromiso con el proyecto.

- **Tener la agenda de las reuniones definida de antemano:** es importante aprovechar los tiempos de las reuniones teniendo bien definidos los temas a tratar, en nuestro caso sentimos que no fuimos eficientes en la gestión de nuestras reuniones con el cliente, no logrando objetivos claros para registrar después de una reunión. Esto genera gastos de tiempo del proyecto en reuniones que no aportan valor.
- **Compromiso de horas del equipo:** al principio del proyecto nos comprometimos como equipo a cumplir con 20 horas semanales cada uno, esto no se cumplió y generaba choques dentro del equipo. Es importante ser realistas a la hora de definir el compromiso con el proyecto.
- **Negociación con el cliente:** el cliente no tenía claridad en las fechas de entrega, por lo que tuvimos que negociar las funcionalidades que irían incluidas en cada una y las fechas esperadas. Si bien parece obvio, esta tarea no es trivial, estas definiciones tienen que quedar claras para ambas partes mediante documentos compartidos.

10. Continuidad del proyecto

Luego de realizada la entrega académica, vamos a enfocarnos en la salida a producción del resultado del segundo reléase, esto es, MVP 2. El objetivo es brindar soporte a este MVP y reparar los bugs que puedan surgir. También incorporaremos pequeños cambios en los *features* que sugiera el cliente.

Para priorizar los *releases* a futuro, se tuvo una reunión de coordinación donde se decidió la siguiente priorización de funcionalidades:

User story	Release
<ul style="list-style-type: none">TES-267: Como administrador quiero poder cambiar la implementación de los servicios de machine learning para utilizar otro modelo si lo deseo DONETES-23: Como usuario quiero que las preguntas puedan catalogarse correctamente en diferentes etiquetas para poder responderlas más fácil en el futuro DONETES-403: Modelado de ML DONE	MVP 2
<ul style="list-style-type: none">TES-10: Como usuario quiero poder usar la aplicacion desde el celular para poder contestar las preguntas desde ahi DONE	MVP 2
<ul style="list-style-type: none">TES-187: Como usuario quiero poder eliminar preguntas para descartar las que perjudiquen mis publicaciones TO DO	Release 3
<ul style="list-style-type: none">TES-196: Como usuario quiero poder visualizar el historial de preguntas realizadas de un usuario en una misma publicación para poder seguir el hilo de una conversación TO DO	Release 3
<ul style="list-style-type: none">TES-385: Como usuario quiero poder tener un dashboard para poder ver estadísticas de mis preguntas TO DO	A definir por el cliente
<ul style="list-style-type: none">TES-193: Como usuario quiero poder contestar preguntas masivamente para poder responder las mismas preguntas una unica vez TO DOTES-194: Como usuario quiero poder eliminar preguntas masivamente para descartar varias preguntas que perjudiquen mis publicaciones TO DO	A definir por el cliente

Ilustración 10.1: reunión con Sagal para definir pasos a seguir.

Las historias de usuario planificadas para MVP 2 ya están completas en su desarrollo, pero falta hacer la salida a producción, por lo que el MVP 2 incluiría las

historias detalladas en Ilustración 10.1 y una historia de transferencia de conocimiento. Las historias planificadas como Release 3 y a definir serán realizados personal de Sagal.

11. Referencias

- [1] "Capability Maturity Model Integration", *Wikipedia*. feb. 26, 2020, Consultado: abr. 12, 2020. [En línea]. Disponible en: https://en.wikipedia.org/w/index.php?title=Capability_Maturity_Model_Integration&oldid=942746146.
- [2] "Sagal". <https://sagal.io/> (consultado abr. 12, 2020).
- [3] "Enterprise resource planning", *Wikipedia*. abr. 10, 2020, Consultado: abr. 12, 2020. [En línea]. Disponible en: https://en.wikipedia.org/w/index.php?title=Enterprise_resource_planning&oldid=950156562.
- [4] "Bienvenidos a MercadoLibre". <http://mercadolibre.com/> (consultado abr. 12, 2020).
- [5] "Producteca, tu negocio omnicanal", *Producteca*. <https://www.producteca.com/> (consultado abr. 12, 2020).
- [6] "Con Astroselling multiplica tus ventas en Mercadolibre". <https://astroselling.com/> (consultado abr. 12, 2020).
- [7] "What is User Experience (UX) Design?", *The Interaction Design Foundation*. <https://www.interaction-design.org/literature/topics/ux-design> (consultado abr. 12, 2020).
- [8] Institute of Electrical and Electronics Engineers, Ed., *IEEE recommended practice for software requirements specifications: approved 25 June 1998*. New York, NY: IEEE, 1998.
- [9] "ESLint - Pluggable JavaScript linter", *ESLint - Pluggable JavaScript linter*. / (consultado abr. 12, 2020).
- [10] *airbnb/javascript*. Airbnb, 2020.
- [11] Node.js, "Node.js", *Node.js*. <https://nodejs.org/en/> (consultado abr. 12, 2020).
- [12] "Empowering App Development for Developers | Docker". <https://www.docker.com/> (consultado abr. 12, 2020).
- [13] "React - A JavaScript library for building user interfaces". <https://reactjs.org/> (consultado abr. 12, 2020).

- [14] “Representational state transfer”, *Wikipedia*. feb. 19, 2020, Consultado: abr. 12, 2020. [En línea]. Disponible en: https://en.wikipedia.org/w/index.php?title=Representational_state_transfer&oldid=941589430.
- [15] “IBM Watson | IBM”. <https://www.ibm.com/watson> (consultado abr. 12, 2020).
- [16] “AWS | Cloud Computing - Servicios de informática en la nube”. <https://aws.amazon.com/es/> (consultado abr. 12, 2020).
- [17] “AWS | Almacenamiento de datos seguro en la nube (S3)”, *Amazon Web Services, Inc*. <https://aws.amazon.com/es/s3/> (consultado abr. 12, 2020).
- [18] styled-components, “styled-components”, *styled-components*. <https://www.styled-components.com> (consultado abr. 12, 2020).
- [19] “TechEmpower Web Framework Performance Comparison”, *www.techempower.com*. <https://www.techempower.com/benchmarks/#section=data-r18&hw=ph&test=plaintext> (consultado abr. 12, 2020).
- [20] “Stack Overflow Developer Survey 2019”, *Stack Overflow*. https://insights.stackoverflow.com/survey/2019/?utm_source=social-share&utm_medium=social&utm_campaign=dev-survey-2019 (consultado abr. 12, 2020).
- [21] “Fastify, Fast and low overhead web framework, for Node.js”. <https://www.fastify.io/> (consultado abr. 12, 2020).
- [22] “Java servlet”, *Wikipedia*. mar. 14, 2020, Consultado: abr. 12, 2020. [En línea]. Disponible en: https://en.wikipedia.org/w/index.php?title=Java_servlet&oldid=945569575.
- [23] “Grails Framework”. <https://grails.org/> (consultado abr. 12, 2020).
- [24]. “NET | Free. Cross-platform. Open Source.”, *Microsoft*. <https://dotnet.microsoft.com/> (consultado abr. 12, 2020).
- [25] wadepickett, “ASP.NET documentation”. <https://docs.microsoft.com/en-us/aspnet/core/> (consultado abr. 12, 2020).
- [26] “API Documentation & Design Tools for Teams | Swagger | Swagger”. <https://swagger.io/> (consultado abr. 12, 2020).

- [27] “The most popular database for modern apps”, *MongoDB*. <https://www.mongodb.com> (consultado abr. 12, 2020).
- [28] “Mongoose ODM v5.9.8”. <https://mongoosejs.com/> (consultado abr. 12, 2020).
- [29] “Progressive Web Apps”, *web.dev*. <https://web.dev/progressive-web-apps/> (consultado abr. 12, 2020).
- [30] “BDD Testing & Collaboration Tools for Teams | Cucumber”. <https://cucumber.io/> (consultado abr. 12, 2020).
- [31] “Jest · 🍌 Delightful JavaScript Testing”. <https://jestjs.io/> (consultado abr. 12, 2020).
- [32] “Introduction · Enzyme”. <https://enzymejs.github.io/enzyme/> (consultado abr. 12, 2020).
- [33] “AWS | Servicio de entrega de contenidos y transferencia de datos”, *Amazon Web Services, Inc.* <https://aws.amazon.com/es/cloudfront/> (consultado abr. 12, 2020).
- [34] “AWS | Elastic compute cloud (EC2) de capacidad modificable en la nube”, *Amazon Web Services, Inc.* <https://aws.amazon.com/es/ec2/> (consultado abr. 12, 2020).
- [35] “lint (software)”, *Wikipedia*. mar. 29, 2020, Consultado: abr. 12, 2020. [En línea]. Disponible en: [https://en.wikipedia.org/w/index.php?title=Lint_\(software\)&oldid=948040920](https://en.wikipedia.org/w/index.php?title=Lint_(software)&oldid=948040920).
- [36] “Top 10 ERP and E-commerce Integration Providers in 2019 | Hacker Noon”. <https://hackernoon.com/top-10-erp-and-e-commerce-integration-providers-in-2019-69bacf1bb7d2> (consultado abr. 12, 2020).
- [37] “What is Scrum?”, *Scrum.org*. <https://www.scrum.org/resources/what-is-scrum> (consultado abr. 12, 2020).
- [38] “Behavior-driven development”, *Wikipedia*. abr. 12, 2020, Consultado: abr. 12, 2020. [En línea]. Disponible en: https://en.wikipedia.org/w/index.php?title=Behavior-driven_development&oldid=950525091.

- [39] “Feature-driven development”, *Wikipedia*. sep. 12, 2019, Consultado: abr. 12, 2020. [En línea]. Disponible en: https://en.wikipedia.org/w/index.php?title=Feature-driven_development&oldid=915290021.
- [40] “Gherkin Reference - Cucumber Documentation”. <https://cucumber.io/docs/gherkin/reference/> (consultado abr. 12, 2020).
- [41] Atlassian, “Gitflow Workflow | Atlassian Git Tutorial”, *Atlassian*. <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow> (consultado abr. 12, 2020).
- [42] “Story Point Estimation - Getting Started With Affinity Estimation”, *Agile Velocity*, jun. 02, 2016. <https://agilevelocity.com/blog/get-started-story-points-via-affinity-estimation-cheat-sheet/> (consultado abr. 12, 2020).
- [43] Atlassian, “Jira | Issue & Project Tracking Software”, *Atlassian*. <https://www.atlassian.com/software/jira> (consultado abr. 12, 2020).
- [44] “The best retrospective for beginners – All About Retrospectives”. <https://retromat.org/blog/best-retrospective-for-beginners/> (consultado abr. 12, 2020).
- [45] P. L. Primandari A y Sholiq, “Effort Distribution to Estimate Cost in Small to Medium Software Development Project with Use Case Points”, *Procedia Comput. Sci.*, vol. 72, pp. 78–85, 2015, doi: 10.1016/j.procs.2015.12.107.
- [46] “Sagal Help Center”. <http://help.sagal.io/es/> (consultado abr. 12, 2020).
- [47] Institute of Electrical and Electronics Engineers y IEEE-SA Standards Board, *IEEE standard for configuration management in systems and software engineering*. New York: Institute of Electrical and Electronics Engineers, 2012.
- [48] Atlassian, “Bitbucket | The Git solution for professional teams”, *Bitbucket*. <https://bitbucket.org/product> (consultado abr. 12, 2020).
- [49] Atlassian, “Confluence - Accomplish more together”, *Atlassian*. <https://www.atlassian.com/software/confluence> (consultado abr. 12, 2020).
- [50] Atlassian, “Atlassian | Software Development and Collaboration Tools”, *Atlassian*. <https://www.atlassian.com> (consultado abr. 12, 2020).
- [51] Atlassian, “Sourcetree | Free Git GUI for Mac and Windows”, *SourceTree*. <https://www.sourcetreeapp.com> (consultado abr. 12, 2020).

- [52] “Página principal de Microsoft Office”. <https://www.office.com/?auth=2>
(consultado abr. 12, 2020).
- [53] “My Drive - Google Drive”. <https://drive.google.com/drive/u/0/my-drive>
(consultado abr. 12, 2020).

12. Anexos

12.1. Backlog

T	Key ↑	Summary	Status	Resolution
	TES-3	Front - Como usuario quiero visualizar todas las preguntas para poder responderlas una tras otra	DONE	Done
	TES-4	Como usuario al visualizar una pregunta quiero poder visualizar todos los datos relacionados para poder contestar fácilmente	DONE	Done
	TES-5	Como usuario al visualizar una pregunta quiero poder visualizar el nombre y apellido del usuario que me consulta para identificar al usuario que pregunta	DONE	Done
	TES-8	Como usuario quiero poder tener un paginado de preguntas a responder para poder navegar entre ellas	DONE	Done
	TES-9	Como usuario quiero poder visualizar todas las preguntas ya respondidas para tener un historial de las mismas además de un seguimiento	DONE	Done
	TES-10	Como usuario quiero poder usar la aplicación desde el celular para poder contestar las preguntas desde ahí	DONE	Done
	TES-11	Como usuario quiero poder visualizar las preguntas más antiguas sin contestar primero para responderlas lo antes posible	DONE	Done
	TES-12	Como usuario quiero al contestar una pregunta que desaparezca para visualizar solo las no respondidas	DONE	Done
	TES-13	Como usuario quiero poder filtrar las preguntas para poder buscar las que son más relevantes	DONE	Done
	TES-14	Como usuario quiero poder contestar una pregunta para responderle a los clientes finales	DONE	Done
	TES-23	Como usuario quiero que las preguntas puedan catalogarse correctamente en diferentes etiquetas para poder responderlas más fácil en el futuro	DONE	Done
	TES-42	Conexión e interfaz con Sagal	DONE	Done
	TES-48	Back - Como usuario quiero visualizar todas las preguntas para poder responderlas una tras otra	DONE	Done
	TES-60	Como administrador quiero que la aplicación se pruebe automáticamente para tener confianza en la aplicación	DONE	Done
	TES-65	Asegurar la performance de la aplicación	TO DO	Unresolved
	TES-66	Asegurar que el frontend de la aplicación sea reutilizable	DONE	Done
	TES-67	Asegurar la usabilidad respetando las métricas y pantallas acordadas	TO DO	Unresolved
	TES-81	Preparar el primer release	DONE	Done
	TES-85	Como usuario quiero agrupar las preguntas por artículo para tener las preguntas más ordenadas	TO DO	Unresolved
	TES-86	Aplicar uso de linter	DONE	Done
	TES-87	Como usuario quiero poder comunicarme con el soporte técnico para resolver mis dudas	DONE	Done

TES-186	Como usuario quiero ver más detalles del artículo a demanda para poder responder las preguntas más específicas	DONE	Done
TES-187	Como usuario quiero poder eliminar preguntas para descartar las que perjudiquen mis publicaciones	TO DO	Unresolved
TES-188	Como usuario quiero poder utilizar hotkeys en el uso del sistema para que mejore mis tiempos de respuesta	DONE	Done
TES-189	Como usuario quiero poder dar de baja templates para dejar de utilizarlos al responder preguntas	DONE	Done
TES-190	Como usuario quiero poder editar templates para poder responder mas rapido	DONE	Done
TES-191	Como usuario quiero poder ver los templates para luego poder administralos	DONE	Done
TES-192	Como usuario quiero poder cambiar de páginas entre los diferentes templates que están creados para poder navegar eficazmente	DONE	Done
TES-193	Como usuario quiero poder contestar preguntas masivamente para poder responder las mismas preguntas una unica vez	TO DO	Unresolved
TES-194	Como usuario quiero poder eliminar preguntas masivamente para descartar varias preguntas que perjudiquen mis publicaciones	TO DO	Unresolved
TES-195	Como usuario quiero poder dar de alta templates para utilizarlos al responder preguntas	DONE	Done
TES-196	Como usuario quiero poder visualizar el historial de preguntas realizadas de un usuario en una misma publicación para poder seguir el hilo de una conversación	TO DO	Unresolved
TES-233	Reuniones de Tutoria	DONE	Done
TES-237	Reuniones con Sagal	DONE	Done
TES-253	Como usuario quiero poder ver quien realizo las acciones sobre las preguntas para poder hacerles un seguimiento	DONE	Done
TES-263	Como usuario quiero recibir notificaciones sobre las acciones que estoy realizando para mejorar la usabilidad	DONE	Done
TES-266	Como usuario quiero poder visualizar los tags asociadas a una pregunta para poder identificar sobre que trata rápidamente	DONE	Done
TES-267	Como administrador quiero poder cambiar la implementación de los servicios de machine learning para utilizar otro modelo si lo deseo	DONE	Done
TES-268	Como usuario quiero ver cuales son los hotkeys disponibles para poder usarlos	DONE	Done
TES-277	Como usuario quiero poder buscar las preguntas con determinado tag para filtrar mejor	DONE	Done
TES-278	Como usuario quiero poder agregar tags a una pregunta para poder clasificarla	DONE	Done
TES-279	Como usuario quiero poder eliminar las tags de una pregunta	DONE	Done
TES-280	Como usuario quiero poder usar los templates para contestar preguntas mas agilmente	DONE	Done
TES-282	Como usuario quiero poder expandir el historico para ver mas detalles	DONE	Done
TES-319	Como administrador quiero poder monitorear la app con sentry.io para saber si hay problemas recurrentes	DONE	Done
TES-365	Como usuario quiero poder instalar la aplicacion en el celular para usarla de forma "nativa"	DONE	Done
TES-378	Como usuario quiero poder configurar una respuesta precargada basada en los templates y en los tags para poder contestar más rápido	DONE	Done
TES-385	Como usuario quiero poder tener un dashboard para poder ver estadísticas de mis preguntas	TO DO	Unresolved
TES-399	Asegurar la portabilidad del sistema	TO DO	Unresolved
TES-400	El sistema debe ser modificable	TO DO	Unresolved

T	Key ↑	Summary	Status	Resolution
	TES-401	Requerimientos de Software	TO DO	Unresolved
	TES-402	Requerimientos de Hardware	TO DO	Unresolved
	TES-403	Modelado de ML	DONE	Done
	TES-417	Documentación Final	TO DO	Unresolved
	TES-433	Utilizar REST como comunicación entre componentes	TO DO	Unresolved

12.2. Issues

A continuación, disponemos de los *issues* ocasionados entre los diferentes *sprints* y *release* uno incluidos. Estos fueron reportados desde Jira.

TES-140	Fix de doble render	DONE	Done
TES-141	En la paginacion, el backend devuelve preguntas de todos los clientes en la primera y la ultima pagina.	DONE	Done
TES-142	La división en páginas del paginado da resultados incorrectos	DONE	Done
TES-172	Vista de historial ordena al revés por fecha	DONE	Done
TES-174	Textos no centrados en vista de historial	DONE	Done
TES-175	Eliminar filtro "Sin responder" en el historial	DONE	Done
TES-176	Información de variaciones se salen del modal	DONE	Done
TES-177	Fecha con menos de 24hs sale como fecha normal	DONE	Done
TES-178	Paginado lento cuando vas a una página anterior desde la última pagina	DONE	Done
TES-179	Paginado, si aplico un filtro en un numero de página posterior al resultado falla	DONE	Done
TES-180	En modal de filtrado de preguntas no debería ejecutarse el filtro a las preguntas si ya está en ese filtro	DONE	Done
TES-181	<u>Al responder preguntas, quedan huecos entre preguntas en la vista de preguntas sin responder</u>	DONE	Done
TES-182	El botón "Responder" falla en algunos casos	DONE	Done
TES-183	Si hay dos preguntas expandidas al mismo tiempo y se contesta la de arriba, queda expandida la que está en la misma posición	DONE	Done
TES-184	Pool de preguntas cuando estoy en última página, tiene pocas preguntas	DONE	Done
TES-262	Al editar template, el botón para editar funciona la primera vez, luego falla	DONE	Done
TES-318	Hotkeys problemas al apretar dos veces la misma combinación	DONE	Done
TES-347	Last user action se ve en questions	DONE	Done
TES-357	En vista de preguntas, el overflow anda mal con muchos clicks	DONE	Done
TES-358	El valor y nombre de los atributos deben ser String	DONE	Done
TES-427	Routing por defecto no funciona	DONE	Done
TES-428	En vista de historial de preguntas, expandir el detalle se muestra vacío	DONE	Done
TES-429	Mostrar Intercom no funciona	DONE	Done
TES-430	A veces el backend devuelve estado 403	DONE	Done
TES-431	La imagen por defecto en artículos sin imágenes no se muestra	DONE	Done

12.3. User Stories

Historia de usuario TES-3	
Título	<i>Front</i> - Como usuario quiero visualizar todas las preguntas para poder responderlas una tras otra
Descripción	-
Criterios de aceptación	<p>Visualización sin estar autenticado en el sistema:</p> <ul style="list-style-type: none"> • Dado que no estoy autenticado en Sagal • Cuando entro a la sección de preguntas • Entonces me redirige a la página principal de Sagal <p>Visualización estando autenticado en el sistema:</p> <ul style="list-style-type: none"> • Dado que estoy autenticado en Sagal • Cuando entro a la sección de preguntas • Entonces visualizo las primeras 20 preguntas sin responder <p>Visualización estando autenticado en el sistema:</p> <ul style="list-style-type: none"> • Dado que estoy autenticado en Sagal • Cuando entro a la sección de preguntas • Entonces visualizo las primeras 20 preguntas de las más antiguas a las más nuevas
Actividad	Desarrollo
Estimación	5
Componentes	<i>Frontend</i>

Historia de usuario TES-4	
Título	Como usuario al visualizar una pregunta quiero poder visualizar todos los datos relacionados para poder contestar fácilmente
Descripción	-
Criterios de aceptación	<p>Frontend</p> <p>Expansión:</p> <ul style="list-style-type: none"> • Dado que estoy visualizando las preguntas • Cuando selecciono una pregunta • Entonces se expande la pregunta con detalle de la misma <p>Visualización de la imagen del artículo:</p> <ul style="list-style-type: none"> • Dado que estoy en la sección de preguntas • Cuando selecciono la primera pregunta • Entonces puedo ver la imagen relacionada al artículo <p>Visualización de los detalles del artículo:</p> <ul style="list-style-type: none"> • Dado que estoy en la sección de preguntas • Cuando selecciono la primera pregunta • Entonces puedo ver los detalles relacionados al artículo <p>Visualización de pregunta:</p> <ul style="list-style-type: none"> • Dado que estoy en la sección de preguntas • Cuando selecciono la primera pregunta

	<ul style="list-style-type: none"> • Entonces puedo ver la pregunta <p>Visualización de campo para responder:</p> <ul style="list-style-type: none"> • Dado que estoy en la sección de preguntas • Cuando selecciono la primera pregunta • Entonces puedo ver el campo para responder una pregunta <p>Visualización de más detalles:</p> <ul style="list-style-type: none"> • Dado que estoy en la sección de preguntas • Cuando selecciono la primera pregunta • Entonces puedo ver más detalles acerca de la pregunta <p>Backend</p> <p>Mostrar información de artículo:</p> <ul style="list-style-type: none"> • Dado que tengo una pregunta para responder • Cuando obtengo la primera página de las preguntas • Entonces obtengo la pregunta con la información del artículo
Actividad	Desarrollo
Estimación	8
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-5	
Título	Como usuario al visualizar una pregunta quiero poder visualizar el nombre y apellido del usuario que me consulta para identificar al usuario que pregunta
Descripción	-
Criterios de aceptación	<p>Frontend</p> <p>Mostrar datos de cliente:</p> <ul style="list-style-type: none"> • Dado que quiero contestar una pregunta • Cuando la despliego • Entonces tengo que poder ver los datos del cliente <p>Backend</p> <p>Mostrar datos de cliente:</p> <ul style="list-style-type: none"> • Dado que tengo 1 pregunta para responder • Cuando obtengo la primera página de preguntas a responder • Entonces puedo ver la pregunta con la información del cliente
Actividad	Desarrollo
Estimación	2
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-8	
Título	Como usuario quiero poder tener un paginado de preguntas a responder para poder navegar entre ellas
Descripción	-
Criterios de aceptación	<p>Backend y Frontend</p> <p>Ir hacia la próxima página:</p> <ul style="list-style-type: none"> • Dado que hay 500 preguntas en la sección y solo se muestran 15 por página y estoy en la primera pagina • Cuando cambio a la segunda pagina • Entonces solo veré las preguntas de la posición 16 a la 30 <p>Ir hacia la página anterior:</p> <ul style="list-style-type: none"> • Dado que hay 500 preguntas en la sección y solo se muestran 15 por página y estoy en la cuarta pagina • Cuando voy a la tercer pagina • Entonces solo veré las preguntas de la posición 31 a la 45 <p>Cambiar a la primera página:</p> <ul style="list-style-type: none"> • Dado que hay 500 preguntas en la sección y solo se muestran 15 por página y estoy en la página 5 • Cuando cambio a la primera pagina

	<ul style="list-style-type: none"> • Entonces solo veré las preguntas de la posición 1 a la 15 <p>Cambiar a la última página:</p> <ul style="list-style-type: none"> • Dado que hay 500 preguntas en la sección y solo se muestran 15 por página y estoy en la página 5 • Cuando cambio a la primera pagina • Entonces solo veré las preguntas de la posición 496 a la 500
Actividad	Desarrollo
Estimación	21
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-9	
Título	Como usuario quiero poder visualizar todas las preguntas ya respondidas para tener un historial de las mismas además de un seguimiento
Descripción	-
Criterios de aceptación	<p>Frontend</p> <p>Visualizar preguntas sin estar autenticado en el sistema:</p> <ul style="list-style-type: none"> • Dado que no estoy autenticado en Sagal • Cuando voy a la sección de historial • Entonces me redirecciona a la página de <i>login</i> <p>Visualizar preguntas estando autenticado en el sistema:</p> <ul style="list-style-type: none"> • Dado que estoy autenticado en Sagal y tengo 20 preguntas respondidas • Cuando voy a la sección de historial • Entonces veo las 20 preguntas <p>Visualizar las preguntas más recientes primero:</p> <ul style="list-style-type: none"> • Dado que estoy autenticado en Sagal y tengo 20 preguntas respondidas • Cuando voy a la sección de historial • Entonces veo las primeras 20 preguntas ordenadas por fecha más recientes primero <p>Backend</p>

	<p>Responder directamente desde Mercado Libre y la pregunta no existe en el <i>backend</i>:</p> <ul style="list-style-type: none"> • Dado que tengo 1 pregunta para responder y no existe en el <i>backend</i> • Cuando respondo la pregunta directamente en Mercado Libre • Entonces Sagal notifica al <i>backend</i> con una pregunta nueva y se guarda en el historial de preguntas <p>Responder pregunta directamente desde Mercado Libre y la pregunta existe en el <i>backend</i>:</p> <ul style="list-style-type: none"> • Dado que tengo 1 pregunta para responder y existe en el <i>backend</i> como “sin responder” • Cuando respondo la pregunta directamente en Mercado Libre • Entonces Sagal notifica al <i>backend</i> con una pregunta nueva, se guarda en el historial de preguntas y se elimina de la sección de preguntas para responder <p>Visualizar preguntas en historial:</p> <ul style="list-style-type: none"> • Dado que tengo 10 preguntas respondidas • Cuando hago un <i>request</i> GET a questions/historical • Entonces puedo ver las 10 preguntas ordenadas por fecha más reciente primero
Actividad	Desarrollo

Estimación	8
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-10	
Título	Como usuario quiero poder usar la aplicación desde el celular para poder contestar las preguntas desde ahí
Descripción	-
Criterios de aceptación	<p>Interfaz adaptada:</p> <ul style="list-style-type: none"> • Dado que estoy accediendo desde un dispositivo móvil • Cuando navego a Sagal • Entonces me abre la sección de preguntas con una interfaz adaptada para dispositivos móviles <p>Mover a la siguiente página de preguntas:</p> <ul style="list-style-type: none"> • Dado que estoy accediendo desde un dispositivo móvil y accedo a la sección de preguntas • Cuando me desplazo hacia abajo y sobrepaso la cantidad de preguntas de la primera pagina • Entonces me carga la próxima página de preguntas <p>Mover a la anterior página de preguntas:</p>

- Dado que estoy accediendo desde un dispositivo móvil y accedo a la sección de preguntas
- Cuando me desplazo hacia arriba y sobrepaso la pregunta de más arriba.
- Entonces me carga la página anterior de preguntas

Visualizar preguntas sin responder:

- Dado que estoy accediendo desde un dispositivo móvil
- Cuando navego a Sagal
- Entonces visualizo un listado de preguntas sin responder.

Seleccionar pregunta:

- Dado que estoy accediendo desde un dispositivo móvil y estoy en la sección de preguntas
- Cuando selecciono una pregunta
- Entonces se abre una nueva vista con los datos de la pregunta.

Filtrar preguntas:

- Dado que estoy accediendo desde un dispositivo móvil y estoy en la sección de preguntas
- Cuando aprieto en el icono de filtros
- Entonces se abre la ventana emergente de filtros de preguntas

	<p>Visualizar detalles de pregunta:</p> <ul style="list-style-type: none"> • Dado que estoy accediendo desde un dispositivo móvil y tengo seleccionada una pregunta • Cuando aprieto en el icono de detalles extras • Entonces se abre la ventana emergente de detalles asociada a la pregunta <p>Responder pregunta:</p> <ul style="list-style-type: none"> • Dado que estoy accediendo desde un dispositivo móvil y tengo seleccionada una pregunta • Cuando aprieto en el botón de responder • Entonces se responde la pregunta <p>Acceder a <i>Intercomm</i>:</p> <ul style="list-style-type: none"> • Dado que estoy accediendo desde un dispositivo móvil • Cuando aprieto en el icono de detalles extras • Entonces se abre la ventana emergente del chat con soporte
Actividad	Desarrollo
Estimación	21
Componentes	<i>Frontend</i>

Historia de usuario TES-11	
Título	Como usuario quiero poder visualizar las preguntas más antiguas sin contestar primero para responderlas lo antes posible
Descripción	-
Criterios de aceptación	<p>Backend y Frontend</p> <p>Orden de preguntas:</p> <ul style="list-style-type: none"> • Dado que hay 20 preguntas pendientes para contestar • Cuando voy a contestar las preguntas • Entonces visualizo las preguntas por orden, siendo las más antiguas primero
Actividad	Desarrollo
Estimación	2
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-12	
Título	Como usuario quiero al contestar una pregunta que desaparezca para visualizar solo las no respondidas
Descripción	-
Criterios de aceptación	<p>Responder una pregunta:</p> <ul style="list-style-type: none"> • Dado que quiero contestar una pregunta sin responder y estoy en la sección de preguntas

	<ul style="list-style-type: none"> • Cuando la contesto • Entonces desaparece con un efecto de la lista de preguntas
Actividad	Desarrollo
Estimación	3
Componentes	<i>Frontend</i>

Historia de usuario TES-13	
Título	Como usuario quiero poder filtrar las preguntas para poder buscar las que son más relevantes
Descripción	-
Criterios de aceptación	<p>Backend y Frontend</p> <p>Filtrado por código:</p> <ul style="list-style-type: none"> • Dado que tengo 10 preguntas con el código de artículo 123 en un total de 500 preguntas • Cuando solicito el filtro por el código de artículo 123 • Entonces veré las 10 preguntas con el código 123 <p>Filtrado por Título:</p> <ul style="list-style-type: none"> • Dado que tengo 10 preguntas con el título "Test" en un total de 500 preguntas • Cuando solicito el filtro por título "Test"

	<ul style="list-style-type: none">• Entonces veré las 10 preguntas con el título "Test" <p>Filtrado por cliente:</p> <ul style="list-style-type: none">• Dado que tengo 10 preguntas del cliente "Pepe" en un total de 500 preguntas• Cuando solicito el filtro por cliente "Pepe"• Entonces veré las 10 preguntas del cliente "Pepe" <p>Filtrado por texto de pregunta:</p> <ul style="list-style-type: none">• Dado que tengo 10 preguntas con el texto "Permutas por libro" en un total de 500 preguntas• Cuando solicito el filtro por texto "Permutas por libro"• Entonces veré las 10 preguntas con el texto "Permutas por libro" <p>Filtrado por estado:</p> <ul style="list-style-type: none">• Dado que tengo 10 preguntas con el estado "Eliminado" en un total de 500 preguntas• Cuando solicito el filtro por estado "Eliminado"• Entonces veré las 10 preguntas del cliente "Eliminado" <p>Filtrado por etiquetas:</p> <ul style="list-style-type: none">• Dado que tengo 10 preguntas con la etiqueta "Stock" en un total de 500 preguntas• Cuando solicito el filtro por etiqueta "Stock"
--	---

	<ul style="list-style-type: none"> • Entonces veré las 10 preguntas de etiqueta "Stock" <p>Filtrado por múltiples campos:</p> <ul style="list-style-type: none"> • Dado que tengo 10 preguntas con: <ul style="list-style-type: none"> ○ Estado "Eliminado" ○ Cliente "Pepe" ○ Título "Horno" ○ Código "123" • Cuando solicito los filtros: <ul style="list-style-type: none"> ○ Estado "Eliminado" ○ Cliente "Pepe" ○ Título "Horno" ○ Código "123" • Entonces tengo que ver solo las preguntas con los filtros seleccionados.
Actividad	Desarrollo
Estimación	2
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-14	
Título	Como usuario quiero poder contestar una pregunta para responderle a los clientes finales
Descripción	-
Criterios de aceptación	<p>Frontend</p> <p>Responder una pregunta:</p> <ul style="list-style-type: none"> • Dado que estoy en la sección de preguntas • Cuando ingreso a la primera pregunta y la pregunta se expande • Entonces puedo escribir la respuesta, se envía y la pregunta desaparece con un efecto en pantalla <p>Cargar más preguntas:</p> <ul style="list-style-type: none"> • Dado que respondí una pregunta • Cuando existen más de 15 preguntas para responder • Entonces una nueva pregunta se carga <p>Backend</p> <p>Responder pregunta:</p> <ul style="list-style-type: none"> • Dado que existe 1 pregunta para responder • Cuando respondo la pregunta con “tenemos stock” • Entonces la pregunta se envía a responder y se guarda en el historial de preguntas y se elimina de las preguntas para contestar

Responder una pregunta ya respondida:

- Dado que existe 1 pregunta en el historial de preguntas
- Cuando respondo esa pregunta con “tenemos stock”
- Entonces devuelve mensaje de que no existe la pregunta y no se responde la pregunta

Responder una pregunta no existente:

- Dado que no tengo preguntas para responder
- Cuando quiero responder una inexistente
- Entonces devuelve mensaje de que la pregunta no existe

Responder una pregunta sin mensaje de respuesta:

- Dado que tengo una pregunta sin responder
- Cuando respondo sin mensaje de respuesta
- Entonces devuelve mensaje de que debe tener una respuesta y la pregunta no se responde

Responder pregunta de otro cliente:

- Dado que el cliente número 900 y tiene una pregunta para responder y yo soy el cliente número 1
- Cuando intento responder la pregunta del otro cliente con “tenemos stock”

	<ul style="list-style-type: none"> Entonces devuelve mensaje que no existe pregunta para responder y la pregunta no se responde.
Actividad	Desarrollo
Estimación	8
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-23	
Título	Como usuario quiero que las preguntas puedan catalogarse correctamente en diferentes etiquetas para poder responderlas más fácil en el futuro
Descripción	-
Criterios de aceptación	Catalogar pregunta: <ul style="list-style-type: none"> Dado que llega una nueva pregunta al sistema Cuando se termina de crear y se manda a catalogar a la interfaz de <i>machine learning</i> Entonces debería predecir las etiquetas con mayor coincidencia del modelo pre aprendido.
Actividad	Desarrollo
Estimación	34
Componentes	<i>Backend</i>

Historia de usuario TES-42	
Título	Conexión e interfaz con Sagal
Descripción	Esta user story hace referencia a la creación y el detalle de las interfaces que realizamos para que Sagal pudiera comunicarse con nosotros.
Criterios de aceptación	<p>Notificación de pregunta inexistente:</p> <ul style="list-style-type: none"> • Dado que se genera una nueva pregunta en Sagal • Cuando se notifica a nuestro <i>Backend</i> • Entonces la pregunta se almacena y se responde a Sagal <p>Notificación de pregunta existente:</p> <ul style="list-style-type: none"> • Dado que se produce un cambio en una pregunta existente en Sagal Y existen en nuestro <i>Backend</i> • Cuando se notifica a nuestro <i>Backend</i> • Entonces la pregunta se actualiza a los datos enviados por Sagal <p>Notificación de artículo existente:</p> <ul style="list-style-type: none"> • Dado que se produce un cambio en un artículo en Sagal Y está asociado a una pregunta de nuestro <i>Backend</i> • Cuando se notifica a nuestro <i>Backend</i> • Entonces las preguntas asociadas a ese artículo se actualizan a los datos enviados por Sagal

	<p>Notificación de artículo inexistente:</p> <ul style="list-style-type: none"> • Dado que se produce un cambio en un artículo en Sagal Y no está asociada a ninguna pregunta almacenada previamente en nuestro <i>Backend</i> • Cuando se notifica a nuestro <i>Backend</i> • Entonces se responde con 404 a Sagal
Actividad	Desarrollo
Estimación	8
Componentes	<i>Backend</i>

Historia de usuario TES-48	
Título	Back - Como usuario quiero visualizar todas las preguntas para poder responderlas una tras otra
Descripción	-
Criterios de aceptación	<p>Acceso sin token:</p> <ul style="list-style-type: none"> • Dado que no tengo token. • Cuando realizo una <i>request</i> a “api/questions”. • Entonces me devuelve un mensaje de token invalido. <p>Acceso con token incorrecto:</p> <ul style="list-style-type: none"> • Dado que tengo token incorrecto. • Cuando realizo una <i>request</i> a “api/questions”. • Entonces me devuelve un mensaje de token invalido. <p>Obtener preguntas:</p> <ul style="list-style-type: none"> • Dado que tengo token valido. • Cuando realizo una <i>request</i> a “api/questions” con el número de página 1 y limite 20. • Entonces respondo con las primeras 20 preguntas.
Actividad	Desarrollo
Estimación	3
Componentes	<i>Backend</i>

Historia de usuario TES-60	
Título	Como administrador quiero que la aplicación se pruebe automáticamente para tener confianza en la aplicación
Descripción	-
Criterios de aceptación	Realización de pruebas en Cucumber, Jest y uso de Gherkin para cada user story de los diferentes sprints.
Actividad	Testing
Estimación	13
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-65	
Título	Asegurar la performance de la aplicación
Descripción	Asegurar el rendimiento de performance <2 segundos para 100 usuarios concurrentes
Criterios de aceptación	-
Actividad	Desarrollo
Estimación	0
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-66	
Título	Asegurar que el <i>frontend</i> de la aplicación sea reutilizable
Descripción	Base para reconstruir el <i>frontend</i> de la aplicación actual. Módulos genéricos y claros para poder ser reutilizados. Incorporar todos los pedidos del cliente con respecto a estilos.
Criterios de aceptación	-
Actividad	Desarrollo
Estimación	0
Componentes	<i>Frontend</i>

Historia de usuario TES-67	
Título	Asegurar la usabilidad respetando las métricas y pantallas acordadas
Descripción	<ul style="list-style-type: none"> • Debe ser <i>responsive</i> • No más de 2 pasos para contestar una pregunta. • La información necesaria para contestar debe ser desplegada.
Criterios de aceptación	-
Actividad	Desarrollo

Estimación	0
Componentes	<i>Frontend</i>

Historia de usuario TES-81	
Título	Preparar el primer <i>release</i>
Descripción	Tareas necesarias para el primer release, esto incluye prueba de embebido de aplicación <i>frontend</i> en un <i>Iframe</i> , generar scripts de <i>deploy</i> para los diferentes componentes, realizar los ajustes necesarios para el <i>enviroment</i> de producción.
Criterios de aceptación	-
Actividad	Desarrollo
Estimación	21
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-85	
Título	Como usuario quiero agrupar las preguntas por articulo para tener las preguntas más ordenadas
Descripción	Esta <i>user story</i> se decidió dejarla para después del proyecto académica por decisión del cliente
Criterios de aceptación	-
Actividad	Desarrollo
Estimación	No estimado hasta ahora, ya que no entro en ninguno de los <i>sprints</i>
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-86	
Título	Aplicar uso de <i>linter</i>
Descripción	-
Criterios de aceptación	Uso de <i>linter</i> para asegurar <i>standards</i> requeridos por el cliente.
Actividad	Desarrollo
Estimación	0
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-87	
Título	Como usuario quiero poder comunicarme con el soporte técnico para resolver mis dudas
Descripción	-
Criterios de aceptación	<p>Comunicarse con <i>intercom</i>:</p> <ul style="list-style-type: none"> • Dado que quiero comunicarme con soporte. • Cuando hago <i>click</i> en el botón de <i>intercom</i>. • Entonces tengo que poder conectarme con el soporte. <p>Abrir <i>helpdesk</i>:</p> <ul style="list-style-type: none"> • Dado que quiero ir al <i>helpdesk</i>. • Cuando hago <i>click</i> en el botón de <i>helpdesk</i>. • Entonces me abre una nueva pestaña con el <i>enlace</i> de <i>helpdesk</i> de Sagal.
Actividad	Desarrollo
Estimación	2
Componentes	<i>Frontend</i>

Historia de usuario TES-186	
Título	Como usuario quiero ver más detalles del artículo a demanda para poder responder las preguntas más específicas
Descripción	-
Criterios de aceptación	<p>Visualizar detalles extra del artículo asociado:</p> <ul style="list-style-type: none"> • Dado que estoy en la sección de preguntas • Cuando selecciono la primera Y hago <i>click</i> en el botón de “más detalles”. • Entonces puedo visualizar más detalles sobre el artículo relacionado a la pregunta
Actividad	Desarrollo
Estimación	3
Componentes	<i>Frontend</i>

Historia de usuario TES-187	
Título	Como usuario quiero poder eliminar preguntas para descartar las que perjudiquen mis publicaciones
Descripción	-
Criterios de aceptación	<p>Frontend</p> <p>Eliminar pregunta:</p> <ul style="list-style-type: none"> • Dado que tengo seleccionada una pregunta • Cuando hago click en eliminar y confirmo la eliminación • Entonces se elimina la pregunta <p>Backend</p> <p>Eliminar pregunta:</p> <ul style="list-style-type: none"> • Dado que tengo una pregunta sin responder • Cuando quiero eliminar esa pregunta • Entonces se elimina en Mercado Libre y se elimina de las preguntas pendientes a responder <p>Eliminar pregunta que no existe:</p> <ul style="list-style-type: none"> • Dado que no tengo preguntas para responder • Cuando quiero eliminar alguna pregunta • Entonces devuelve mensaje de pregunta no existente y no se borra ninguna pregunta <p>Eliminar pregunta con estado invalido:</p>

	<ul style="list-style-type: none"> • Dado que tengo una pregunta con estado respondida • Cuando quiero eliminar esa pregunta • Entonces devuelve mensaje de pregunta no existente y no elimina la pregunta
Actividad	Desarrollo
Estimación	3
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-188									
Título	Como usuario quiero poder utilizar <i>hotkeys</i> en el uso del sistema para que mejore mis tiempos de respuesta								
Descripción	-								
Criterios de aceptación	Uso de <i>hotkey</i>: <ul style="list-style-type: none"> • Dado que estoy utilizando la aplicación • Cuando aprieto la hotket "key-1" y "key-2" • Entonces se produce la acción "action" <table border="1" data-bbox="628 1520 1366 1865"> <thead> <tr> <th>Key-1</th> <th>Key-2</th> <th>action</th> </tr> </thead> <tbody> <tr> <td>Ctrl</td> <td>enter</td> <td>Realizar la acción del formulario seleccionado</td> </tr> </tbody> </table>			Key-1	Key-2	action	Ctrl	enter	Realizar la acción del formulario seleccionado
Key-1	Key-2	action							
Ctrl	enter	Realizar la acción del formulario seleccionado							

	Shift	c	Collapsar todas las preguntas
	Shift	e	Expandir todas las preguntas
	Ctrl	d	Mostrar los detalles de la pregunta seleccionada
	Ctrl	f	Abrir los filtros
	ctrl	arrowdown	Moverse a la siguiente pregunta a contestar
	Ctrl	arrowup	Moverse a la pregunta anterior a contestar
	Ctrl	p	Ir a la pregunta anterior a contestar
	Ctrl	space	Ir a la primer pregunta a contestar
	escape		Cierra los <i>modals</i> abiertos

Actividad	Desarrollo
Estimación	8
Componentes	<i>Frontend</i>

Historia de usuario TES-189	
Título	Como usuario quiero poder dar de baja <i>templates</i> para dejar de utilizarlos al responder preguntas
Descripción	-
Criterios de aceptación	<p><i>Frontend</i></p> <p>Eliminar una plantilla:</p> <ul style="list-style-type: none"> • Dado que estoy en la seccion de plantillas y tengo una plantilla creada • Cuando hago <i>click</i> en el icono de eliminar de la plantilla • La plantilla desaparece y es eliminada <p><i>Backend</i></p> <p>Eliminar plantilla:</p> <ul style="list-style-type: none"> • Dado que tengo una plantilla • Cuando la elimino • Entonces devuelve mensaje de eliminado y la plantilla es eliminada <p>Eliminar plantilla no existente:</p> <ul style="list-style-type: none"> • Dado que no tengo plantillas creadas

	<ul style="list-style-type: none"> • Cuando quiero eliminar una • Entonces responde con mensaje de error y no se elimina ninguna plantilla
Actividad	Desarrollo
Estimación	3
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-190	
Título	Como usuario quiero poder editar <i>templates</i> para poder responder más rápido
Descripción	-
Criterios de aceptación	<p>Backend</p> <p>Actualizar <i>template</i></p> <ul style="list-style-type: none"> • Dado que hay 1 <i>template</i> creado • Cuando edito el <i>template</i> con otro texto • Entonces se responde que el <i>template</i> fue actualizado correctamente <p>Actualizar <i>template</i> no existente</p> <ul style="list-style-type: none"> • Dado que no hay <i>templates</i> • Cuando envié a editar un <i>template</i> • Entonces se responde con el error de que no existe el <i>template</i>

	<p>Actualizar <i>template</i> con nombre de <i>template</i> existente</p> <ul style="list-style-type: none"> • Dado que hay 2 <i>templates</i> creados, uno tiene "stock" como nombre y otro "precio" • Cuando cuando envié a cambiar el nombre del <i>template</i> "precio" a "stock" • Entonces se responde que el nombre del <i>template</i> ya existe <p>Frontend</p> <p>Actualizar <i>template</i></p> <ul style="list-style-type: none"> • Dado que estoy en la sección de <i>templates</i> • Cuando hago click en editar el primer <i>template</i> • Entonces se abre el model con los datos del <i>template</i> a editar
Actividad	Desarrollo
Estimación	3
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-191	
Título	Como usuario quiero poder ver los <i>templates</i> para luego poder administrarlas
Descripción	-
Criterios de aceptación	<p>Frontend</p> <p>Visualizar plantillas:</p> <ul style="list-style-type: none"> • Dado que estoy autenticado en Sagal y tengo 5 plantillas creadas • Cuando ingreso a la sección de plantillas • Entonces puedo ver las 5 plantillas <p>Backend</p> <p>Obtener plantillas:</p> <ul style="list-style-type: none"> • Dado que tengo 2 plantillas creadas • Cuando hago un <i>request</i> a <i>/api/templates</i> • Entonces me devuelve la información de las 2 plantillas
Actividad	Desarrollo
Estimación	3
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-192	
Título	Como usuario quiero poder cambiar de páginas entre los diferentes <i>templates</i> que están creados para poder navegar eficazmente
Descripción	-
Criterios de aceptación	<p>Ir a la primera pagina</p> <ul style="list-style-type: none"> • Dado que tengo 19 <i>templates</i> creados • Cuando voy a la primera página con límite de 10 • Entonces se responde con los primeros 10 <i>templates</i> <p>Ir a la última pagina</p> <ul style="list-style-type: none"> • Dado que tengo 19 <i>templates</i> creados • Cuando voy a la última página con límite de 10 • Entonces se responde con los últimos 9 <i>templates</i> <p>Ir fuera de las posibles paginas</p> <ul style="list-style-type: none"> • Dado que tengo 19 <i>templates</i> creados • Cuando voy a la tercera página con límite de 10 • Entonces se responde sin <i>templates</i>
Actividad	Desarrollo
Estimación	2
Componentes	<i>Backend</i>

Historia de usuario TES-193	
Título	Como usuario quiero poder contestar preguntas masivamente para poder responder las mismas preguntas una única vez
Descripción	Esta <i>user story</i> se decidió dejarla para después del proyecto académica por decisión del cliente
Criterios de aceptación	-
Actividad	Desarrollo
Estimación	No estimado hasta ahora, ya que no entro en ninguno de los <i>sprints</i>
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-194	
Título	Como usuario quiero poder eliminar preguntas masivamente para descartar varias preguntas que perjudiquen mis publicaciones
Descripción	Esta <i>user story</i> se decidió dejarla para después del proyecto académica por decisión del cliente
Criterios de aceptación	-
Actividad	Desarrollo

Estimación	No estimado hasta ahora, ya que no entro en ninguno de los <i>sprints</i>
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-195	
Título	Como usuario quiero poder dar de alta <i>templates</i> para utilizarlos al responder preguntas
Descripción	-
Criterios de aceptación	<p>Frontend</p> <p>Visualizar crear una plantilla:</p> <ul style="list-style-type: none"> • Dado que estoy en la sección de plantillas • Cuando hago <i>click</i> en el botón de crear • Entonces voy a ver el formulario de creación de plantillas <p>Plantilla invalida:</p> <ul style="list-style-type: none"> • Dado que estoy en el formulario para crear plantillas y no pongo nombre de plantilla • Cuando hago <i>click</i> en el botón de guardar • Entonces devuelve un mensaje de error y la plantilla no es creada <p>Guardar plantilla:</p> <ul style="list-style-type: none"> • Dado que estoy en el formulario para crear plantillas y completo los datos correctamente • Cuando hago <i>click</i> en el botón de guardar

- Entonces devuelve un mensaje de éxito y puedo ver la plantilla creada

Backend

Crear plantilla:

- Dado que no tengo plantilla
- Cuando creo una plantilla con el nombre “stock”
- Entonces la plantilla es creada

Crear una plantilla existente:

- Dado que tengo una plantilla creada con el nombre “stock”
- Cuando creo otra plantilla con el mismo nombre
- Entonces devuelve mensaje de que la plantilla ya existe y la plantilla no es creada

Crear una plantilla sin contenido:

- Dado que no tengo plantillas
- Cuando creo una plantilla de nombre “stock” pero sin contenido
- Entonces devuelve mensaje de error y la plantilla no es creada

Crear plantilla sin nombre:

- Dado que no tengo plantillas
- Cuando quiero crear una plantilla sin nombre

	<ul style="list-style-type: none"> Entonces devuelve mensaje de error y la plantilla no es creada
Actividad	Desarrollo
Estimación	5
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-196	
Título	Como usuario quiero poder visualizar el historial de preguntas realizadas de un usuario en una misma publicación para poder seguir el hilo de una conversación
Descripción	-
Criterios de aceptación	<p>Frontend</p> <p>Visualizar botón para mostrar conversación previa</p> <ul style="list-style-type: none"> Dado que estoy en la sección de preguntas y la primera pregunta pertenece a un artículo que ya recibió preguntas de ese usuario Cuando selecciono la primera pregunta Entonces visualizo el botón para expandir la conversación <p>Expandir conversación</p> <ul style="list-style-type: none"> Dado que estoy visualizando una pregunta que posee una conversación previa

	<ul style="list-style-type: none"> • Cuando hago <i>click</i> mostrar conversación • Entonces visualizo la conversación anterior, es decir las preguntas realizadas, así como mis respuestas previas. <p>Backend</p> <p>Obtener preguntas del mismo cliente:</p> <ul style="list-style-type: none"> • Dado que tengo 2 preguntas respondidas del cliente “Pepe” sobre el artículo 1 y una pregunta sin responder del mismo cliente sobre el artículo 1 • Cuando obtengo las preguntas relacionadas de la pregunta sin responder • Entonces me devuelve las preguntas respondidas
Actividad	Desarrollo
Estimación	8
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-233	
Título	Reuniones de tutoría
Descripción	Las reuniones realizadas con nuestro tutor Álvaro Ortas
Criterios de aceptación	-
Actividad	Gestión

Estimación	1
Componentes	-

Historia de usuario TES-237	
Título	Reuniones con Sagal
Descripción	Reuniones realizadas junto a la gente de Sagal, generalmente realizamos una cada 1 o 2 semanas
Criterios de aceptación	-
Actividad	Gestión
Estimación	1
Componentes	-

Historia de usuario TES-253	
Título	Como usuario quiero poder ver quien realizo las acciones sobre las preguntas para poder hacerles un seguimiento
Descripción	-
Criterios de aceptación	<p>Frontend</p> <p>Visualizar acciones del usuario en el historial</p> <ul style="list-style-type: none"> • Dado que estoy en la sección del historial

	<ul style="list-style-type: none"> • Cuando hago <i>click</i> en la primera pregunta historial y luego <i>click</i> en el botón de historial de acciones • Entonces visualizo cada acción realizada sobre la pregunta y el usuario asociado a la misma <p>Backend</p> <p>Almacenar la última acción de una pregunta</p> <ul style="list-style-type: none"> • Dado que tengo 1 pregunta a responder • Cuando el usuario de id 100 responde la pregunta con "Tenemos stock" • Entonces la pregunta es almacenada con la última acción como respondida y con el id de usuario como 100 <p>Visualizar las acciones sobre una pregunta</p> <ul style="list-style-type: none"> • Dado que tengo una pregunta con el id 1 respondida por el usuario de id 100 • Cuando se recibe una consulta a <code>"/api/questions/:1/actions"</code> • Entonces se responde las acciones asociadas a la pregunta 1
Actividad	Desarrollo
Estimación	5
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-263	
Título	Como usuario quiero recibir notificaciones sobre las acciones que estoy realizando para mejorar la usabilidad
Descripción	-
Criterios de aceptación	<p>Mostrar errores</p> <ul style="list-style-type: none"> • Dado que estoy autenticado • Cuando se produce un error en el sistema • Entonces se abre una notificación de error abajo a la derecha <p>Mostrar acciones realizadas correctamente</p> <ul style="list-style-type: none"> • Dado que estoy autenticado • Cuando realizo una acción correcta en el sistema • Entonces se abre una notificación abajo a la derecha
Actividad	Desarrollo
Estimación	8
Componentes	<i>Frontend</i>

Historia de usuario TES-266	
Título	Como usuario quiero poder visualizar los tags asociadas a una pregunta para poder identificar sobre que trata rápidamente
Descripción	-
Criterios de aceptación	Visualizar tags de las preguntas: <ul style="list-style-type: none"> • Dado que estoy autenticado en Sagal • Cuando voy a la sección de preguntas • Entonces puedo ver los tags asociadas a las preguntas
Actividad	Desarrollo
Estimación	5
Componentes	<i>Frontend</i>

Historia de usuario TES-267	
Título	Como administrador quiero poder cambiar la implementación de los servicios de <i>machine learning</i> para utilizar otro modelo si lo deseo
Descripción	Esta <i>user story</i> fue incluida en TES-23, decidimos dejarla ya que en un principio pensamos en realizarla como una <i>user story</i> separada
Criterios de aceptación	-
Actividad	Desarrollo

Estimación	0
Componentes	<i>Backend</i>

Historia de usuario TES-268	
Título	Como usuario quiero ver cuáles son los <i>hotkeys</i> disponibles para poder usarlos
Descripción	-
Criterios de aceptación	<p>Mostrar ayuda de <i>hotkeys</i></p> <ul style="list-style-type: none"> • Dado que estoy en la sección de preguntas • Cuando voy al botón de ayuda de <i>hotkeys</i> • Entonces puedo visualizar las <i>hotkeys</i> disponibles
Actividad	Desarrollo
Estimación	3
Componentes	<i>Frontend</i>

Historia de usuario TES-277	
Título	Como usuario quiero poder buscar las preguntas con determinado tag para filtrar mejor
Descripción	-
Criterios de aceptación	<p>Frontend</p> <p>Filtrar preguntas por tags:</p> <ul style="list-style-type: none"> • Dado que estoy en la sección de preguntas y tengo 10 preguntas con tag stock y 10 preguntas con tag garantía • Cuando elijo el filtro de tag y elijo el valor stock • Entonces veo solo las preguntas que tengan el tag stock <p>Backend</p> <p>Filtrar por tag de la pregunta:</p> <ul style="list-style-type: none"> • Dado que tengo 10 preguntas con el tag stock y 10 preguntas con el tag garantía • Cuando filtro las preguntas por el tag de stock • Entonces visualizo 10 preguntas con el tag stock
Actividad	Desarrollo
Estimación	8
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-278	
Título	Como usuario quiero poder agregar tags a una pregunta para poder clasificarla
Descripción	-
Criterios de aceptación	<p>Backend</p> <p>Agregar etiqueta a pregunta existente</p> <ul style="list-style-type: none"> • Dado que tengo 1 pregunta creada • Cuando agrego la etiqueta “stock” a la pregunta • Entonces visualizo la pregunta con la nueva etiqueta <p>Agregar etiqueta inexistente a pregunta existente</p> <ul style="list-style-type: none"> • Dado que existe 1 pregunta creada • Cuando agrego la etiqueta “Testing” a la pregunta • Entonces se retorna error de que la etiqueta no existe <p>Agregar etiqueta a pregunta inexistente</p> <ul style="list-style-type: none"> • Dado que no existen preguntas generadas • Cuando trato de agregar la etiqueta “stock” a una pregunta • Entonces se retorna error de que la pregunta no existe <p>Frontend</p> <p>Agregar etiqueta</p>

	<ul style="list-style-type: none"> • Dado que estoy en la sección de preguntas • Cuando abro el panel de etiquetas de la primera pregunta Y agrego una nueva etiqueta Y hago <i>click</i> en guardar • Entonces visualizo las etiquetas viejas más la nueva
Actividad	Desarrollo
Estimación	8
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-279	
Título	Como usuario quiero poder eliminar las tags de una pregunta
Descripción	-
Criterios de aceptación	<p><i>Frontend</i></p> <p>Eliminar etiqueta de pregunta</p> <ul style="list-style-type: none"> • Dado que estoy en la sección de preguntas Y la primera pregunta posee la etiqueta "stock" • Cuando abro las etiquetas de la primera pregunta Y elimino la etiqueta "stock" y luego hago <i>click</i> en el botón de guardar • Entonces visualizo que la pregunta no posee más la etiqueta de 'stock"

	<p>Backend</p> <p>Eliminar etiqueta de pregunta existente</p> <ul style="list-style-type: none"> • Dado que existe una pregunta con la etiqueta “stock” • Cuando remuevo la etiqueta “stock” de la pregunta existente • Entonces podre visualizar la pregunta sin la etiqueta “stock”
Actividad	Desarrollo
Estimación	5
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-280	
Título	Como usuario quiero poder usar los <i>templates</i> para contestar preguntas más ágilmente
Descripción	-
Criterios de aceptación	<p>Frontend</p> <p>Responder pregunta utilizando plantilla:</p> <ul style="list-style-type: none"> • Dado que tengo una plantilla creada y una pregunta para responder y estoy en la pregunta expandida de esa pregunta • Cuando escribo “@” en el campo de respuesta y selecciono la plantilla creada y respondo

	<ul style="list-style-type: none"> • Entonces la pregunta es respondida con la información contenida en la plantilla <p>Backend</p> <p>Responder pregunta con plantilla:</p> <ul style="list-style-type: none"> • Dado que tengo una pregunta para responder y una plantilla creada con el texto “Hola” • Cuando respondo la pregunta utilizando la plantilla • Entonces la pregunta se envía al historial y desaparece de preguntas pendientes para responder y la pregunta es respondida con el texto “Hola”
Actividad	Desarrollo
Estimación	5
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-282	
Título	Como usuario quiero poder expandir el histórico para ver más detalles
Descripción	-
Criterios de aceptación	<p>Mostrar detalles de pregunta respondida</p> <ul style="list-style-type: none"> • Dado que estoy en el historial Y la primera pregunta esta respondida • Cuando abro la primera pregunta • Entonces puedo ver los detalles de quien la respondió Y que respondió
Actividad	Desarrollo
Estimación	5
Componentes	<i>Frontend</i>

Historia de usuario TES-319	
Título	Como administrador quiero poder monitorear la app con sentry.io para saber si hay problemas recurrentes
Descripción	-
Criterios de aceptación	<p>Visualización de información pertinente en la cuenta asociada a <i>Sentry.io</i></p> <ul style="list-style-type: none"> • Dado que quiero administrar la aplicación <i>frontend</i> • Cuando voy a mi cuenta de <i>Sentry.io</i>

	<ul style="list-style-type: none"> Entonces puedo visualizar todas las solicitudes de mi aplicación <i>frontend</i> así como los errores posibles
Actividad	Desarrollo
Estimación	2
Componentes	<i>Frontend</i>

Historia de usuario TES-365	
Título	Como usuario quiero poder instalar la aplicación en el celular para usarla de forma "nativa"
Descripción	-
Criterios de aceptación	<p>Instalación aplicación en escritorio</p> <ul style="list-style-type: none"> Dado que estoy utilizando la aplicación en un navegador web Cuando selecciono instalar Entonces queda un instalada en el escritorio <p>Instalación aplicación en dispositivo móvil</p> <ul style="list-style-type: none"> Dado que estoy utilizando la aplicación en un dispositivo móvil Cuando selecciono instalar Entonces queda un instalada como una aplicación más

Actividad	Desarrollo
Estimación	3
Componentes	<i>Frontend</i>

Historia de usuario TES-378	
Título	Como usuario quiero poder configurar una respuesta precargada basada en los <i>templates</i> y en los tags para poder contestar más rápido
Descripción	-
Criterios de aceptación	<p><i>Frontend</i></p> <p>Obtener configuración:</p> <ul style="list-style-type: none"> • Dado que estoy autenticado en Sagal y tengo una configuración creada • Cuando ingreso a la sección de configuración • Entonces puedo ver la configuración <p>Guardar configuración:</p> <ul style="list-style-type: none"> • Dado que estoy en la sección de configuración y tengo 2 plantillas creadas • Cuando creo una configuración con una plantilla como bienvenida y la otra plantilla como despedida • Entonces la configuración se guarda

	<p>Backend</p> <p>Guardar configuración de un cliente:</p> <ul style="list-style-type: none"> • Dado que tengo un <i>token</i> valido y tengo 2 plantillas creadas • Cuando quiero guardar una nueva configuración utilizando esas 2 plantillas • Entonces la configuración es guardada
Actividad	Desarrollo
Estimación	8
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-385	
Título	Como usuario quiero poder tener un <i>dashboard</i> para poder ver estadísticas de mis preguntas
Descripción	Esta <i>user story</i> se decidió dejarla para después del proyecto académica por decisión del cliente
Criterios de aceptación	-
Actividad	Desarrollo
Estimación	No estimado hasta ahora, ya que no entro en ninguno de los <i>sprints</i>
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-399	
Título	Asegurar la portabilidad del sistema
Descripción	<ul style="list-style-type: none"> • Se debe poder usar en Edge, safari, chrome, firefox • Se debe poder usar en IOs y Android
Criterios de aceptación	-
Actividad	Desarrollo
Estimación	0
Componentes	<i>Frontend</i>

Historia de usuario TES-400	
Título	El sistema debe ser modificable
Descripción	<ul style="list-style-type: none"> • Base para reconstruir el <i>frontend</i> de la aplicación actual. • Módulos genéricos y claros para poder ser reutilizados. • El cliente hace <i>code review</i> del código del <i>frontend</i> para garantizar esto. • Usar ESLint para seguir estándar de codificación. • Seguir los lineamientos propuestos por AirBnB.
Criterios de aceptación	-

Actividad	Desarrollo
Estimación	0
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-401	
Título	Requerimientos de Software
Descripción	<ul style="list-style-type: none"> • El <i>backend</i> debe ser con Node.js • El <i>frontend</i> debe ser con React
Criterios de aceptación	-
Actividad	Desarrollo
Estimación	0
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-402	
Título	Requerimientos de Hardware
Descripción	Los servicios deben de correr en servidores de AWS
Criterios de aceptación	-
Actividad	Desarrollo
Estimación	0
Componentes	<i>Backend, Frontend</i>

Historia de usuario TES-403	
Título	Modelado de ML
Descripción	<ul style="list-style-type: none"> • Uso de IBM Watson para ML • Modelo que pueda predecir etiquetas para las preguntas con una confianza del 70% • Herramientas para que el cliente pueda reentrenar el modelo
Criterios de aceptación	-
Actividad	Desarrollo
Estimación	0
Componentes	<i>Backend, Frontend</i>

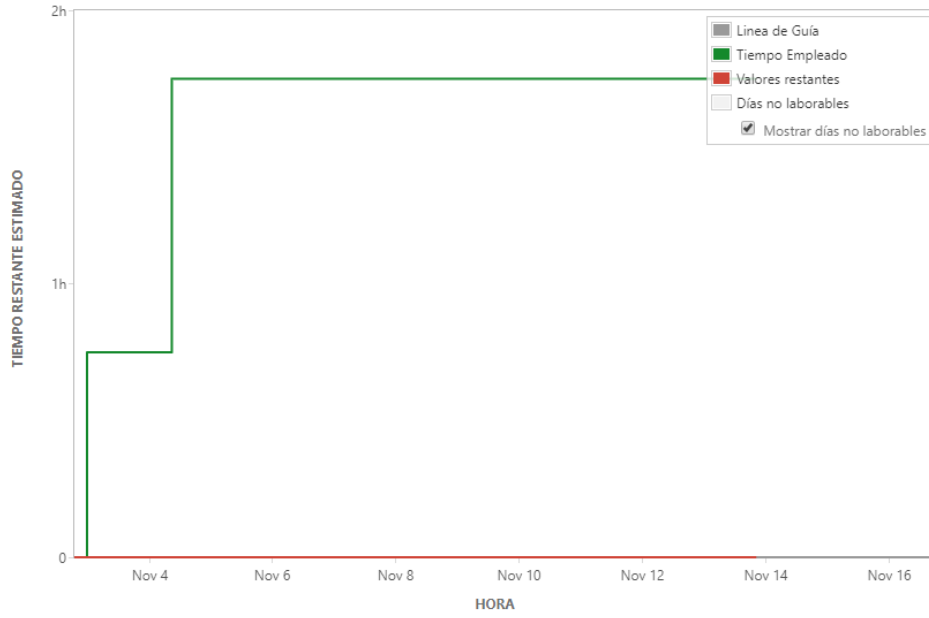
Historia de usuario TES-417	
Título	Documentación Final
Descripción	Proceso de creación del documento final a entregar en la instancia académica
Criterios de aceptación	-
Actividad	Documentación
Estimación	0

Componentes	<i>Backend, Frontend</i>
--------------------	--------------------------

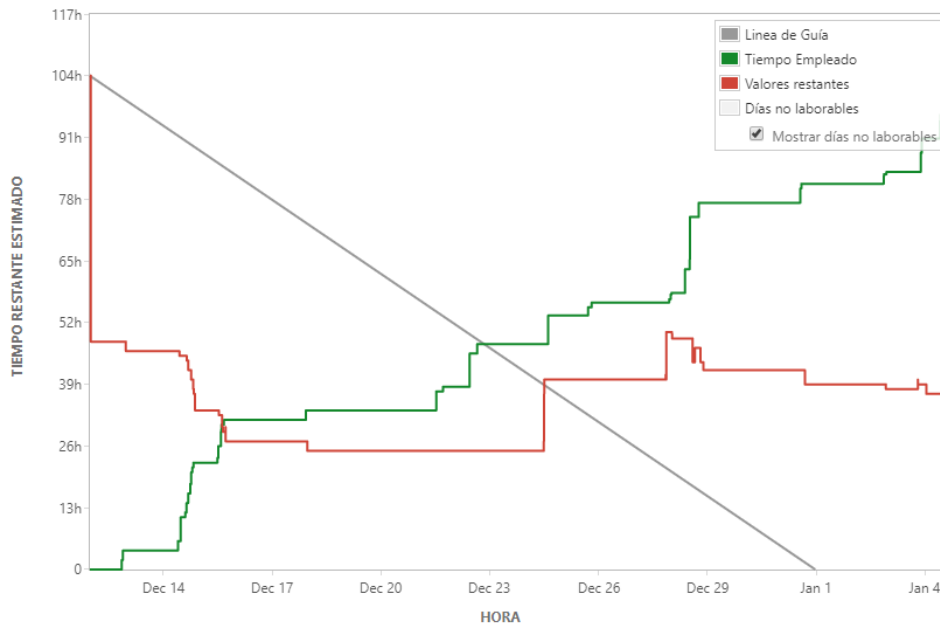
Historia de usuario TES-433	
Título	Utilizar REST como comunicación entre componentes
Descripción	-
Criterios de aceptación	-
Actividad	Desarrollo
Estimación	0
Componentes	<i>Backend, Frontend</i>

12.4. Burndown

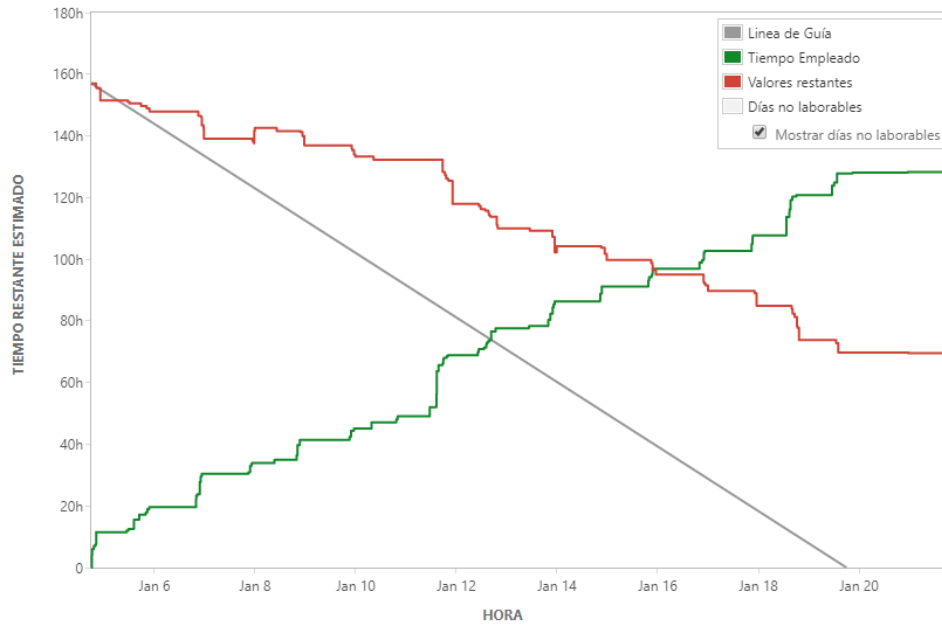
Sprint - Spike



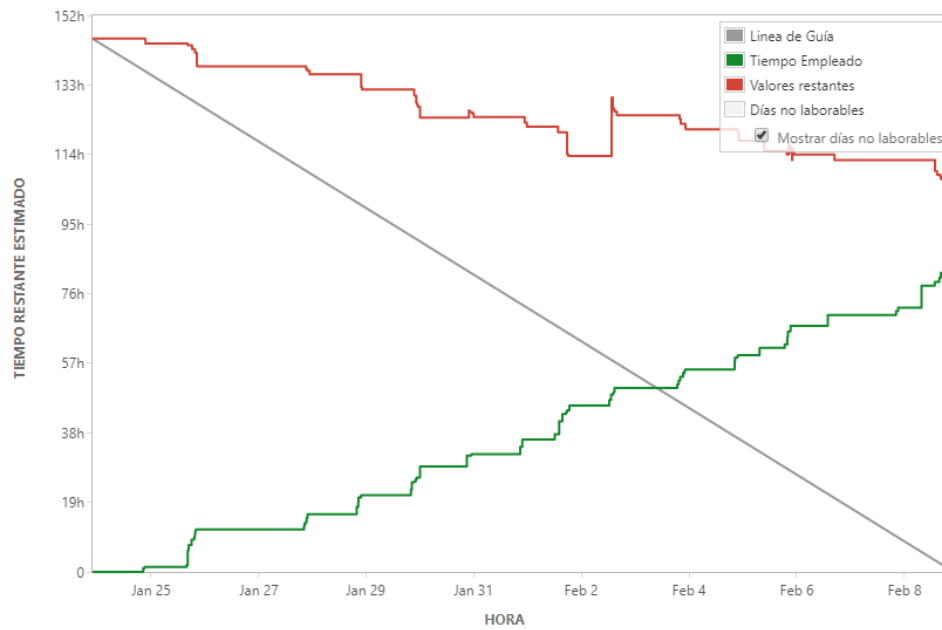
Sprint 1



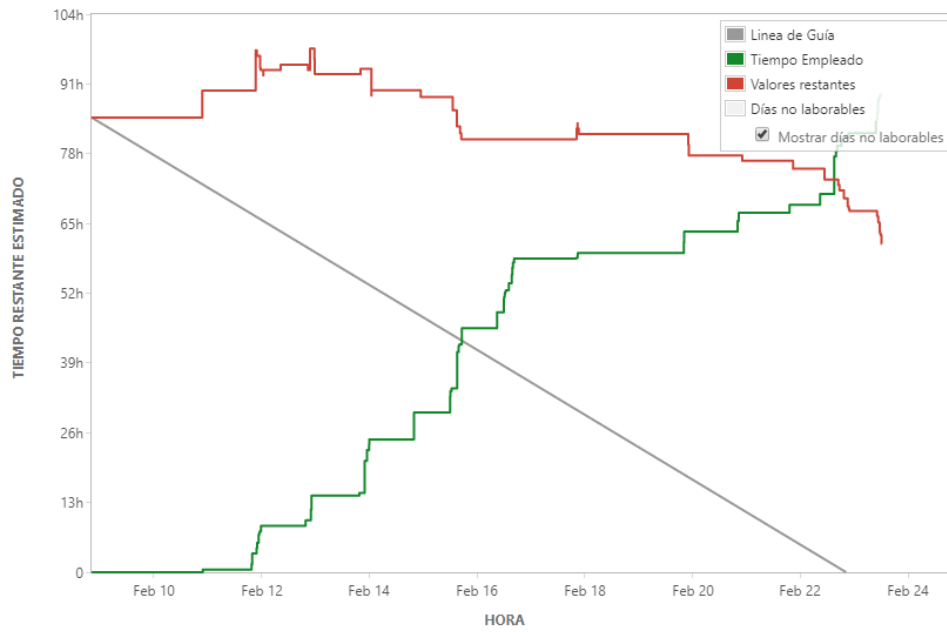
Sprint 2



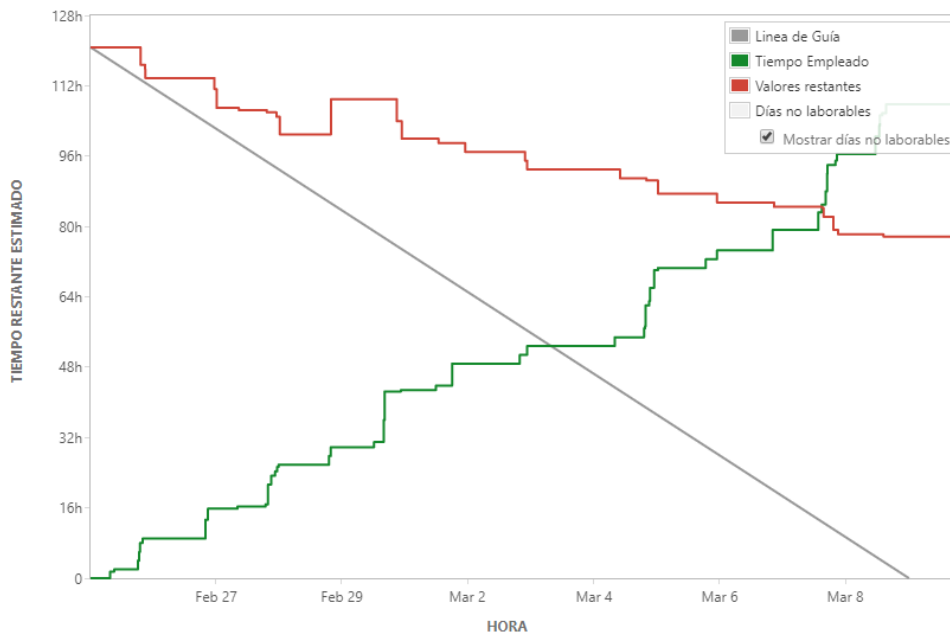
Sprint 3



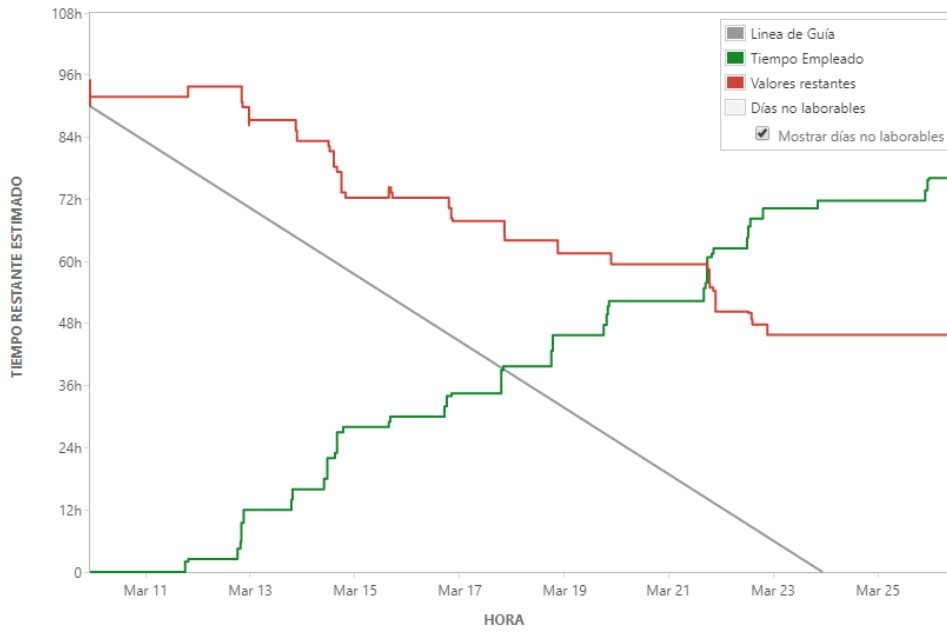
Sprint 4



Sprint 5



Sprint 6



12.5. Sprint reports

Sprint – spike

Status Report

* Issue added to sprint after start time

Completed Issues

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (-)
TES-21 *	Crear el proyecto para front	☑ Task	↑ Medium	DONE	-
TES-22 *	Crear el proyecto para back	☑ Task	↑ Medium	DONE	-

Issues Not Completed

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (-)
TES-19 *	Investigar conectividad	☑ Task	↑ Medium	TO DO	-
TES-23 *	Como usuario quiero que las preguntas puedan catalogarse correctamente en diferentes etiquetas para poder responderlas más fácil en el futuro	📌 Story	↑ Highest	TO DO	-
TES-28 *	Investigar la creacion de una PWA	☑ Task	↑ Medium	IN PROGRESS	-
TES-34 *	Autenticación de app	☑ Task	↑ Medium	TO DO	-

Issues Removed From Sprint

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (-)
TES-3	Front - Como usuario quiero visualizar todas las preguntas para poder responderlas una tras otra	📌 Story	↑ Highest	TO DO	-
TES-14	Como usuario quiero poder contestar una pregunta para responderle a los clientes finales	📌 Story	↑ Highest	TO DO	-

Sprint 1

Status Report

* Issue added to sprint after start time

Completed Issues

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (9 → 21)
TES-3	Front - Como usuario quiero visualizar todas las preguntas para poder responderlas una tras otra	Story	Highest	DONE	5
TES-48	Back - Como usuario quiero visualizar todas las preguntas para poder responderlas una tras otra	Story	Medium	DONE	4 → 3
TES-57 *	Preparacion de primera revision	Task	Highest	DONE	-
TES-60 *	Como administrador quiero que la aplicación se pruebe automáticamente para tener confianza en la aplicación	Story	High	DONE	- → 13
TES-86 *	Aplicar uso de linter	Story	Medium	DONE	-

Issues Not Completed

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (- → 14)
TES-4 *	Como usuario al visualizar una pregunta quiero poder visualizar todos los datos relacionados para poder contestar fácilmente	Story	Highest	TO DO	- → 2
TES-5 *	Como usuario al visualizar una pregunta quiero poder visualizar el nombre y apellido del usuario que me consulta para identificar al usuario que pregunta	Story	Highest	TO DO	- → 2
TES-8 *	Como usuario quiero poder tener un paginado de preguntas a responder para poder navegar entre ellas	Story	Medium	TO DO	- → 3
TES-14 *	Como usuario quiero poder contestar una pregunta para responderle a los clientes finales	Story	Highest	TO DO	- → 2
TES-65 *	Asegurar la performance de la aplicacion	Story	Medium	IN PROGRESS	- → 5

Issues Removed From Sprint

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (-)
TES-23	Como usuario quiero que las preguntas puedan catalogarse correctamente en diferentes etiquetas para poder responderlas más fácil en el futuro	Story	Highest	TO DO	-

Sprint 2

Status Report

Completed Issues

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (71)
TES-4	Como usuario al visualizar una pregunta quiero poder visualizar todos los datos relacionados para poder contestar fácilmente	Story	Highest	DONE	8
TES-8	Como usuario quiero poder tener un paginado de preguntas a responder para poder navegar entre ellas	Story	Medium	DONE	21
TES-9	Como usuario quiero poder visualizar todas las preguntas ya respondidas para tener un historial de las mismas ademas de un seguimiento	Story	Medium	DONE	8
TES-11	Como usuario quiero poder visualizar las preguntas mas antiguas sin contestar primero para responderlas lo antes posible	Story	Medium	DONE	2
TES-12	Como usuario quiero al contestar una pregunta que desaparezca para visualizar solo las no respondidas	Story	Medium	DONE	3
TES-14	Como usuario quiero poder contestar una pregunta para responderle a los clientes finales	Story	Highest	DONE	8
TES-81	Preparar el primer release	Story	High	DONE	21

Issues Not Completed

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (16)
TES-13	Como usuario quiero poder filtrar las preguntas para poder buscar las que son mas relevantes	Story	Highest	TO DO	13
TES-87	Como usuario quiero poder comunicarme con el soporte técnico para resolver mis dudas	Story	Highest	TO DO	3

Issues Removed From Sprint

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (- → 8)
TES-42	Conexión e interfaz con Sagal	Story	High	TO DO	- → 8

Sprint 3

Completed Issues

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (70)
TES-13	Como usuario quiero poder filtrar las preguntas para poder buscar las que son mas relevantes	Story	Highest	DONE	2
TES-23	Como usuario quiero que las preguntas puedan catalogarse correctamente en diferentes etiquetas para poder responderlas más fácil en el futuro	Story	Highest	DONE	34
TES-66	Asegurar que el frontend de la aplicación sea reutilizable	Story	Highest	DONE	3
TES-87	Como usuario quiero poder comunicarme con el soporte técnico para resolver mis dudas	Story	Highest	DONE	2
TES-185	Hacer un update con el tutor	Task	Medium	DONE	-
TES-186	Como usuario quiero ver más detalles del articulo a demanda para poder responder las preguntas más específicas	Story	Highest	DONE	3
TES-188	Como usuario quiero poder utilizar hotkeys en el uso del sistema para que mejore mis tiempos de respuesta	Story	Medium	DONE	8
TES-189	Como usuario quiero poder dar de baja templates para dejar de utilizarlos al responder preguntas	Story	Medium	DONE	3
TES-190	Como usuario quiero poder editar templates para poder responder mas rapido	Story	Medium	DONE	3
TES-191	Como usuario quiero poder ver los templates para luego poder administralos	Story	High	DONE	3
TES-192	Como usuario quiero poder cambiar de páginas entre los diferentes templates que están creados para poder navegar eficazmente	Story	Medium	DONE	2
TES-195	Como usuario quiero poder dar de alta templates para utilizarlos al responder preguntas	Story	Medium	DONE	5
TES-233	Reuniones de Tutoria	Story	Medium	DONE	1
TES-237	Reuniones con Sagal	Story	Medium	DONE	1
TES-252 *	Agregar datos extra a las questions	Task	Medium	DONE	-
TES-264 *	Documentacion de interfaces y cosas necesarias para dejar Release 2 en produccion	Task	High	DONE	-

Issues Not Completed

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (6)
TES-65	Asegurar la performance de la aplicacion	Story	Medium	IN PROGRESS	1
TES-229	Hacer el merge contra master del release	Task	Medium	TO DO	-
TES-253 *	Como usuario quiero poder ver quien realizo las acciones sobre las preguntas para poder hacerles un seguimiento	Story	Medium	TO DO	5

Issues completed outside of this sprint

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (-)
TES-228	Hacer el merge contra master de release 1	Task	Medium	DONE	-

Sprint 4

Completed Issues

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (44)
TES-263	Como usuario quiero recibir notificaciones sobre las acciones que estoy realizando para mejorar la usabilidad	Story	Low	DONE	8
TES-266	Como usuario quiero poder visualizar los tags asociadas a una pregunta para poder identificar sobre que trata rápidamente	Story	Medium	DONE	5
TES-277	Como usuario quiero poder buscar las preguntas con determinado tag para filtrar mejor	Story	Medium	DONE	8
TES-278	Como usuario quiero poder agregar tags a una pregunta para poder clasificarla	Story	Medium	DONE	8
TES-279	Como usuario quiero poder eliminar las tags de una pregunta	Story	Low	DONE	5
TES-280	Como usuario quiero poder usar los templates para contestar preguntas mas agilmente	Story	Medium	DONE	5
TES-281	Hacer ajustes gráficos de la aplicación para salir a producción	Task	Highest	DONE	-
TES-282	Como usuario quiero poder expandir el historico para ver mas detalles	Story	Medium	DONE	5
TES-315	Sprint planning y pre planning	Task	Medium	DONE	-
TES-316	Sprint retrospective - Sprint 3	Task	Medium	DONE	-
TES-336 *	Endpoint para actualizar datos de los articulos	Task	Medium	DONE	-
TES-345 *	Cambiar forma de mandar una respuesta a Sagal	Task	High	DONE	-
TES-347 *	Last user action se ve en questions	Bug	Highest	DONE	-
TES-357 *	En vista de preguntas, el overflow anda mal con muchos clicks	Bug	Medium	DONE	-
TES-358 *	El valor y nombre de los atributos deben ser String	Bug	Medium	DONE	-
TES-360 *	Pasar respuesta a las preguntas sincronico	Task	Medium	DONE	-

Issues Not Completed

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (1 → 0)
TES-65	Asegurar la performance de la aplicacion	Story	Medium	IN PROGRESS	1 → 0
TES-84	Preparar el primer release	Task	Highest	TO DO	-
TES-229	Hacer el merge contra master del release	Task	Medium	TO DO	-
TES-276	Hacer ajustes en la presentación para la segunda revisión	Task	Highest	TO DO	-
TES-286	Hacer benchmarking del modelo de watson	Task	Medium	IN PROGRESS	-
TES-317 *	Sprint retrospective - Sprint 4	Task	Medium	TO DO	-
TES-320 *	Reunion de tutoria	Task	Medium	TO DO	-
TES-321 *	Team replanning para ajustes de reunion de tutoria	Task	Medium	TO DO	-

Sprint 5

Completed Issues

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (42)
TES-10	Como usuario quiero poder usar la aplicacion desde el celular para poder contestar las preguntas desde ahi	Story	High	DONE	34
TES-84	Preparar el primer release	Task	Highest	DONE	-
TES-229	Hacer el merge contra master del release	Task	Medium	DONE	-
TES-268	Como usuario quiero ver cuales son los hotkeys disponibles para poder usarlos	Story	Medium	DONE	3
TES-276	Hacer ajustes en la presentación para la segunda revisión	Task	Highest	DONE	-
TES-286	Hacer benchmarking del modelo de watson	Task	Medium	DONE	-
TES-319	Como administrador quiero poder monitorear la app con sentry.io para saber si hay problemas recurrentes	Story	Medium	DONE	2
TES-321	Team replanning para ajustes de reunion de tutoria	Task	Medium	DONE	-
TES-361	Reunión de tutoria	Task	Medium	DONE	-
TES-363	Reajustar la interfaz con Watson	Task	High	DONE	-
TES-364	Cambios solicitados por el cliente	Task	Medium	DONE	-
TES-365	Como usuario quiero poder instalar la aplicacion en el celular para usarla de forma "nativa"	Story	Medium	DONE	3
TES-373	Segunda revision	Task	Medium	DONE	-
TES-395	Sprint 5 Planning	Task	Medium	DONE	-

Issues Not Completed

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (8)
TES-65	Asegurar la performance de la aplicacion	Story	Medium	TO DO	0
TES-362	Generar interfaces para conexion con Sagal	Task	High	TO DO	-
TES-369	Hacer mocks para el diseño del dashboard	Task	Medium	TO DO	-
TES-378	Como usuario quiero poder configurar una respuesta precargada basada en los templates y en los tags para poder contestar más rápido	Story	Medium	TO DO	8
TES-396 *	Investigacion de metodos de autenticacion para mitigar riesgo	Task	Medium	TO DO	-

Issues completed outside of this sprint

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (-)
TES-317	Sprint retrospective - Sprint 4	Task	Medium	DONE	-

Sprint 6

Completed Issues

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (13)
TES-253	Como usuario quiero poder ver quien realizo las acciones sobre las preguntas para poder hacerles un seguimiento	Story	Medium	DONE	5
TES-362	Generar interfaces para conexion con Sagal	Task	High	DONE	-
TES-378	Como usuario quiero poder configurar una respuesta precargada basada en los templates y en los tags para poder contestar más rápido	Story	Medium	DONE	8
TES-403 *	Modelado de ML	Story	Medium	DONE	-
TES-404	Sprint 5 retrospective	Task	Medium	DONE	-
TES-405	Sprint 6 planning	Task	Medium	DONE	-
TES-410	Reunion de tutoria primera semana	Task	Medium	DONE	-
TES-411	Reunion de tutoria segunda semana	Task	Medium	DONE	-
TES-412 *	Crear un manual del usuario	Task	Medium	DONE	-

Issues Not Completed

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (21)
TES-65	Asegurar la performance de la aplicacion	Story	Medium	TO DO	0
TES-369	Hacer mocks para el diseño del dashboard	Task	Medium	TO DO	-
TES-385	Como usuario quiero poder tener un dashboard para poder ver estadísticas de mis preguntas	Story	Medium	TO DO	21
TES-409	Ajustes de presentación	Task	Medium	IN PROGRESS	-
TES-413 *	Obtener cantidad de preguntas recibidas por SAGAL por mes	Task	Medium	TO DO	-

12.6. Sprint retrospectives

Retrospective – 04/01/2020

Date	4 ene. 2020
Team	Tesis
Participants	@Nicolas Gonzalez @Nicolás Caballero @Renzo Delfino

Resultados y Productividad

	Sprint 1	Acumulado
Story Points	21	21
Horas Trabajadas	115	125
Productividad	0,18	0,168

Background

- Primer sprint
- Primera planning
- Cerrar el sprint, ver velocidad
- Redefinir testing, planning, scope y manejo de comentarios del cliente
- Reglas de trabajo y aseguramiento de la calidad

Retrospective

Start doing	Stop doing	Keep doing
<ul style="list-style-type: none">• Seguir las definiciones de la distribución de trabajo• Dejar documentado las reglas de las branches• Cumplir las horas semanales	<ul style="list-style-type: none">• Agregar US al sprint• Subestimar tareas• Estimar mal• Asignar responsables a la US y a las tareas de ante mano• Trabajar desordenado con gitflow	<ul style="list-style-type: none">• Usar las herramientas correctamente• Mantener la comunicación y resolución de problemas• Reuniones con el cliente periódicas

Action items

- Recomendaciones de Simon, anotar en una US con subtareas para próximo sprint
- Pasar las reglas del linter a Warn
- Agregar linter al backend
- Ajustar el flujo de las branches
- Definir estructuras de riesgo
- Master(release), Develop(rolling release), Feature(on work features)
- Remove el in testing flow
- Agregar build de master y develop
- Definir registro y tratamiento de bugs, Kanban de issues pendientes

Retrospective – 21/01/2020

Date	21 ene. 2020
Team	Tesis
Participants	@Nicolas Gonzalez @Renzo Delfino @Nicolás Caballero

Productividad

	Sprint 2	Acumulado
Story Points	71	92
Horas Trabajadas	150	275
Productividad	0,47	0,334

Background

Segundo sprint terminado del proyecto

Retrospective

Start doing	Stop doing	Keep doing
<ul style="list-style-type: none">• Agregar horas de trabajos de gestión• Agregar horas cuando se trabaja de "a pares"	<ul style="list-style-type: none">• Resolver tareas fuera del US al que le corresponde• No mergear features sin tests	<ul style="list-style-type: none">• Mantener buena comunicación• Reuniones con el cliente periódicas•

Action items

- Reajustar el compromiso de horas de 25 semanales a 20 semanales.
- Dejar los domingos libres
- Definir las estructuras de riesgo
- Agregar linter al backend @Renzo Delfino
- Planning meeting el 23 ene. 2020

Retrospective – 08/02/2020

Date	8 feb. 2020
Team	Tesis
Participants	@Nicolas Gonzalez @Renzo Delfino @Nicolás Caballero

	Sprint 3	Acumulado
Story Points	70	162
Horas Trabajadas	80	355
Productividad	0,87	0,45

Background

Revisión del tercer sprint del proyecto.

Retrospective

Start doing	Stop doing	Keep doing
<ul style="list-style-type: none">• Hacer status breves a mitad de la semana• Discutir mas temas de diseño• Documentar las decisiones de diseño	<ul style="list-style-type: none">• Revisar las estimaciones para evitar agregar nuevas tareas al sprint•	<ul style="list-style-type: none">• Reuniones con el cliente• Mantener el compromiso con las horas de trabajo.

Action items

- Fijar los miercoles como dia de status semanal

Retrospective – 24/02/2020

Date	24 feb. 2020
Team	Tesis
Participants	@Nicolas Gonzalez @Renzo Delfino @Nicolás Caballero

	Sprint 4	Acumulado
Story Points	44	206
Horas Trabajadas	125	480
Productividad	0,35	0,43

Background

Revisión de cuarto sprint, se cumplieron con todas las US del sprint.

Retrospective

Start doing	Stop doing	Keep doing
<ul style="list-style-type: none">• Presionar al cliente con los entregables• Gestionar adecuadamente los riesgos	<ul style="list-style-type: none">• Discutir temas que están fuera del alcance en el sprint actual• Agregar tareas al sprint	<ul style="list-style-type: none">• Compromiso de horas• Gestión de las decisiones de diseño

Action items

- Agregar la discusión de los riesgos en las planning meetings
- Tener reuniones de "presión" con el cliente

Retrospective – 09/03/2020

Date	9 mar. 2020
Team	Tesis
Participantes	@Nicolas Gonzalez @Nicolás Caballero @Renzo Delfino

	Sprint 5	Acumulado
Story Points	42	248
Horas Trabajadas	111	679
Horas de desarrollo	65	329
Productividad	0.64	0.75

Retrospective

Start doing	Stop doing	Keep doing
<ul style="list-style-type: none"> • Monitorear al beta tester 	<ul style="list-style-type: none"> • Esperar que el cliente cumpla con el compromiso 	<ul style="list-style-type: none"> • Mantener el nivel de compromiso •

Positive & True

Nicolas C

- Mitigar la puesta en producción
- Contemplar las horas del examen
- Ayudar al equipo

Renzo

- Organizar las reuniones con el cliente para resolver problemas
- Dedicación de horas

Nicolas G

- Dedicarle horas a la presentación
- Segunda revisión
- Presión al cliente

+Compromiso de estar casi todos los días online

+Compromiso del equipo con las tareas

Cumplimiento de horas

Documentacion de diseño aclara como atacar un problema

Tarea de sobrecarga deberiamos haberla hecho hace varios sprint atras

Hacer pruebas de stress en las tecnologías antes de aplicarlas

Priorizar las pruebas

Hacer presentación con todos los features y no solo con la mayoría.

Practicar la presentacion

-Manejar el cliente de forma mas fluida

Tener reuniones entre todos

Retrospective – 24/03/2020

Date	24 mar. 2020
Team	Tesis
Participants	@Nicolas Gonzalez @Renzo Delfino @Nicolás Caballero

	Sprint 6	Acumulado
Story Points	13	261
Horas Trabajadas	87	766
Horas de desarrollo	48	377
Productividad	0.27	0.69

Luego de analizar los números vemos que la mayoría de las horas de desarrollo se gastaron en mitigación del riesgo de generación de interfaces lo que bajo la productividad radicalmente.

Retrospectiva

Start doing	Stop doing	Keep doing
<ul style="list-style-type: none">Darle mas seguimiento al beta tester	<ul style="list-style-type: none">	<ul style="list-style-type: none">Trabajo en equipo en issues complejas o rediseños

Positive & True

Debido a que tuvimos que hacer varias mitigaciones de riesgo, los tres coincidimos en que el trabajo en equipo para manejar algunos problemas de performance fue lo mas clave.

Como estamos sobre el fin del proyecto creemos que es la parte clave que nos queda es poder plasmar todo nuestro trabajo en la documentación final.

Action items

- Dedicar tiempo de documentación

12.7. Reuniones con el cliente

Reunión con el cliente - 21/01/2020

Participants

- @Nicolás Caballero
- @Renzo Delfino
- Alejandro Tocar
- Alan Bukstein

› Goals

- Mostrarle el proyecto hasta el momento
- Obtener feedback de diseño y funcionalidad

Discussion topics

- Se presento la aplicación para que la vieran y pudieran probarla
- Se anotaron los comentarios correspondientes y se registraron como acción items para los próximos sprints

Action items

- Mover el link de la publicación a la izquierda
- Modal de filtros que no tenga la línea
- Indicar como esta ordenado el date y que se pueda cambiar el orden
- Boton de filtrado con el mismo estilo que SAGAL
- Codigo SKU que sea solo CODIGO
- Filtros por estado, que aparezcan todos los estados marcados desde el arranque
- Mostrar los filtros aplicados
- Cambiar "Bajo revision" a "En revision"
- Agregar respuesta en el historial
- Dejar respirar el texto de abajo @Nicolás Caballero

Decisions

- ✔ Hacer cambios de diseño en la web
- ✔ Si el articulo esta pausado no se puede responder

Reunión con empresa X - 22/01/2020

Participants

- @Renzo Delfino
- @Nicolás Caballero
- Alejandro Tocar
- Gerencia de Marketing de empresa X

Goals

Obtener feedback de un cliente de Sagal con el uso del nuevo modulo de preguntas.

Action items

Pendientes marcados por el cliente.

- Viabilidad de hacer cola de mensajes agrupados por cliente al visualizar las preguntas
- Blacklist de usuarios que preguntan en MercadoLibre no lo utilizan. No se considera como prioritario

Decisions

Se tomaron las siguientes decisiones:

- ✔ Templates al responder
- ✔ Agregar datos extra a los artículos. (Tipo y costos de envío, condicion del artículo y atributos)
- ✔ Datos para el dashboard (cantidad de preguntas eliminadas, quien usuario lo elimino, promedio de tiempo de respuesta, que articulo tuvo la mayor cantidad de preguntas, cuantas preguntas respondió un usuario)
- ✔ Empresa X acepto ser el beta tester de nuestra aplicación y ser evaluado por el impacto del uso de la misma.

Feedback de release 1 – 21/02/2020

- ✓ Alan - Quedar marcado el menú
- ✓ Alan - Tooltip en las secciones del menú
- ✓ Alan - Buscar lugar nuevo para tooltip de hotkeys
- ✓ Alan - Focus en objetos funciona raro
- ✓ Alan - Barrita entre card y header anularla para cohesion entre ambos
- ✓ Alan - Falta alineacion entre detalles, ver publicacion y responder
- ✓ Alan - Text area responder alineado a la pregunta para que siga el mismo lineamiento
- ✓ Alan - Agregar padding left a los detalles
- ✓ Alan - Cruz arreevez el css de cerrar los popups
- ✓ Alan - Poner algun estilo al mention puesto
- ✓ Alan - Mostrar que key usar para las mentions con placeholder
- ✓ Alan - Actualizar etiquetas, faltan tildes
- ✓ Ale - Titulo y sku. Color negro, sacar el sombreado. Negrita solo al titulo
- ✓ Ale - más detalles centrado igual que foto y botones
- ✓ Ale - Responder como el boton de ingresar de sagal
- ✓ Ale - Detalles y ver publicaciones → mismo boton de filtros
- ✓ Ale - Misma fuente que en app.sagal.io (Lato)
- ✓ Ale - Filtros(titulo) + chiquito sin negrito y cruz mas chiquito

12.8. Reuniones del equipo

Definición del trabajo – 04/01/2020

Status	DECIDED
Impact	HIGH
Driver	@Nicolas Gonzalez
Contributors	@Nicolás Caballero @Renzo Delfino
Due date	Jan 4, 2020
Outcome	Tomando en cuenta la primera revision documentamos la forma de trabajo

Background

Dado que tuvimos un mes de bajo rendimiento optamos por documentar nuestra estructura de trabajo y dejar bien definidos algunas estructuras de como vamos a trabajar.

Pre planning meeting (sprint)

Se toman como inputs las reuniones con el cliente. De ahí se obtiene una priorizacion de features para el proximo release.

1. Basandonos en las features requeridas armamos un backlog de User stories para el release.
2. Seleccionamos algunas de esas user stories y estimamos los story points de las mismas
3. Cuando vemos que estamos cerca de 70 story points pasamos a la planning meeting.

Planning meeting

Nos basamos en las user stories estimadas por la pre planning.

1. Vamos eligiendo las US que estan estimadas en story points y las dividimos en tareas.
2. Estimamos las tareas en horas de trabajo
3. Volvemos a revisar la estimacion en story points y corregimos si nos parece adecuado basandonos en las tareas definidas.
4. Si estamos de acuerdo la pasamos al sprint.
5. Iniciamos el sprint

Sprint retrospective

Una vez finalizado el sprint, analizamos como venimos trabajando como equipo con 3 preguntas: Start, stop, continúe.

Luego nos definimos acciones de equipo a tomar.

Release review

Incluye al cliente y al tester.

Hacemos una demo de la aplicación y evaluamos los cambios efectuados hasta el momento en comparación al release anterior.

Revisamos las prioridades para el próximo release y las ajustamos en caso de ser necesario.

Gitflow y ambientes

Dividimos nuestro flujo de trabajo en 3 ramas

- Master - Ambiente de producción
- Develop - Ambiente de desarrollo y stage
- Features - Desarrollo de US

El desarrollo se hace en la rama feature/UserStory, una vez terminado el desarrollo se hace una pull request para ser revisada y mergeada a Develop.

Una vez mergeada a develop, Jenkins se encarga de hacer un build y dejar el servidor levantado para poder probarlo.

Cuando se define un release, se hace un merge contra master donde Jenkins nuevamente hace un build para usar.

El cliente solicita ser reviewer de las pull requests del frontend, por lo que todos los comentarios de su parte son registrados y tomados en cuenta para ajustar en el sprint siguiente.

Outcome

Con esta estructura bien definida se nos hizo más fácil estructurar el trabajo para el sprint siguiente.

Ingreso de *bugs/issues* de uso - 28/03/2020

Ingreso de bugs/issues de uso



Created by Nicolas Gonzalez
Mar 28, 2020

Toda la gestión del cliente se va a manejar a través de SAGAL, por lo que los clientes van a comunicarse directamente con soporte de SAGAL. Una vez soporte detecte un bug, es responsable de generar el ticket dentro de Jira para que pueda ser evaluado, priorizado y resuelto.

<https://cobre.atlassian.net/secure/Createlssue!default.jspa>

Datos para el ingreso en Jira:

- Issue Type: Bug
- Summary: Titulo descriptivo del problema
- Reporter: Dejar el por defecto
- Componentes: Si se identifica de que componente puede provenir el error se puede agregar. Se divide en Frontend y Backend.
- Descripción: Lo mas detallado posible para poder reproducir el bug, todos los bugs que no se puedan reproducir correctamente no serán agregados a los próximos sprints para solucionar.
- Attachments: En caso de ser necesario, se pueden adjuntar imágenes de los bugs encontrados.
- Resto de campos: Se pueden dejar en los valores por defecto.

Solicitamos que se ingrese un Ticket por bug, y no agregar varios dentro de un mismo ticket ya que dificulta el seguimiento.

Ajustes de alcance sprint 3 - 30/01/2020

Status	DECIDED
Impact	LOW
Driver	@Renzo Delfino
Approver	@Nicolás Caballero
Contributors	@Nicolás Caballero @Nicolas Gonzalez
Informed	
Due date	Jan 30, 2020
Outcome	Se decidio agregar esta tarea ya que es fundamental contar con todos los datos necesarios para responder las preguntas.

Options considered


	Option 1:	Option 2:
Description	Dejarlo para el proximo Sprint	Estimarlo y realizarlo en este Sprint
Pros and cons	<ul style="list-style-type: none"> + Cumplimiento con nuestra metodologia de trabajo - Retrabajo a nivel de frontend 	<ul style="list-style-type: none"> + Se evita el retrabajo a nivel de frontend + Tarea que no consume mucho tiempo y puede encajar en el Sprint sin necesidad de reestimar lo anterior. - Incumplimiento con nuestra metodologia de trabajo
Estimated cost	MEDIUM	LOW

Ajustes de alcance sprint 3 - 02/02/2020

El recurso encargado del Backend finalizo todas las tareas relacionadas al mismo, por lo que se decidio en conjunto con el equipo, agregar mas tareas que involucren un impacto mayor del lado del Backend.

Status	DECIDED
Impact	HIGH
Driver	@Renzo Delfino
Approver	@Nicolás Caballero
Contributors	@Nicolás Caballero @Nicolas Gonzalez
Informed	
Due date	Feb 2, 2020
Outcome	Se agrega User Story para registrar acciones de los usuarios

Options considered

	Option 1:	Option 2:
Description	Planificarlo para el proximo Sprint	Planificarlo para el Sprint actual
Pros and cons	<p> Se cumple con la metodologia de trabajo</p> <p> Recurso con pocas tareas</p>	<p> Recurso con tareas para realizar</p> <p> No se cumple con la metodologia de trabajo</p>
Estimated cost	LARGE	MEDIUM

Decisión de diseño Watson – 08/02/2020

Status	DECIDED
Impact	HIGH / MEDIUM / LOW
Driver	@Nicolas Gonzalez
Due date	Feb 8, 2020
Outcome	Como pasarle las preguntas para re entrenamiento de watson

Background

Discusion sobre como elegir las preguntas y tags para re entrenar el modelo

Options considered

	Option 1:	Option 2:
Description	Pasar solo las tags por personas	Pasar las tags de watson y las de las personas cuando hay un cambio
Pros and cons	<ul style="list-style-type: none">+ Facil implementacion- Posibles fallas del modelo en malas clasificaciones	<ul style="list-style-type: none">+ Mejor entrenamiento del modelo- Tracking de cambios
Estimated cost	MEDIUM	MEDIUM

Outcome

Option 2.

Cuando un usuario agrega una tag manualmente o elimina una generada por Watson, TODAS las tags de la pregunta se guardan para enviar al re-entrenamiento del modelo.

Ej:

"Hay stock?"

- Tags de Watson: Stock, Precio
- Tags de la Persona: Metodo de pago (elimina Precio de Watson)
- Tags enviadas a re-entrenar: Stock, Metodo de pago

Ajustes de alcance *sprint 4* – reunión en conjunto con *Product owner* - 02/02/2020

Status	DECIDED
Impact	MEDIUM
Driver	@Renzo Delfino
Approver	@Alan Bukstein
Contributors	@Nicolás Caballero @Nicolas Gonzalez @Alan Bukstein
Informed	
Due date	Feb 12, 2020
Outcome	En vez de mandar a una Queue de AWS, vamos a esperar por la respuesta de Sagal para saber si la question se guarda en el historial o queda en la coleccion de questions.

Background

Sagal se encuentra implementando las interfaces que les brindamos a traves de la documentacion y nos dieron feedback sobre la implementacion del mismo.

Relevant data

El Product Owner realizo una reunion con nosotros para determinar la mejor alternativa para mandar las respuestas de las preguntas.

Options considered

	Option 1:	Option 2:
Description	Mandar respuesta a una Queue de AWS	Mandar directamente a Sagal y esperar por su respuesta para determinar el flujo siguiente.
Pros and cons	<ul style="list-style-type: none"> ⊕ La implementacion solo seria configuraciones para mandar a una queue ⊖ Puede darse casos en donde la accion de responder una pregunta demore si la queue es muy grande. 	<ul style="list-style-type: none"> ⊕ Al ser sincronico, el cliente ve reflejada si la pregunta se respondio o no mas rapido ⊖ Re trabajo ya que hay que cambiar partes de la implementacion realizada
Estimated cost	LOW	MEDIUM

Ajustes de alcance *sprint* 5 y pasos a seguir 02/02/2020

Participants

- @Nicolás Caballero
- @Renzo Delfino
- Sagal

Goals

Hacer una revisión de los pendientes y los ajustes de Scope necesarios para poder terminar con el proyecto en el tiempo correcto.

Discussion topics

Interfaces pendientes

En caso de que no se entreguen las interfaces en el correr de esta semana **Feb 28, 2020** , para poder poner en producción el sistema para que lo puedan probar, tomaremos nosotros ese trabajo recortando parte del scope de la aplicación.

Puesta en producción

Falta documentar las fechas de inicio de uso por parte del cliente. Si no se pone en "beta" el sistema, no podemos continuar agregando features sin tenerlas probadas.

Como fecha proponemos: **Mar 1, 2020** . De esta manera nos da tiempo para poder trabajar en algún bug que surja del uso.

La segunda versión sería lanzada el **Mar 16, 2020** .

Nosotros dejaremos disponible un formulario para reportar bugs por parte del cliente, trabajando inmediatamente en bugs que rompan la funcionalidad del sistema.

En caso de que sean ajustes o detalles que no rompan el proceso de uso del sistema, se dejarán para el sprint final.

Modelos predictivos

En caso de que no estén listas las preguntas para el **Feb 28, 2020** , entrenaremos el modelo con las preguntas catalogadas hasta el momento. Bajando la confianza del modelo.

Dashboard

Queda la US para el sprint siguiente, **Mar 9, 2020** al **Mar 23, 2020** . Este sprint incluirá los arreglos que surjan del uso y ajustes de performance y gráficos que se consideren necesarios y estén reportados antes del **Mar 9, 2020**

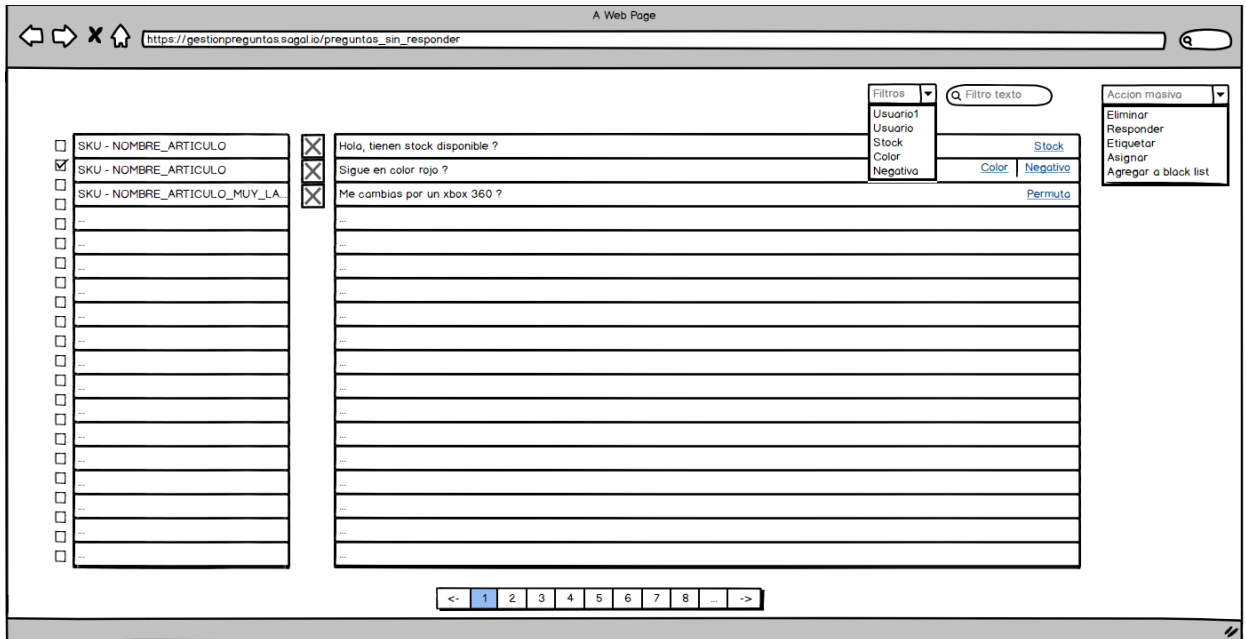
12.9. Endpoints de backend con Swagger

GET	/api/questions
POST	/api/questions
GET	/api/pools/question
GET	/api/questions/history
POST	/api/questions/answers
GET	/api/questions/{id}/actions
PUT	/api/questions/{id}/tags
POST	/api/templates
GET	/api/templates
PUT	/api/templates/{id}
DELETE	/api/templates/{id}
PUT	/api/learning/config
GET	/api/learning/config
GET	/api/health
PUT	/api/articles/{sagalCentralId}
POST	/api/questions/train

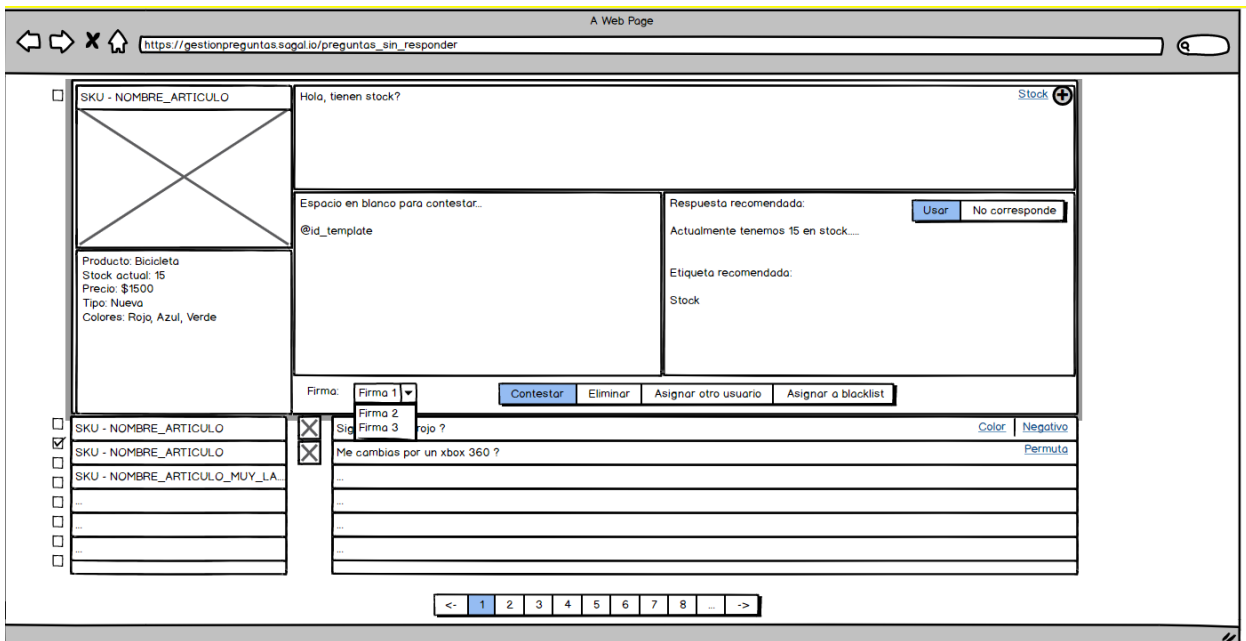
12.10. Mockups

Mockup versión de escritorio

Preguntas sin responder



Pregunta expandida



Pregunta detallada

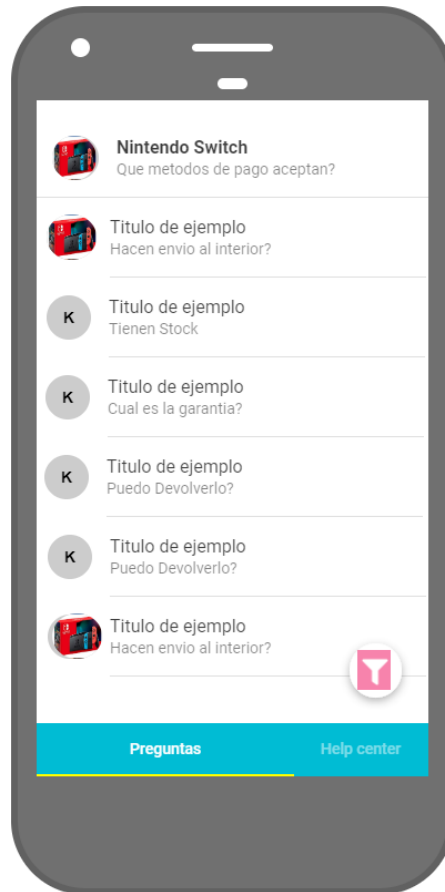
A Web Page

https://

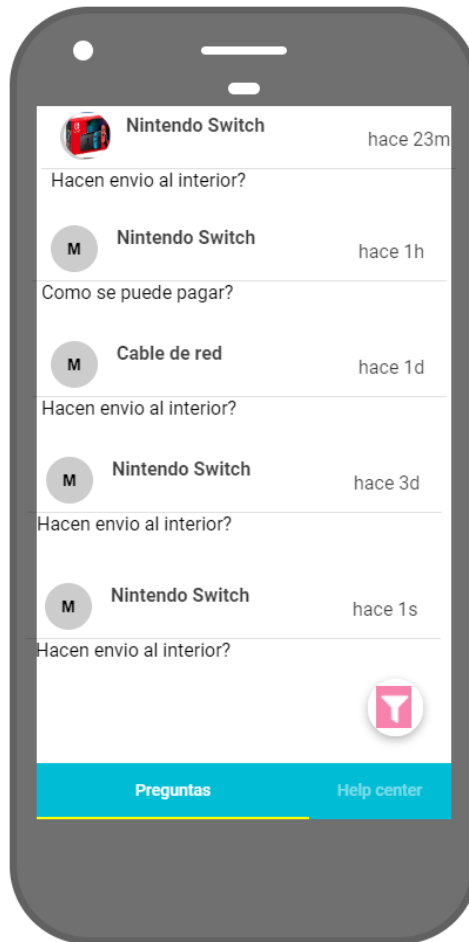
SKU - NOMBRE_ARTICULO	Hola, tienen stock?	Stock
	Espacio en blanco para contestar	Respuesta recomendada: <input type="button" value="Usar"/> <input type="button" value="No corresponde"/>
Producto: Bicicleta Stock actual: 15 Precio: \$1500 Tipo: Nueva Colores: Rojo, Azul, Verde		Actualmente tenemos 15 en stock...
		Etiqueta recomendada: Stock
Firma 1 Firma 2 Firma 3	<input type="button" value="Contestar"/> <input type="button" value="Eliminar"/> <input type="button" value="Asignar otro usuario"/> <input type="button" value="Asignar a blacklist"/>	

Mockup responsive

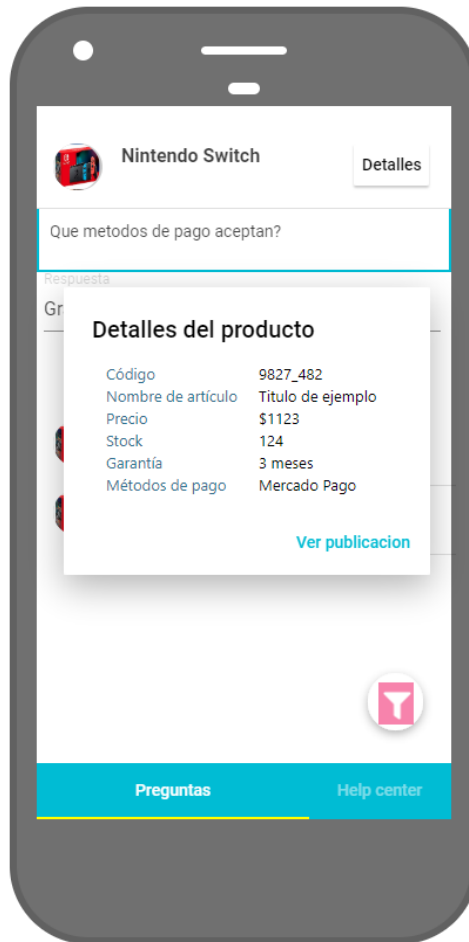
Vista de preguntas primera versión:



Vista de preguntas segunda versión:



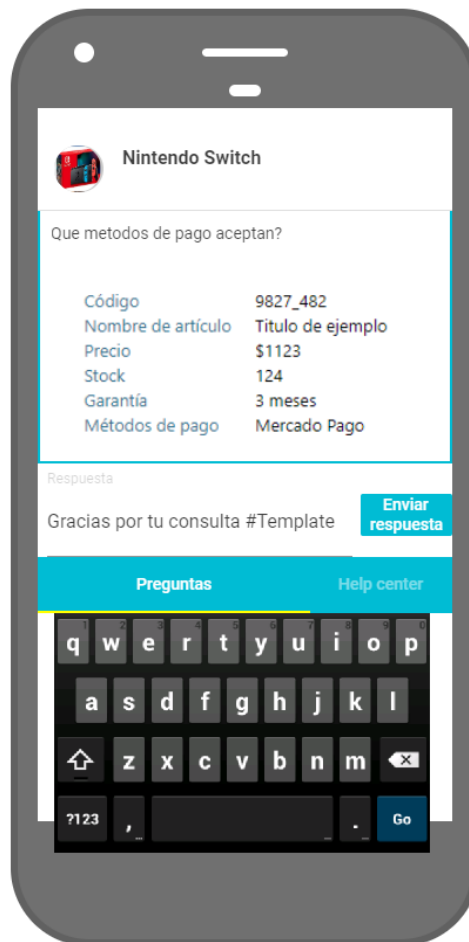
Detalles de pregunta:



Contestar pregunta primera versión:



Contestar pregunta segunda versión:



Feedback de experto en aplicaciones móviles

- No sean robustas, simples y claras.
- Al ser espacio menor, ocupar espacio correctos y que sean claros.
- Header arriba.
- Dos iconos, help-center y filtros en menú.
- Vista de grid como Intercomm, pero bastante acertado.
- Abrir nueva vista completa para los detalles con flecha para atrás.
- Enviar respuesta con icono.
- Iconos abajo para acciones de la pregunta.
- Detalles con popup y ver publicación.
- Botón de detalles con icono.
- Paginado debería permitir hacerlo.

12.11. Reporte tests

Resultados de *testing* del frontend

```
PASS src/__tests__/step_definitions/tags.update.js (11.989s)
PASS src/__tests__/step_definitions/template.create.js
PASS src/__tests__/step_definitions/history.pagination.js (14.626s)
PASS src/__tests__/step_definitions/question.answer.js (14.86s)
PASS src/__tests__/step_definitions/question.details.js (15.459s)
PASS src/__tests__/step_definitions/hotkeys.js
PASS src/__tests__/step_definitions/question.visualization.js
PASS src/__tests__/step_definitions/history.visualization.js
PASS src/__tests__/step_definitions/learning.config.js
PASS src/__tests__/step_definitions/question.pagination.js (18.253s)
PASS src/__tests__/step_definitions/tags.visualization.js
PASS src/__tests__/step_definitions/history.details.js
PASS src/__tests__/step_definitions/support.js
PASS src/__tests__/step_definitions/history.actions.js
PASS src/__tests__/step_definitions/notifications.js
PASS src/__tests__/step_definitions/template.visualization.js
PASS src/__tests__/step_definitions/template.edit.js
PASS src/__tests__/step_definitions/template.delete.js
PASS src/__tests__/step_definitions/filter.history.js
PASS src/__tests__/step_definitions/filter.question.js (22.688s)

Test Suites: 20 passed, 20 total
Tests:       57 passed, 57 total
Snapshots:   0 total
Time:        24.498s
Ran all test suites.
```

Resultados de *testing* del *backend*

```
PASS src/__tests__/step_definitions/article.update.steps.js (6.926s)
PASS src/__tests__/step_definitions/template.visualization.steps.js (6.969s)
PASS src/__tests__/step_definitions/template.update.steps.js (7.018s)
PASS src/__tests__/step_definitions/action.visualization.steps.js (7.052s)
PASS src/__tests__/step_definitions/automatic.answer.steps.js (7.051s)
PASS src/__tests__/step_definitions/question.details.steps.js (7.038s)
PASS src/__tests__/step_definitions/template.delete.steps.js (7.062s)
PASS src/__tests__/step_definitions/template.creation.steps.js (7.11s)
PASS src/__tests__/step_definitions/tag.creation.steps.js (7.261s)
PASS src/__tests__/step_definitions/question.visualization.steps.js (7.323s)
PASS src/__tests__/step_definitions/question.history.steps.js (7.432s)
PASS src/__tests__/step_definitions/template.pagination.steps.js (7.534s)
PASS src/__tests__/step_definitions/question.answer.steps.js (7.559s)
PASS src/__tests__/step_definitions/question.pagination.steps.js (8.739s)
PASS src/__tests__/step_definitions/question.filter.steps.js (10.001s)

Test Suites: 15 passed, 15 total
Tests:       55 passed, 55 total
Snapshots:   0 total
Time:        10.727s, estimated 11s
Ran all test suites matching /step_definitions/i.
```

Ejemplos de escenarios en Gherkin

Ejemplo de escenarios en Gherkin para agregar tags a preguntas en *backend*

```
Feature: As a user i want to add tags into questions so i can classify them

Scenario: Add tag to existing question
  Given i have 1 question created
  When i add the tag Stock to the question
  Then i see the question with the tag added

Scenario: Add no existing tag to existing question
  Given i have 1 question created
  When i add the tag Testing_123 to the question
  Then returns tag Testing_123 not exists message

Scenario: Add tag to no existing question
  Given i do not have questions created
  When i try to add the tag Stock
  Then returns question not exists message

Scenario: Remove tag to existing question
  Given i have 1 question created with Stock tag
  When i remove the Stock tag from the question
  Then i see the question without the tag
```

Ejemplo de escenarios en Gherkin para *hotkeys* en *frontend*

```
Feature: As a user I want to use differents hotkeys

Scenario Outline: User using a hotkey
  Given I am in the question section, logged in.
  When I'm press <key1> + <key2>
  Then hotkey is executed

Examples:
  | key1 | key2 |
  | ctrl | enter |
  | shift | c |
  | shift | e |
  | ctrl | d |
  | ctrl | f |
  | ctrl | arrowdown |
  | ctrl | arrowup |
  | ctrl | p |
  | ctrl | |
```

12.12. Reporte pruebas de concurrencia y rendimiento

Resultados de concurrencia realizando 5000 *requests* repartidos entre 50 usuarios concurrentes usando la librería *loadtest* impactando directamente contra un servidor de pruebas ubicado en Estados Unidos que dispone de 2gb de RAM y 1 núcleo de CPU a 2ghz.

La *request* solicitada fue: `/api/questions`, la cual obtiene las preguntas para el cliente.

```
{
  "totalRequests": 5000,
  "totalErrors": 0,
  "totalTimeSeconds": 117.99262780000001,
  "rps": 42,
  "meanLatencyMs": 1174.9,
  "maxLatencyMs": 1968,
  "minLatencyMs": 440,
  "percentilesMs": {
    "50": 1166,
    "90": 1451,
    "95": 1527,
    "99": 1693
  }
}
```

12.13. Estadísticas de las preguntas actuales de Sagal

Resumen, preguntas de Sagal:

Descripción	Cantidad
Total, preguntas por día (27/03/2020)	6497
Total, preguntas por mes (diciembre 2019)	192291
Promedio de preguntas de los últimos 6 meses	158761

Cantidad de preguntas últimos 6 meses en Sagal:

Mes	Cantidad
Octubre 2019	135356
Noviembre 2019	165885
Diciembre 2019	192291
Enero 2020	161985
Febrero 2020	139687
Marzo 2020	157364

Resumen, preguntas de empresa que probó el sistema:

Descripción	Cantidad
Total, preguntas por día (27/03/2020)	82
Total, preguntas por mes (diciembre 2019)	5342
Promedio de preguntas de los últimos 6 meses	4248

Cantidad de preguntas últimos 6 meses de empresa que probó el sistema:

Mes	Cantidad
Octubre 2019	3595
Noviembre 2019	5112
Diciembre 2019	5342
Enero 2020	3426
Febrero 2020	4190
Marzo 2020	3828

12.14. Interfaces requeridas para salida a producción

Documento generado 05/02/2020

Resumen

- Adquirir subscripcion Watson (no necesario para la primer puesta en produccion).
- Definir y catalogar 200 preguntas de cada categoria (no necesario para la primer puesta en produccion).
- Crear queue SQS para que Sagal-Questions envíe las respuestas a las preguntas.
- Crear endpoint para que Sagal-Questions envíe las respuestas a las preguntas
- Crear cuenta en sentry.io y pasar credenciales para usarlas
- Hacer que back loggee en el cloud watch
- Cambios al JWT de Sagal.
- Implementacion para que Sagal-Questions reciba Questions con datos de Article (Cualquier cambio en las Questions).
- Implementacion para que Sagal-Questions reciba datos de Articles (Cualquier cambio a los Articles).
- Tener un servidor disponible para poner Sagal-Questions
- Implementar IFrame en Sagal-Central para Sagal-Questions
- Datos de server para hacer deploy continuo con jenkins
- Datos de aws para subir bundle a s3
- Configurar los enviroments

Crear queue SQS para que Sagal-Questions envíe las respuestas a las preguntas

```
1 {
2   "sagalCentralId": 10, //Number
3   "answer": "Tenemos stock, saludos" //String
4   "action": "ANSWER" //Enum String, only ANSWER for the moment.
5 }
```

Crear endpoint para que Sagal-Questions envíe las respuestas a las preguntas

```
1 POST -> https://app.sagal.io/api/questions/{{sagalQuestionId}}/answers
```

```
1 {
2   "answer": "Tenemos stock, saludos" //String
3 }
```

Cambios al JWT de login de Sagal

```
1 {
2   "clientEcommerceId": 70, //Number (optional field)
3   "clientId": 1, //Number
4   "sub": "luisito" //String
5 }
```

Implementación para que Sagal-Questions reciba Questions con datos de Article

```
1 POST -> http://{{domain}}:{{port}}/api/questions
```

```

1 {
2   "sagalCentralId": 1000, //Number -> Question Id in Sagal
3   "clientId": 123, //Number -> Client Id in Sagal
4   "storeId": 321, //Number -> Store Id in Sagal
5   "clientEcommerceId": 70, //Number -> Client Ecommerce Id in Sagal
6   "customer": {
7     "id": 1, //Number -> Customer Id in Sagal (used for make actions like block User)
8     "name": "PEPE" //String -> Customer name
9   },
10  "text": "Tiene stock ?", //String -> Question text
11  "answer": null, //String -> Question answer
12  "status": "UNANSWERED", //Enum String['UNANSWERED', 'ANSWERED', 'CLOSED_UNANSWERED', 'UNDE
13  "dateCreated": 1578525656883, //Number -> Date Created in timestamp format
14  "article": {
15    "sagalCentralId": 100, //Number -> Article Id in Sagal
16    "code": "Ejemplo-123", //String -> Article SKU
17    "condition": "NEW", //Enum String['NEW', 'USED', 'RECONDITIONED']
18    "title": "Articulo ejemplo", //String -> Article title
19    "price": 123, //Number -> Article Price
20    "currency": "$", //String -> Article currency
21    "stock": 10, //Number -> Article stock
22    "warranty": "Sin garantia", //String -> Article warranty
23    "paymentMethods": [ //Array String -> Payment methods
24      "Mercado Pago"
25    ],
26    "shippingMethods": [
27      {
28        "typeName": "Montevideo", //String -> Shipping type name
29        "cost": 500 //Number -> Shipping cost
30      },
31      {
32        "typeName": "Interior",
33        "cost": 800
34      },
35      {
36        "typeName": "Mercado Envios",
37        "cost": 0
38      }
39    ],
40    "attributes": [
41      {
42        "name": "Taco", //String -> Article attribute name
43        "value": "10" //String -> Article attribute value
44      }
45    ],
46    "variations": [
47      {
48        "code": "Variation123", //String -> Variation SKU
49        "attributes": [
50          {
51            "name": "Taco", //String -> Variation attribute name
52            "value": "12" //String -> Variation attribute value
53          }
54        ],
55        "stock": 102, //String -> Variation stock
56        "color": "Rojo", //String -> Variation color
57        "size": 30 //Number -> Variation size
58      }
59    ],
60    "urlMeli": "https://www.mercadolibre.com.uy/", //String -> Link to Meli Publication
61    "urlPicture": "https://sagal.io/images/logo-header.png", //String -> Article first pi
62    "status": "PUBLISHED", //Enum String['PUBLISHED', 'PAUSED', 'DELETED', 'UNDER_REVIEW'] -
63    "dateCreated": 1578525656883 //Number -> Date Created in timestamp format
64  }
65 }

```

Implementacion para que Sagal-Questions reciba datos de Articles

1 PUT -> http://{{domain}}:{{port}}/api/articles/:sagalCentralId

```
1 {
2     "code": "Ejemplo-123", //String -> Article SKU
3     "condition": "NEW", //Enum String['NEW', 'USED', 'RECONDITIONED']
4     "title": "Articulo ejemplo", //String -> Article title
5     "price": 123, //Number -> Article Price
6     "currency": "$", //String -> Article currency
7     "stock": 10, //Number -> Article stock
8     "warranty": "Sin garantia", //String -> Article warranty
9     "paymentMethods": [ //Array String -> Payment methods
10        "Mercado Pago"
11    ],
12    "shippingMethods": [
13        {
14            "typeName": "Montevideo", //String -> Shipping type name
15            "cost": 500 //Number -> Shipping cost
16        },
17        {
18            "typeName": "Interior",
19            "cost": 800
20        },
21        {
22            "typeName": "Mercado Envios",
23            "cost": 0
24        }
25    ],
26    "attributes": [
27        {
28            "name": "Taco", //String -> Article attribute name
29            "value": "10" //String -> Article attribute value
30        }
31    ],
32    "variations": [
33        {
34            "code": "Variation123", //String -> Variation SKU
35            "attributes": [
36                {
37                    "name": "Taco", //String -> Variation attribute name
38                    "value": "12" //String -> Variation attribute value
39                }
40            ],
41            "stock": 102, //String -> Variation stock
42            "color": "Rojo", //String -> Variation color
43            "size": 30 //Number -> Variation size
44        }
45    ],
46    "urlMeli": "https://www.mercadolibre.com.uy/", //String -> Link to Meli Publication
47    "urlPicture": "https://sagal.io/images/logo-header.png", //String -> Article first picture
48    "status": "PUBLISHED", //Enum String['PUBLISHED', 'PAUSED', 'DELETED', 'UNDER_REVIEW']
49    "dateCreated": 1578525656883 //Number -> Date Created in timestamp format
50 }
```

12.15. Manual de usuario

Preguntas

Desde la pestaña **PREGUNTAS** podrás ver y responder de forma sencilla todas las consultas de los clientes.

Cada pregunta es una línea, con un click se despliega mostrándote la información relevante que te ayudara al momento de responder.

Una vez que hagas click en el botón de **"Responder"** la pregunta se ira de la lista y pasara al **Historial**.

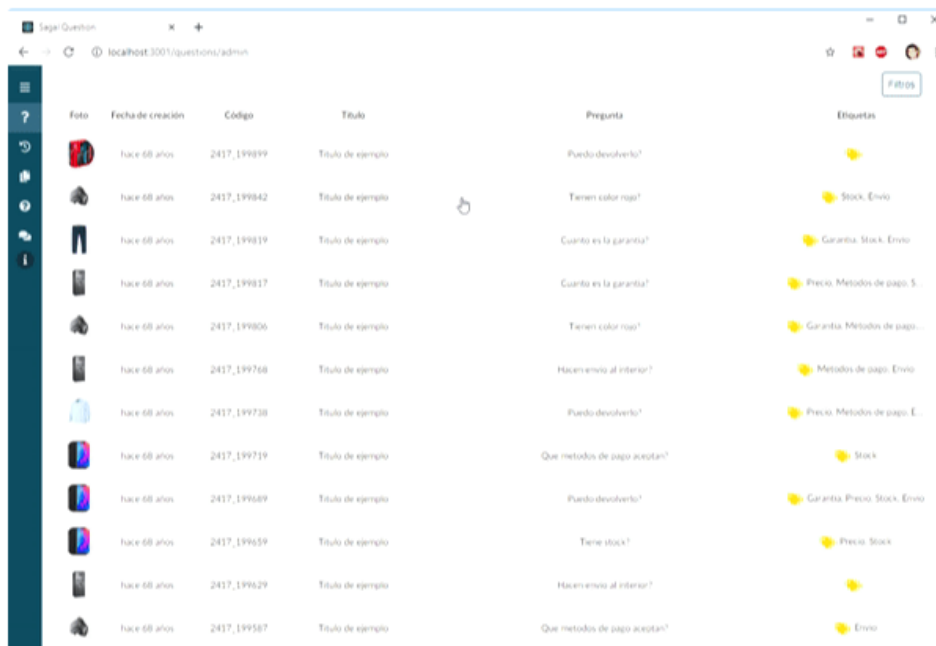
























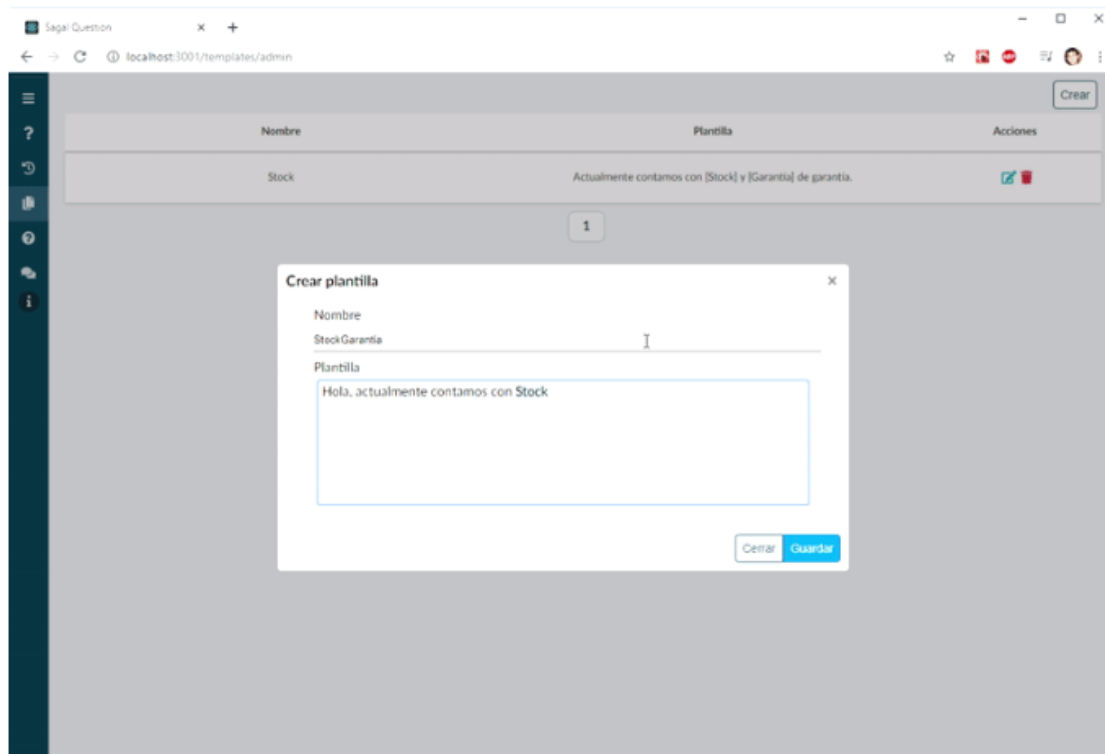


Foto	Fecha de creación	Código	Título	Pregunta	Etiquetas
	hace 68 años	2417_199899	Título de ejemplo	Puedo devolverlo?	
	hace 68 años	2417_199942	Título de ejemplo	Tienen color rojo?	 Stock, Envío
	hace 68 años	2417_199919	Título de ejemplo	Cuanto es la garantía?	 Garantía, Stock, Envío
	hace 68 años	2417_199817	Título de ejemplo	Cuanto es la garantía?	 Precio, Métodos de pago, S...
	hace 68 años	2417_199806	Título de ejemplo	Tienen color rojo?	 Garantía, Métodos de pago...
	hace 68 años	2417_199768	Título de ejemplo	Hacemos al interior?	 Métodos de pago, Envío
	hace 68 años	2417_199738	Título de ejemplo	Puedo devolverlo?	 Precio, Métodos de pago, E...
	hace 68 años	2417_199719	Título de ejemplo	Que métodos de pago aceptan?	 Stock
	hace 68 años	2417_199689	Título de ejemplo	Puedo devolverlo?	 Garantía, Precio, Stock, Envío
	hace 68 años	2417_199659	Título de ejemplo	Tiene stock?	 Precio, Stock
	hace 68 años	2417_199629	Título de ejemplo	Hacemos al interior?	
	hace 68 años	2417_199587	Título de ejemplo	Que métodos de pago aceptan?	 Envío

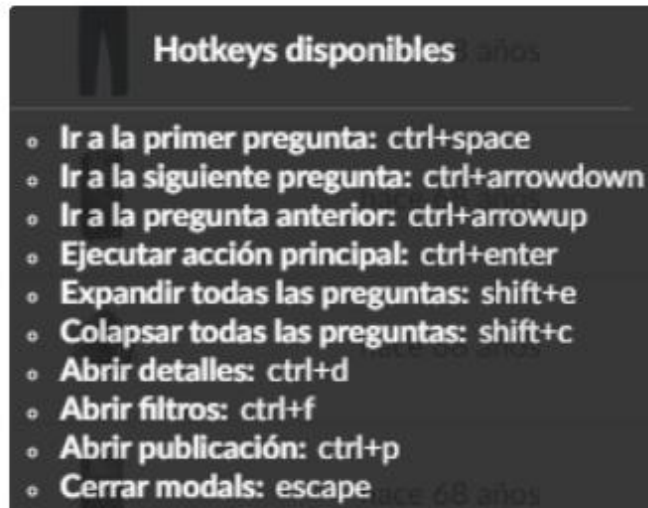
Plantillas

Como forma de facilitar la respuesta de estas preguntas, tenemos una funcionalidad donde se puede guardar una "plantilla" de texto que queremos usar de forma de ser mas rápidos a la hora de contestar las preguntas. Para esto, en el menú de "Plantillas" podemos crear atajos de texto que se pueden usar con @Atajo en la respuesta de forma de agilizar la misma.



Atajos de teclado

Existen varios atajos de teclado que nos permiten acceder a las funciones más rápido.



Hotkeys disponibles

- Ir a la primer pregunta: ctrl+space
- Ir a la siguiente pregunta: ctrl+arrowdown
- Ir a la pregunta anterior: ctrl+arrowup
- Ejecutar acción principal: ctrl+enter
- Expandir todas las preguntas: shift+e
- Colapsar todas las preguntas: shift+c
- Abrir detalles: ctrl+d
- Abrir filtros: ctrl+f
- Abrir publicación: ctrl+p
- Cerrar modals: escape

Historial

Dentro del historial podrás ver todas las preguntas y las respuestas que se dieron en su momento. También cuenta con filtros para poder ordenar y buscarlas fácilmente.

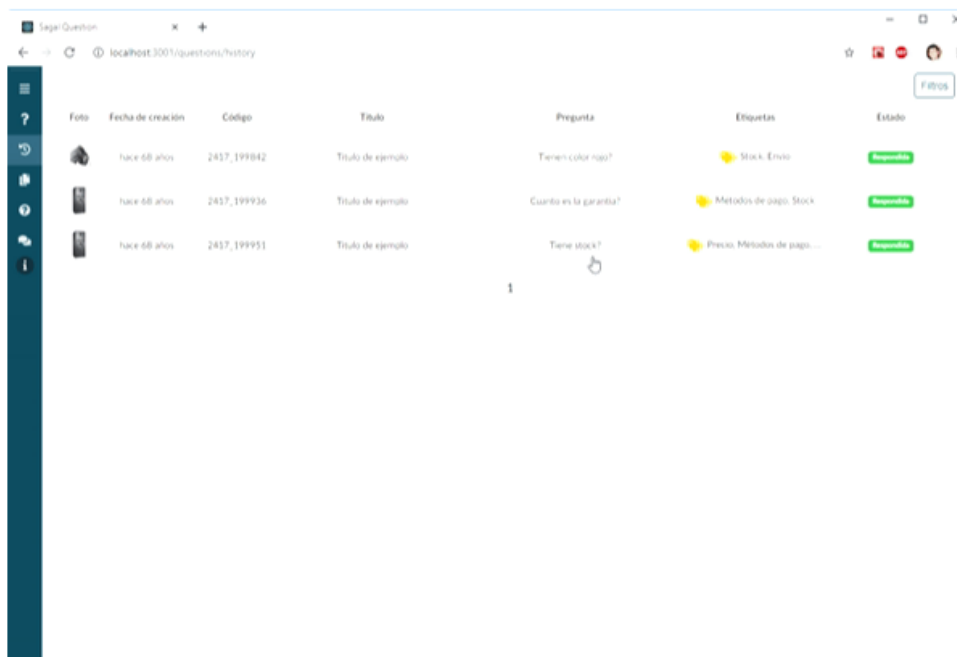





Foto	Fecha de creación	Código	Título	Pregunta	Etiquetas	Estado
	hace 68 años	2437,199842	Título de ejemplo	Tienen color rojo?	Stock, Erro	Responde
	hace 68 años	2437,199936	Título de ejemplo	Cuanto es la garantía?	Metodos de pago, Stock	Responde
	hace 68 años	2437,199951	Título de ejemplo	Tiene stock?	Preco, Metodos de pago, ...	Responde

Encuesta de evaluación

Agradecemos puedan completar esta encuesta de evaluación del uso del sistema para poder mejorar.

https://docs.google.com/forms/d/e/1FAIpQLSdajYHGpUk_RQpYRO8QXpSqhmPFnIK0wuDHG6k9MkFspxc7vg/viewform?usp=sf_link

Fallas del sistema

En caso de detectar fallas en el uso del sistema, nos pueden contactar por email o whatsapp (datos) en caso de ser una falla grave, sino por el formulario xx.

12.16. Reporte de riesgos

Agregamos a este documento los riesgos de cada Sprint, su nivel y sus planes de mitigación. Una vez finalizado el sprint revisamos si se materializaron y si los planes de mitigación fueron exitosos.

Sprint 1, 2 y 3

Para estos primeros sprints tuvimos una gestión de riesgo "ad hoc".

Detectábamos posibles riesgos y generábamos alguna estrategia de mitigación "intuitiva" como puede ser análisis, prueba de concepto o re-definición del diseño actual.

Luego del Sprint 3 detectamos que algunos riesgos eran grandes y el tamaño de la aplicación requería que se manejaran adecuadamente, así como dejarlos debidamente documentados para poder analizar el impacto de la materialización de los mismos.

Dentro de los riesgos manejados ad hoc estuvieron los siguientes:

- Implementar la aplicación como PWA
 - Análisis en el sprint 0
- Uso de Watson como plataforma de machine learning
 - Sobre estimación de la tarea de implementación
- Entrenamiento del modelo
 - Sobre estimación para categorización de preguntas iniciales

Sprint 4

1. Viaje de trabajo de @Nicolas Gonzalez - ALTO
 - a. Como forma de mitigación planificamos menos tareas dentro del sprint ya que la cantidad de horas destinadas al proyecto podía ser menor.
2. Entregables de interfaces por parte del cliente - ALTO
 - a. Fijamos una fecha limite.
3. Deploy en S3 usando pagina estática - BAJO
 - a. En caso de que no sea posible, lo subimos a un servidor propio.

Cierre de sprint:

1. No tuvo gran impacto en el proyecto. Se mitigo correctamente.
2. Se transformo en un problema por no planificar correctamente la mitigacion.
3. No se materializo el riesgo.

Sprint 5

1. Entrega de interfaces por parte del cliente - **PROBLEMA**
 - a. Al no gestionarse correctamente se convirtió en un problema. Para gestionar este problema estimamos una tarea para este sprint que implique la realización de las interfaces por nuestra parte, recortando el scope para poder poner en producción el sistema.

2. Exámenes pendientes - **BAJO**
 - a. Reducimos la cantidad de horas estimadas para llegar al sprint debido a exámenes pendientes que bajarían la cantidad de horas disponibles para el proyecto

3. Clasificación de preguntas - **BAJO**
 - a. El cliente se comprometió a clasificar las preguntas antes del **Feb 29, 2020**
 - b. En caso de que no cumplan, se clasificara el modelo con lo que se entregue. El modelo quedara menos preciso en ese caso pero no insumirá trabajo de parte del equipo.

4. No uso de la aplicación por parte del usuario final - **MEDIO**
 - a. Una vez puesta en producción, monitorearemos el uso y contestación de preguntas de forma diaria. En caso de que no veamos uso por parte del usuario "beta", agendaremos una reunión para revisar cuales son los puntos que habría que ajustar para que se use.
 - b. En caso de que los clientes designados para las pruebas no respondan, solicitaremos a Sagal que ponga el modulo en "beta" para todos los clientes. De esta manera tendremos mas alcance para que los clientes lo usen.

5. Dificultades de configuración para compartir el método de autenticacion JWT - **ALTO**
 - a. Se detectan posibles dificultades a la hora de configurar el cookie sharing para autenticar nuestra aplicación.
 - b. Como forma de mitigacion agregamos una tarea de investigación para otros posibles metodos de autenticacion dentro de la app

Sprint 6

1. Reunión con Beta tester **ALTO**
 - a. Dificultades de las generar las reuniones
 - i. Para eso vamos a tener reuniones cada 3 días con el cliente para ver el estado de la coordinación
2. Corona virus **ALTO**
 - a. Dificultades para reunirse con el cliente y el beta tester, en caso de que se impongan bloqueos pasaremos a coordinar las reuniones telefónicas o mediante vídeo conferencia

Sprint 7 - Documentacion

1. Reunión con Beta tester - PROBLEMA - **ALTO**
 - a. Dificultades de las generar las reuniones
 - i. Generado por el riesgo 2, aun no logramos coordinar las reuniones efectivamente.
 - ii. Puede generar dificultades con la completitud de la documentacion, en caso de que no sea posible obtener algunas metricas, se dejaran para la defensa y se recabaran luego de la entrega de la documentacion.
2. Corona virus **ALTO**
 - a. Dificultades para reunirse con el cliente y el beta tester, en caso de que se impongan bloqueos pasaremos a coordinar las reuniones telefónicas o mediante vídeo conferencia.
 - b. Envíos a seguro de paro dificultan el uso.

Riesgos de sprints anteriores

1. Entrega de interfaces por parte del cliente - **PROBLEMA** - Se mitigo correctamente con TES-362: [Generar interfaces para conexion con Sagal](#) **DONE**

Los demás riesgos no se materializaron.

12.17. Encuestas de satisfacción

Encuesta realizada:

Encuesta de satisfacción

Nos gustaría conocer su opinión sobre la calidad de la nueva sección de preguntas. Por favor, indique hasta qué punto está de acuerdo o en desacuerdo con los siguientes enunciados.

Algunos enunciados son similares entre si de forma de asegurarnos que podemos determinar de forma más precisa su opinión respecto al sistema.

- 1 - No estoy de acuerdo en absoluto (NA)
- 2 - Estoy en desacuerdo (D)
- 3 - No estoy de acuerdo, ni tampoco en desacuerdo (N)
- 4 - Estoy de acuerdo (A)
- 5 - Estoy muy de acuerdo (MA)

	1 - NA	2 - D	3 - N	4 - A	5 - MA
Soy capaz de llevar a cabo mi trabajo con la nueva versión	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Me es posible ejecutar funciones con precisión	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Fui capaz de aprender a usar el sistema en un corto período de tiempo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es fácil arreglar un error en el programa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es fácil identificar un error en el programa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Fue fácil comprobar el programa para asegurarnos que ejecutaba sus funciones	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Soy capaz de hacer cambios de configuración en el programa con muy poco esfuerzo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Estoy muy contento con el software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
El software cumple con mis expectativas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Resultados:

	1 - NA	2 - D	3 - N	4 - A	5 - MA
Soy capaz de llevar a cabo mi trabajo con la nueva versión	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Me es posible ejecutar funciones con precisión	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Fui capaz de aprender a usar el sistema en un corto período de tiempo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Es fácil arreglar un error en el programa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Es fácil identificar un error en el programa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Fue fácil comprobar el programa para asegurarnos que ejecutaba sus funciones	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Soy capaz de hacer cambios de configuración en el programa con muy poco esfuerzo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Estoy muy contento con el software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
El software cumple con mis expectativas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

Evaluación del sistema *

	1 - NA	2 - D	3 - N	4 - A	5 - MA
Soy capaz de llevar a cabo mi trabajo con la nueva versión	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Me es posible ejecutar funciones con precisión	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Fui capaz de aprender a usar el sistema en un corto período de tiempo	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es fácil arreglar un error en el programa	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es fácil identificar un error en el programa	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Fue fácil comprobar el programa para asegurarnos que ejecutaba sus funciones	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Soy capaz de hacer cambios de configuración en el programa con muy poco esfuerzo	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Estoy muy contento con el software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
El software cumple con mis expectativas	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

En total obtuvimos un promedio de 4.10 de satisfacción entre Sagal y el usuario final con respecto al MVP 1.

12.18. Tabla con modelo de distribución de esfuerzo

Table 4 Result of Effort Distribution (%) per activity

Activities	% Effort (this research)	% Effort (medium-large) [4]	Activities	% Effort (this research)	% Effort (medium-large) [4]
Software Phases			Configuration management	4.3	4.16
Requirements	1.6	7.5	Documentation	8.4	4.16
Specifications	7.5	7.5	Training and support	1.0	4.16
Design	6.0	10	Total	17.5	20.82
Implementation	52.0	10	Quality and testing phases		
Acceptance & deployment	5.5	7.5	Integration testing	7.0	7.5
Total	72.6	42.5	Quality assurance	0.9	8.34
Ongoing life-cycle activities			Evaluation and Testing	2.0	20.84
Project management	3.8	8.34	Total	9.9	36.68