Plan de Acciones y Actividades de Equipo

PIS, Grupo 10

Facultad de Ingeniería Udelar

[**Descripción General del Proyecto**](#_rhsrovpmq5gy) **3**

[Visión global del producto](#_qp1v1j2d6s6w) 3

[Introducción](#_w19jgh2c0sv5) 3

[Visión del producto](#_9utttzxdsedf) 3

[Descripción del problema](#_bzrecxu23nw5) 4

[Introducción](#_nif6q9q2fj5f) 4

[Diagrama de Problemas Central](#_y7r4rdcgprea) 5

[Diagrama de Problemas](#_lmfzr6e5i4cr) 6

[**Productos principales a entregar**](#_kwqq3hcbb0uh) **6**

[Descripción general](#_vzft8gh5t9f9) 6

[Descripción funcional](#_qgugut1ixpdr) 7

[Descripción técnica](#_rt0z94r7eqwn) 8

[**Proceso de desarrollo**](#_1nzik4hfa2t) **8**

[**Organización del equipo e integrantes**](#_k7ogxwby2tmo) **8**

[**Acuerdos de trabajo**](#_lg6uqyqo27yv) **10**

[Duración de las iteraciones](#_1hp4fsd778zp) 10

[Tablero de seguimiento de la iteración](#_lt3wfcum69z0) 10

[**Métodos de colaboración**](#_dj5o64x2md6h) **10**

[**Seguimiento y estimación**](#_6noib77zlmul) **10**

[Seguimiento](#_tw73u9ja0fjk) 10

[Estimación](#_cvdvlcrtx1he) 11

[**Recursos necesarios**](#_8sfpwff1qjs9) **11**

[Herramienta para la gestión del Backlog del producto y el alcance de cada iteración](#_3yzyrd2how17) 11

[Herramienta para registro de esfuerzo](#_35philt2iona) 11

[Herramienta para la gestión del tablero de seguimiento](#_5554b8s62jes) 11

## Descripción General del Proyecto

## Visión global del producto

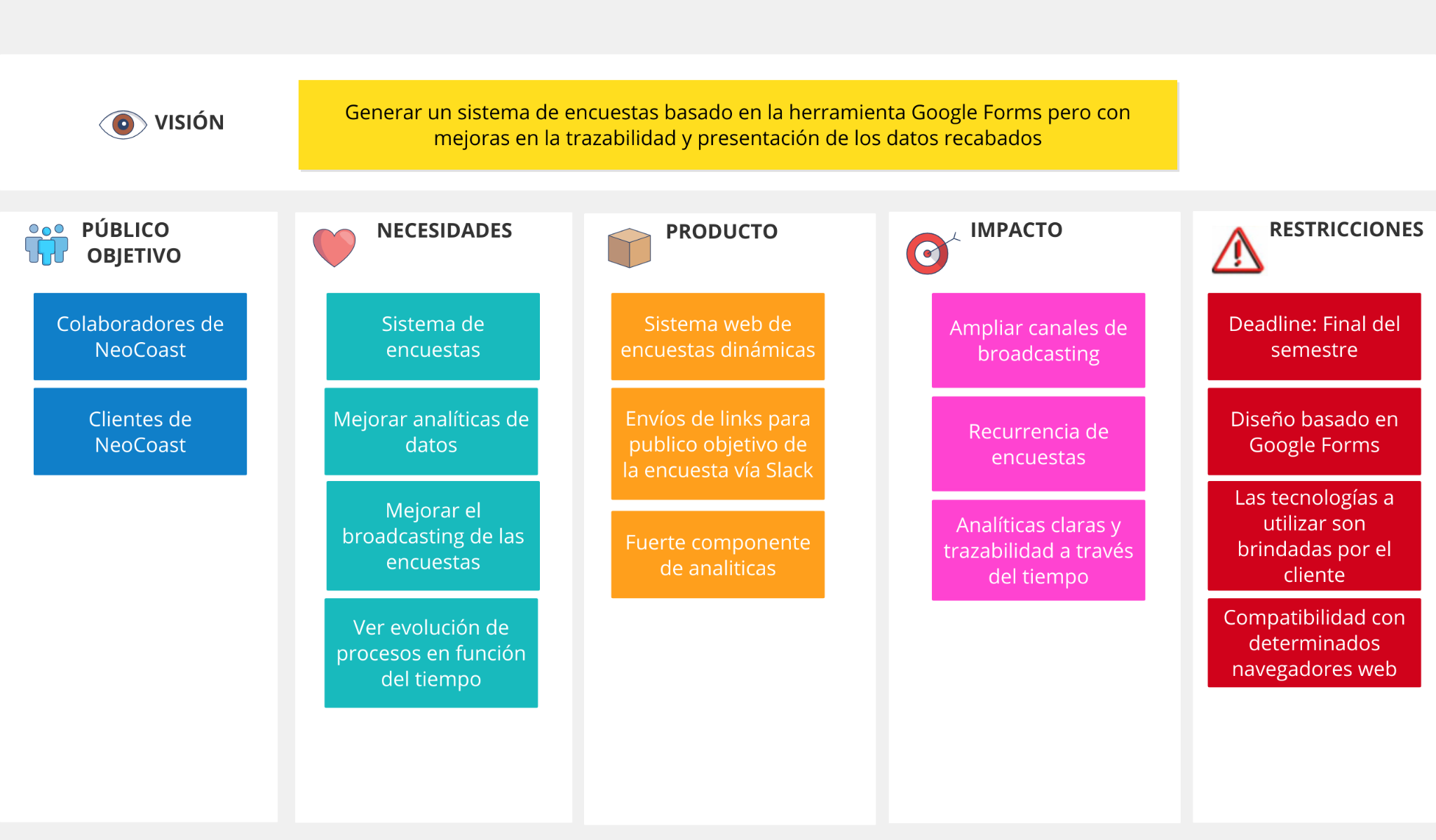
### Introducción

Dado el problema planteado por NeoCoast, el objetivo principal es crear una plataforma web de encuestas para satisfacer sus necesidades, que a su vez abarcan a las de sus clientes y sus empleados/colaboradores.

La principal necesidad del cliente es la resolución eficiente del manejo de respuestas de encuestas obtenidas a través de sus clientes/empleados, con el objetivo de poner énfasis en la mejora de procesos, organización interna y áreas de trabajo de la empresa.

Para ello, creen necesario tener más precisión en la manipulación de datos generados, planteando el acceso a un historial de respuestas de encuestas, la generación de un análisis gráfico o porcentual, y la obtención de detalles específicos o grupales de las mismas.

### Visión del producto



Mencionaremos los principales puntos a tener en cuenta al momento de planificar la solución a presentar:

* La plataforma debe contar con tres tipos de usuarios:
  + Administradores
  + Empleados de NeoCoast
  + Invitados.
* Los administradores pueden realizar encuestas, y enviarlas a grupos de usuarios y/o usuarios específicos. Los datos obtenidos deben guardarse y poder ser accedidos sin un límite de tiempo, ésta es una de las principales funcionalidades, ya que forma parte del problema planteado por el cliente.
* Dado que el sistema anterior utilizado por el cliente era Google Forms, se debe tener en cuenta el brindarle una solución que le facilite el uso en términos de interfaz de usuario, y a su vez en funcionalidades. Estos detalles nos dan una visión sobre el formato que debe tener la plataforma en relación a las encuestas, y nos permite sugerir al cliente mejoras/extensiones de funcionalidades.
* Los administradores tendrán acceso a información sobre analíticas de encuestas, las cuales podrán ser representadas mediante gráficas o tablas. Dichas analíticas pueden estar relacionadas a una encuesta en específico, o a un grupo de encuestas, y permitir una serie de filtros para hilar lo más fino posible en la información que se desea analizar.
* Los usuarios pueden recibir un recordatorio sobre una encuesta a responder a través de la aplicación Slack, o a través de su email. Esta funcionalidad puede ser programada para que ocurra con una determinada frecuencia.

## Descripción del problema

### Introducción

NeoCoast es una empresa desarrolladora de software a medida que provee soluciones web y móviles utilizando metodologías ágiles y las últimas tecnologías en mercado. Además provee un servicio de ampliación de staff, donde pueden acoplarse a otras empresas para sacar adelante su proyecto.

NeoCoast se preocupa por la opinión de sus clientes y el clima laboral en la oficina. Con este fin, desde el año 2018 presentan regularmente diversas encuestas de satisfacción utilizando la aplicación Google Forms. Dado que actualmente se genera un volumen muy alto de encuestas, Google Forms no resuelve el objetivo fundamental de las mismas, que es proveer un mecanismo sencillo para obtener información y así lograr el refinamiento del procesos de trabajo y diversas áreas.

En resumen, el análisis de las encuestas generadas se ha vuelto tedioso y actividades como filtrar y obtener un histórico de resultados resulta muy complejo para NeoCoast. Esta actividad pretende elaborar un diagrama de problema central y otro de problemas generales utilizando la técnica de “5 porqués”.

### Diagrama de Problemas Central

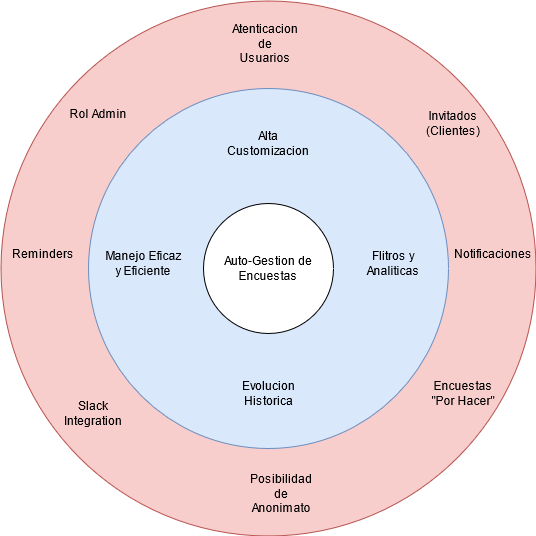


Diagrama del Problema Central

* **Círculo Blanco:** Problema Principal de Neo Coast. Contar con una plataforma para gestionar las encuestas que se realicen.
* **Círculo Azul**: Razones que generan la necesidad. Se desean obtener analíticas de las encuestas y contar un registro histórico de las mismas para filtrar y obtener la información necesaria.
* **Círculo Rojo**: Impacto del problema en los usuarios (internos y externos). Los usuarios podrán loguearse y completar encuestas que les hayan sido enviadas o ver sus respuestas en encuestas pasadas. Serán notificados mediante integraciones con Slack y también por email. Un usuario en rol de administrador podrá programar encuestas, enviarlas al grupo deseado, etc.

### Diagrama de Problemas

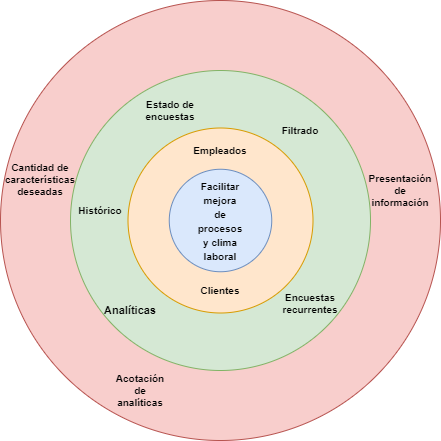


Diagrama de Problemas?

**Círculo azul:** ¿Para qué? La necesidad viene dada de poder mejorar los procesos tanto con el cliente como dentro de la empresa de una manera sencilla.

**Círculo naranja**: ¿Para quiénes? Los empleados de NeoCoast y los clientes de la empresa serán los usuarios del sistema.

**Círculo verde**: ¿Cómo? Manteniendo un histórico de las encuestas realizadas y generando diversas analíticas sobre estas se podrá obtener mejor información para detectar posibles mejoras en los procesos.

**Círculo rojo**: ¿Problemas? Puede que se generen muchas analíticas haciendo más difícil poder obtener información relevante. En esta misma línea se debe tener consideración a la hora de mostrar la información procesada y no presentar gráficas o datos que no colaboren con la necesidad planteada.

# Productos principales a entregar

## Descripción general

Se deberá constuir una API y una aplicación web que haga uso de la misma, la cual trendrá 3 tipos de usuarios:

* Administrador
* Empleado de NeoCoast
* Invitado

Los administradores de la plataforma serán capaces de poder crear encuestas y enviarlas a los demás tipos de usuarios, además podrá listarlas, filtrarlas por distintos parámetros y hacerles seguimiento.

Los empleados podrán iniciar sesión en la plataforma y visualizar todas las encuestas que deben responder así como las ya finalizadas.

Se espera que los clientes de la empresa puedan acceder a la plataforma como

invitados para responder encuestas.

## Descripción funcional

A continuación se detallarán los requisitos funcionales:

| **#** | **Requisito funcional** |
| --- | --- |
| 1 | El empleado debería poder responder a una encuesta. |
| 2 | El empleado debería poder ver y editar una respuesta a una encuesta que realizó. |
| 3 | El empleado debería poder loguearse al sistema. |
| 4 | El empleado debería poder ver encuestas que están programadas. |
| 5 | El administrador debería poder ingresar grupos de destinatarios para enviar una encuesta. |
| 6 | El administrador debería poder ingresar manualmente destinatarios para enviar una encuesta. |
| 7 | El administrador debería poder crear una encuesta recurrente. |
| 8 | El administrador debería poder loguearse al sistema. |
| 9 | El administrador debería poder crear una encuesta programada (incluye instantáneas). |
| 10 | El administrador debería poder editar una encuesta programada que no fue enviada. |
| 11 | El administrador debería poder ver el listado de encuestas existentes. |
| 12 | El administrador debería poder filtrar en la lista de respuestas existentes. |
| 13 | El administrador debería poder establecer recordatorios asociados a encuestas. |
| 14 | El administrador debería poder crear recordatorios para administradores/as. |
| 15 | El administrador debería poder crear, ver, editar y borrar grupos de usuarios/as. |
| 16 | El administrador debería poder agregar integrantes a grupos de usuarios/as. |
| 17 | El administrador debería poder sacar integrantes de grupos de usuarias/os. |
| 18 | El administrador debería poder ver estadísticas de una encuesta. |
| 19 | El administrador debería poder agregar, quitar y editar preguntas de una versión de una encuesta recurrente que aún no fue enviada. |
| 20 | El administrador debería poder ver estadísticas referentes a la creación de encuestas. |
| 21 | El administrador debería poder ver quiénes respondieron y quiénes no respondieron a una encuesta. |
| 22 | El administrador debería poder pausar la recurrencia de una encuesta recurrente. |
| 23 | El administrador debería poder reanudar la recurrencia de una encuesta recurrente pausada. |
| 24 | El administrador debería poder dar de baja una encuesta programada antes de que se envíe. |
| 25 | El administrador debería poder volver a programar una encuesta programada que fue dada de baja. |
| 26 | El administrador debería poder responder a una encuesta. |
| 27 | El administrador debería poder ver y editar una respuesta a una encuesta que realizó. |
| 28 | El administrador debería poder enviar invitaciones a registrarse para empleados o administradores. |
| 29 | El usuario debería poder registrarse |
| 30 | El invitado debería poder responder una encuesta |
| 31 | El sistema debería enviar las encuestas programadas que no hayan sido dadas de baja cuando sea la fecha y hora de envío. |
| 32 | El sistema debería pasar las entrevistas al estado “respondidas” cuando el plazo de respuestas haya transcurrido. |
| 33 | El sistema debería enviar recordatorios cuando sea la fecha y hora de envío. |
| 34 | El sistema debería loguear cada interacción frontend/backend que un/a administrador/a tenga con él. |
| 34 | El sistema debería enviar las encuestas recurrentes que no hayan sido dadas de baja cuando se cumpla una nueva iteración de recurrencia. |

## Descripción técnica

Para el desarrollo de la API se estará utilizando Ruby 2.7, Rails 6 y RSpec para los tests. Para el caso de la base de datos estaremos utilizando Postgresql.

Para el desarrollo de la aplicación web estaremos utilizando React 17.0.

La aplicación web y la API serán alojadas en Heroku.

# Proceso de desarrollo

Se decidió utilizar el proceso PISAgile, un proceso creado específicamente en el marco de la asignatura Proyecto de Ingeniería de Software, el cual tiene un fuerte componente ágil con incorporaciones de otros enfoques. PISAgile tiene como objetivo adaptarse a las prácticas de la Ingeniería de software vinculadas al desarrollo evolutivo; experimentar de forma temprana para agregar valor al cliente y obtener feedback lo antes posible. Para su definición se consideran aspectos de Scrum, Kanban, Rational Unified Process (RUP), eXtreme Programing (XP), Design Thinking, Gestión 3.0 y prácticas generales de la Ingeniería de software.

Se decidió utilizar este proceso de desarrollo debido a que el equipo se siente más cómodo utilizando metodologías ágiles, así como también al conocer que el cliente también suele utilizar metodologías ágiles, se definió que PISAgile sería el mejor proceso para este proyecto.

Por otra parte, la posibilidad de tener 3 semanas de ciclo inicial, como lo establece esta metodología, repercutió positivamente a la hora de tomar la decisión. En nuestro caso, al no tener mucho conocimiento de las tecnologías a utilizar en el proyecto, se decidió poder tener un plazo inicial mayor para poder tener conocimiento de dichas tecnologías, así como también tener mayor tiempo para poder realizar un análisis de la solución.

# Organización del equipo e integrantes

Analizando los roles establecidos por la metodología PISAgile, se procedió a realizar determinados cambios en algunos de los mismos por cual se decidió la creación de nuevos roles los cuales están detallados a continuación: **Ayudantes**  
En la fase inicial decidimos ayudantes para los responsables de seguimiento y SCM, pero la intención del equipo es que el rol de “Ayudante” pueda estar habilitado para cualquiera de los responsables. Un ayudante, como lo dice la palabra, será de soporte y ayudará a los responsables de cada área con sus obligaciones.

**Desarrollador (Análisis/Diseño) y Desarrollador (Construcción)**  
En la definición de PISAgile, un desarrollador forma parte de todas estas áreas del proceso de software. Nuestro equipo optó por separarlos por dos roles distintos según el “enfoque” del desarrollador.   
  
La decisión está basada principalmente en dos motivos: (1) La fase inicial del proceso se caracteriza por tener un fuerte enfoque en las áreas de Diseño y Análisis que consideramos necesitan tener una atención dedicada, y su vez, el equipo también necesita que otros integrantes se dediquen más a experimentar con las tecnologías a utilizar durante la etapa de construcción. El equipo siente que de esta manera no solo está respetando las necesidades actuales del proyecto sino que también (2) podemos dedicarnos en profundidad a las áreas que más nos interesan a cada uno.  
  
No obstante, tanto un “Desarrollador (Análisis/Diseño)” como un “Desarrollador (Construcción)” son ambos “Desarrollador” y podrían seguir siendo considerados como tal en la definición original del rol del PISAgile.

**Resp. de Diseño y Resp. de Construcción**  
  
Motivado también por las razones listadas para el rol de desarrollador, se decidió separar estos roles que son originalmente uno, en dos, ayuda a una repartición más “saludable” de las responsabilidades, de forma que cada uno se pueda enfocar en su área durante esta etapa del proceso.   
  
La intención del grupo seguramente sea de mantener esta separación para fases futuras, pero sin duda durante la fase de construcción estos roles trabajarán muy en conjunto.

Con la inclusión de los nuevos roles de ayudante, desarrollador de análisis/diseño, desarrollador de construcción, responsable de diseño y responsable de construcción, se establecieron los siguientes roles para cada uno de los integrantes del equipo:

| **Integrante** | **Rol 1** | **Rol 2** |
| --- | --- | --- |
| Agustin Brozia | Dev (Analisis/Disenio) | Dev (Construccion) |
| Bruno De Simone | Resp. Calidad | Ayud. SCM |
| Ciro Sayagués Laso | Dev (Analisis/Disenio) | Dev (Construccion) |
| Federico Gelso Gonzalez | Facilitador | Dev (Analisis/Disenio) |
| Gianluca Giordano Pignatta | Resp. SCM | Ayud. Seguimiento |
| Javier Gómez | Dev (Construccion) | Dev (Analisis/Disenio) |
| Lorenzo Motta | Dev (Construccion) | -- |
| Lucas Fontes | Resp. Construccion | Dev (Analisis/Disenio) |
| Manuel Estefanell | Dev (Construccion) | Dev (Analisis/Disenio) |
| Marcelo Marenales | Dev (Construccion) | -- |
| Martin Larroca | Dev (Construccion) | Dev (Analisis/Disenio) |
| Rodrigo Gonzalez | Dev (Construccion) | -- |
| Santiago Alaniz | Resp. de Diseno | -- |
| Santiago Hernández | Resp. Seguimiento | Dev (Analisis/Disenio) |

# 

# Acuerdos de trabajo

## Duración de las iteraciones

Basándonos en la metodología PISAgile, las iteraciones tendrán una duración de dos semanas, realizándose 5 diferentes iteraciones a lo largo del desarrollo del proyecto.

## Tablero de seguimiento de la iteración

El tablero de seguimiento de la iteración estará implementado en Jira y en el mismo se podrán observar todas las historias de usuario a desarrollar en la iteración así como también los diferentes entregables a lo largo del sprint.

Para poder organizarnos a la hora de reconocer el estado de avance de cada elemento del tablero, se utilizaran los diferentes estados:

* Por hacer: indica que el ítem no se ha comenzado a realizar.
* Bloqueado: indica que el ítem no debe ser realizado.
* En curso: indica que el ítem está en proceso de creación.
* Revisión: indica que el ítem está finalizado y puede ser revisado.
* Listo: indica que el ítem ya fue finalizado y revisado.

# Métodos de colaboración

Para la comunicación con el cliente el equipo realizará meetings semanalmente y para comunicaciones informales o aclaración de dudas simples se utilizara Slack.

# Seguimiento y estimación

## Seguimiento

Para el seguimiento en cuanto a lo realizado en la semana se utilizará un tablero realizado en una hoja de cálculo de Google, en ella cada uno de los integrantes debe indicar cuánto tiempo dedicó a cada tarea dia a dia. Además en dicha planilla se indica la estimación que se le asignó, por lo que el equipo debe comparar si está cumpliendo con lo estimado.

## Estimación

Para estimar el tiempo de dedicación que se le debe otorgar a cada tarea(entregables o ítems del backlog), se va a utilizar una técnica basada en el consenso grupal llamada Planning Poker. Para este se utilizara un mazo de tarjetas que muestre la secuencia de Fibonacci, donde cada número equivale a horas de estimación.

El equipo espera poder utilizar las dos herramientas en conjunto para poder tener una mejor retrospectiva de cada iteración y poder analizar en qué se puede mejorar para futuros sprints.

# Recursos necesarios

## Herramienta para la gestión del Backlog del producto y el alcance de cada iteración

Para poder gestionar el Backlog de producto de forma correcta, estaremos utilizando la herramienta Jira, la cual nos permite definir las historias de usuario a realizar en cada sprint y poder establecer el estado de las mismas. Con estas herramientas, es posible asignar un responsable a cada una de las historias de usuario y poder definir el equipo que también va a estar participando de las mismas. Con esto podemos observar el estado actual del backlog en cada momento del sprint y poder reasignar tareas en caso de que el progreso no sea el esperado.

## Herramienta para registro de esfuerzo

A la hora de registrar el esfuerzo, estamos utilizando la herramientas Google Sheets donde en una determinada hoja de cálculo estaremos logrando detallar todo el esfuerzo semanal. En una de las hojas, estaremos creando una tabla de esfuerzo donde cada integrante del equipo deberá ingresar la cantidad de horas semanales totales realizadas en cada una de las actividades. Por otra parte, para tener un registro de horas diario, estaremos definiendo otra tabla en la cual cada integrante del equipo deberá detallar las horas diarias para cada una de las tareas realizadas en dicho día.

Con esto podremos saber el progreso del equipo a lo largo de las horas y nos ayudará a poder estimar semana a semana la cantidad de horas que nos llevará cada una de las tareas e items del backlog.

## Herramienta para la gestión del tablero de seguimiento

Para gestionar el tablero de seguimiento también estaremos utilizando Jira donde podremos reflejar el estado actual de cada uno de los documentos y tareas que tengamos que realizar a lo largo de cada iteración. Además, cada una de las tareas tendrá un responsable, que será la persona encargada de reflejar el status actual de cada tarea y actividad en el tablero de seguimiento.