HEALTHTECH

**Plan de Gestión de la Configuración del Software**

Integrantes

Chávez Gave Jose Luis

Lozano Paredes Renzo Jeanpier Santos

Mendoza Gómez, Carlos Daniel

Rosillo Ramirez Jeremy Andrew

Obando Salinas Enmanuel Jose

Torres Rivera Richard Maycol

Vera Alva Miguel Angel

Versión 1.5

**Control de versión**

| **Fecha** | **Autor** | **Descripción** |
| --- | --- | --- |
| 19/09/2024 | Migue Angell Vera Alva | Creación del documento, avances del punto 1 hasta el punto 2.2 |
| 26/09/2024 | Richard Maycol Torres Rivera | Avance del punto 3.1 “Identificación” del apartado “Actividades de GCS” |
| 07/11/2024 | Jeremy Andrew Rosillo Ramirez | Avance del punto 3.2 “Control de la GCS” |
| 27/11/2024 | Jeremy Andrew Rosillo Ramirez | Avance del punto 3.3 al 3.5 |
| 28/11/2024 | Mendoza Gomez, Carlos Daniel | Avance del punto 3.6 |
|  |  |  |

1. **Introducción**
   1. **Situación de la empresa y problemática**

HealthTech S.A es una consultoría de software la cual lleva desde el año 2019 dedicada al desarrollo de aplicaciones web para pequeñas y medianas empresas especializadas en el sector salud. Durante estos años, asesoramos a quienes buscan una solución a la tardía ejecución y administración de sus operaciones clínicas y administrativas. Esto permite seguir eficientes estrategias de optimización, ajustadas a los contextos de cada cliente. Además, contamos con el compromiso del sector público y privado para la creación de diversos proyectos de software en paralelo, los cuales están enfocados a ofrecer una mejor gestión de citas para pacientes con problemas de salud mental. Los proyectos que hemos desarrollado anteriormente son **FisiMental**, **DigitalHealth**, **MindCare** y **CognitivaPro**. Mientras que el proyecto en desarrollo actual es **MentalSync** para la organización De-Mentes, una asociación pionera en el campo de la salud mental en Latinoamérica.

Actualmente, dichos proyectos y sus archivos principales se encuentran centralizados en computadoras, las cuales deben ser únicamente accedidas por los miembros de cada equipo de desarrollo. Por otra parte, cada equipo tiene disposición para usar las herramientas, dependencias y librerías que les parezcan más convenientes. Al tratarse de pocas unidades, se establecieron horarios y turnos para que cada desarrollador avance las actividades correspondientes a sus iteraciones. Este panorama ha generado en estos últimos meses problemas en la gestión de los proyectos de software, así como la falta de estandarización en herramientas, y un grave peligro de sobreescribir los avances aprobados, ocasionando una mayor cantidad de errores y la inutilidad de funcionalidades de las aplicaciones.

* 1. **El objetivo del plan**

Con el fin de abordar las problemáticas detectadas en la gestión de proyectos de HealthTech S.A., se propone implementar un sistema centralizado en la nube que permita a todos los equipos de desarrollo acceder de manera segura y organizada a los recursos necesarios. Este sistema integrará herramientas de control de versiones y gestión colaborativa, garantizando que los equipos puedan trabajar de manera eficiente y sin interferencias. Además, se establecerá un estándar común para el uso de librerías y dependencias, lo que permitirá una mayor coherencia en el desarrollo, reducirá los errores de sobreescritura y mejorará la estabilidad de las funcionalidades aprobadas.

La selección de una herramienta de control de versiones adecuada será clave para evitar conflictos de código y asegurar la correcta aprobación de cambios, manteniendo así la integridad y calidad de los proyectos. Este enfoque busca optimizar el desarrollo, minimizando retrasos y confusiones, y asegurando una mayor confiabilidad en los productos entregados a los clientes.

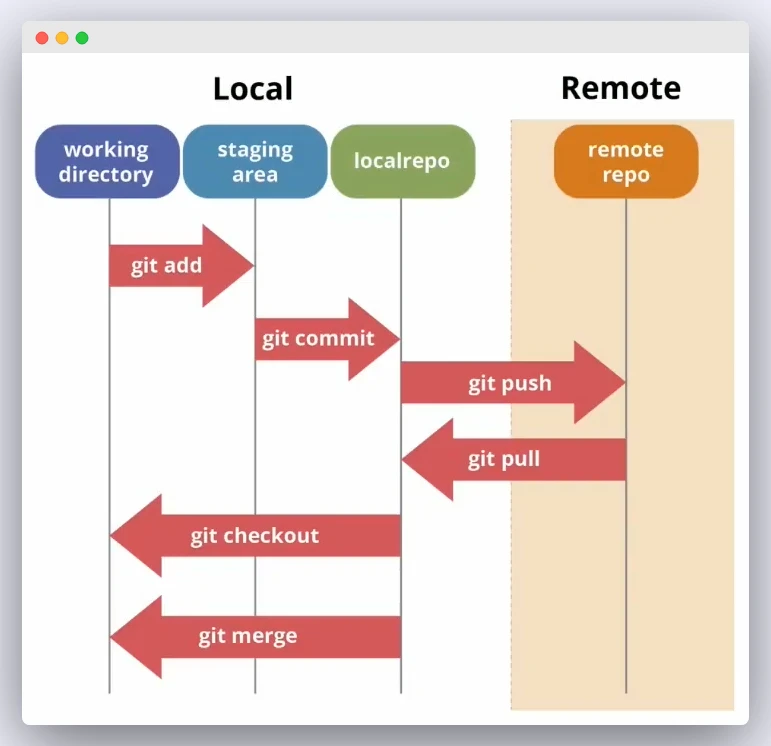
1. **Gestión de Configuración de Software**
   1. **Roles y responsabilidades**

| **Rol** | **Responsabilidad (es)** |
| --- | --- |
| **Project Manager** | Supervisar el funcionamiento de la Gestión de la Configuración. |
| **Gestor de Configuración** | Ejecutar todas las tareas de Gestión de la Configuración. |
| **Inspector de Aseguramiento de Calidad** | Auditar la Gestión de la Configuración. |
| **Bibliotecario** | Mantener un repositorio centralizado y seguro donde se almacenen los artefactos de configuración, como el código fuente, documentos de diseño, etc. |
| **Miembros del Equipo del Proyecto** | Seguir los procedimientos y políticas de configuración establecidos por el Gestor de Configuración y el Project Manager |

* 1. **Herramientas, entorno e Infraestructura**

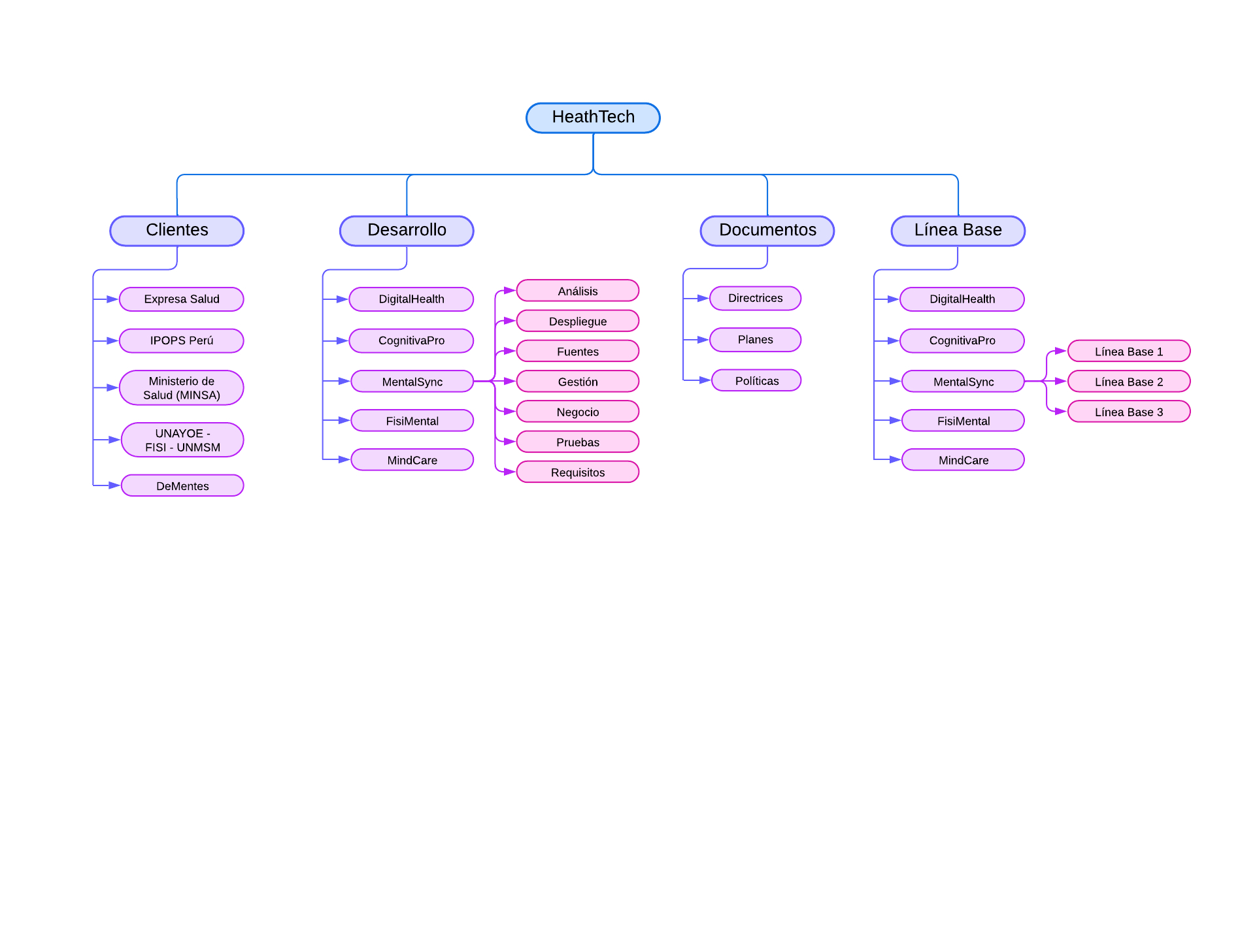
| **CRITERIOS** | **Git Hub** | **Git Lab** | **Bitbucket** |
| --- | --- | --- | --- |
| **Gratuito** | Repositorios privados y públicos, con limitaciones en funcionalidades avanzadas como la integración continua (CI) y límites de almacenamiento. | Incluye integración continua y despliegue continuo | Repositorios públicos y privados ilimitados de max. 5 colaboradores. Integracion con Jira, Confluence y productos de Atlassian |
| **Curva de aprendizaje** | Moderada, posee una interfaz web amigable | Un poco avanzada debido a la inclusión de muchas más funcionalidades integradas. | Moderada, interfaz moderna. La integración con Jira facilita procesos. |
| **Centralizado** | Repositorios alojados en servidores de GitHub | Repositorios alojados en su plataforma o en un self-hosted. | Repositorios alojados en los servidores de Atlassian |
| **Distribuido** | Posibilidad de clonar repositorios en máquinas locales, realizar commits offline, y subir cambios a un servidor central. | Los desarrolladores pueden trabajar localmente y sincronizar cambios con el servidor remoto. | Los colaboradores pueden trabajar localmente y posteriormente subir los cambios al repositorio remoto. |

La herramienta elegida para este proyecto será GitHub, debido a su equilibrio entre funcionalidad y accesibilidad. GitHub ofrece un plan gratuito que cubre las necesidades básicas del control de versiones, tanto para repositorios públicos como privados. Su interfaz intuitiva facilita su adopción, incluso para aquellos con poca experiencia en Git, y permite una gestión eficiente de proyectos.



**Imagen: Arquitectura de trabajo de GitHub**

1. **Actividades de la GCS**
   1. **Identificación**
      1. **Definición de la estructura de las librerías**

****

* + 1. **Definición de la línea base**

**Nombre del Proyecto:** Sistema de Gestión de Citas de Salud Mental (SGCSM)

| **Hito** | **Nombre** | **Nomenclatura** |
| --- | --- | --- |
| Hito 1: 29 /09/24 | Línea Base 1 | SGCSM-PC.docx  SGSCM-CP.xlsx  SGCSM-DN.docx  SGSCM-RFNF.docx  SGCSM-DDCU.docx  SGCSM-DECU001.docx  SGCSM-DECU002.docx  SGCSM-DECU003.docx  SGCSM-DECU004.docx  SGCSM-DECU005.docx  SGCSM-DECU006.docx  SGCSM-DECU007.docx  SGCSM-DECU008.docx  SGCSM-DECU009.docx  SGCSM-DECU010.docx  SGCSM-DECU011.docx  SGCSM-DECU012.docx  SGCSM-DECU013.docx |
| Hito 2:  24/10/24 | Línea Base 2 | SGCSM-DMBD.docx  SGCSM-DAS.docx  SGCSM-DDW.docx  SGCSM-DDUI.docx |
| Hito 3:  14/11/24 | Línea Base 3 | SGCSM-DDCP.docx  SGCSM-DICC.docx  SGCSM-DECP.xslx  SGCSM-MU.docx  SGCSM-MDS.docx  \\Cliente  \\Servidor |

* + 1. **Definición de nomenclaturas de ítems**
* **Caso 1: Para ítems que pertenecen a un proyecto**

*Acrónimo del proyecto + “-” + Acrónimo del ítem + “.” + extensión*

* **Caso 2: Para ítems que pertenecen a un proyecto, los nombres se repiten y los ítems se prestan para una enumeración**

*Acrónimo del proyecto + “-” + Acrónimo del ítem + etiqueta numerada + “.” + extensión*

* **Caso 3: Para ítems que pertenecen a un proyecto y los nombres se repiten**

*Acrónimo del proyecto + “-” + Acrónimo del ítem + Primeras 3 letras de la última palabra del acrónimo + “.” extensión*

* **Caso 4: Para ítems generales**

*Acrónimo del ítem + “.” + extensión*

* **Caso 5: Para ítems generales y los nombres se repiten**

*Acrónimo del ítem + “.” + Primeras 3 letras de la última palabra del acrónimo + “.” extensión*

* + 1. **Lista de ítem con la nomenclatura**

**Nombre del Proyecto:** Sistema de Gestión de Citas de Salud Mental (SGCSM)

| **Tipo** | **Nombre** | **Nomenclatura** | **Extensión** | **Proyecto** |
| --- | --- | --- | --- | --- |
| E | Plan de la Gestión de la Configuración | PGC | .docx | - |
| E | Project Charter | SGCSM-PC | .docx | SGCSM |
| E | Cronograma de Proyecto | SGCSM-CP | .xslx | SGCSM |
| E | Documento de Negocio | SGCSM-DN | .docx | SGCSM |
| E | Requisitos funcionales y no funcionales | SGCSM-RFNF | .docx | SGCSM |
| E | Documento de Diagrama de casos de uso | SGCSM-DDCU | .docx | SGCSM |
| E | Documento de Especificación de casos de uso 1 | SGCSM-DECU001 | .docx | SGCSM |
| E | Documento de Especificación de casos de uso 2 | SGCSM-DECU002 | .docx | SGCSM |
| E | Documento de Especificación de casos de uso 3 | SGCSM-DECU003 | .docx | SGCSM |
| E | Documento de Especificación de casos de uso 4 | SGCSM-DECU004 | .docx | SGCSM |
| E | Documento de Especificación de casos de uso 5 | SGCSM-DECU005 | .docx | SGCSM |
| E | Documento de Especificación de casos de uso 6 | SGCSM-DECU006 | .docx | SGCSM |
| E | Documento de Especificación de casos de uso 7 | SGCSM-DECU007 | .docx | SGCSM |
| E | Documento de especificación de casos de uso 8 | SGCSM-DECU008 | .docx | SGCSM |
| E | Documento de especificación de casos de uso 9 | SGCSM-DECU009 | .docx | SGCSM |
| E | Documento de especificación de casos de uso 10 | SGCSM-DECU010 | .docx | SGCSM |
| E | Documento de especificación de casos de uso 11 | SGCSM-DECU011 | .docx | SGCSM |
| E | Documento de especificación de casos de uso 12 | SGCSM-DECU012 | .docx | SGCSM |
| E | Documento de Modelo de Base de datos | SGCSM-DMBD | .docx | SGCSM |
| E | Documento de Arquitectura de Software | SGCSM-DAS | .docx | SGCSM |
| E | Documento de Diseños de Wireframes | SGCSM-DDW | .docx | SGCSM |
| E | Documento de Diseños de UI | SGCSM-DDUI | .docx | SGCSM |
| E | Documento de diseño de casos de prueba | SGCSM-DDCP | .docx | SGCSM |
| E | Documento de ejecución de casos de prueba | SGCSM-DECP | .xslx | SGCSM |
| E | Documento de Informe de Calidad de Codigo | SGCSM - DICC | .docx | SGCSM |
| F | Código fuente del lado del cliente | \\Client | .html, .css y .js | SGCSM |
| F | Código fuente del lado del servidor | \\Server | .js | SGCSM |
| S | Manual de despliegue y soporte | SGCSM-MDS | .docx | SGCSM |
| S | Documento de manual de usuario | SGCSM-MU | .docx | SGCSM |

* 1. **Control de GCS**
     1. **Fases del proceso de gestión de cambios**

Se ha considerado un total de 8 fases:

1. **Recepción y análisis de la solicitud:** Se recibe la solicitud de cambio y se procede a evaluar su viabilidad y requisitos.
2. **Clasificación del cambio:** Tras revisar la solicitud, se clasifica en una de las tres categorías posibles: estándar, urgente o preaprobado.
3. **Evaluación del impacto y riesgos:** Un Comité de Control de Cambios (CCC) es asignado para evaluar los posibles riesgos y el impacto que el cambio tendrá sobre el proyecto.
4. **Aprobación del cambio:** El CCC toma la decisión de aprobar o rechazar la solicitud.
5. **Planificación y calendarización:** Se establecen las fechas para implementar y verificar el cambio, y se comunica a todas las personas involucradas sobre la implementación de este cambio.
6. **Implementación:** Se asignan tareas a los miembros del equipo, se garantiza que se siga un plan adecuado y se lleva a cabo el cambio.
7. **Verificación de la implementación:** Se evalúan los efectos del cambio en el proyecto y se mide la satisfacción de los usuarios y stakeholders.
8. **Cierre:** Una vez finalizada la verificación de la implementación del cambio, la solicitud se considera concluida.

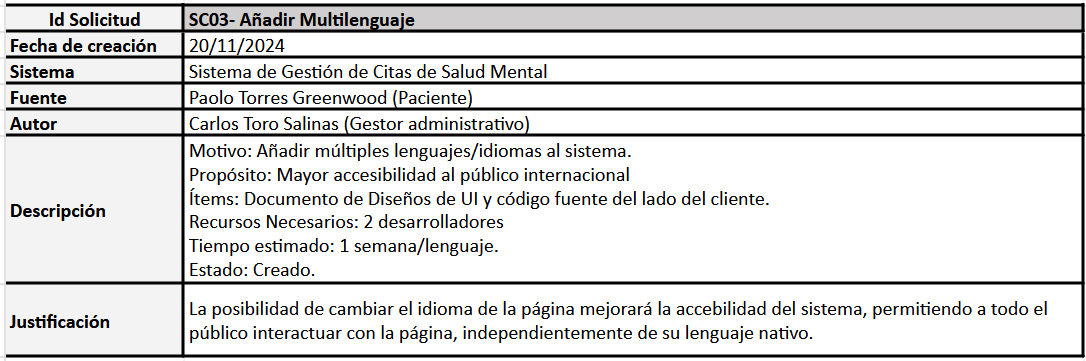
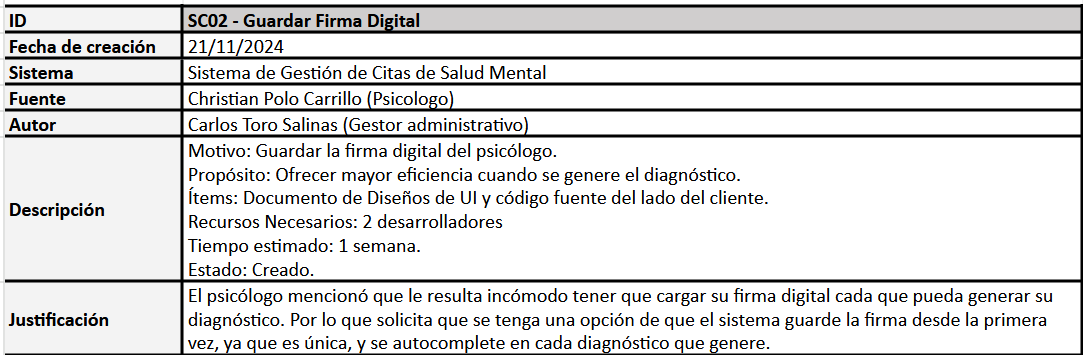
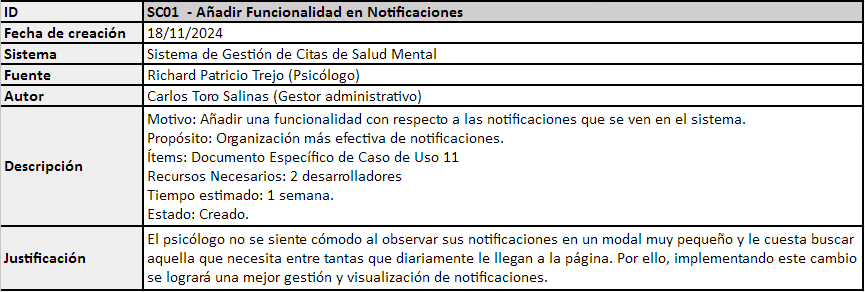
****

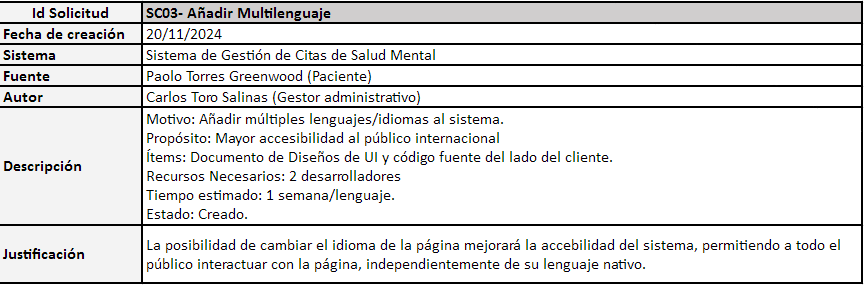
* + 1. **Lista de estados de las solicitudes de cambio**

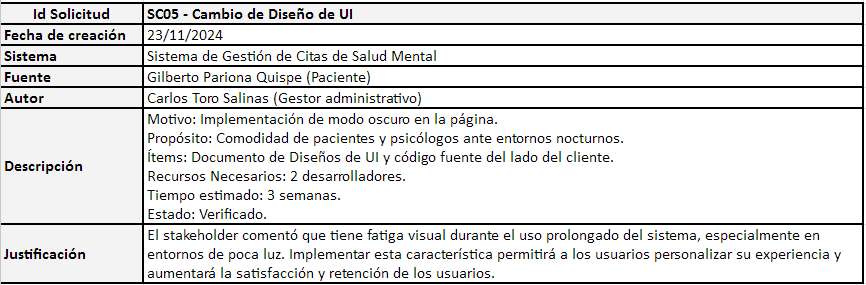
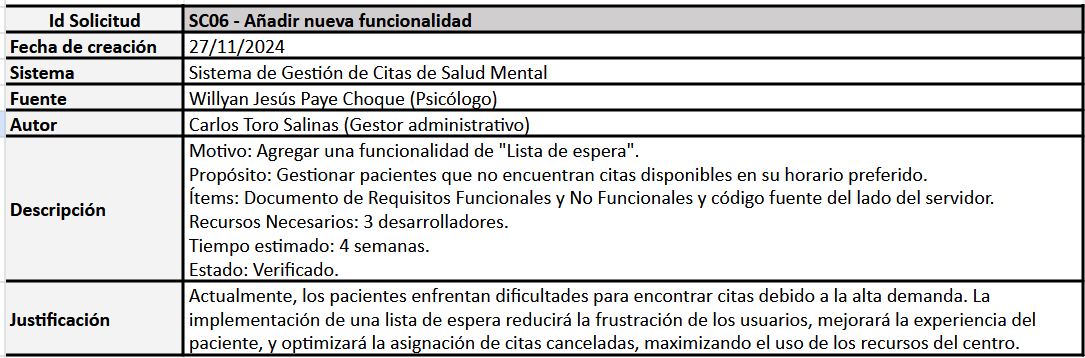
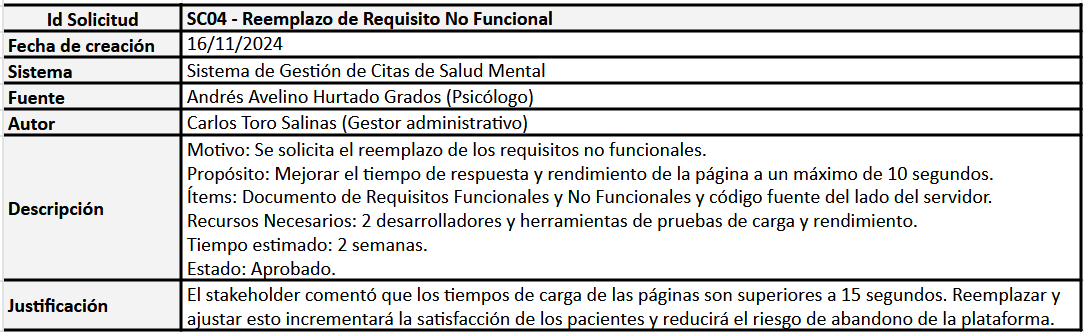
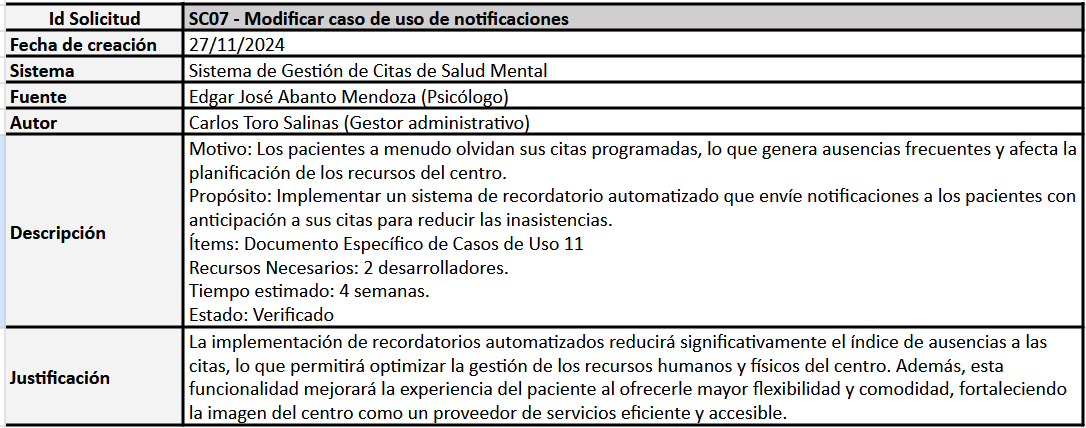
| **Estado** | **Descripción** | **Etapa** |
| --- | --- | --- |
| Creado | Estado designado a aquellas peticiones de cambios que han sido creados, pero que todavía no han sido enviados para su revisión preliminar. | 0 |
| Recibido | Estado para aquellas peticiones de cambios que después de enviados han sido recibidos para su análisis preliminar. | 1 |
| Rechazado | Estado para aquellas peticiones de cambios que después de enviados y después de su análisis preliminar, fueron rechazados por no cumplir con los parámetros necesarios o su formato es incorrecto. | 1 |
| En clasificación | Estado para aquellas peticiones de cambio que después de ser recibidos y analizados, se encuentran clasificando su tipo de cambio | 2 |
| En evaluación | Estado para aquellas peticiones de cambios que después de enviados y después de su análisis preliminar, fueron aceptados para su evaluación de impacto y riesgos. | 3 |
| Aprobado | Estado para aquellas peticiones de cambios que después de evaluar su impacto y los riesgos que contraen, fueron aprobados para su ejecución. | 4 |
| Desaprobado | Estado para aquellas peticiones de cambios que después de evaluar su impacto y los riesgos que contraen, fueron desaprobados y descartados. | 4 |
| En planificación | Estado para aquellas peticiones de cambios que después de aprobarse se encuentran planificando y gestionando su calendario. | 5 |
| En implementación | Estado para aquellas peticiones de cambios que después de planificarse pasaron a su ejecutarse e implementarse su cambio. | 6 |
| Verificado | Estado para aquellas peticiones de cambios que han sido implementadas y verificadas, si sus cambios fueron hechos según lo planificado. | 7 |
| En corrección | Estado que es la implementación de las correcciones que se indicaron al verificarse y que volverá al estado de verificación. | 7 |
| Cerrado | Estado para aquellas peticiones de cambio que culminaron su ciclo | 8 |

* + 1. **Solicitudes de Cambio**

En las siguientes tablas se muestran las solicitudes que se hicieron por parte de los usuarios del sistema, como los pacientes y psicólogos, para realizar añadir o hacer cambios de ciertas funcionalidades que se tienen.







* + 1. **Seguimiento de las solicitudes de cambio**

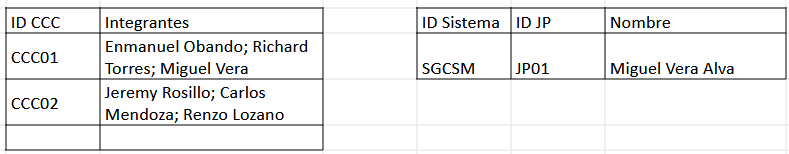
Para realizar el seguimiento de las solicitudes de cambio se ha considerado los siguientes campos:

* **ID**: Identificador de la solicitud de cambio.
* **Fecha de creación**: Fecha de creación de la solicitud de cambio.
* **ID del sistema**: Identificador del sistema al que pertenece la solicitud de cambio.
* **Fuente**: Stakeholder o usuario que reporta la solicitud de cambio.
* **Autor**: Persona encargada de enviar la solicitud de cambio.
* **Descripción**: Descripción de la solicitud de cambio.
* **Justificación**: Justificación de la solicitud de cambio.
* **Tipo de petición**: Clasificación que se le da a la solicitud de cambio. Puede ser: estándar, urgente y pre-aprobado.
* **Estado**: Estado actual de la solicitud de cambio.
* **Fecha de recepción**: Fecha de recepción de la solicitud para su análisis preliminar.
* **Fecha de evaluación**: Fecha donde se evalúan los impactos y riesgos de la solicitud de cambio.
* **Fecha de aprobación**: Fecha de aprobación para la ejecución de la solicitud de cambio.
* Fecha de planificación: Fecha donde se planea el cronograma de la ejecución de la solicitud de cambio.
* **Fecha de implementación**: Fecha de inicio de la implementación de cambios de la solicitud.
* **Fecha de verificación**: Fecha de verificación por parte de los stakeholders de la implementación de la solicitud.
* **Fecha de corrección**: Fecha de corrección, si hubiese una observación por parte de los stakeholders.
* **Fecha de cierre**: Fecha del fin del proceso de gestión de cambios para la solicitud.
* **ID CCC**: Identificador del equipo de Comité de Control de Cambios.
* **ID Jefe de Proyecto**: Identificador del jefe del proyecto.
* **Plazo**: Tiempo aproximado de la implementación de la solicitud de cambio.
* **Prioridad**: Nivel de prioridad de la solicitud. Puede ser: bajo, medio y alto.

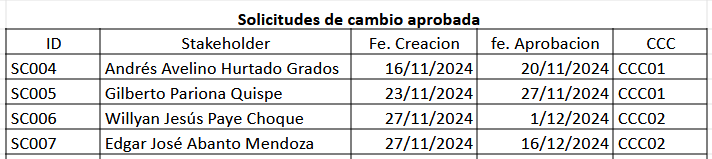
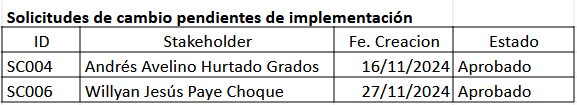


* + 1. **Comité de Control de Cambios**

Se divide en 2 comités, el primero formado por los estudiantes Obando, Torres y Vera, mientras que el segundo está formado por Rosillo, Mendoza y Lozano. Siendo Miguel Vera Alva nuestro

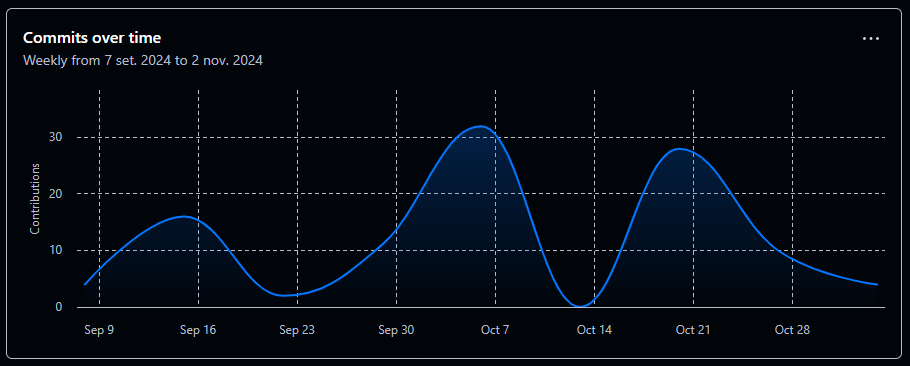


* + 1. **Reportes de las solicitudes de cambio**



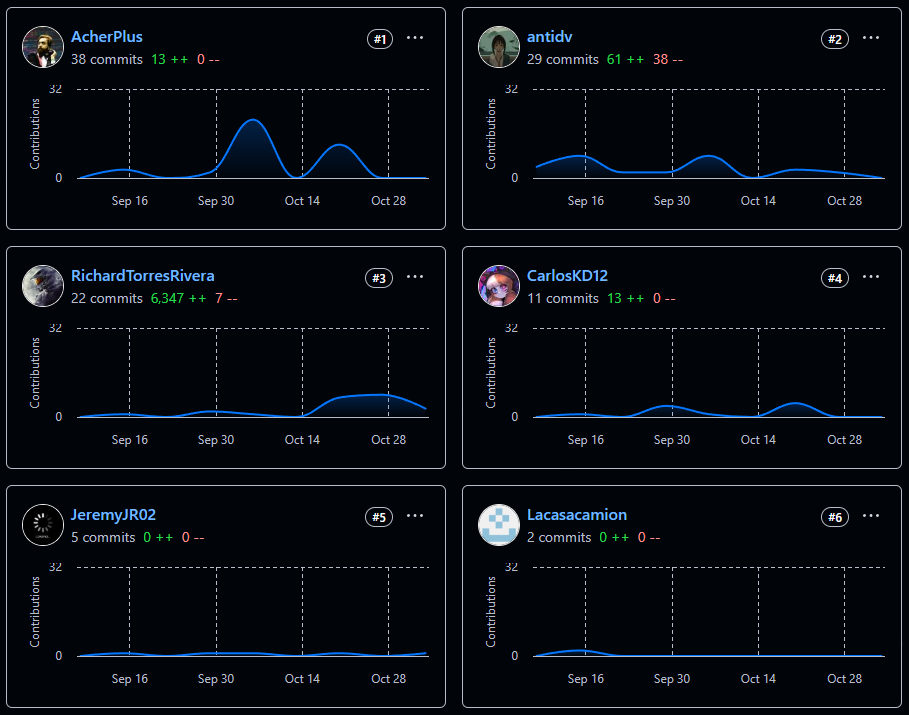
* 1. **Estado de la GCS**
     1. **Gráfica General**

Esta gráfica muestra los Commits a lo largo del tiempo en nuestro repositorio de código, abarcando el periodo desde el 7 de septiembre de 2024 hasta el 2 de noviembre de 2024. La gráfica indica el número de Commits que se realizaron semanalmente y los picos señalan los días que hubo más interacción de Commits, que cuadran con las semanas de finalizar los hitos.



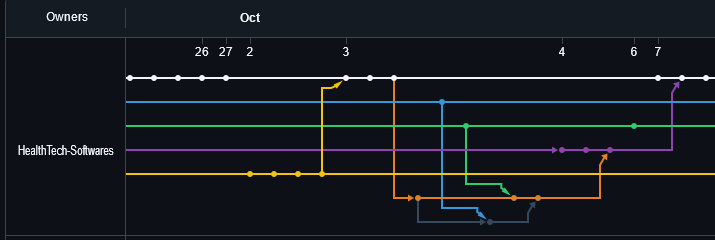
* + 1. **Gráfica de Contribuciones**

Esta captura muestra un desglose de las contribuciones de nuestro equipo en el repositorio, en el mismo periodo de tiempo que la gráfica anterior. Cada cuadro representa las contribuciones individuales de cada miembro del equipo de desarrollo. Aquellas personas que tienen pocas contribuciones con pocas líneas modificadas, significa que han trabajo en documentos y no código; además de no tener asignados muchos ítems para los primeros hitos.

****

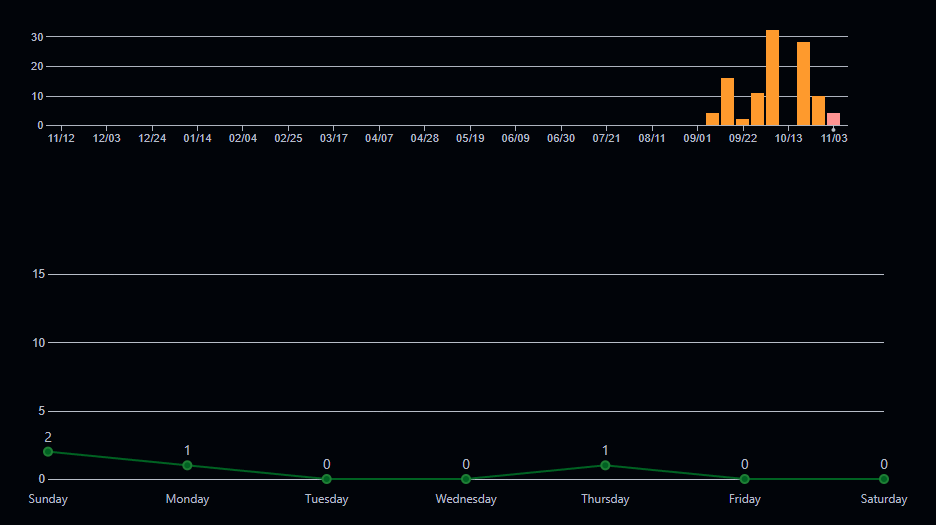
* + 1. **Gráfica Network**

Esta gráfica representa la actividad de las ramas de cada uno de los integrantes del proyecto. Los puntos indican los commits realizados y las flechas, los merges.



* + 1. **Gráfica de Commits**

Esta imagen muestra que la mayor parte del trabajo en el proyecto se concentró en octubre y se espera aumente en noviembre en la última fase del proyecto (Hito 3). Durante estos meses, hubo un periodo de mucha actividad, especialmente en una semana donde el equipo hizo la mayoría de los cambios.



* 1. **Auditoría de la GCS**
     1. **Auditoría Física**

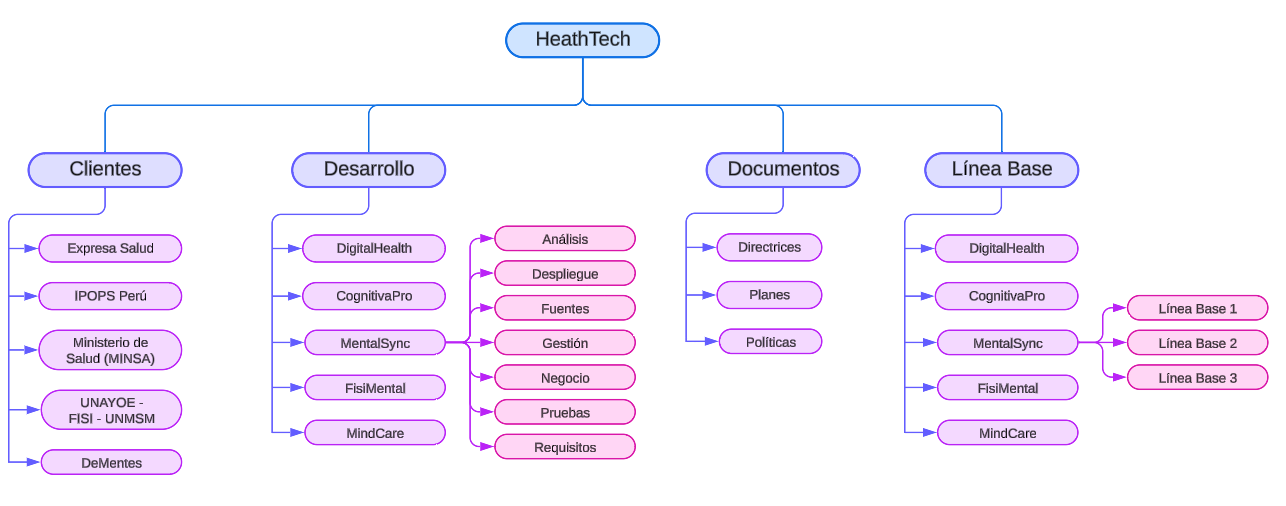
| Ítem | Nombre del Documento | Fecha de Revisión | Auditor | Estado |
| --- | --- | --- | --- | --- |
| 1 | SGCSM-DDCP | 12/11/2024 | Vera Alva, Miguel Angel | Conforme |
| 2 | SGCSM-DECP | 12/11/2024 | Vera Alva, Miguel Angel | Conforme |
| 3 | SGCSM-MU | 12/11/2024 | Vera Alva, Miguel Angel | Conforme |
| 4 | SGCSM-MDS | 14/11/2024 | Vera Alva, Miguel Angel | Conforme |
| 5 | Código Fuente Cliente | 14/11/2024 | Vera Alva, Miguel Angel | Conforme |
| 6 | Código Fuente Servidor | 14/11/2024 | Vera Alva, Miguel Angel | Conforme |

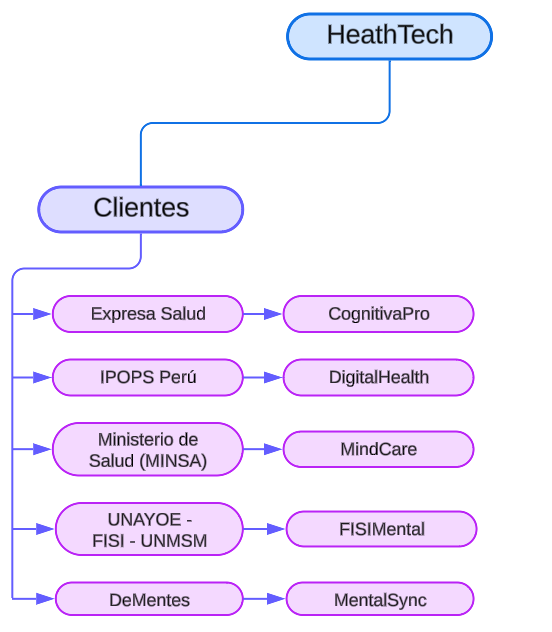
* + 1. **Auditoria Funcional**

| **Caso de Uso** | **Fecha de Auditoría** | **Auditor** | **Estado** |
| --- | --- | --- | --- |
| **CU001 - Iniciar Sesión** | 25/11/2024 | Mendoza Gómez, Carlos Daniel | Aprobado |
| **CU002 - Ver psicólogos** | 25/11/2024 | Mendoza Gómez, Carlos Daniel | Aprobado |
| **CU003 - Reservar Cita** | 25/11/2024 | Rosillo Ramirez Jeremy Andrew | Aprobado |
| **CU004 -Realizar Pago** | 25/11/2024 | Obando Salinas Enmanuel Jose | Aprobado |
| **CU005 - Realizar encuesta** | 25/11/2024 | Torres Rivera Richard Maycol | Aprobado |
| **CU006 - Ver Citas** | 25/11/2024 | Vera Alva Miguel Angel | Aprobado |
| **CU007 - Asistir cita online** | 25/11/2024 | Mendoza Gómez, Carlos Daniel | Aprobado |
| **CU008 - Generar Diagnóstico** | 25/11/2024 | Vera Alva Miguel Angel | Aprobado |
| **CU009 - Gestionar Historial de citas** | 25/11/2024 | Mendoza Gómez, Carlos Daniel | Aprobado |
| **CU010 - Modificar Datos** | 25/11/2024 | Torres Rivera Richard Maycol | Aprobado |
| **CU011 - Ver notificación** | 25/11/2024 | Obando Salinas Enmanuel Jose | Aprobado |
| **CU012 - Gestionar Perfil** | 25/11/2024 | Vera Alva Miguel Angel | Aprobado |
| **CU013 - Gestionar psicólogo** | 25/11/2024 | Rosillo Ramirez Jeremy Andrew | Aprobado |

* 1. **Gestión de Release**

La estrategia de releases para el proyecto **MentalSync** está diseñada para adaptarse a las necesidades específicas de cada cliente. En el repositorio, cada cliente tiene su propia carpeta dentro de la sección **Clientes**, donde se organiza y almacena el entregable correspondiente al software solicitado. Esta estructura permite gestionar de manera eficiente las personalizaciones, configuraciones y versiones específicas de cada implementación, asegurando que cada cliente reciba solo los componentes relevantes para su entorno.

****

****

* 1. **Entrega de Software**
     1. **Preparación previa**
* Primero revisamos que todos los casos de uso clave (login, gestión de citas, etc.) estén aprobados según el PGC y confirmamos que tanto el cliente como el servidor pasaron las auditorías funcionales y físicas.
* Revisamos la rama principal del repositorio (main o master) asegurando que solo contiene código aprobado.
* Creamos un release tag en GitHub para identificar el estado exacto del código que pasará a producción (ejemplo: v1.0.0).
* Por último, se realizan pruebas en un entorno de staging (copia exacta del entorno de producción) para validar el despliegue.

#### **Configuración del Entorno de Producción**

* Por parte de la infraestructura de despliegue, contamos con Render, y para ello configuramos el API en el Backend con [**https://healthtech-f0yh.onrender.com**](https://healthtech-f0yh.onrender.com) y verificamos que el frontend está correctamente vinculado.
* Verificamos la variables de entorno en en el Frontend en el archivo .env: **VITE\_BACKEND\_URL=**[**https://healthtech-f0yh.onrender.com**](https://healthtech-f0yh.onrender.com)**/api**
* Nos aseguramos de que las claves de autenticación JWT, credenciales de base de datos, y demás secretos están protegidos y no comprometidos.

#### **Despliegue a Producción**

* Generamos el build del frontend ejecutando el comando en bash: **npm run build**
* Subimos los archivos estáticos generados (build/) al servidor de producción.
* Confirmamos que la API y sus rutas (por ejemplo, /api/psicologos) funcionan correctamente con herramientas como Postman.
* Por parte de la base de datos, verificamos las migraciones necesarias y realizamos el backup antes de ejecutar cualquier cambio.
* Validamos las funcionalidades principales (login, creación de citas, etc.) directamente en producción.
* Presentar la aplicación al cliente para su revisión inicial.