Índice

[Introducción 3](#_heading=h.xdaz7v5tlh71)

[Product Owner 3](#_heading=h.k5p1us9jyyvx)

[Equipo del Proyecto 3](#_heading=h.o306k4sc5s4n)

[Scrum Master 3](#_heading=h.446u8qr9noos)

[Equipo de Desarrollo 3](#_heading=h.lwdg4fg01y6j)

[Acuerdos 4](#_heading=h.inea3z6i0yx9)

[Definición del Product Backlog Inicial 4](#_heading=h.prg9ebi9sqmy)

[Técnica de estimación a utilizar 4](#_heading=h.acieu61h5ksl)

[Definición de la Tecnología a utilizar en el desarrollo del producto 5](#_heading=h.x01lycoxnd9s)

[Métricas del Proyecto 5](#_heading=h.1piq4hc6zy9q)

[Tipos de métricas del proyecto 5](#_heading=h.1vejd32cy2oj)

[Trabajo pendiente de los sprints 5](#_heading=h.pxkcszwcnvul)

[Trabajo pendiente de epics y publicaciones 6](#_heading=h.u2k681c25t9h)

[Velocidad 6](#_heading=h.ok26npyke3vg)

[Gráfico de control 6](#_heading=h.wyih7z39sqgx)

[Diagrama de flujo acumulado 6](#_heading=h.d03qul44yiqd)

[Métricas a utilizar 6](#_heading=h.yx612blwv2tq)

[Burntdown Chart 6](#_heading=h.z00wz0a2f7fp)

[Velocity Chart 7](#_heading=h.onz2lsequtkb)

[Gestion de Configuracion de Software 8](#_heading=h.hpkgeg4sdixm)

[Estructura del repositorio 8](#_heading=h.qdna95xa6frv)

[Ítems de configuración 9](#_heading=h.75uedu16v3ig)

[Glosario 10](#_heading=h.elubpma86ij1)

# 

# 

# Introducción

En este informe se expone la solución propuesta para el proyecto del turnero de castración; tratará de concretar un seguimiento del desarrollo de la app, teniendo en cuenta tanto las características funcionales como técnicas del sistema. Se utilizará la metodología Scrum para su desarrollo, se incluirá un Product Backlog que detalla las historias de usuario, lo cual facilitará la comprensión de las funcionalidades del proyecto. Con el fin de lograr una mejor productividad en las realizaciones de estas historias de usuario, definiremos los roles de cada miembro de nuestro grupo; esto permitirá mejorar nuestra organización y facilitará la asignación de tareas. Además analizaremos nuestro desenvolvimiento y eficiencia con gráficas e informes propios de esta metodología.

# Product Owner

**María Celeste Ávila, Directora de Fauna Doméstica**

# Equipo del Proyecto

## Scrum Master

* **Juan Ignacio Camargo**

## Equipo de Desarrollo

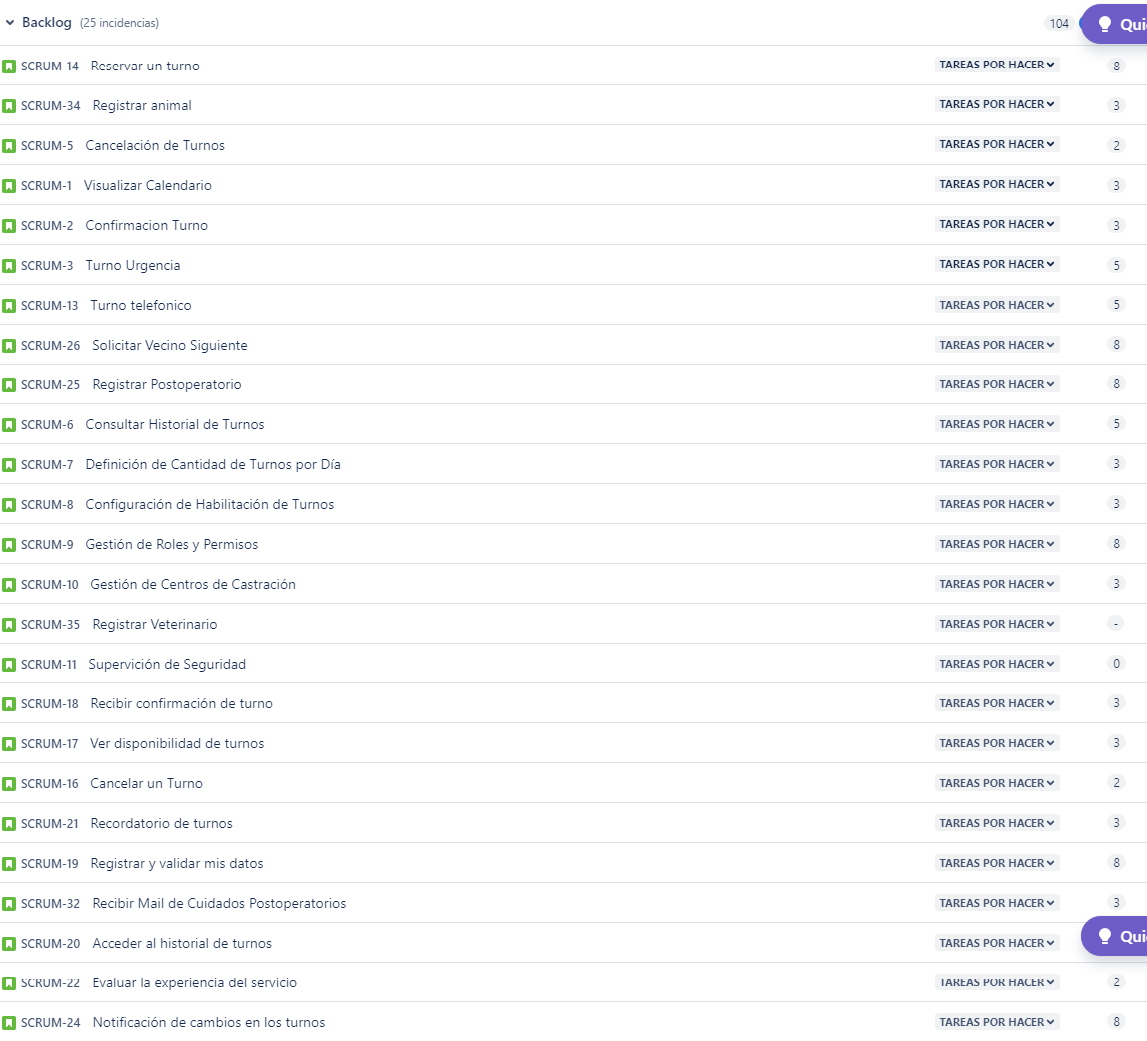
* **Sharon Puente (Product Owner)**
* **Agustín Colodny**
* **Juan Sebastian Veron**
* **Valentin Zabala**
* **Martin Pignata**

## Acuerdos

**Meets semanales**

* Domingos 19:00 hs - 22:00 hs
* Martes 18:00 hs - 21:00 hs
* Jueves 19:00 hs

# Definición del Product Backlog Inicial



# Técnica de estimación a utilizar

La técnica que utilizaremos para el proceso de estimación y planificación para nuestro proyecto de metodología ágil, va a ser el poker planning y los story points. Primeramente para aclarar los conceptos, se definen como:

Story Points son una unidad de medida utilizada para cuantificar el esfuerzo requerido para completar una historia de usuario. A diferencia de las horas o días, los story points se basan en una escala relativa que considera la complejidad, el riesgo y la incertidumbre asociados a una tarea. Por ejemplo, si una historia de usuario se estima en 8 story points, significa que se considera que es significativamente más compleja o requiere más esfuerzo que una tarea de 5 story points.

Poker Planning, por otro lado, es una técnica de estimación colaborativa utilizada en metodologías ágiles, como Scrum, para prever el esfuerzo necesario para completar una tarea o historia de usuario. En esta técnica, el equipo de desarrollo discute los requisitos de una historia y cada miembro del equipo asigna una estimación en puntos de historia usando cartas con valores predefinidos. Las estimaciones se basan en la complejidad, el tiempo y el esfuerzo necesario. El proceso se repite hasta que se alcanza un consenso sobre la estimación final.

Estas técnicas ayudan a mejorar la precisión, colaboración y flexibilidad en la gestión de tareas del proyecto, de manera que se fomenta la discusión y el consenso entre los miembros del equipo, lo que suele llevar a estimaciones más precisas; permite capturar diferentes perspectivas y conocimientos, lo que resulta en una evaluación más completa de las tareas. Además, los equipos pueden planificar mejor la capacidad de trabajo para cada sprint, las estimaciones en puntos de historia permiten establecer una velocidad promedio del equipo y ajustar las cargas de trabajo en consecuencia, facilitando la gestión de expectativas y la asignación de tareas. Esta última técnica también es útil para medir la capacidad del equipo de forma continua y ajustar las estimaciones según la evolución del proyecto.

En definitiva, creemos que la metodología de estimación que llevaremos a cabo, será una actividad colaborativa que involucre a todo el equipo, promoviendo el diálogo y la alineación sobre los objetivos y requisitos del proyecto. Esto fortalece el trabajo en equipo y es una herramienta que nos beneficiará en la planificación del proceso y ayudará a la retroalimentación del mismo.

# Definición de la Tecnología a utilizar en el desarrollo del producto

Para el desarrollo del producto utilizaremos una combinación de tecnologías que garantizan un buen rendimiento y escalabilidad. Para el backend, emplearemos ASP.NET, un framework que nos permitirá gestionar de manera eficiente la lógica de negocio y las APIs del sistema. En el frontend, utilizaremos React, una librería de JavaScript que nos ayudará a crear interfaces de usuario interactivas y dinámicas, mejorando la experiencia del usuario. Para el almacenamiento de datos, optamos por SQL Server como base de datos relacional, lo que nos asegura un manejo eficiente y seguro de la información, con capacidad para escalar según las necesidades del sistema.

# Métricas del Proyecto

Las métricas ágiles proporcionan información sobre la productividad en las distintas fases de un ciclo de vida de desarrollo de software. Esto ayuda a evaluar la calidad de un producto y realizar un seguimiento del rendimiento de un equipo.

Supervisar y compartir métricas ágiles sólidas puede reducir la confusión y arrojar luz sobre el progreso (y retrasos) del equipo a lo largo del ciclo de desarrollo.

## Tipos de métricas del proyecto

### Trabajo pendiente de los sprints

Al inicio del sprint, el equipo plantea un pronóstico de la cantidad de trabajo que puede completar durante el sprint. Un informe de evolución de sprints supervisa la finalización del trabajo a lo largo del sprint. El eje de abscisas representa el tiempo y el eje de ordenadas se refiere a la cantidad de trabajo que queda por completar, medido en puntos de historia u horas.

### Trabajo pendiente de epics y publicaciones

Los diagramas de trabajo pendiente de epics y versiones sirven para realizar el seguimiento del progreso del desarrollo a lo largo de una muestra más amplia de trabajo que en el diagrama de trabajo pendiente de sprints, y guía el desarrollo de los equipos de Scrum y Kanban.

### Velocidad

La velocidad es la cantidad media de trabajo que un equipo de scrum lleva a cabo durante un sprint, medida en puntos de historia u horas, y es muy útil para los pronósticos. Un descenso de la velocidad media generalmente es un indicador de que alguna parte del desarrollo del equipo se ha vuelto ineficiente y debería tratarse en la siguiente retrospectiva.

### Gráfico de control

Los gráficos de control se centran en el tiempo del ciclo de las incidencias individuales, es decir, el tiempo total para pasar de "en curso" a "finalizado". Medir la duración del ciclo es un modo flexible y eficiente de mejorar los procesos de un equipo porque los resultados de los cambios se detectan casi instantáneamente, de modo que se permiten ajustes inmediatos.

### Diagrama de flujo acumulado

Un diagrama de flujo acumulativo muestra la acumulación de progreso a lo largo del tiempo. El diagrama de flujo acumulado debería ser una curva suave de izquierda a derecha. Las burbujas o huecos en cualquier color indican limitaciones y cuellos de botella.

## Métricas a utilizar

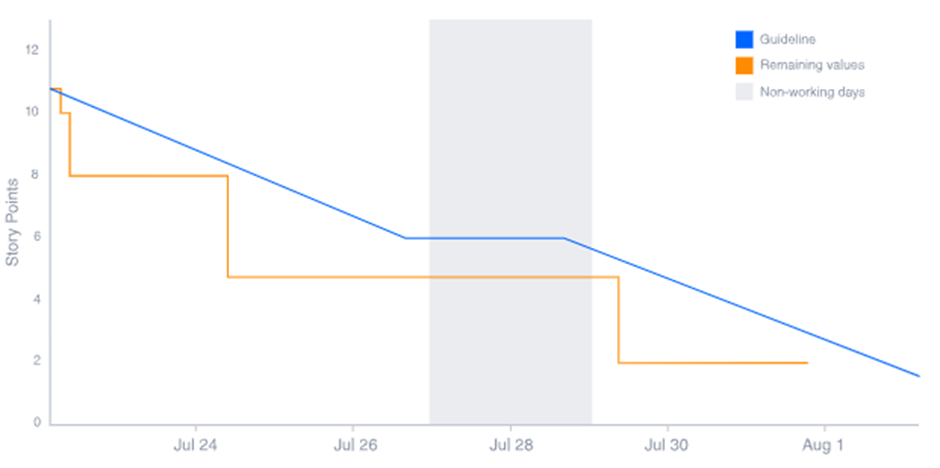
### Burntdown Chart

El burntdown chart muestra la cantidad de trabajo que se ha completado en una epic o sprint y el trabajo total restante. Los burntdown charts se utilizan para predecir la probabilidad de que el equipo complete su trabajo en el tiempo disponible. También son excelentes para mantener al equipo al tanto de cualquier aumento del alcance que se produzca.

Los burntdown charts son útiles porque permiten comprender cómo trabaja el equipo. Por ejemplo:

* Si notas que el equipo termina constantemente el trabajo antes, esto podría ser una señal de que no se están comprometiendo a trabajar lo suficiente durante la planificación del sprint.
* Si constantemente no cumplen con sus pronósticos, esto podría ser una señal de que se han comprometido a trabajar demasiado.
* Si el gráfico de evolución muestra una caída pronunciada durante el sprint, esto podría ser una señal de que el trabajo no se ha estimado con precisión o no se ha desglosado correctamente.

**Gráfico de Burntdown Chart**

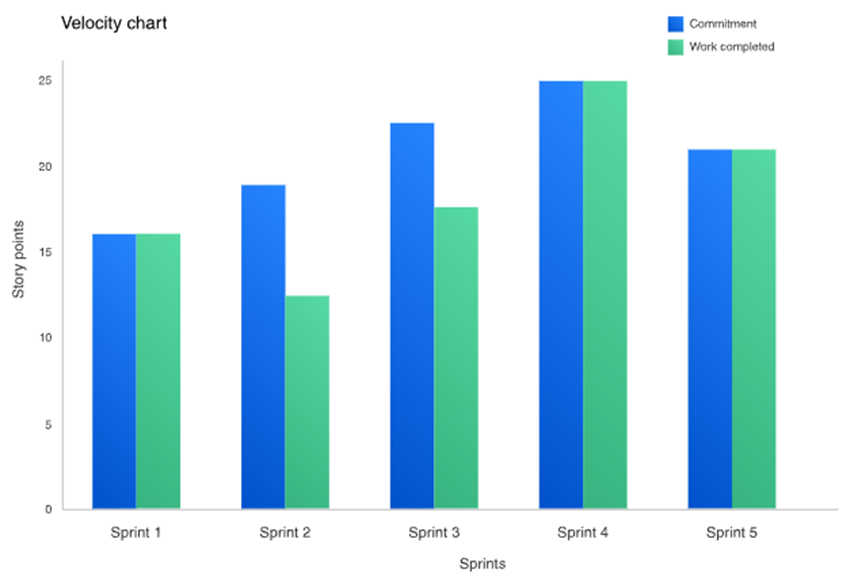
****

### Velocity Chart

El velocity chart muestra la cantidad promedio de trabajo que un equipo Scrum completa durante un sprint. Los equipos pueden usar la velocidad para predecir qué tan rápido pueden trabajar con el trabajo pendiente, ya que el informe rastrea el trabajo pronosticado y completado durante varios sprints. Cuantos más sprints, más preciso será el pronóstico.

La velocidad se calcula tomando el promedio de las estimaciones totales completadas durante los últimos sprints.

**Gráfico de Velocity Chart**

****

# Gestion de Configuracion de Software

## Estructura del repositorio

* Turnero de castración
  + Sprint
    - Documentación
      * Planificación y seguimiento
        + Planificación
        + Resultados
      * Definición del producto
        + Listado de Epics y US
        + Diagramas
  + Desarrollo del producto
    - Backend
      * Código
    - Frontend
      * Código
  + Manual de usuario

## Ítems de configuración

| **Nombre del ítem de configuración** | **Regla de nombrado** | **Ubicación** |
| --- | --- | --- |
| Sprint Número | Sprint\_N<número de sprint> | /Turnero de castración/ |
| Planificación | Planificación\_S<número de sprint>.pdf | /Turnero de castración/Sprint Número/Documentación/Planificación y Seguimiento |
| Resultados | Resultados\_S<número de sprint>.pdf | /Turnero de castración/Sprint Número/Documentación/Planificación y Seguimiento |
| Listado de Epics y US | Listado\_EpicsUS\_S<número de sprint>.pdf | /Turnero de castración/Sprint Número/Documentación/Definición del Producto |
| Diagramas | Diagrama\_de\_<tipo de diagrama>\_S<número de sprint>.pdf | /Turnero de castración/Sprint Número/Documentación/Definición del Producto |
| Código | <Nombre de componente>.<extensión> | /Turnero de castración/Sprint Número/Desarrollo del producto |
| Manual de Usuario | Manual\_de\_Usuario.pdf | /Turnero de castración/ |

## Glosario

| **Sigla** | **Significado** |
| --- | --- |
| N<número de sprint> | Abreviación de Número de sprint ‘X’. Ejemplo: “N2” = Número de Sprint 2. |
| S<número de sprint> | Abreviación de Sprint ‘X’. Ejemplos: “S4” : Sprint 4. |
| <tipo de diagrama> | Tipo de diagrama UML. Ejemplos: “Clase”, “Comunicacion”, “Secuencia”, etc. |
| <extensión> | Nombre de la extensión de un archivo identificado como ítem de configuración |