# 

[**Capítulo I: Sprint 0 3**](#_heading=h.1fob9te)

[1.1 Configuración del Entorno (Docker) 3](#_heading=h.3znysh7)

[1.2 Producto Inicial 5](#_heading=h.2et92p0)

[1.2.1 Definición del Producto Inicial 5](#_heading=h.tyjcwt)

[1.2.2 Visión inicial del producto 6](#_heading=h.3dy6vkm)

[1.2.3 Identificar las funcionalidades básicas 6](#_heading=h.1t3h5sf)

[1.2.4 Borrador product Backlog 7](#_heading=h.4d34og8)

[1.3 Creación de Equipos 7](#_heading=h.2s8eyo1)

[1.4 Capacitación 8](#_heading=h.17dp8vu)

[1.5 Definición de las Reglas del Equipo 8](#_heading=h.3rdcrjn)

[1.6 Planificación de Infraestructura 9](#_heading=h.lnxbz9)

[1.7 Gestión de Proyectos con enfoque Kanban 9](#_heading=h.1ksv4uv)

[1.7.1 Tableros Kanban 9](#_heading=h.44sinio)

[1.7.2 Gestión de proyectos 9](#_heading=h.2jxsxqh)

[1.7.3 Seguimiento del progreso 10](#_heading=h.z337ya)

[1.7.4 Flujo de trabajo. 10](#_heading=h.3j2qqm3)

[1.8 Gestión Efectiva del Product Backlog 10](#_heading=h.1y810tw)

[1.8.1 Refinar el product backlog 11](#_heading=h.4i7ojhp)

[1.8.2 Priorizar el product backlog 11](#_heading=h.2xcytpi)

[1.8.3 Historias de usuario según INVEST 11](#_heading=h.1ci93xb)

[1.09 Planificación y Ejecución del Primer Sprint 11](#_heading=h.3whwml4)

[1.09.1 Historias de usuario del primer sprint 11](#_heading=h.2bn6wsx)

[1.10 Gestión de Reuniones en Scrum 22](#_heading=h.qsh70q)

[1.10.1 Preparación para las Reuniones de Scrum: 22](#_heading=h.3as4poj)

[Para seleccionar las historias de usuario y definir las tareas. 22](#_heading=h.1pxezwc)

[1.10.2 Preparación de las reuniones diarias de Scrum: Para el seguimiento del progreso y la identificación de obstáculos. 23](#_heading=h.49x2ik5)

[1.10.3 Preparación de la reunión de planificación del sprint: Para demostrar las funcionalidades completadas. 23](#_heading=h.2p2csry)

[1.10.4 Preparación de la reunión de revisión del sprint 23](#_heading=h.147n2zr)

[1.10.5 Preparación de la reunión de la Retrospectiva del Sprint: 23](#_heading=h.3o7alnk)

[Para revisar el sprint y planificar mejoras. 23](#_heading=h.23ckvvd)

[**Capítulo II: Sprint 1 23**](#_heading=h.32hioqz)

[2.1 Acta de Constitución 23](#_heading=h.1hmsyys)

[2.2 Backlog del Producto 23](#_heading=h.41mghml)

[2.3 Planificación de Lanzamiento 24](#_heading=h.2grqrue)

[2.4 Registro de Interesados 24](#_heading=h.vx1227)

[**Capítulo III: Sprint 2 24**](#_heading=h.3fwokq0)

[3.1 Crear repositorio para el proyecto 24](#_heading=h.1v1yuxt)

[3.2 Gestión de Documentación de Proyectos en un Repositorio 28](#_heading=h.4f1mdlm)

[3.2.1 Proyecto completo en el repositorio 28](#_heading=h.2u6wntf)

[3.2.1 Documentación del proyecto en DOC 28](#_heading=h.19c6y18)

[3.2.2 Los nombres de los documentos con el patrón recomendado 28](#_heading=h.3tbugp1)

# 

# **Capítulo I: Sprint 0**

## **1.1 Configuración del Entorno (Docker)**

**Proyecto:** Desarrollo de un sistema web para estudiantes academica de odontologia, con un servidor web local para pruebas y luego desplegar en un servidor en la nube (INMOTION HOSTING).

**Paso 0: Configuración de Docker**

Descargar e instalar Docker

<https://www.docker.com/products/docker-desktop/>

Descargar e instalar WSL2

<https://learn.microsoft.com/es-es/windows/wsl/install>

**Paso 1: Entorno de desarrollo integrado (IDE)**

Descargar e instalar un IDE como Visual Studio Code (VS Code), que es ampliamente utilizado y ofrece muchas extensiones útiles para el desarrollo web.

<https://code.visualstudio.com/docs/?dv=win>

Descargar e instalar el motor de base de datos MySql

<https://sourceforge.net/projects/xampp/files/XAMPP%20Windows/8.2.4/xampp-windows-x64-8.2.4-0-VS16-installer.exe/download>

**Paso 2: Configuración GitHub**

Instalar Git en tu sistema para el control de versiones.

<https://github.com/git-for-windows/git/releases/download/v2.42.0.windows.2/Git-2.42.0.2-64-bit.exe>

Configurar tu nombre de usuario y dirección de correo electrónico en Git.

git config --global user.name "Tu Nombre"

git config --global user.email "tu@email.com"

**Paso 3: Instalar Node.js y npm**

Descargar e instalar Node.js, que incluye npm (Node Package Manager), para gestionar paquetes y dependencias de JavaScript.

<https://code.visualstudio.com/docs/?dv=win>

**Paso 4: Crear una carpeta de proyecto**

Crea una carpeta en tu sistema dónde alojar tu proyecto web.

**Paso 5: Inicializar un repositorio Git o clona el repositorio**

Inicia un repositorio Git en la carpeta del proyecto.

git init

git clone “repositorio” “nombre que quieres darle”

Repositorio back:

Repositorio front:

<https://github.com/proyecto-clinica-odontologica/muelitas-front.git>

**Paso 6: Abrir el proyecto en tu IDE**

Abre la carpeta del proyecto en tu IDE (VS Code).

**Paso 7: Instala los framework tanto para frontend y backend**

Segun corresponda al rol instala el framework requerido frontend:

npm install -g @angular/cli

Segun corresponda al rol instala el framework requerido backend:

npm install -g @nestjs/cli

**Paso 8: Crear el proyecto backend y frontend**

Crea el proyecto mediante los comandos frontend:

ng new “nombre del proyecto”

backend:

nest new “nombre del proyecto”

**Paso 9: Ingresar a la carpeta del proyecto backend y frontend**

Crea el proyecto mediante los comandos frontend:

cd ./muelitas-front

backend:

cd ./muelitas-back

**Paso 10: Instalar dependencias y dependencias de desarrollo en el proyecto**

Instala mediante los comandos frontend:

npm i instalar todas las dependencias

npm i “dependencia” instalar nueva dependencia

npm i “dependencia de desarrollo” –dev instalar dependencia de desarrollo

backend:

yarn i “dependencia”

yarn i “dependencia de desarrollo” –dev

**Paso 11: Crear módulos, componentes, interfaces, servicios, etc**

Crea mediante los comandos frontend:

ng g “tipo” “ruta/nombre”

ejemplo:

ng g m auth --route --routing --module —> crear módulo

ng g interface models/usuario —> crear interfaz o modelo

ng g service services/usuario --skip-tests —> crear servicio

ng g c auth --skip-tests —> crear componente

backend:

nest g m <path/nombre> —> Crea un módulo

nest g co <path/nombre> —> Crea un controlador

nest g service <path/nombre> —> Crea un servicio

nest g gu <path/nombre> —> Crea un guard

nest g resource <nombre> —> Crea un recurs completo CRUD

**Paso 12: Inicializa el proyecto**

Inicializa el proyecto mediante el comando

frontend:

ng serve -o

backend:

yarn start:dev

**Paso 13: Actualiza el repositorio**

Antes de empezar ejecuta el siguiente comando

git pull origin “rama asignada”

**Paso 14: Sube tus cambios**

git add . guardar todo

git commit -m “comentario breve de lo que hiciste” comentar lo hecho

git checkout -b “nueva rama creada” crear una rama

git push origin “rama asignada” subir a la rama

**Paso 15: Desarrolla y sigue probando**

Inicializa el proyecto mediante el comando

frontend:

ng serve -o

backend:

yarn start:dev

**Paso 16: Unir ramas (solo encargados)**

git switch “nombre rama destino”

git checkout “nombre rama destino”

git merge “nombre rama origen”

ojo: si hay conflictos de merge resolverlo y publicar en otra rama “de preferencia dev”

**Paso 17: Crear el proyecto para subirlo a un servidor en la nube**

Comando para crear el proyecto para subirlo en un hosting o vps

frontend:

ng build –prod

backend:

yarn build

frontend: angular - bootstrap

backend: typescript - Nest js

orm: TypeOrm

base de datos: mysql

## **1.2 Producto Inicial**

### **1.2.1 Definición del Producto Inicial**

La definición del Producto Inicial es un paso fundamental en el desarrollo ágil de productos, ya que establece las bases para la planificación y ejecución del proyecto. Permite al equipo de desarrollo y al Product Owner tener una comprensión compartida de lo que se espera lograr y cuáles son las funcionalidades clave que deben entregarse en las primeras etapas del desarrollo.

**Proyecto:** Desarrollo de un sistema web para estudiantes academica de odontologia, con un servidor web local para pruebas y luego desplegar en un servidor en la nube (INMOTION HOSTING).

**Sistema web odontológico “Muelitas felices”**

Mudar el actual sistema que se tiene y mejorarlo lo más que se pueda para tener un seguimiento de los pacientes por las sesiones que pase en los consultorios y ver el progreso de mejora mediante el odontograma.

### **1.2.2 Visión inicial del producto**

**Visión del Proyecto**

Desarrollar un sistema que, a través de la vanguardia en innovación tecnológica, garantice la seguridad de los datos, optimice la eficiencia operativa y promueva la formación avanzada de estudiantes en odontología. Este sistema busca proporcionar una atención de primera categoría, mejorar la calidad de vida de los pacientes y preparar a los profesionales del futuro para los desafíos de la odontología.

**Visión de la Carrera de Odontología**

Servir y solucionar los problemas de salud de su entorno, al romper esquemas y proponer nuevas alternativas en los diagnósticos y tratamientos de la odontología, mostrando el liderazgo profesional.

### **1.2.3 Identificar las funcionalidades básicas**

1. Gestión de Historias Clínicas: Permite registrar información detallada sobre los pacientes, incluyendo datos personales, antecedentes médicos y odontológicos.
2. Odontograma Interactivo: Facilita la creación y visualización de odontogramas que ayudan a planificar y llevar a cabo procedimientos odontológicos de manera efectiva.
3. Diagnóstico y Tratamiento: Permite registrar diagnósticos, tratamientos y seguimientos de los pacientes, garantizando un registro completo de la atención.
4. Generación de Informes: Puede generar informes detallados que respaldan la toma de decisiones clínicas informadas.
5. Gráficos de Evolución: Ayuda a seguir el progreso de la salud bucal del paciente a lo largo del tiempo, mediante gráficos y registros visuales.
6. Facilidad de Acceso y Búsqueda: Permite acceder rápidamente a la información de los pacientes y buscar registros específicos.
7. Cumplimiento Normativo: Cumple con los estándares y regulaciones de privacidad de datos para garantizar la confidencialidad de la información del paciente.

### **1.2.4 Borrador product Backlog**

[Sprint 0 - Oficial Presentacion hoy](https://docs.google.com/document/d/15GVwpcgiFX5d2BDhj4B2QlWIP4cMfhy8SM0R6-_Kd5I/edit#heading=h.x74rjche73gu)

## **1.3 Creación de Equipos**

| **INTEGRANTES** | **ROLES** |
| --- | --- |
| Rodriguez Mallqui Maycol Rolando | **Scrum master I** |
| Huaman Verastein Anibal Jhamil | **Scrum master II** |
| Cusicuna Mucha Grozny | **Scrum master III** |
| Gutarra Romo Flor de Maria | **Scrum master IV** |
| Chavez Perez Hector | **Scrum master V** |
| Suasnabar Gaspar Alexis | **Scrum master VI** |
| Suazo Vilca Jean Carlos | **Equipo de desarrollo** |
| Aguilar Condor Cristopher | **Equipo de desarrollo** |
| Santa María Astuhuamán Carla Ariana | **Equipo de desarrollo** |
| Lavado Ayala Carlos | **Equipo de desarrollo** |
| Soto Montes Jesús | **Equipo de desarrollo** |
| Rodríguez Santana Renato | **Equipo de desarrollo** |
| Coronado Martinez Mathyas | **Equipo de desarrollo** |
| Muñico Tadeo Layoned | **Equipo de desarrollo** |
| Cangalaya Antezana Jheycit | **Equipo de desarrollo** |
| Flores Lapa Yeric Luis | **Equipo de desarrollo** |
| Torres Ricse Jean | **Equipo de desarrollo** |
| Oré Tarazona Erick Anton | **Equipo de desarrollo** |
| Huaman Sanchez Juan Anthony | **Equipo de desarrollo** |
| Dolorier Díaz Dennys | **Equipo de desarrollo** |
| Hoces Ricse Alfredo | **Equipo de desarrollo** |

## **1.4 Capacitación**

La capacitación es un componente esencial en la gestión de proyectos, y en este caso, la familiaridad de la gran mayoría de estudiantes con Scrum representa una ventaja significativa. Scrum es una metodología ágil ampliamente reconocida y utilizada en la industria de la gestión de proyectos, lo que facilita su implementación en este proyecto mediano. Al contar con un equipo que ya posee conocimientos sobre Scrum, se puede aprovechar este recurso para garantizar una mejor planificación, ejecución y control del proyecto.

## **1.5 Definición de las Reglas del Equipo**

## Estas reglas del equipo proporcionan un marco claro para la colaboración y la comunicación efectiva, lo que contribuye a un ambiente de trabajo productivo y colaborativo en el equipo de desarrollo de software. Es importante que todos los miembros del equipo estén al tanto de estas reglas y se adhieran a ellas para lograr un funcionamiento armonioso y eficiente.

**Reglas de Generales**

* Los encargados son los responsables de entregar el trabajo solicitado a su grupo en el tiempo solicitado y de manera eficiente si no se les bajará el porcentaje de participación.
* Los avances de los integrantes según rol será asignada por porcentaje por el encargado del rol.
* Los encargados darán las actividades a su grupo.
* Todos los miembros del equipo deben fomentar una comunicación abierta y transparente. Esto incluye compartir información relevante, problemas, avances y obstáculos de manera regular y sin ocultar nada que pueda afectar al proyecto.
* Establecer reuniones periódicas, como reuniones de equipo, reuniones de seguimiento o reuniones de retroalimentación, para asegurarse de que todos estén alineados en los objetivos del proyecto y para abordar cualquier preocupación o problema de manera oportuna.
* Entrar a los módulos según su rol (administrador, docente o estudiante).
* Comentar cualquier funcionalidad para entender de qué se trata.
* Cuando surjan desacuerdos o conflictos, el equipo debe abordarlos de manera constructiva y profesional.
* Todos los miembros del equipo deben seguir los estándares y buenas prácticas de desarrollo de software establecidos por la organización.
* Los miembros del equipo deben asegurarse de documentar adecuadamente su trabajo, incluyendo el código, los diseños y los procesos.
* Antes de entregar cualquier trabajo o funcionalidad, se deben realizar pruebas exhaustivas para asegurarse de que el software funcione correctamente y cumpla con los requisitos.

**Reglas de Documentación**

* El equipo debe establecer estándares claros para la documentación, incluyendo formatos, estructuras y herramientas a utilizar. Esto garantiza que la documentación sea coherente en todo el proyecto.
* Antes de que comience el desarrollo de una nueva funcionalidad, se debe crear una documentación detallada de las definiciones de usuario o historias de usuario. Esto incluye los criterios de aceptación y los requisitos funcionales.
* Todas las reuniones de Scrum, como las reuniones diarias de stand-up, las reuniones de planificación de sprint y las retrospectivas, deben estar documentadas adecuadamente. Esto ayuda a rastrear el progreso y las decisiones tomadas durante el sprint.
* Cada tarea o funcionalidad desarrollada debe incluir documentación detallada de las pruebas realizadas, tanto las pruebas de unidad como las pruebas de integración. Esto asegura que el software se haya probado adecuadamente antes de su implementación.
* La documentación debe mantenerse actualizada a lo largo del proyecto. Cualquier cambio en los requisitos, la arquitectura o el diseño debe reflejarse de manera oportuna en la documentación correspondiente.
* Programar revisiones regulares de la documentación para garantizar su calidad y precisión. Estas revisiones pueden ser parte de las tareas de revisión de sprint o llevarse a cabo de manera separada.
* Se debe de asegurarse de que todos los miembros del equipo tengan acceso a la documentación relevante y establecer una política clara de gestión de permisos para garantizar que la información esté disponible solo para aquellos que la necesiten.

**Reglas de Backend**

* Ser claros con las ubicaciones donde se realizarán las modificaciones.
* Implementar medidas de seguridad adecuadas para garantizar el acceso seguro a las funcionalidades.
* Generar una documentación clara, que incluya ejemplos de solicitudes y respuestas, para facilitar la integración con el frontend
* Escribir pruebas unitarias para asegurar la calidad del código. Automatizar las pruebas siempre que sea posible.
* Manejar los errores de manera adecuada y proporcionar respuestas de error claras y útiles
* Colaborar con el equipo de Base de Datos para asegurarse de que el diseño de la base de datos sea coherente con las necesidades del backend.

**Reglas de Frontend: Diseño**

* Los colores serán de acuerdo a la empresa que colabore con nosotros.
* Priorizar lo estético al momento de diseñar (no mucho texto).
* Al realizar listas solo colocar unas cinco de ejemplo debido a que al unir con lo programado solo se necesita uno para que el bucle corra en eso.
* Tratar de poner barras de estado cuando ya estemos en los procesos.
* Se tiene que poder tener responsividad excepto cuando tengamos el odontograma.
* El dashboard si o si tendrá un navegador, sidebar y su contenido, por supuesto.
* Para el odontograma diseñar solo para desktop.

**Reglas de Frontend: Programación**

* Comenzar con el crud para los usuarios(administradores, odontólogos, recepcionista, etc).
* Las tareas asignadas serán según el rol (administrador, docente o estudiante) que elijas para no desviarte.
* Basaremos los modelos y servicios de acorde al backend.
* Crear toast “mensaje de estado” para probar tu avance o con console.log(“mensaje”).
* Esperaremos a los de diseño para luego unir lo programado con lo visual.
* Para los procesos de la empresa nos basaremos en bpmn que se diseñe.

**Reglas de Base de Datos**

* Programar revisiones del progreso del diseño de la base de datos.
* Utilizar la misma versión del software para el desarrollo de la base de datos.
* Crear usuarios para todos (según rol elegido administrador, doctor, recepcionista, etc).

## **1.6 Planificación de Infraestructura**

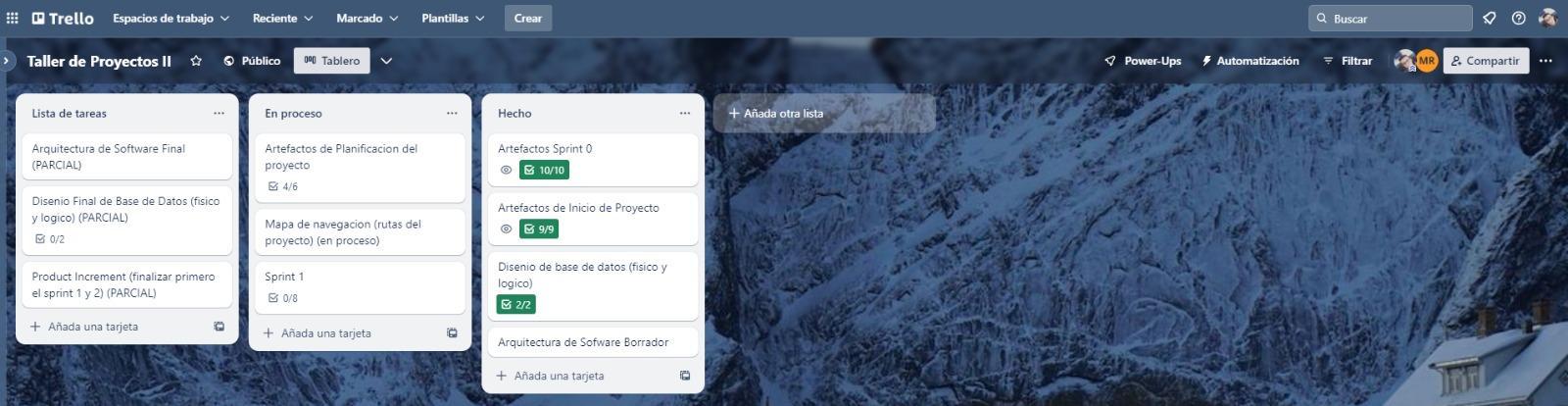
**Entornos de Desarrollo y Pruebas:**

* ***Identificación de Entornos:***
  + **frontend:** Angular - TypeScript - Bootstrap
  + **backend:** Nest js - TypeScript
  + **orm:** TypeOrm
  + **base de datos:** Mysql
* ***Herramientas:***
  + IDE: Visual Studio Code
  + Modelado: Bizagi y Erwin Data Modeler
  + Diseño: Adobe XD, Figma, Photoshop, Adobe Illustrator.
  + Presentación: Canva
  + Organización: Jira Software (DE PAGA)

## 

## **1.7 Gestión de Proyectos con enfoque Kanban**

### **1.7.1 Tableros Kanban**



### **1.7.2 Gestión de proyectos**

La gestión de proyectos en un software odontológico es esencial para coordinar y ejecutar con éxito tratamientos complejos o procedimientos. Los proyectos pueden incluir la planificación de cirugías, la coordinación entre diferentes especialidades dentales y la gestión de equipos de atención al paciente. Al utilizar tableros Kanban, los equipos pueden organizar y dar seguimiento a estos proyectos de manera efectiva. Cada proyecto se representa como una tarjeta en el tablero, y las columnas reflejan las etapas del proyecto, como planificación, ejecución y finalización. Esto permite una gestión proactiva y la identificación temprana de problemas potenciales.

### **1.7.3 Seguimiento del progreso**

Solo los jefes encargados, en las reuniones con el cada equipo pueden modificar el progreso del proyecto en el tablero Kanban.

### **1.7.4 Flujo de trabajo.**

Creación de Citas (Por Hacer): El proceso comienza cuando un paciente agenda una cita en la clínica dental. La recepcionista ingresa los detalles de la cita en el software odontológico y crea una tarjeta en la columna "Por Hacer" del tablero Kanban. Esta tarjeta incluye la información del paciente y la cita programada.

Confirmación de Cita (En Espera de Confirmación): La tarjeta se mueve a la columna "En Espera de Confirmación". El sistema envía automáticamente un recordatorio al paciente para confirmar la cita por correo electrónico o mensaje de texto. El personal de la clínica dental también puede comunicarse directamente con el paciente si es necesario.

Cita Confirmada (En Tratamiento): Una vez que el paciente confirma la cita, la tarjeta se traslada a la columna "En Tratamiento". El odontólogo y el asistente dental pueden acceder a la tarjeta para prepararse para la cita y registrar los procedimientos planeados.

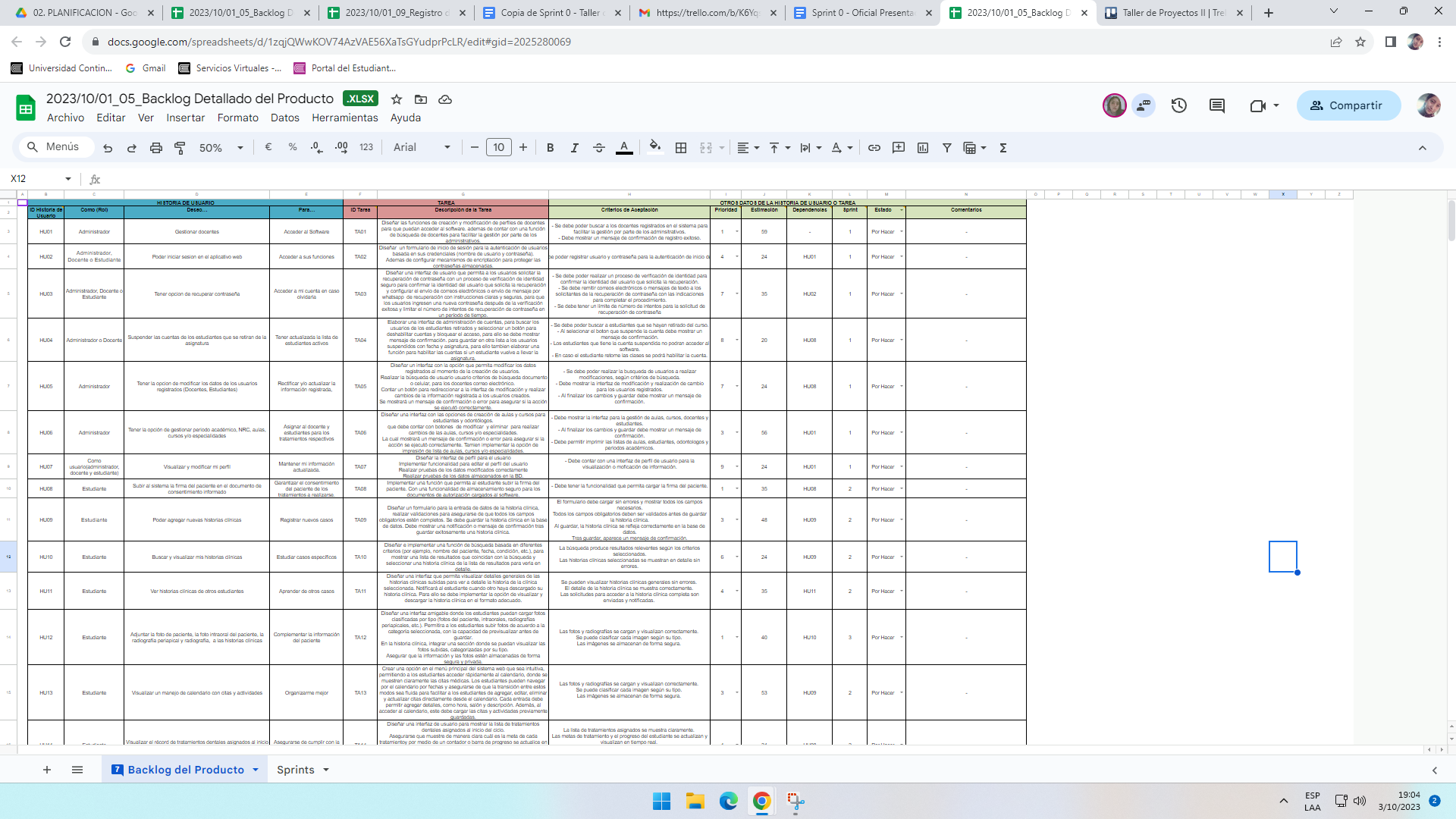
Durante el Tratamiento (En Tratamiento): Mientras se lleva a cabo el tratamiento, la tarjeta permanece en la columna "En Tratamiento". El personal dental actualiza la tarjeta con detalles sobre los procedimientos realizados y cualquier observación relevante.

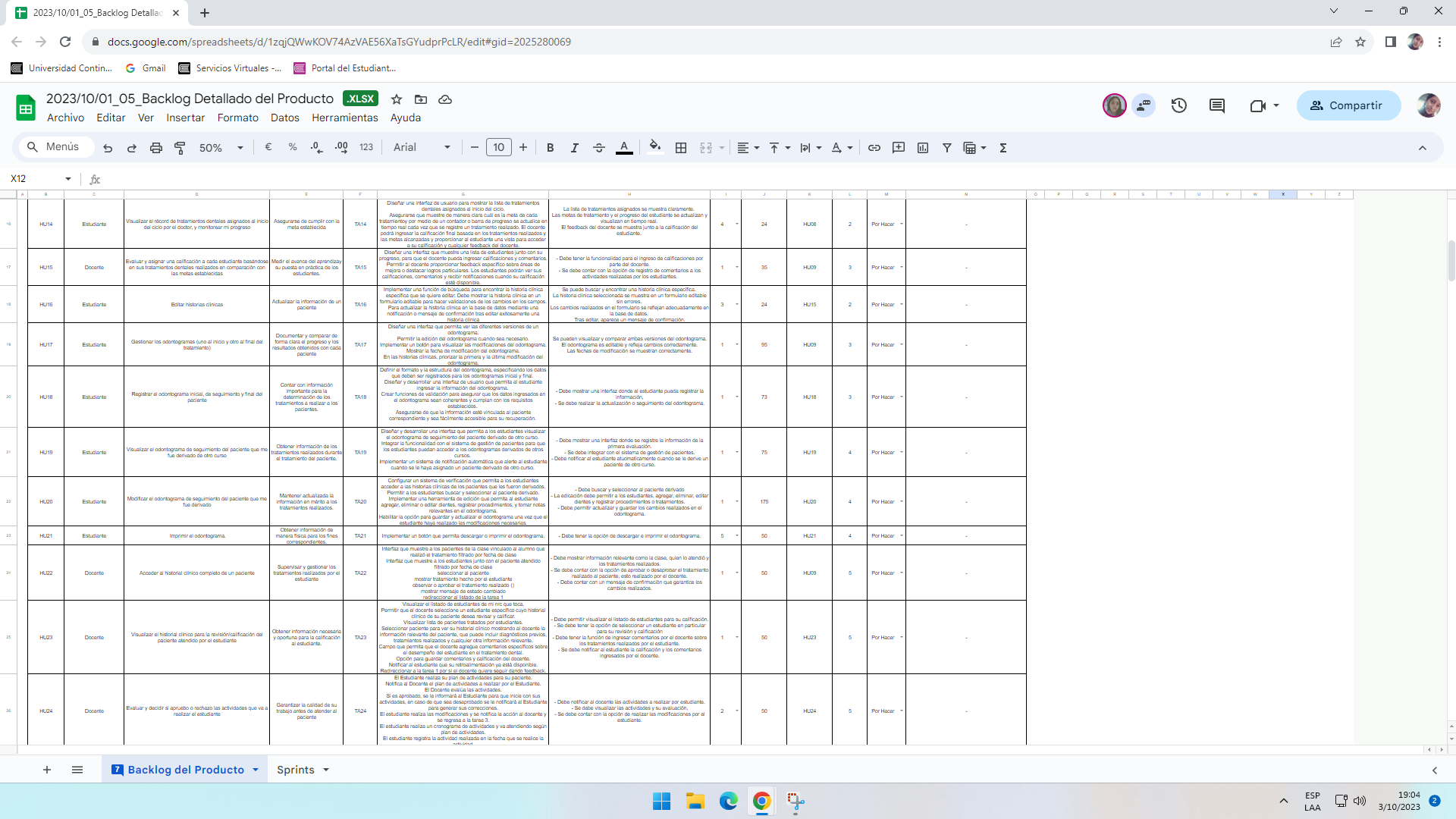
Finalización del Tratamiento (Completado): Una vez que se completa el tratamiento, la tarjeta se mueve a la columna "Completado". En esta etapa, se registran los procedimientos realizados, se actualiza la información del paciente y se prepara para la facturación.

Cierre (Completado):. Una vez que se completa la facturación y se recibe el pago, la tarjeta se archiva o se elimina del tablero Kanban.

## **1.8 Gestión Efectiva del Product Backlog**

[2023/10/01\_05\_Backlog Detallado del Producto.xlsx](https://docs.google.com/spreadsheets/d/1zqjQWwKOV74AzVAE56XaTsGYudprPcLR/edit#gid=2025280069)





### **1.8.1 Refinar el product backlog**

### **1.8.2 Priorizar el product backlog**

### **1.8.3 Historias de usuario según INVEST**

**HU001 - Gestionar docentes:**

* **I**ndependiente: Esta historia puede ser implementarse sin depender de otras historias.
* **N**egociable: Los detalles específicos de cómo gestionar docentes pueden ser discutidos y acordados con el equipo.
* **V**aluable: Gestionar docentes es valioso para permitirles acceder al software y utilizarlo de manera efectiva.
* **E**stimable: Es posible estimar el tiempo y recursos necesarios para esta tarea.
* **S**mall (Pequeña): La historia es lo suficientemente pequeña como para ser completada en una iteración.
* **T**estable: Se pueden definir criterios de aceptación claros para evaluar si la historia está completa.

**HU002 - Iniciar sesión:**

* **I**ndependiente: Puede implementarse de manera independiente.
* **N**egociable: Los detalles sobre cómo iniciar sesión pueden ser discutidos y acordados con el equipo.
* **V**aluable: El inicio de sesión es valioso ya que permite a administradores, docentes y estudiantes acceder al sistema.
* **E**stimable: Es posible estimar el esfuerzo requerido para implementar el inicio de sesión.
* **S**mall (Pequeña): La historia es lo suficientemente pequeña como para completarse en una sola iteración.
* **T**estable: Se pueden definir criterios de aceptación para evaluar si el inicio de sesión funciona correctamente.

**HU003 - Recuperación de contraseña:**

* **I**ndependiente: Puede implementarse sin depender de otras historias.
* **N**egociable: Los detalles de la recuperación de contraseña pueden ser discutidos y acordados con el equipo.
* **V**aluable: La recuperación de contraseña es valiosa para usuarios que olviden sus contraseñas.
* **E**stimable: Es posible estimar el esfuerzo requerido para implementar esta funcionalidad.
* **S**mall (Pequeña): La historia es lo suficientemente pequeña como para completarse en una iteración.
* **T**estable: Se pueden definir criterios de aceptación claros para evaluar si la recuperación de contraseña funciona correctamente.

**HU004 - Verificar desempeño de estudiantes:**

* **I**ndependiente: Puede implementarse de manera independiente.
* **N**egociable: Los detalles específicos de cómo verificar el desempeño de los estudiantes pueden ser discutidos y acordados con el equipo.
* **V**aluable: Verificar el desempeño de los estudiantes es valioso para los docentes.
* **E**stimable: Es posible estimar el tiempo y recursos necesarios para esta funcionalidad.
* **S**mall (Pequeña): La historia es lo suficientemente pequeña como para completarse en una iteración.
* **T**estable: Se pueden definir criterios de aceptación para evaluar si la funcionalidad funciona correctamente.

**HU005 - Autorizar derivaciones de pacientes:**

* **I**ndependiente: Puede ser implementada sin depender de otras historias.
* **N**egociable: Los detalles de cómo autorizar las derivaciones de pacientes pueden ser discutidos y acordados con el equipo.
* **V**aluable: Autorizar derivaciones de pacientes es valioso para la gestión de tratamientos en el sistema.
* **E**stimable: Es posible estimar el esfuerzo necesario para implementar esta funcionalidad.
* **S**mall (Pequeña): La historia es lo suficientemente pequeña como para completarse en una iteración.
* **T**estable: Se pueden definir criterios de aceptación claros para evaluar si la funcionalidad de autorización de derivaciones funciona correctamente.

## **1.09 Planificación y Ejecución del Primer Sprint**

[2023/10/01-08Backlog del Sprint.xlsx](https://docs.google.com/spreadsheets/d/17p75Y2BCijwFRAfmfYGLwMe0f4tZKyKf/edit#gid=691451340)

### **1.09.1 Historias de usuario del primer sprint**

| **Sprint** | **ID** | **Historia de Usuario** |
| --- | --- | --- |
| 1 | HU001 | Como administrador, quiero gestionar docentes para que puedan acceder al software. |
| 1 | HU002 | Como administrador, docente o estudiante, quiero poder iniciar sesión en el aplicativo web para acceder a sus funciones. |
| 1 | HU003 | Como administrador, docente o estudiante, deseo tener la opción de recuperar mi contraseña para acceder a mi cuenta en caso de olvidarla. |
| 1 | HU004 | Como docente, quiero poder verificar el desempeño de los estudiantes. |
| 1 | HU005 | Como docente quiero poder autorizar a los estudiantes de Clínica 1 de derivar a sus pacientes a Clínica 2 para continuar con su tratamiento. |
| 1 | HU006 | Como Administrador o docente, quiero poder suspender las cuentas de los estudiantes que se retiren de la asignatura, para tener actualizada la lista de estudiantes activos. |
| 1 | HU007 | Como administrador deseo la opción de modificar los datos de los usuarios registrados. |
| 1 | HU008 | Como administrador deseo tener la opción de gestionar periodo académico, NRC, aulas, cursos y/o especialidades para asignar al docente y estudiantes para los tratamientos respectivos. |
| 1 | HU009 | Como estudiante, quiero poder agregar nuevas historias clínicas para registrar nuevos casos. |
| 1 | HU010 | Como estudiantes, quiero editar historias clínicas para actualizar la información de un paciente. |
| 1 | HU011 | Como estudiante, quiero buscar y visualizar mis historias clínicas para estudiar casos específicos. |
| 1 | HU012 | Como estudiante, quiero gestionar dos odontogramas (uno al inicio y otro al final del tratamiento) para documentar y comparar de forma clara el progreso y los resultados obtenidos con cada paciente. |
| 1 | HU013 | Como estudiante, quiero ver historias clínicas de otros estudiantes para aprender de otros casos. |
| 1 | HU014 | Como estudiante, quiero adjuntar la foto de paciente, la foto intraoral del paciente, la radiografía periapical y radiografía, a las historias clínicas para complementar la información del paciente. |
| 1 | HU015 | Como estudiante, quiero visualizar un manejo de calendario con citas y actividades para organizarme mejor. |
| 1 | HU016 | Como estudiante, quiero visualizar el récord de tratamientos dentales asignados al inicio del ciclo por el doctor, y monitorear mi progreso para asegurarse de cumplir con la meta establecida. |
| 1 | HU017 | Como docente, quiero evaluar y asignar una calificación a cada estudiante basándose en sus tratamientos dentales realizados en comparación con las metas establecidas. |
| 1 | HU018 | Como docente, quiero acceder al historial clínico completo de un paciente para supervisar y gestionar los tratamientos realizados por el estudiante. |
| 1 | HU019 | Como docente, quiero visualizar el historial clínico para la revisión/calificación del paciente atendido por el estudiante. |
| 1 | HU020 | Como docente, quiero evaluar y decidir si apruebo o rechazo las actividades que va a realizar el estudiante para garantizar la calidad de su trabajo antes de atender al paciente. |
| 1 | HU021 | Como docente, quiero recibir notificaciones de historias clínicas donde se requiera de una firma. |
| 1 | HU022 | Como usuario(administrador, docente y estudiante), quiero visualizar y modificar mi perfil para mantener mi información actualizada. |
| 1 | HU023 | Como administrador, quiero ser el único quien agregue cuentas de docentes. |
| 1 | HU024 | Como administrador, quiero suspender (restringir el acceso al sistema) la cuenta de docentes no activos(retirado) |
| 1 | HU025 | Como usuario (docente, estudiante), quiero cambiar mi contraseña predeterminada para aumentar mi seguridad. |
| 1 | HU026 | Como estudiante quiero registrar el odontograma inicial, de seguimiento y final del paciente. |
| 1 | HU027 | Como estudiante quiero visualizar el odontograma de seguimiento del paciente que me fue derivado de otro curso. |
| 1 | HU028 | Como estudiante de semestre superior quiero modificar el odontograma de seguimiento del paciente que me fue derivado. |
| 1 | HU029 | Como estudiante quiero poder imprimir el odontograma |
| 1 | HU030 | Como estudiante quiero subir al sistema la firma del paciente en el documento de consentimiento informado. |
| 1 | HU031 | Como docente, quiero poder subir un registro de celdas con los estudiantes y que el sistema cree sus cuentas respectivamente. |

**1.09.2 Tareas de las historias de usuario del primer Sprint**

**HU001:** Como administrador, quiero gestionar docentes para que puedan acceder al software

* + **Tarea 1:** Diseñar las funciones de creación y modificación de perfiles de docentes.
  + **Tarea 2:** Implementar una función de búsqueda de docentes para facilitar la gestión por parte de los administrativos.
  + **Tarea 3:** Diseñar un proceso de confirmación para agregar nuevos docentes al sistema.

**HU002:** Como administrador, docente o estudiante, quiero poder iniciar sesión en el aplicativo web para acceder a sus funciones.

* + - **Tarea 1:** Diseñar un formulario de inicio de sesión.
    - **Tarea 2:** Implementar la autenticación de usuarios basada en sus credenciales (nombre de usuario y contraseña).
    - **Tarea 3:** Configurar mecanismos de encriptación para proteger las contraseñas almacenadas.

**HU003:** Como administrador, docente o estudiante, deseo tener la opción de recuperar mi contraseña para acceder a mi cuenta en caso olvidarla.

* + - **Tarea 1:** Diseñar una interfaz de usuario que permita a los usuarios solicitar la recuperación de contraseña.
    - **Tarea 2:** Desarrollar un proceso de verificación de identidad seguro para confirmar la identidad del usuario que solicita la recuperación.
    - **Tarea 3:** Configurar el envío de correos electrónicos o envío de mensaje por whatsapp de recuperación de contraseña con instrucciones claras y seguras.
    - **Tarea 4:** Crear una página para que los usuarios ingresen una nueva contraseña después de la verificación exitosa.
    - **Tarea 5:** Implementar medidas de seguridad adicionales, como limitar el número de intentos de recuperación de contraseña en un período de tiempo.

**HU004:** Como docente, quiero poder verificar el desempeño de los estudiantes

* + - **Tarea 1:**Mostrar los detalles de la historia clínica.
    - **Tarea 2:** Diseñar una interfaz de usuario para verificar.
    - **Tarea 3 :** Diseñar un botón para que el docente verifique los trabajos.

**HU005:** Como docente quiero poder autorizar a los estudiantes de Clínica 1 de derivar a sus pacientes a Clínica 2 para continuar con su tratamiento.

* **Tarea 1:** Elaborar una interfaz que permita autorizar la derivación de pacientes.
* **Tarea 2:** Completar formulario de registro para solicitar autorización
* **Tarea 3:** Mensaje de validación de formulario guardado.
* **Tarea 4:** Recibir notificación para validar que el paciente derivado va a ser atendido.
* **Tarea 5:** Notificar a los docentes de clínica 1 cuando los pacientes derivados hayan terminado su tratamiento en clínica 2.

**HU006:** Como Administrador o docente, quiero poder suspender las cuentas de los estudiantes que se retiren de la asignatura, para tener actualizada la lista de estudiantes activos.

* **Tarea 1:** Elaborar una interfaz de administración de cuentas de usuarios.
* **Tarea 2**: Buscar los usuarios de los estudiantes retirados.
* **Tarea 3:** Seleccionar un botón para deshabilitar cuentas y bloquear el acceso.
* **Tarea 4:** Mostrar mensaje de confirmación.
* **Tarea 5:** Guardar en una lista los usuarios suspendidos con fecha y asignatura
* **Tarea 6:** Elaborar una función para habilitar las cuentas si un estudiante vuelve a llevar la asignatura

**HU007:** Como administrador deseo la opción de modificar los datos de los usuarios registrados.

* **Tarea 1:** Diseñar un interfaz con la opción que permita modificar los datos registrados al momento de la creación de usuarios.
* **Tarea 2:** Realizar la búsqueda de usuario usuario criterios de búsqueda documento o celular, para los docentes correo electrónico.
* **Tarea 3:** Contar un botón para redireccionar a la interfaz de modificación y realizar cambios de la información registrada a los usuarios creados.
* **Tarea 4:** Se mostrará un mensaje de confirmación o error para asegurar si la acción se ejecutó correctamente.

**HU008:** Como administrador deseo tener la opción de gestionar periodo académico, NRC, aulas, cursos y/o especialidades para asignar al docente y estudiantes para los tratamientos respectivos.

* **Tarea 1:** Diseñar una interfaz con las opciones de creación de aulas y cursos para estudiantes y odontólogos.
* **Tarea 2:** Contar con botones de modificar y eliminar para realizar cambios de las aulas, cursos y/o especialidades.
* **Tarea 3:** Se mostrará un mensaje de confirmación o error para asegurar si la acción se ejecutó correctamente.
* **Tarea 4:** Implementar la opción de impresión de lista de aulas, cursos y/o especialidades.

**Registro de Historias Clínicas (como estudiantes):**

**Gestión de historias clínicas**

**HU009:** Como estudiante, quiero poder agregar nuevas historias clínicas para registrar nuevos casos.

* **Tarea 1:** Diseñar un formulario para la entrada de datos de la historia clínica.
* **Tarea 2:** Implementar validaciones para asegurarse de que todos los campos obligatorios estén completos.
* **Tarea 3** Registrar odontograma del paciente
* **Tarea 4:** Crear una base de datos o estructura de almacenamiento para las historias clínicas.
* **Tarea 5:** Implementar la lógica para guardar la historia clínica en la base de datos.
* **Tarea 6:** Diseñar e implementar una notificación o mensaje de confirmación tras guardar exitosamente una historia clínica.

**HU010:** Como estudiantes, quiero editar historias clínicas para actualizar la información de un paciente.

* + - **Tarea 1:** Implementar una función de búsqueda para encontrar la historia clínica específica que se quiere editar.
    - **Tarea 2:** Mostrar la historia clínica en un formulario editable.
    - **Tarea 3:** Implementar validaciones para cambios en los campos.
    - **Tarea 4:** Crear la lógica para actualizar la historia clínica en la base de datos.
    - **Tarea 5:** Diseñar e implementar una notificación o mensaje de confirmación tras editar exitosamente una historia clínica.

**HU011:** Como estudiante, quiero buscar y visualizar mis historias clínicas para estudiar casos específicos.

* + - **Tarea 1:** Diseñar e implementar una función de búsqueda basada en diferentes criterios (por ejemplo, nombre del paciente, fecha, condición, etc.).
    - **Tarea 2:** Mostrar una lista de resultados que coincidan con la búsqueda.
    - **Tarea 3:** Permitir seleccionar una historia clínica de la lista de resultados para verla en detalle.

**Documentación y visualización**

**HU012:** Como estudiante, quiero gestionar dos odontogramas (uno al inicio y otro al final del tratamiento) para documentar y comparar de forma clara el progreso y los resultados obtenidos con cada paciente.

* + - **Tarea 1:** Diseñar una interfaz que permita ver las diferentes versiones de un odontograma.
    - **Tarea 2:** Permitir la edición del odontograma cuando sea necesario.
    - **Tarea 3:** Implementar un botón para visualizar las modificaciones del odontograma.
    - **Tarea 4:** Mostrar la fecha de modificación del odontograma.
    - **Tarea 5:** En las historias clínicas, priorizar la primera y la última modificación del odontograma.

**HU013:** Como estudiante, quiero ver historias clínicas de otros estudiantes para aprender de otros casos.

* + - **Tarea 1:** Diseñar una interfaz que permita visualizar detalles generales de las historias clínicas subidas a la base de datos.
    - **Tarea 2:** Implementar la función de visualizar a detalle la historia de la clínica seleccionada.
    - **Tarea 3:** Implementar una función que notifique al estudiante cuando otro estudiante haya descargado su historia clínica.
    - **Tarea 4:** Implementar la opción de visualizar y descargar la historia clínica en el formato adecuado.

**HU014:** Como estudiante, quiero adjuntar la foto de paciente, la foto intraoral del paciente, la radiografía periapical y radiografía, a las historias clínicas para complementar la información del paciente.

* + - **Tarea 1:** Diseñar una interfaz amigable donde los estudiantes puedan cargar fotos clasificadas por tipo (fotos del paciente, intraorales, radiografías periapicales, etc.).
    - **Tarea 2:** Implementar una función que permita a los estudiantes subir fotos de acuerdo a la categoría seleccionada, con la capacidad de previsualizar antes de guardar.
    - **Tarea 3:** En la historia clínica, integrar una sección donde se puedan visualizar las fotos subidas, categorizadas por su tipo.
    - **Tarea 4:** Asegurar que la información y las fotos estén almacenadas de forma segura y privada.

**Gestión de actividades y evaluación**

**HU015:** Como estudiante, quiero visualizar un manejo de calendario con citas y actividades para organizarme mejor.

* **Tarea 1:** Crear una opción en el menú principal del sistema web que sea intuitiva, permitiendo a los estudiantes acceder rápidamente al calendario.
* **Tarea 2:** Diseñar una interfaz de visualización para el calendario donde se muestren claramente las citas médicas.
* **Tarea 3:** Permitir a los estudiantes navegar por el calendario por fechas y asegurarse de que la transición entre estos modos sea fluida.
* **Tarea 4:** Facilitar a los estudiantes la opción de agregar, editar, eliminar y actualizar citas directamente desde el calendario. Cada entrada debe permitir agregar detalles, como hora, salón y descripción.
* **Tarea 5:** Desarrollar la lógica necesaria para que, al realizar cambios en el calendario, estos se reflejen en la base de datos en tiempo real. Además, al acceder al calendario, este debe cargar las citas y actividades previamente guardadas.

**HU016:** Como estudiante, quiero visualizar el récord de tratamientos dentales asignados al inicio del ciclo por el doctor, y monitorear mi progreso para asegurarse de cumplir con la meta establecida.

* + - **Tarea 1:** Diseñar una interfaz de usuario para mostrar la lista de tratamientos dentales asignados al inicio del ciclo.
    - **Tarea 2:** Desarrollar una función que recupere desde la base de datos los tratamientos específicos asignados por el docente a cada estudiante.
    - **Tarea 3:** Asegurarse de que la interfaz muestre de manera clara cuál es la meta de cada tratamiento.
    - **Tarea 4:** Desarrollar un contador o barra de progreso que se actualice en tiempo real cada vez que se registre un tratamiento realizado.
    - **Tarea 5:** Permitir al docente ingresar la calificación final basada en los tratamientos realizados y las metas alcanzadas.
    - **Tarea 6:** Proporcionar al estudiante una vista para acceder a su calificación y cualquier feedback del docente.

**HU017:** Como docente, quiero evaluar y asignar una calificación a cada estudiante basándose en sus tratamientos dentales realizados en comparación con las metas establecidas.

* + - **Tarea 1:** Diseñar una interfaz que muestre una lista de estudiantes junto con su progreso.
    - **Tarea 2:** Implementar una función para que el docente pueda ingresar calificaciones y comentarios.
    - **Tarea 3:** Permitir al docente proporcionar feedback específico sobre áreas de mejora o destacar logros particulares.
    - **Tarea 4:** Crear una sección donde los estudiantes puedan ver sus calificaciones y comentarios.
    - **Tarea 5:** Implementar notificaciones para informar a los estudiantes cuando su calificación esté disponible.

**Revisión de Historias Clínicas (Docente):**

**HU018:** Como docente, quiero acceder al historial clínico completo de un paciente para supervisar y gestionar los tratamientos realizados por el estudiante.

* + - **Tarea 1:** Interfaz que muestre a los pacientes de la clase vinculado al alumno que realizó el tratamiento filtrado por fecha de clase
    - **Tarea 1:** Interfaz que muestre a los estudiantes junto con el paciente atendido filtrado por fecha de clase
    - **Tarea 2:** seleccionar al paciente
    - **Tarea 3:** mostrar tratamiento hecho por el estudiante
    - **Tarea 4:** observar o aprobar el tratamiento realizado
    - **Tarea 5:** mostrar mensaje de estado cambiado
    - **Tarea 6:** redireccionar al listado de la tarea 1

**HU019:** Como docente, quiero visualizar el historial clínico para la revisión/calificación del paciente atendido por el estudiante.(HU0XX)

* + - **Tarea 1:** Visualizar el listado de estudiantes de mi nrc que toca.
    - **Tarea 2:** Permitir que el docente seleccione un estudiante específico cuyo historial clínico de su paciente desea revisar y calificar.
    - **Tarea 3:** Visualizar lista de pacientes tratados por estudiantes.
    - **Tarea 4:** Seleccionar paciente para ver su historial clínico mostrando al docente la información relevante del paciente, que puede incluir diagnósticos previos, tratamientos realizados y cualquier otra información relevante.
    - **Tarea 5:** Campo que permita que el docente agregue comentarios específicos sobre el desempeño del estudiante en el tratamiento dental.
    - **Tarea 6:** Opción para guardar comentarios y calificación del docente.
    - **Tarea 7:** Notificar al estudiante que su retroalimentación ya está disponible.
    - **Tarea 8:** Redireccionar a la tarea 1 por si el docente quiere seguir dando feedback.

**HU020:** Como docente, quiero evaluar y decidir si apruebo o rechazo las actividades que va a realizar el estudiante para garantizar la calidad de su trabajo antes de atender al paciente.

* **Tarea 1:** El Estudiante realiza su plan de actividades para su paciente.
* **Tarea 2:** Notifica al Docente el plan de actividades a realizar por el Estudiante.
* **Tarea 3:** El Docente evalúa las actividades.
* **Tarea 4:** Si es aprobado, se le informará al Estudiante para que inicie con sus actividades, en caso de que sea desaprobado se le notificará al Estudiante para generar sus correcciones.
* **Tarea 5:** El estudiante realiza las modificaciones y se notifica la acción al docente y se regresa a la tarea 3.
* **Tarea 6:** El estudiante realiza un cronograma de actividades y va atendiendo según plan de actividades.
* **Tarea 7:** El estudiante registra la actividad realizada en la fecha que se realice la actividad.

**HU021:** Como docente, quiero recibir notificaciones de historias clínicas donde se requiera de una firma.

* + - **Tarea 1:** Estudiante termina de registrar datos que requieren firma.
    - **Tarea 2:** Asociar al registro función de notificaciones de requerimiento de firma por parte del docente.
    - **Tarea 3:** Docente recibe la notificación en su bandeja de notificaciones del sistema(debatible).
    - **Tarea 4:** Ver listado en la bandeja de notificaciones del sistema.
    - **Tarea 5:** Seleccionar actividad de la bandeja de notificaciones del sistema y redireccionar donde corresponda.
    - **Tarea 6:** Funcionalidad para firmar u observar la actividad realizada.
    - **Tarea 7:** Testear.

**Configuración de cuenta y Odontograma:**

**HU022:** Como usuario(administrador, docente y estudiante), quiero visualizar y modificar mi perfil para mantener mi información actualizada.

* + - **Tarea 1:** Diseñar la interfaz de perfil para el usuario.
    - **Tarea 2:** Implementar funcionalidad para editar el perfil del usuario.
    - **Tarea 3:** Realizar pruebas de los datos modificados correctamente.
    - **Tarea 4:** Realizar pruebas de los datos almacenados en la BD.

**HU023:** Como administrador, quiero ser el único quien agregue cuentas de docentes.

* + - **Tarea 1:** Diseñar la interfaz de administrador para la creación de cuentas.
    - **Tarea 2:** Implementar la funcionalidad de agregar cuenta para los docentes.
    - **Tarea 3:** Verificar los datos de la cuenta.

**HU024:** Como administrador, quiero suspender (restringir el acceso al sistema) la cuenta de docentes no activos (retirados).

* + - **Tarea 1:** Diseñar la interfaz que permita seleccionar las cuentas de docentes a suspender (cambio de estado).
    - **Tarea 2:** Implementar la funcionalidad de suspensión cuenta de docentes.
    - **Tarea 3:** Implementar una ventana emergente de confirmación de suspensión de cuenta.
    - **Tarea 4:** Permitir al administrador seleccionar un docente libre.
    - **Tarea 5:** Realizar pruebas del funcionamiento de la eliminación de cuenta.

**HU025:** Como usuario (docente, estudiante), quiero cambiar mi contraseña predeterminada para aumentar mi seguridad.

* + - **Tarea 1:** Diseñar la interfaz que permita el cambio de contraseña
    - **Tarea 2:** Implementar una ventana emergente de confirmación de cambio de contraseña.
    - **Tarea 3:** Implementar funcionalidad de cambio de contraseña.
    - **Tarea 4:** Validar el cambio de contraseña.

**HU026:** Como estudiante quiero registrar el odontograma inicial, de seguimiento y final del paciente.

* + - **Tarea 1:** Definir el formato y la estructura del odontograma, especificando los datos que deben ser registrados para los odontogramas inicial y final.
    - **Tarea 2:** Diseñar y desarrollar una interfaz de usuario que permita al estudiante ingresar la información del odontograma.
    - **Tarea 3:** Crear funciones de validación para asegurar que los datos ingresados en el odontograma sean coherentes y cumplan con los requisitos establecidos.
    - **Tarea 4:** Asegurarse de que la información esté vinculada al paciente correspondiente y sea fácilmente accesible para su recuperación.

**HU027:** Como estudiante quiero visualizar el odontograma de seguimiento del paciente que me fue derivado de otro curso.

* **Tarea 1:** Diseñar y desarrollar una interfaz que permita a los estudiantes visualizar el odontograma de seguimiento del paciente derivado de otro curso.
* **Tarea 2:** Integrar la funcionalidad con el sistema de gestión de pacientes para que los estudiantes puedan acceder a los odontogramas derivados de otros cursos.
* **Tarea 3:** Implementar un sistema de notificación automática que alerte al estudiante cuando se le haya asignado un paciente derivado de otro curso.

**HU028:**Como estudiante de semestre superior quiero modificar el odontograma de seguimiento del paciente que me fue derivado.

* **Tarea 1:** Configurar un sistema de verificación que permita a los estudiantes acceder a las historias clínicas de los pacientes que les fueron derivados.
* **Tarea 2:** Permitir a los estudiantes buscar y seleccionar al paciente derivado.
* **Tarea 3:** Implementar una herramienta de edición que permita al estudiante agregar, eliminar o editar dientes, registrar procedimientos, y tomar notas relevantes en el odontograma.
* **Tarea 6:** Habilitar la opción para guardar y actualizar el odontograma una vez que el estudiante haya realizado las modificaciones necesarias.

**HU029:** Como estudiante quiero poder imprimir el odontograma

* **Tarea 1:** Implementar un botón que permita descargar o imprimir el odontograma.

**HU030:** Como estudiante quiero subir al sistema la firma del paciente en el documento de consentimiento informado.

* **Tarea 1:** Implementa una función que permita al estudiante subir la firma del paciente.
* **Tarea 3:** Implementa una funcionalidad de almacenamiento seguro para los documentos de autorización cargados.

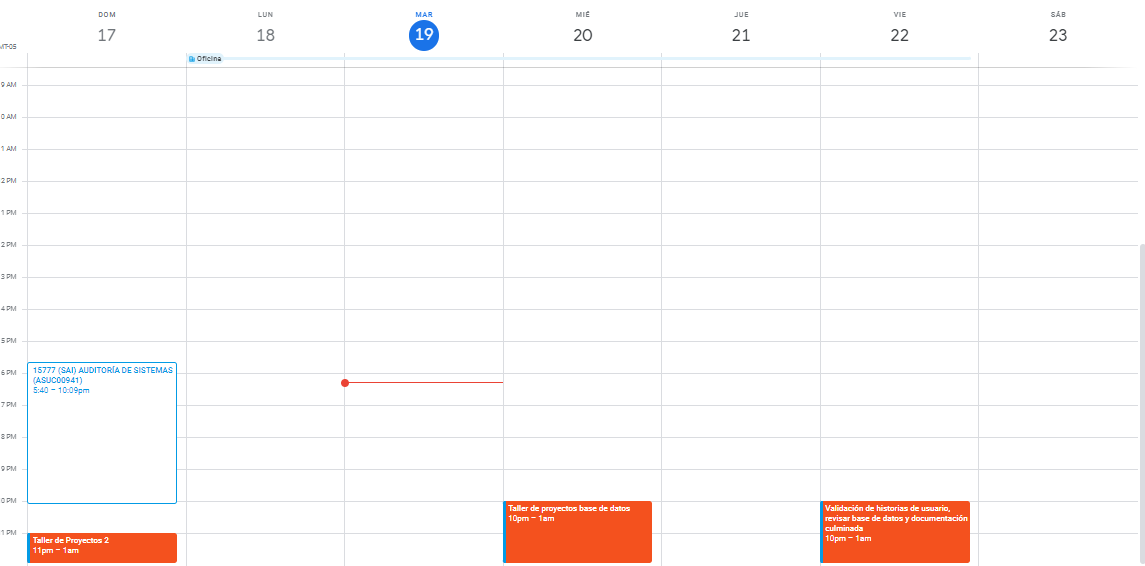
**HU031:**Como docente, quiero poder subir un registro de celdas con los estudiantes y que el sistema cree sus cuentas respectivamente.

* **Tarea 1:** Definir el formato del archivo de registro que el docente utilizará para cargar la información de los estudiantes.
* **Tarea 2:** Desarrolla una interfaz de usuario o apartado donde el docente pueda cargar el archivo de registro.
* **Tarea 3:** Implementa una función de validación que verifique si el archivo cumple con el formato especificado.
* **Tarea 4:** Utiliza la información del archivo de registro para crear cuentas de usuario para cada estudiante en el sistema. Esto incluye la generación automática de nombres de usuario y contraseñas seguras.
* **Tarea 5:** Notificar al estudiante mediante mensajes automáticos al correo institucional una vez creada la cuenta.

## **1.10 Gestión de Reuniones en Scrum**

### **1.10.1 Preparación para las Reuniones de Scrum:**

### Para seleccionar las historias de usuario y definir las tareas.

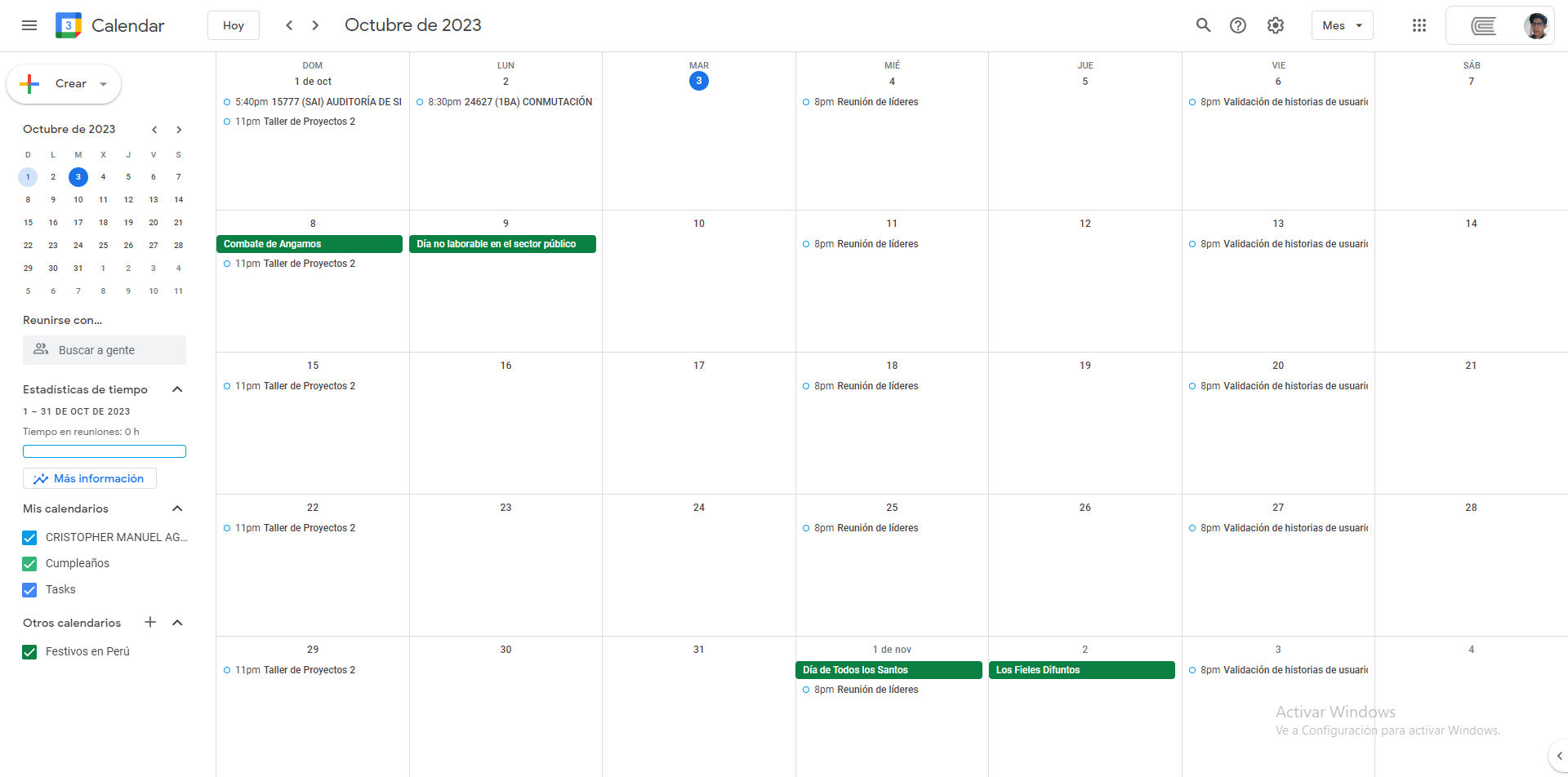
****

<https://drive.google.com/drive/folders/1eb6pKntJphmC5Ye3B20N6hqKFAsDakmQ>

### **1.10.2 Preparación de las reuniones diarias de Scrum:** Para el seguimiento del progreso y la identificación de obstáculos.

### **1.10.3 Preparación de la reunión de planificación del sprint:** Para demostrar las funcionalidades completadas.

### **1.10.4 Preparación de la reunión de revisión del sprint**



### **1.10.5 Preparación de la reunión de la Retrospectiva del Sprint:**

### Para revisar el sprint y planificar mejoras.

Este Sprint Backlog detallado incluye tareas específicas para cada historia de usuario seleccionada. El equipo se enfocará en completar estas tareas durante el sprint para asegurarse de que las funcionalidades estén listas para su revisión y prueba al final del sprint.

### 

# **Capítulo II: Sprint 1**

## **2.1 Acta de Constitución**

[2023/09/17\_02\_Acta de Constitución.docx](https://docs.google.com/document/d/1zCqXMNb_2fjxD1oFz3F4Ta5gcwZVUhWG/edit#heading=h.gjdgxs)

## **2.2 Backlog del Producto**

[2023/10/01\_05\_Backlog Detallado del Producto.xlsx](https://docs.google.com/spreadsheets/d/1zqjQWwKOV74AzVAE56XaTsGYudprPcLR/edit#gid=2025280069)

## **2.3 Planificación de Lanzamiento**

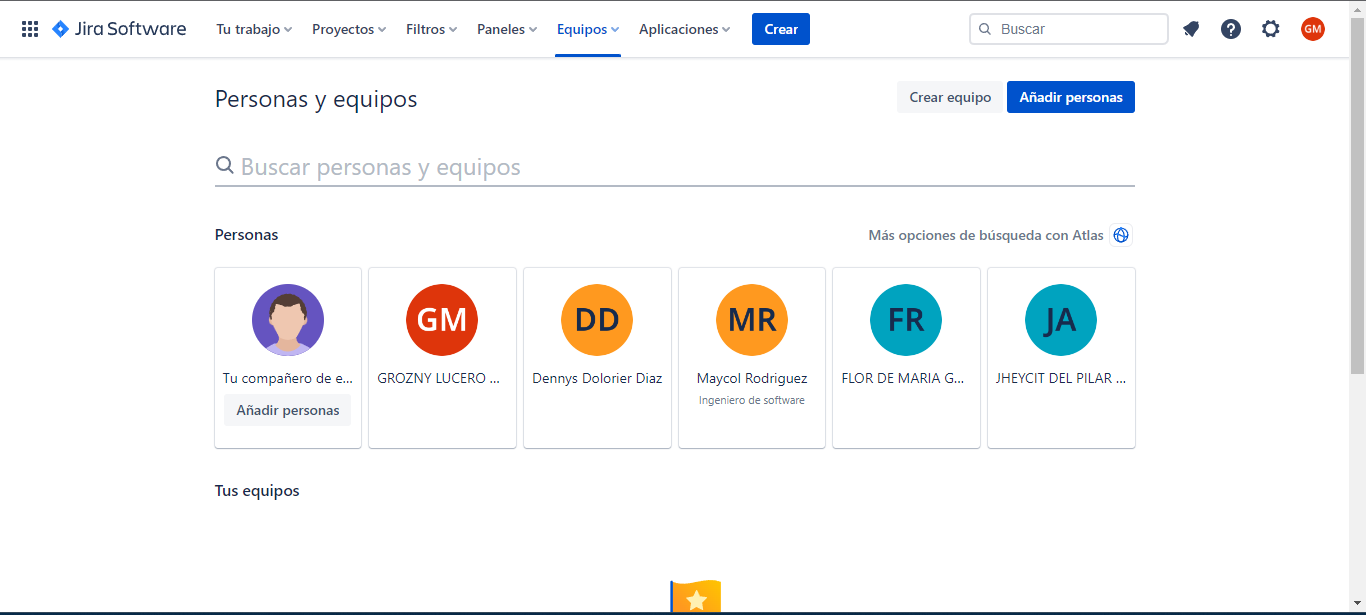
[2023/09/17\_06\_Planificación de Lanzamiento.pptx](https://docs.google.com/presentation/d/1ZVXTfd6l4YRZx8kn4DmNI4sy2dKY85nd/edit#slide=id.p1)

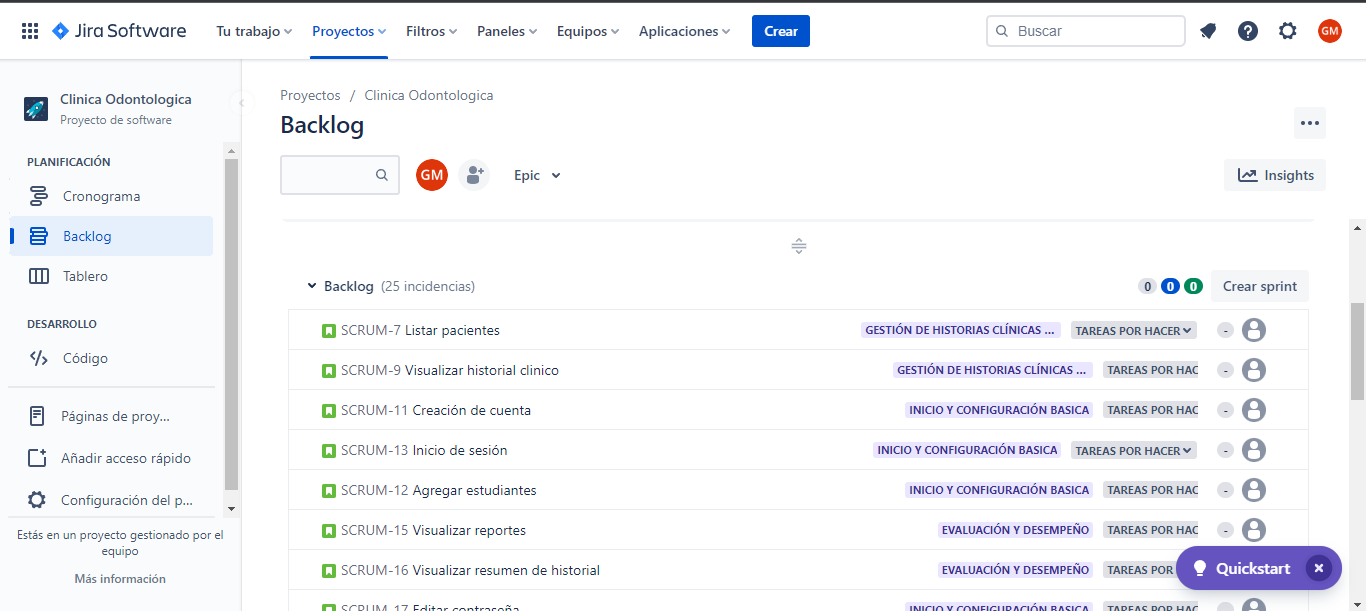
## **2.4 Registro de Interesados**

[2023/09/17\_01\_Registro de Interesados.xlsx](https://docs.google.com/spreadsheets/d/1by-seY2TeThtRDTtE00ieu3oDg_rgPSO/edit#gid=1472978789)

# **Capítulo III: Sprint 2**

## **3.1 Crear repositorio para el proyecto**





**Control de Versiones:**

* Sistema de Control de Versiones: Git se utilizará como sistema de control de versiones principal.
* Repositorios:
  + **Repositorio Central:** Un repositorio centralizado se aloja en una plataforma como GitHub.
  + **Repositorio Frontend:**

<https://github.com/proyecto-clinica-odontologica/muelitas-front-end.git>

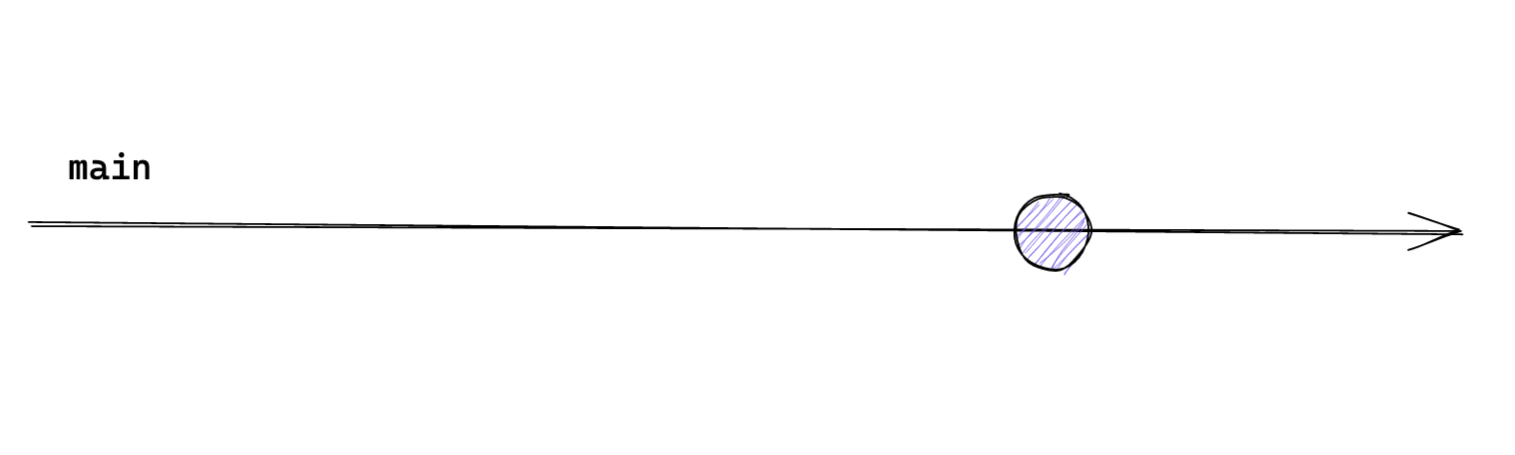
* + **Repositorio Backend:**

<https://github.com/proyecto-clinica-odontologica/muelitas-back-end.git>

* + **Ramas de Desarrollo:** Se seguirá una estrategia de ramificación (**ship/show/ask**) para gestionar el desarrollo de características y correcciones de errores.

**SHIP:**

Ship significa que vamos a hacer un cambio directamente a la rama principal. No esperamos revisiones de código, ni integración. Vamos directos a producción (aunque antes del despliegue sí se harán los tests o checks pertinentes para evitar errores)



Resumen:

Cambios directos en la rama master o main.

Rama master o main.

Pensado para:

* He añadido una nueva funcionalidad con un patrón establecido.
* He arreglado de forma sencilla un bug por un error.
* Actualizaciones de documentación.
* Mejora de código por feedback del equipo o la comunidad.
* Se añaden nuevos tests para evitar errores.

**SHOW:**

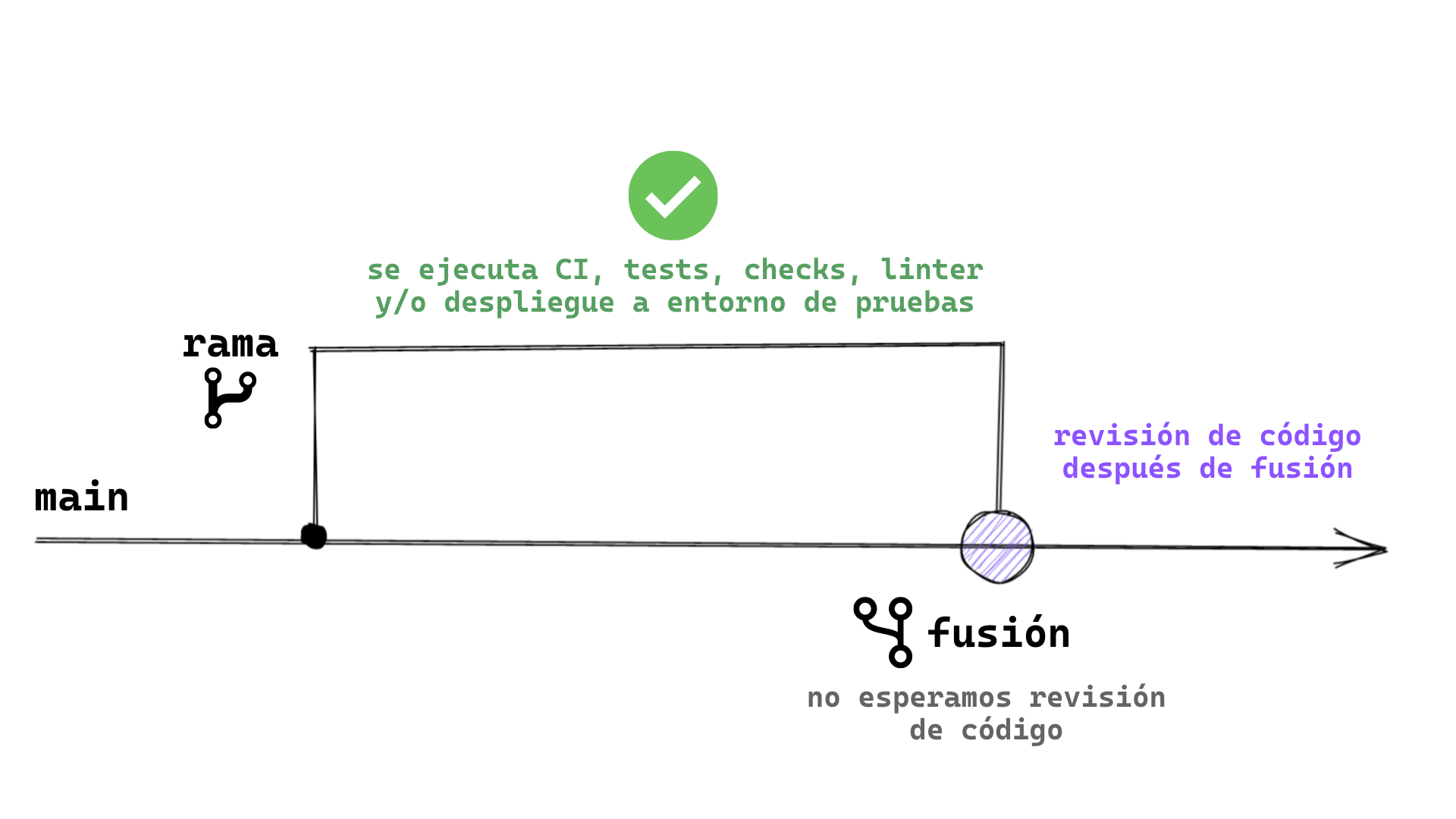
En este caso sí usamos Pull Request pero no esperamos revisiones manuales del código. Es decir, esperamos que los tests automatizados, pruebas de cobertura y validación de código sean exitosos pero no que otra persona revise el código.

Esto no quiere decir que no ocurran conversaciones sobre el código. La diferencia es que ocurrirán después de hacer la fusion de los cambios.

La idea es que el trabajo fluya hacia adelante, con el menor número de bloqueos, pero que sigan existiendo espacios dónde se pueda hablar y discutir sobre cómo mejorar las prácticas de desarrollo y el código que se crea.

El equipo es notificado que se creó una Pull Request, la revisan posteriormente y después se hacen las observaciones que sean necesarias. Lo interesante es que hace que esa PR queda fácilmente identificable y separada.

Las Pull Request muchas veces se usan como una forma de forzar la conversación dentro del equipo y compartir conocimiento. A veces, puede ser una buena idea. Pero nunca deben ser una sustitución a buenas dinámicas de trabajo en equipo y usar programación a pares o en grupo.



**Resumen:**

Cambios en otra rama para luego juntarlo realizando un pull con master y luego un merge al master.

Rama master con merge de rama “nombre desarrollador”.

Pensado para:

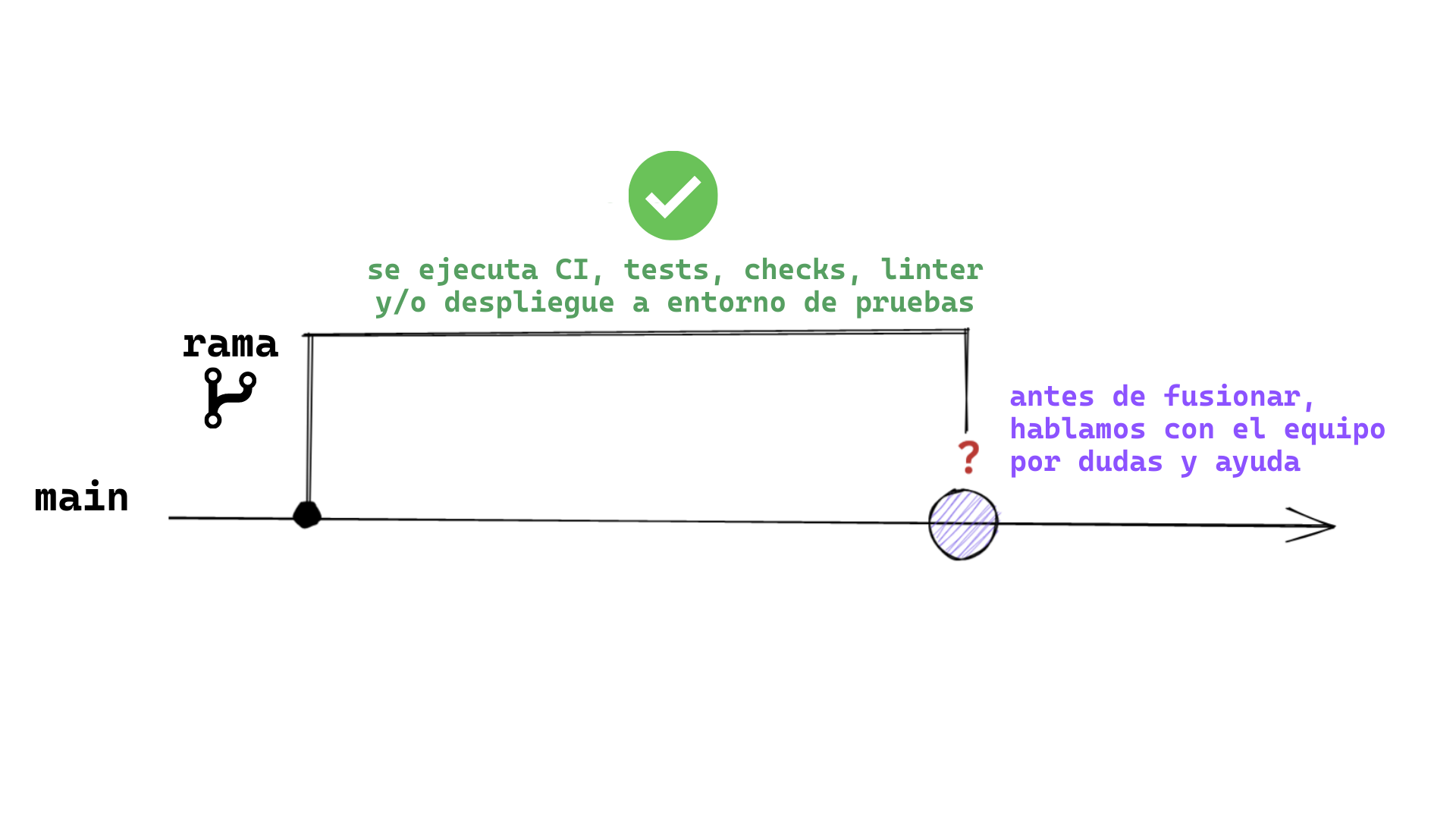
* Hacer arreglos necesarios para los bugs y dejar constancia para que se aprenda.
* Crear pequeñas mejoras de código o refactors.
* Añadir nuevas funcionalidades siguiendo estructuras ya acordadas.
* Funcionalidades con pruebas automáticas.

**ASK:**

Esta categoría es similar a Show pero aquí sí esperamos al feedback de nuestro equipo antes de fusionar la rama. Lo hacemos porque existe algo de incertidumbre: bien porque la solución es complicada, no sabemos implementarla, existen dudas…

La idea es que la rama dure el mínimo tiempo posible para no bloquear el trabajo de otros miembros del equipo.

Una cosa importante a destacar es que decidir usar la categoría de Ask no quiere decir que esperemos la aprobación de nuestros colegas. Estamos abriendo una vía de conversación y debate antes de fusionar la rama ya que existe algún bloqueo o duda, pero es posible que no estemos buscando una revisión general de los cambios (si es así, deberíamos indicarlo).



**Resumen:**

Cambios en otra rama que afecta buena parte del código y para realizar el merge se debe de avisar a los demás que se tocó código que a otros correspondía.

Rama master con merge de rama “nombre desarrollador”.

Pensado para:

* Cuando es un trabajo muy grande y se necesita ayuda.
* Hay dudas sobre cómo hacerlo funcionar o la calidad del código.
* Existe incertidumbre sobre lo que estamos haciendo.
* Estamos esperando a que algo ocurra para poder fusionar la rama.
* Políticas de Confirmación:
  + Convenciones de mensajes de confirmación significativos.
  + Uso de ramas de características para el desarrollo.

**Mantenimiento y Actualizaciones:**

* Política de Mantenimiento: Establecer un calendario de mantenimiento regular para parches y actualizaciones de software previo acuerdo con el product owner si es posible.

**Documentación:**

* Documentación Técnica: Mantener documentación técnica actualizada para la configuración de la infraestructura, procedimientos de implementación y flujos de trabajo.

## **3.2 Gestión de Documentación de Proyectos en un Repositorio**

### **3.2.1 Proyecto completo en el repositorio**

### **3.2.1 Documentación del proyecto en DOC**

### **3.2.2 Los nombres de los documentos con el patrón recomendado**