

**FACULTAD DE INGENIERÍA**

**ASIGNATURA:** TALLER DE PROYECTOS 2

**SEMESTRE:** 2024-10

**NOMBRE DEL PROFESOR:** GAMARRA MORENO DANIEL

**INTEGRANTES:**

1. GRANDE CONTRERAS, Mario
2. ROLDAN HINOJOSA, Jhon
3. VERA CHINCHAY, Javier
4. MALPARTIDA ZEVALLOS, Franco Luis
5. BALDEON LARA, Jhonatan Moisés
6. COLLACHAGUA VENTURA, Junior
7. SANABRIA TINOCO, Williams Anthony
8. JAUREGUI PONCE, Jhon Kennedy
9. POVIS URIBE, Jean Anthony

10. DIAZ LOAYZA, Diego Juan de Dios

**FECHA DE ENTREGA : 9/04/2024**

**HUANCAYO – PERU**

**2024**

[**Capítulo I: Sprint 0 3**](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.1fob9te)

[1.1 Configuración del Entorno 3](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.3znysh7)

[1.2 Producto Inicial 5](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.2et92p0)

[1.2.1 Definición del Producto Inicial 5](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.tyjcwt)

[1.2.2 Visión inicial del producto 6](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.3dy6vkm)

[1.2.3 Identificar las funcionalidades básicas 6](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.1t3h5sf)

[1.2.4 Borrador product Backlog 7](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.4d34og8)

[1.3 Creación de Equipos 7](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.2s8eyo1)

[1.4 Capacitación 8](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.17dp8vu)

[1.5 Definición de las Reglas del Equipo 8](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.3rdcrjn)

[1.6 Planificación de Infraestructura 9](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.lnxbz9)

[1.7 Gestión de Proyectos con enfoque Kanban 9](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.1ksv4uv)

[1.7.1 Tableros Kanban 9](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.44sinio)

[1.7.2 Gestión de proyectos 9](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.2jxsxqh)

[1.7.3 Seguimiento del progreso 10](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.z337ya)

[1.7.4 Flujo de trabajo. 10](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.3j2qqm3)

[1.8 Gestión Efectiva del Product Backlog 10](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.1y810tw)

[1.8.1 Refinar el product backlog 11](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.4i7ojhp)

[1.8.2 Priorizar el product backlog 11](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.2xcytpi)

[1.8.3 Historias de usuario según INVEST 11](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.1ci93xb)

[1.09 Planificación y Ejecución del Primer Sprint 11](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.3whwml4)

[1.09.1 Historias de usuario del primer sprint 11](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.2bn6wsx)

[1.10 Gestión de Reuniones en Scrum 22](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.qsh70q)

[1.10.1 Preparación para las Reuniones de Scrum: 22](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.3as4poj)

[Para seleccionar las historias de usuario y definir las tareas. 22](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.1pxezwc)

[1.10.2 Preparación de las reuniones diarias de Scrum: Para el seguimiento del progreso y la identificación de obstáculos. 23](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.49x2ik5)

[1.10.3 Preparación de la reunión de planificación del sprint: Para demostrar las funcionalidades completadas. 23](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.2p2csry)

[1.10.4 Preparación de la reunión de revisión del sprint 23](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.147n2zr)

[1.10.5 Preparación de la reunión de la Retrospectiva del Sprint: 23](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.3o7alnk)

[Para revisar el sprint y planificar mejoras. 23](https://docs.google.com/document/d/1kAxVQCNpOhQ1cMLHk9fqBly_IE9Fto0tqAYbRRMpTiU/edit#heading=h.23ckvvd)

# **Capítulo I: Sprint 0**

## 1.1 Configuración del Entorno

**Proyecto:** Desarrollo de un sistema web para estudiantes academica de odontologia, con un servidor web local para pruebas.

**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://dev.mysql.com/downloads/mysql/8.0.html>

**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:

<https://github.com/MarioJGC/proyecto-historia-clinica.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**

Leer archivo README.md adjuntado en el repositorio de github.

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

Crea mediante los comandos frontend:

| > ng new nombre-del-proyecto // Crea un nuevo proyecto Angular.  > ng generate module nombre-del-modulo // Genera un nuevo módulo Angular.  > ng generate component nombre-del-componente // Crea un nuevo componente.  > ng generate service nombre-del-servicio // Genera un nuevo servicio Angular.  > ng generate class nombre-de-la-clase // Crea una nueva clase TypeScript.  > ng generate interface nombre-de-la-interfaz // Genera una nueva interfaz TypeScript.  > ng generate pipe nombre-del-pipe // Crea un nuevo pipe (filtro) Angular.  > ng generate directive nombre-de-la-directiva // Genera una nueva directiva Angular.  > ng generate guard nombre-del-guard // Crea un nuevo guardia para la protección de rutas.  > ng generate application nombre-de-la-aplicacion // Genera una nueva aplicación dentro del proyecto Angular existente para permitir múltiples aplicaciones en un solo proyecto. |
| --- |

**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”**

Migrar 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.

**Funcionalidades básicas**

1. Registro y Monitorización de Signos Vitales: El sistema debe permitir a los profesionales de la salud registrar y mantener un seguimiento de los signos vitales de los pacientes, como la frecuencia cardíaca, la presión arterial, la temperatura corporal, la frecuencia respiratoria y el nivel de saturación de oxígeno en la sangre.
2. Gestión de Pacientes: Capacidad para registrar información detallada sobre los pacientes, incluidos datos personales, historial médico, alergias
3. Registro de Antecedentes Médicos: Esta función permite a los profesionales de la salud registrar y mantener un registro completo de los antecedentes médicos de los pacientes
4. Historias Clínicas Digitales: Funcionalidad para crear, almacenar y gestionar historias clínicas electrónicas de los pacientes de manera segura y organizada.
5. Registro de Medicamentos: Posibilidad de prescribir medicamentos a los pacientes y registrar esa información en sus historias clínicas. Debe incluir datos como dosis, frecuencia, duración del tratamiento
6. Funcionalidades de Búsqueda y Filtrado: Herramientas que permitan a los usuarios buscar y filtrar información en las historias clínicas de los pacientes de manera eficiente, por ejemplo, por fecha, diagnóstico

## 1.3 Creación de equipos

| INTEGRANTE | ROL |
| --- | --- |
| Baldeon Lara Jhonatan Moises | **Equipo de desarrollo (base de datos)** |
| Sanabria Tinoco Wiliams Anthony | **Equipo de desarrollo (backend)** |
| Vera Chinchay Javier | **Equipo de desarrollo (fronted)** |
| Povis Uribe Jean Anthony | **Equipo de desarrollo (backend)** |
| Jauregui Ponce Jhon Kennedy | **Equipo de desarrollo (fronted)** |
| Grande Contreras Mario | **Equipo de desarrollo (fronted)** |
| Roldan Hinojosa Jhon | **Equipo de desarrollo (base de datos)** |
| Malpartida Zevallos Franco | **Equipo de desarrollo (backend)** |
| Diaz Loayza Diego Juan de Dios | **Equipo de desarrollo (fronted)** |
| Collachagua Ventura Junior Jesus | **Equipo de desarrollo (backend)** |

| INTEGRANTE | ROL |
| --- | --- |
| Roldan Hinojosa Jhon | Product owner |
| Grande Contreras Mario | Scrum master |

**1.4 Capacitación**

La capacitación juega un papel crucial en la gestión de proyectos, y en este contexto, el hecho de que la mayoría de los estudiantes estén familiarizados con Scrum representa una ventaja considerable. Scrum es una metodología ágil ampliamente aceptada 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 tiene experiencia en Scrum, podemos aprovechar este conocimiento para asegurar una planificación, ejecución y control del proyecto más efectivos.

## 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 miembros del equipo deben comprometerse a contribuir de manera activa y responsable al proyecto.
* Se espera que los miembros del equipo se comuniquen de manera abierta y honesta. Esto incluye compartir actualizaciones sobre el progreso del trabajo, expresar cualquier impedimento o problema que puedan enfrentar, y solicitar ayuda cuando sea necesario.
* Establecer reuniones periódicas, para asegurarse de que todos estén alineados en los objetivos del proyecto y para abordar cualquier preocupación o problema de manera oportuna.
* Cada miembro del equipo es responsable de completar sus tareas asignadas en tiempo y forma. Si un miembro del equipo no cumple con esta responsabilidad, se aplicarán consecuencias, como la disminución del porcentaje de participación en el proyecto de 3% por cada tarea no entregada y no se le va a considerar en las presentaciones de las tareas.
* Si un miembro del equipo no puede cumplir con un plazo, debe informar al equipo lo antes posible para que se puedan tomar las medidas necesarias.
* Todos los miembros del equipo deben seguir los estándares y buenas prácticas de desarrollo de software establecidos por la organización.
* Comentar cualquier funcionalidad para entender de qué se trata.
* Cuando surjan desacuerdos o conflictos, el equipo debe abordarlos de manera constructiva y profesional.
* 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**

* Establecer pautas claras para la documentación, incluyendo formatos, estructuras y herramientas, para garantizar coherencia en todo el proyecto.
* Antes de iniciar el desarrollo de una nueva funcionalidad, es crucial elaborar una documentación exhaustiva de las definiciones de usuario o historias de usuario, que abarque los criterios de aceptación y los requisitos funcionales.
* La documentación debe actualizarse continuamente a lo largo del proyecto para reflejar cualquier cambio en requisitos, arquitectura o diseño de manera oportuna.
* Es importante programar revisiones regulares de la documentación para garantizar su calidad y precisión, ya sea como parte de las tareas de revisión de sprint o como procesos separados.
* Se debe garantizar que todos los miembros del equipo tengan acceso a la documentación relevante y establecer una política clara de gestión de permisos para controlar el acceso a la información.

**Reglas de Backend**

* Implementar medidas de seguridad efectivas para asegurar un acceso seguro a las funcionalidades.
* Elaborar una documentación comprensible que contenga ejemplos de solicitudes y respuestas, con el fin de simplificar la integración con el frontend.
* Crear pruebas unitarias para garantizar la calidad del código.
* Gestionar los errores de manera apropiada y ofrecer respuestas de error claras y útiles

**Reglas de Frontend: Diseño**

* Mantener un diseño limpio y minimalista para facilitar la navegación y comprensión de la interfaz.
* Proporcionar retroalimentación visual clara cuando los usuarios interactúan con el software, como cambios de color o animaciones, para confirmar que sus acciones se han realizado correctamente.
* Los colores serán de acuerdo a la empresa que colabore con nosotros.
* 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.
* Implementar medidas de seguridad adecuadas, como control de acceso basado en roles y privilegios, para proteger la confidencialidad y la integridad de los datos.
* Establecer procedimientos regulares de respaldo de datos y planes de recuperación.
* Mantener una documentación clara y actualizada de la estructura de la base de datos

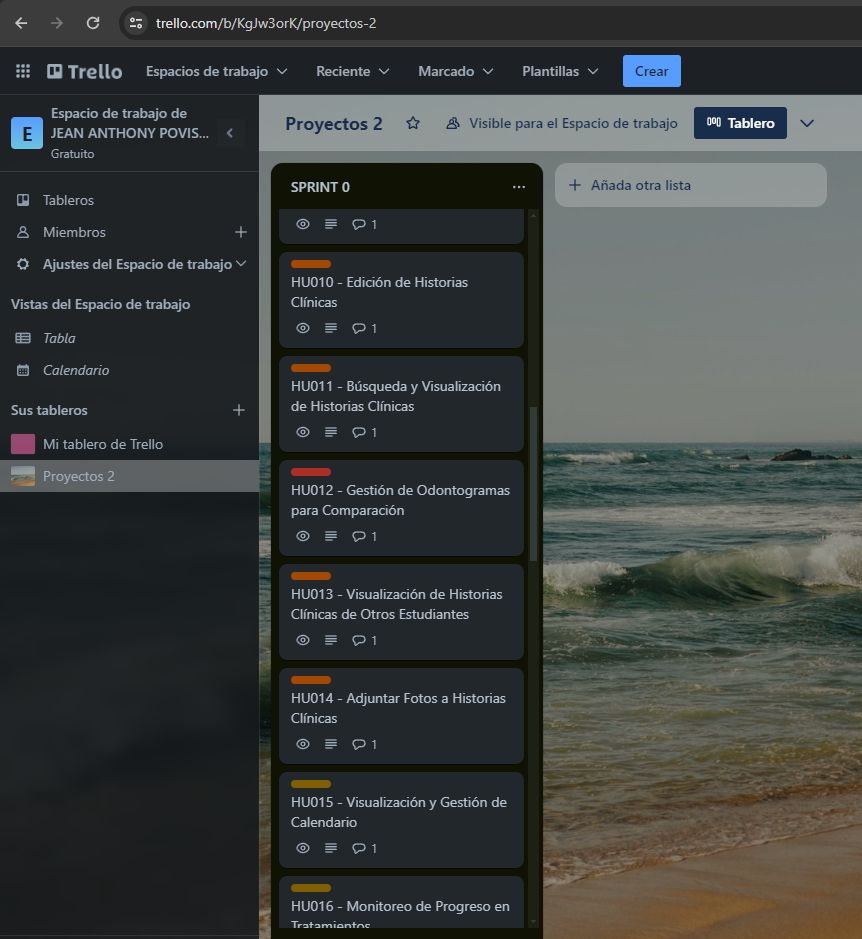
## 1.6 Planificación de Infraestructura

**Entornos de Desarrollo y Pruebas:**

* ***Identificación de Entornos:***
  + **frontend:** Angular - TypeScript - Bootstrap
  + **backend:** Nest js - TypeScript
  + **orm:** TypeOrm
  + **base de datos:** Mysql
  + **Cloud:** Docker
* ***Herramientas:***
  + IDE: Visual Studio Code
  + Modelado: Bizagi y Erwin Data Modeler
  + Diseño: Photoshop, Adobe XD
  + Presentación: Canva
  + Organizador : Trello

**1.7 Tablero de Desarrollo de Historias de Usuario**

[**https://trello.com/b/KgJw3orK/proyectos-2**](https://trello.com/b/KgJw3orK/proyectos-2)



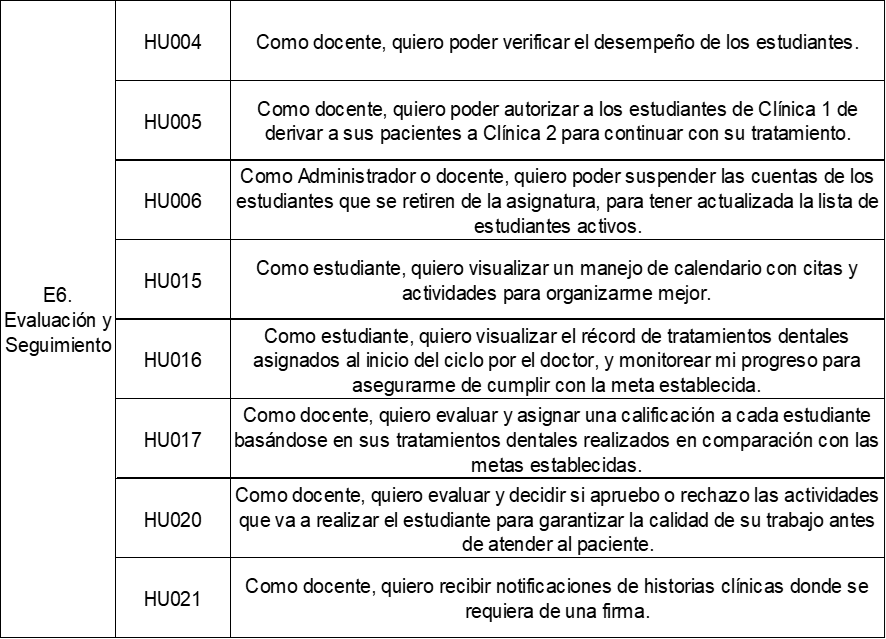
**Borrador del Product Backlog:**

## 1.8 Gestión Efectiva del Product Backlog

[2023/10/01\_05\_Backlog Detallado del Producto.xlsx](https://docs.google.com/spreadsheets/d/1b8H_ldKDUJ8alllc0YawxfVkoRzhP5XpO2T1i5R1FuM/edit?usp=sharing)

## 

## 



## 1.9 Planificación y Ejecución de los Sprint

### 1.9.1 Historias de usuario del primer sprint

| **ID** | **Historia de Usuario** |
| --- | --- |
| HU001 | Como estudiante, quiero poder agregar nuevas historias clínicas para registrar nuevos casos. |
| HU002 | Como estudiantes, quiero editar historias clínicas para actualizar la información de un paciente. |
| HU003 | Como estudiante, quiero buscar y visualizar mis historias clínicas para estudiar casos específicos. |
| HU004 | Como estudiante, quiero ver historias clínicas de otros estudiantes para aprender de otros casos. |
| HU005 | 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. |
| HU006 | Como docente, quiero evaluar y asignar una calificación a cada estudiante basándose en sus tratamientos dentales realizados en comparación con las metas establecidas. |
| HU007 | Como docente, quiero acceder al historial clínico completo de un paciente para supervisar y gestionar los tratamientos realizados por el estudiante. |
| HU008 | Como docente, quiero visualizar el historial clínico para la revisión/calificación del paciente atendido por el estudiante. |
| HU009 | 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. |
| HU010 | Como estudiante quiero subir al sistema la firma del paciente en el documento de consentimiento informado. |
| HU011 | Como estudiante quiero poder ingresar los signos vitales del paciente para asi llevar un registro de evolución |
| HU012 | Como estudiante deseo poder rellenar la historia clínica digitalmente, que tenga el formato de las historias físicas |

**1.09.2 Tareas de las historias de usuario y clasificación por sprints**

**Sprint 1: Funcionalidades básicas de historias clínicas**

Historia 1:

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

Tareas:

* Diseñar la interfaz de usuario para la creación de nuevas historias clínicas.
* Implementar campos necesarios para el registro de la historia clínica (nombre del paciente, fecha de nacimiento, motivo de consulta, etc.).
* Crear la lógica de backend para almacenar nuevas historias clínicas en la base de datos.
* Implementar validaciones para los campos requeridos.
* Realizar pruebas para asegurar que las historias clínicas se agregan correctamente.

Historia 2:

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

Tareas:

* Diseñar la interfaz de usuario para la creación de nuevas historias clínicas.
* Implementar campos necesarios para el registro de la historia clínica (nombre del paciente, fecha de nacimiento, motivo de consulta, etc.).
* Crear la lógica de backend para almacenar nuevas historias clínicas en la base de datos.
* Implementar validaciones para los campos requeridos.
* Realizar pruebas para asegurar que las historias clínicas se agregan correctamente.

Historia 3:

Como estudiante, quiero editar historias clínicas para actualizar la información de un paciente.

Tareas:

* Diseñar la interfaz de usuario para la edición de historias clínicas.
* Implementar la lógica de backend para actualizar los registros en la base de datos.
* Crear validaciones para los campos editables.
* Realizar pruebas para asegurar que la edición de historias clínicas funciona correctamente.

**Sprint 2: Mejoras en la visualización y aprendizaje colaborativo**

Historia 4:

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

Tareas:

* Diseñar la interfaz de usuario para listar historias clínicas de otros estudiantes.
* Implementar permisos y restricciones para asegurar que solo se puede ver la información permitida.
* Crear la lógica de backend para recuperar historias clínicas de otros estudiantes.
* Realizar pruebas para asegurar que las historias clínicas de otros estudiantes se muestran correctamente

Historia 5:

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

Tareas:

* Diseñar la interfaz de usuario para adjuntar imágenes y documentos.
* Implementar la funcionalidad de carga de archivos en el frontend.
* Crear la lógica de backend para almacenar y vincular las imágenes y documentos con las historias clínicas.
* Implementar la visualización de las imágenes y documentos adjuntos.
* Realizar pruebas para asegurar que los archivos se cargan y se muestran correctamente.

**Sprint 3: Evaluación y supervisión por parte del docente**

Historia 6:

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

Tareas:

* Diseñar la interfaz de usuario para la evaluación y asignación de calificaciones.
* Implementar la lógica de backend para almacenar las calificaciones y comentarios del docente.
* Crear formularios de evaluación con criterios específicos.
* Realizar pruebas para asegurar que las calificaciones se guardan y se muestran correctamente.

Historia 7:

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

Tareas:

* Diseñar la interfaz de usuario para que los docentes puedan acceder al historial clínico completo.
* Implementar la lógica de backend para recuperar los historiales clínicos completos.
* Asegurar que la interfaz muestra todos los datos relevantes de manera clara.
* Realizar pruebas para asegurar el acceso y la correcta visualización del historial clínico.

Historia 8:

Como docente, quiero visualizar el historial clínico para la revisión/calificación del paciente atendido por el estudiante.

Tareas:

* Diseñar la interfaz de usuario para la revisión y calificación del historial clínico.
* Implementar la funcionalidad para que los docentes puedan agregar notas y comentarios.
* Crear la lógica de backend para guardar las notas y comentarios del docente.
* Realizar pruebas para asegurar que los docentes pueden revisar y calificar correctamente.

**Sprint 4: Aprobación de actividades y firmas digitales**

Historia 9:

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.

Tareas:

* Diseñar la interfaz de usuario para la aprobación o rechazo de actividades.
* Implementar la lógica de backend para guardar el estado de aprobación o rechazo.
* Crear notificaciones para los estudiantes sobre la decisión del docente.
* Realizar pruebas para asegurar que la funcionalidad de aprobación/rechazo funciona correctamente.

Historia 10:

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

Tareas:

* Diseñar la interfaz de usuario para la captura y subida de firmas.
* Implementar la funcionalidad para cargar y almacenar la firma digital del paciente.
* Asociar la firma con el documento de consentimiento informado en la historia clínica.
* Realizar pruebas para asegurar que las firmas se capturan y almacenan correctamente.

**Sprint 5: Registros adicionales y formatos de historias clínicas**

Historia 11:

Como estudiante, quiero poder ingresar los signos vitales del paciente para así llevar un registro de evolución.

Tareas:

* Diseñar la interfaz de usuario para la entrada de signos vitales.
* Implementar la lógica de backend para almacenar los signos vitales en la base de datos.
* Crear la funcionalidad para visualizar los signos vitales como parte del historial clínico.
* Realizar pruebas para asegurar que los signos vitales se registran y se muestran correctamente.

Historia 12:

Como estudiante, deseo poder rellenar la historia clínica digitalmente, que tenga el formato de las historias físicas.

Tareas:

* Analizar y replicar el formato físico de las historias clínicas en una interfaz digital.
* Implementar los campos y secciones necesarias en la interfaz digital.
* Crear la lógica de backend para almacenar la historia clínica en el formato especificado.
* Realizar pruebas para asegurar que la historia clínica digital se rellena y se visualiza correctamente según el formato físico.

## 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.

### 1.10.2 Preparación de las reuniones semanales de Scrum: Para el seguimiento del progreso y la identificación de obstáculos los días Domingos a las 6pm.

### 1.10.3 Preparación de la reunión de planificación del sprint: Para demostrar las funcionalidades completadas los días martes durante el horario de clases.

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

* **Revisión de objetivos del sprint anterior:** Evaluar si se lograron los objetivos establecidos para el sprint anterior, lo cual proporciona contexto para la discusión sobre el progreso del proyecto.
* **Revisión de incremento:** Muestra el trabajo completado durante el sprint (incremento) a los stakeholders relevantes, lo que puede incluir demostraciones de las características desarrolladas.
* **Preparación de la demostración:** Se prepara una demostración del incremento para presentar durante la reunión de revisión del sprint, lo cual puede implicar la preparación de entornos de demostración y datos de prueba.
* **Revisión de la lista de productos:** Revisar la lista de productos (Product Backlog) para determinar si necesita ajustes basados en el aprendizaje del sprint anterior o comentarios de los stakeholders.
* **Preparación de retroalimentación:** Prepararnos para recibir retroalimentación sobre el incremento mostrado durante la reunión de revisión, estando listo para responder preguntas y tomar notas sobre los comentarios recibidos.
* **Identificación de lecciones aprendidas:** Reflexionar sobre el sprint anterior para identificar lo que funcionó bien, lo que no funcionó tan bien y las lecciones aprendidas que se pueden aplicar al próximo sprint.

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

### Revisión del sprint pasado: Antes de la reunión, revisar el sprint pasado para tener una comprensión clara de lo que se logró, los desafíos enfrentados y cualquier problema que haya surgido durante el sprint.

### Recopilación de datos y métricas: Recopilar datos relevantes del sprint, como métricas de rendimiento del equipo, velocidad de entrega, calidad del trabajo realizado, tiempo dedicado a diferentes tareas, etc. Estos datos proporcionan una base objetiva para la discusión durante la retrospectiva.

### Solicitud de retroalimentación: Se puede solicitar retroalimentación a todos los miembros del equipo sobre el sprint pasado, pidiéndoles que identifiquen aspectos positivos, áreas de mejora y sugerencias para el futuro.

### Preparación de la agenda: Se elabora una agenda para la reunión de retrospectiva, que incluye puntos de discusión como "Lo que salió bien", "Lo que no salió tan bien", "Oportunidades de mejora" y "Acciones a seguir". Tener una agenda clara ayuda a mantener la reunión enfocada y productiva.

### 