**Sprint 0**

**INTEGRANTES:**

1. Camposano Iriarte, Giovanny Jhon

2. Capacoila Apolinario, Jhon Carlos

3. Chavez Pino, Max Dextre

4. De la calle Coz, Carlos Daniel

5. De la cruz Taza, Geovany Ernesto

6. Galvan Durand, Michael Tommy

7. Machuca Barzola, Juan Rafael

8. Palacios Pacheco Javier Luis

9. Rupay Ricaldi Daniel Antony

10. Rivera Varje, Mishell Maricielo

**COORDINADORES**

* Camposano Iriarte, Giovanny Jhon

**BackEnd - JAVASCRIPT**

* Capacoila Apolinario, Jhon Carlos
* Camposano Iriarte, Giovanny Jhon
* De la calle Coz, Carlos Daniel

**FrontEnd -PROGRAMACIÓN**

* Machuca Barzola, Juan Rafael

**Documentación - DRIVE**

* Rivera Varje, Mishell Maricielo
* Rupay Ricaldi Daniel Antony

**Bases de datos - Mysql**

* De la calle Coz, Carlos Daniel
* De la cruz Taza, Geovany Ernesto

**Diseño FrontEnd - CSS OR BOOTSTRAP**

* Machuca Barzola, Juan Rafael
* Rupay Ricaldi Daniel Antony

**1.- Configuración del Entorno  
visual studio code.  
jhon.**

**2 .- 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 para múltiples clínicas odontológicas.

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.

**3 .- Creación de Equipos:**

La formación de equipos en el contexto de un proyecto es de vital importancia para optimizar el aprovechamiento del potencial de cada uno de sus miembros y, en última instancia, alcanzar el éxito. Cuando individuos con diversas habilidades y capacidades colaboran de manera efectiva, se experimenta un aumento significativo en la eficiencia operativa. Además, la sinergia resultante estimula la generación fluida de ideas y la toma de decisiones más sólidas y fundamentadas. La organización meticulosa de un equipo también desempeña un papel crucial al permitir que el grupo aborde los desafíos con creatividad y enfrente obstáculos con resiliencia. En este entorno colaborativo, los logros individuales convergen en un objetivo compartido, lo que se traduce en resultados más efectivos y eficientes en el proyecto a realizar, se han conformado 5 grupos, cada uno con un responsable a cargo, el rol de cada integrante se detalla a continuación:

| **INTEGRANTES** | **ROLES** |
| --- | --- |
| Armando Carrillo | **Product Owner (Empresa)** |
| Camposano Iriarte Giovanny Jhon | **SCRUM MASTER Scrum Developer (Diseño FrontEnd - FrontEnd)** |
| De la calle Coz Carlos Daniel | **Scrum Developer (BackEnd - Bases de datos)** |
| De la cruz Taza, Geovany Ernesto | **Scrum Developer (Bases de datos)** |
| Galvan Durand, Michael Tommy | **Scrum Developer (BackEnd)** |
| Machuca Barzola, Juan Rafael | **Scrum Developer (Diseño FrontEnd - FrontEnd)** |
| Chavez Pino, Max Dextre | **Scrum Developer (Diseño FrontEnd - FrontEnd)** |
| Palacios Pacheco Javier Luis | **Scrum Developer (BackEnd)** |
| Rupay Ricaldi Daniel Antony | **Scrum Developer (BackEnd)** |
| Rivera Varje, Mishell Maricielo | **Scrum Developer (Diseño FrontEnd - FrontEnd)** |
| Capacoila Apolinario Jhon Carlos | **Scrum Developer (BackEnd)** |

**4.- Capacitación**

**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.
* 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.
* Es importante que cada miembro del equipo comprenda claramente sus responsabilidades y roles en el proyecto. Esto evita confusiones y duplicación de esfuerzos.
* Antes de considerar una tarea como completada, debe pasar por un proceso de revisión y aprobación. Esto garantiza que se cumplan los estándares de calidad y que las partes interesadas estén satisfechas con el trabajo realizado.
* Si surge un problema o un obstáculo, es importante que los miembros del equipo asuman la responsabilidad de resolverlo o de informar a la persona adecuada para que se tomen medidas.
* En casos de incumplimiento leve, el primer paso podría ser una advertencia verbal. El miembro del equipo se reuniría con su supervisor o el responsable del proyecto para discutir el problema y acordar un plan de acción para mejorar.
* En casos graves de incumplimiento, una opción podría ser suspender temporalmente al miembro del equipo del proyecto. Durante este período, no se le asignarían tareas relacionadas con el proyecto.
* En casos extremos en los que el incumplimiento sea persistente y cause un daño significativo al proyecto, se podría considerar la terminación de su participación o la relación con los miembros del equipo para el proyecto.

**Reglas de Documentacion**

* 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**

* Todos los datos confidenciales deben ser almacenados de manera segura y encriptada.
* Se deben implementar medidas de autenticación y autorización para proteger los datos y recursos del sistema.
* Se debe realizar una validación exhaustiva de los datos de entrada para prevenir ataques de inyección (por ejemplo, SQL injection) y otros tipos de ataques.
* Diseñar la arquitectura para que sea escalable y pueda manejar un aumento en el tráfico o la carga de trabajo.
* Utilizar servicios en la nube o contenedores para facilitar la escalabilidad.
* Implementar mecanismos de recuperación ante fallos para garantizar que la aplicación siga funcionando incluso en caso de fallos en el sistema.
* Implementar mecanismos de recuperación ante fallos para garantizar que la aplicación siga funcionando incluso en caso de fallos en el sistema.
* Realizar pruebas exhaustivas antes de implementar cambios en producción.
* Seguir un proceso de despliegue controlado que minimice el impacto en los usuarios finales.

**Reglas de Frontend: Diseño**

* Los colores serán de acuerdo a la empresa que colabore con nosotros.
* Entrar a los módulos según su rol (administrador, odontólogos, recepcionista, etc).
* Priorizar lo estético al momento de diseñar.
* No acoplar mucho texto dentro del diseño.
* 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).
* Ingresar como usuario y rol para seguir con la línea de desarrollo.
* Las tareas asignadas serán según el rol 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”).
* Programaremos reuniones para ver avances y posibles cuestiones.
* Cada historia de usuario se avanzará junto al otro grupo frontend: diseño.
* Para los procesos de la empresa nos basaremos en bpmn que se diseñe.
* Esperaremos a los de diseño para luego unir lo programado con lo visual.

**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).
* Optimizar las consultas a la base de datos y reducir la carga de procesamiento en la medida de lo posible.

**8. Planificación de Infraestructura:**

**Entornos de Desarrollo y Pruebas:**

* ***Identificación de Entornos:***
  + **frontend:** html- bootstrap - css
  + **backend:** javascript
  + **base de datos:** mysql
* ***Herramientas:***
  + IDE: Visual Studio Code
  + Modelado: Bizagi.
  + PHPMyadmin
  + colocar las herramientas que están utilizando

**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.: https://github.com/TallerProII/SoftwareOdontologiaII
  + Ramas de Desarrollo: Se seguirá una estrategia de ramificación (por ejemplo, Gitflow) para gestionar el desarrollo de características y correcciones de errores.

rama principal main

rama de desarrollo dev

rama de backend: backend

rama de frontend: frontend (se unen ambos frontend)

rama de frontend: frontend-programación

rama de frontend: frontend-diseño

rama HU#: frontend- HU# (solo si corresponde a frontend)

rama HU#: backend - HU# (solo si corresponde a backend)

rama HU#: frontend-programación - HU# (solo si corresponde a frontend-programación)

rama HU#: frontend-diseño - HU# (solo si corresponde a frontend-programación)

rama HU#: HU# (solo si enlaza a backend y a frontend)

* 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.

**7.- Configuración de Herramientas:**

**github kanban git**

**10.- Preparación del Product Backlog:**

**Preparación del Product Backlog para un Proyecto de Desarrollo de Software**

**Objetivo:** Refinar y priorizar el Product Backlog para tener una lista de historias de usuario claras, estimadas y listas para su desarrollo en el primer Sprint.

**Pasos de Preparación:**

**Reunión de Planificación Inicial:**

* Reunir al equipo de desarrollo, el Product Owner y, si es necesario, otros interesados clave.
* Revisar la visión del producto y los objetivos a largo plazo para comprender la dirección general del proyecto.

**Identificación de Historias de Usuario:**

* Invitar a los interesados a proponer características y funcionalidades para el producto.
* Utilizar técnicas como entrevistas, encuestas o talleres de diseño para recopilar ideas.

**Creación de Historias de Usuario:**

* Traducir las ideas en historias de usuario. Cada historia debe ser:
  + ***Independiente:*** No debe depender de otras historias para ser implementada.
  + ***Negociable:*** Debe ser lo suficientemente flexible como para permitir ajustes.
  + ***Valiosa:*** Debe agregar valor al producto.
  + ***Estimable:*** Debe ser lo suficientemente claro para poder estimar su tamaño y complejidad.

**Estimación de Historias:**

* Invitar al equipo de desarrollo a estimar el esfuerzo necesario para implementar cada historia de usuario.
* Puedes utilizar la técnica de puntos de historia o cualquier otra que sea adecuada para tu equipo.

**Refinamiento de Historias:**

* Realizar reuniones de refinamiento de historias para aclarar cualquier ambigüedad y asegurarse de que las historias sean comprensibles para todos.
* Actualizar las historias según sea necesario.

La preparación del Product Backlog de esta manera garantiza que las historias de usuario estén bien definidas, estimadas y priorizadas antes de comenzar el primer Sprint. Esto ayuda a que el equipo de desarrollo y el Product Owner tengan una comprensión clara de lo que se debe trabajar en cada Sprint y asegura que el producto se desarrolle de manera eficiente y en línea con las necesidades del cliente.

| ID | Historia de Usuario | Aceptado | Sprint | Prioridad | Comentarios |
| --- | --- | --- | --- | --- | --- |
| HU01 | Poder ver mi agenda de citas actualizada en tiempo real para conocer mi carga de trabajo diaria y prepararse adecuadamente para las consultas con los pacientes. | SI |  | ALTA |  |
| HU02 | Poder confirmar o rechazar citas programadas por los pacientes para tener control sobre mi agenda y garantizar que se ajuste a mis disponibilidades. | SI |  | BAJA |  |
| HU03 | Recibir notificaciones de cambios en las citas para estar al tanto de cualquier modificación en mi agenda y poder hacer los ajustes necesarios. | SI |  | MODERADA |  |
| HU04 | Acceder a la información de las citas de los pacientes antes de cada cita para estar enterado sobre su historial médico y proporcionar un tratamiento personalizado y seguro. | SI |  |  |  |
| HU05 | Poder crear y editar las historias clínicas de mis pacientes en el sistema para mantener un registro detallado de su historial médico y tratamientos anteriores. | SI |  |  |  |
| HU06 | Buscar y acceder a las historias clínicas de otros pacientes en el sistema cuando lo requiera para proporcionar un tratamiento más efectivo y personalizado. | SI |  |  |  |
| HU07 | Visualizar un listado de todas las historias clínicas según la fecha en forma ascendente. | SI |  |  |  |
| HU08 | Realizar búsquedas en las historias clínicas utilizando una función de búsqueda avanzada. | SI |  |  |  |
| HU09 | Acceder al sistema y buscar la información de un paciente existente para poder revisar su historial clínico antes de la consulta. | SI |  |  |  |
| HU10 | Crear un nuevo registro de paciente en el sistema cuando un nuevo paciente se presente en la clínica por primera vez. | SI |  |  |  |
| HU11 | Registrar la información médica relevante de un paciente, como antecedentes de salud, alergias o medicamentos actuales, para proporcionar un tratamiento dental seguro y personalizado. | SI |  |  |  |
| HU12 | Actualizar o Eliminar toda la información del paciente en su perfil, si hay cambios en su historial médico, o de contacto para mantener la información actualizada y precisa. | SI |  |  |  |
| HU13 | Agregar una nueva especialidad al sistema para que la clínica pueda ofrecer una gama completa de servicios dentales. | SI |  |  |  |
| HU14 | Modificar la descripción o los detalles de una especialidad existente para reflejar con precisión los servicios que se ofrecen. | SI |  |  |  |
| HU15 | Asignar o cambiar la especialidad de un colega odontólogo en el sistema para asegurarme de que los pacientes sean referidos al profesional adecuado. | SI |  |  |  |
| HU16 | Eliminar una especialidad que ya no se ofrece en la clínica para mantener actualizada la lista de especialidades disponibles. | SI |  |  |  |
| HU17 | Poder ver una lista completa de todas las especialidades de la clínica para estar al tanto de los servicios disponibles y poder referir a los pacientes según sus necesidades específicas. | SI |  |  |  |
| HU18 | Agregar nuevos médicos al sistema y asignarles su especialidad correspondiente para asegurarse de que el personal médico esté correctamente registrado. | SI |  |  |  |
| HU19 | Modificar y/o eliminar la especialidad asignada a un médico existente para reflejar con precisión su área de experiencia actual. | SI |  |  |  |
| HU20 | Tener acceso a la información de contacto de mis colegas con diferentes especialidades en la clínica para coordinar y comunicarse eficazmente cuando se requiere una colaboración interdisciplinaria. | SI |  |  |  |
| HU21 | Poder ver una lista completa de todos los médicos y sus especialidades asignadas para facilitar la gestión del personal médico y la asignación de pacientes según sus necesidades. | SI |  |  |  |
| HU22 | Crear un nuevo registro de odontograma para un paciente en el sistema para documentar su estado dental inicial. | SI |  |  |  |
| HU23 | Poder ingresar información detallada sobre la condición dental del paciente en el odontograma para registrar problemas específicos, como caries, restauraciones o ausencias de dientes. | SI |  |  |  |
| HU24 | Actualizar el odontograma existente de un paciente después de cada consulta para mantener un registro continuo de su salud bucal. | SI |  |  |  |
| HU25 | Buscar a mi paciente y visualizar toda la información actualizada en tiempo real, además de poder exportar en pdf el odontograma. | SI |  |  |  |
| HU26 | Registrar un nuevo tratamiento para un paciente en el sistema para documentar el procedimiento que se realizará. | SI |  |  |  |
| HU27 | Poder detallar la información sobre el tratamiento, incluyendo la descripción, la fecha de realización y los materiales utilizados para mantener un registro preciso de la atención proporcionada. | SI |  |  |  |
| HU28 | Actualizar el registro de tratamiento existente de un paciente si se realizan tratamientos adicionales o de seguimiento para mantener un historial completo de su atención dental. | SI |  |  |  |
| HU29 | Consultar el historial de tratamientos previos de un paciente en el sistema para tomar decisiones informadas sobre el tratamiento actual y planificar futuras intervenciones. | SI |  |  |  |
| HU30 | Un calendario en el sistema que me permita visualizar mis citas programadas de manera clara y organizada para poder administrar mi agenda eficientemente. | SI |  |  |  |
| HU31 | Tener la opción de agregar nuevas citas al calendario directamente para programar las consultas de mis pacientes de manera efectiva. | SI |  |  |  |
| HU32 | Recibir notificaciones automáticas en el calendario para recordarme mis próximas citas y asegurarme de estar preparado para cada consulta. | SI |  |  |  |
| HU33 | Reprogramar o cancelar citas existentes desde el calendario en caso de cambios en la agenda o emergencias. | SI |  |  |  |
| HU34 | Registrar los pagos de los pacientes en el sistema para llevar un registro preciso de los servicios prestados y las transacciones financieras. | SI |  |  |  |
| HU35 | Generar facturas y recibos de pago en el sistema para proporcionar a los pacientes un comprobante de su pago por los tratamientos dentales recibidos. | SI |  |  |  |
| HU36 | Realizar un seguimiento de los pagos pendientes y vencidos en el sistema para garantizar una gestión eficiente de las cuentas por cobrar. | SI |  |  |  |
| HU37 | Realizar reembolsos o ajustes de pagos en el sistema en caso de errores o situaciones especiales para mantener registros precisos y justos de las transacciones financieras. | SI |  |  |  |

**9.- Establecimiento de Objetivos del Primer Sprint:**

**esto es solo un ejemplo, ya en clase se escogerá para el primer sprint**

**Objetivo del Sprint:** Implementar la funcionalidad de registro de pacientes y programación de citas en el sistema clínico odontológico.

**Historias de Usuario Seleccionadas:**

**Historia de Usuario (HU1):** Como recepcionista, quiero registrar la información personal de un paciente, incluyendo nombre, dirección y datos de contacto.

Tarea 1: Diseñar la interfaz de registro de pacientes.

Tarea 2: Implementar la funcionalidad de ingreso de datos personales.

Tarea 3: Validar y almacenar los datos en la base de datos.

Tarea 4: Implementar una función de búsqueda de pacientes existentes durante el registro.

**Historia de Usuario (HU2):** Como recepcionista, quiero programar citas para pacientes, asignando fechas y horas.

Tarea 1: Diseñar la interfaz de programación de citas.

Tarea 2: Implementar la funcionalidad de selección de fecha y hora.

Tarea 3: Conectar la programación de citas con el registro de pacientes.

Tarea 4: Validar la disponibilidad de horarios y fechas para las citas.

**Historia de Usuario (HU3):** Como paciente, quiero recibir confirmaciones automáticas de citas.

Tarea 1: Configurar un sistema de notificaciones automáticas por correo electrónico.

Tarea 2: Conectar el sistema de notificaciones a las citas programadas.

Tarea 3: Implementar opciones para que los pacientes elijan recibir confirmaciones por correo electrónico o mensaje de texto.

Tarea 4: Realizar pruebas exhaustivas de notificaciones y confirmaciones automáticas.

**Historia de Usuario (HU4):** Como odontólogo, quiero ver una lista de mis próximas citas en orden cronológico.

Tarea 1: Diseñar la interfaz de lista de citas para el odontólogo.

Tarea 2: Implementar la visualización de las citas programadas en orden cronológico.

Tarea 3: Agregar filtros para ver citas específicas (por paciente, fecha, etc.).

Tarea 4: Realizar pruebas de la lista de citas.

**Historia de Usuario (HU5):** Como administrador, quiero un panel de control que muestre la carga de trabajo actual del personal.

Tarea 1: Diseñar la interfaz del panel de control.

Tarea 2: Implementar la funcionalidad para mostrar la carga de trabajo actual del personal.

Tarea 3: Agregar gráficos e informes para visualizar la carga de trabajo.

Tarea 4: Realizar pruebas exhaustivas del panel de control.

**Duración del Sprint:** 3 semanas. **preferible por tarea un dia**

**Reuniones del Sprint:**

**Reunión de Planificación del Sprint:** Para seleccionar las historias de usuario y definir las tareas.

**Reuniones diarias de Scrum:** Para el seguimiento del progreso y la identificación de obstáculos.

**Reunión de Revisión del Sprint:** Para demostrar las funcionalidades completadas.

**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.

**10 .- Preparación para las Reuniones de Scrum:**

**definir en la clase en una reunión de encargados**

**Objetivo:** Asegurarse de que el equipo comprenda las reuniones diarias de Scrum, la Reunión de Planificación del Sprint, la Reunión de Revisión del Sprint y la Retrospectiva del Sprint.

**Pasos de Preparación:**

**Reunión de Planificación del Sprint:**

**Convocatoria y Duración:**

Establecer la fecha y la hora para la Reunión de Planificación del Sprint, generalmente programada al inicio de cada Sprint y con una duración máxima de 1 horas para un Sprint de 3 semanas.

**Propósito de la Reunión:**

Explicar que esta reunión se centra en definir qué trabajo se realizará durante el Sprint y cómo se hará.

**Revisión del Product Backlog:**

Asegurarse de que el Product Owner haya priorizado el Product Backlog antes de la reunión y que esté listo para presentarlo.

**Encargados:** Dentro de la clase.

**Documentación:** Después de la reunión de encargados y estar presente el las demás.

**Base de Datos:** Después de la reunión de encargados.

**Backend:** Después de la reunión de encargados.

**Diseño:** Después de la reunión de encargados.

**Frontend:** Después de la reunión de encargados.

**Reunión de Revisión del Sprint:**

**Convocatoria y Audiencia:**

Establecer la fecha y la hora para la Reunión de Revisión del Sprint, programada al final de cada Sprint.

Indicar que esta reunión se realiza para presentar lo que se ha completado en el Sprint a los interesados y obtener su retroalimentación.

**Demostración del Producto:**

Informar al equipo que se llevará a cabo una demostración del producto funcional al final del Sprint.

**Base de Datos:** Sabado de 6:00 pm hasta acabar

**Frontend (Diseño y programación):** Sábado de 7:00 pm hasta acabar

**Backend:** Sábado de 8:00 pm hasta acabar

**Encargados:** Sábado de 9:00 pm hasta acabar (aqui estaran los de **documentación**)

**Retrospectiva del Sprint:**

**Convocatoria y Duración:**

Programar la Retrospectiva del Sprint después de la Reunión de Revisión del Sprint y antes de la próxima Reunión de Planificación del Sprint.

**Propósito de la Retrospectiva:**

Explicar que la Retrospectiva se enfoca en reflexionar sobre el Sprint y mejorar los procesos y la colaboración del equipo.

**Herramientas de Retroalimentación:**

Preparar pizarras, tarjetas adhesivas u otras herramientas que se utilizarán durante la Retrospectiva para recopilar comentarios y retroalimentación.

**Facilitador de la Retrospectiva:**

Designar a un Scrummaster o facilitador para dirigir la Retrospectiva y asegurarse de que se sigan los pasos adecuados.

**Todos:** Domingo a las 8:00 pm hasta acabar esto será por un meet para ver los avances y tener una retroalimentación.

Esta preparación ayudará al equipo a entender la importancia y los objetivos de cada reunión de Scrum, lo que fomentará la participación activa y la eficiencia en el proceso de desarrollo ágil. Además, se asegurará de que todas las partes involucradas tengan claridad sobre cómo se llevan a cabo estas reuniones y cuál es su propósito.

Asegurarse de que el equipo comprenda las reuniones diarias de Scrum, la Reunión de Planificación del Sprint, la Reunión de Revisión del Sprint y la Retrospectiva del Sprint.