Melody Labs

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

Integrantes

* Sarmiento Quispe, Sofía Fernanda
* Chipana Arellano, Franz Esteban
* Contreras Quispe, Harumi Nayeli
* Aranda Vega, Joar Jesús
* Flores Gómez José Benjamín
* Parrales Torres, José Luis
* Palacios Aguilar Adrian Alessandro

Versión 1.0

# 1. Introducción

## 1.1 Situación de la empresa y problemática

### 1.1.1 Situación de la empresa

MelodyLabs es una empresa en crecimiento que se especializa en la creación y desarrollo de software innovador en el ámbito de la tecnología musical. A lo largo de los años, ha destacado por la creación de soluciones personalizadas, tanto para empresas como para usuarios individuales, enfocadas en la producción, organización y distribución de contenido musical. Actualmente, la empresa ha expandido su línea de productos y se encuentra desarrollando un ambicioso proyecto denominado TheVault, un sistema avanzado de reproducción de música diseñado para almacenar, gestionar y ofrecer acceso a grandes catálogos de música de manera eficiente y fluida.

### 1.1.2 Problemática

* Sobreescritura de partes del código

Actualmente, varios desarrolladores trabajan de manera simultánea en diferentes módulos de TheVault, y debido a la falta de un sistema de control de versiones robusto y centralizado, se han presentado numerosos casos de sobrescritura involuntaria de código.

* Falta de actualizaciones consistentes entre los equipos

Otro grave problema es la falta de sincronización entre los equipos de desarrollo. Algunos equipos o individuos no cuentan con la versión más actualizada del código fuente al comenzar su trabajo, lo que provoca que implementen sus cambios sobre una versión antigua o incompleta del software.

* Ausencia de historial claro de cambios

La falta de un sistema formalizado para la gestión de versiones ha provocado que el equipo de desarrollo no tenga acceso a un historial claro de los cambios realizados. Esto genera dificultades para rastrear quién introdujo una modificación específica o cuándo y por qué se hizo, lo que complica la identificación de errores recientes y la toma de decisiones informadas.

## 1.2 El objetivo del plan

Se debe implementar una herramienta que organice y gestione bien las versiones de código ( como git) para evitar que el equipo de desarrolladores sobreescriben código, asegurarse que todos trabajen con la versión más actualizada y llevar un registro de todos los cambios.

# 2. Gestión de Configuración de Software

## 2.1 Roles y responsabilidades

### Project Manager

**Responsabilidad**: Establecer las metas generales del proyecto, coordinar al equipo y asegurarse de que se sigan las normas del control de versiones. El Project Manager también tiene la responsabilidad de velar por el cumplimiento de los plazos y la correcta entrega del producto.

**Miembros involucrados**:

**Sofía Fernanda Sarmiento Quispe**: Jefe de proyecto, Programador Back, Diseñador UX.

### Gestor de Configuración

### **Responsabilidad**: Definir las políticas para la creación, fusión y revisión de ramas de código. Asegurarse de que el equipo siga las buenas prácticas de gestión de versiones y control de cambios.

**Miembros involucrados**:

**Adrian Alessandro Palacios Aguilar**: Programador Back, Tester.

**José Luis Parrales Torres**: Programador Back, Tester, DBA.

### Inspector de Aseguramiento de Calidad

**Responsabilidad**: Trabajar junto con el Gestor de Configuración para verificar el historial de cambios, revisar el código y asegurar que los estándares de calidad estén siendo respetados en todas las etapas del desarrollo.

**Miembros involucrados**:

**Franz Esteban Chipana Arellano**: Tester, Analista QA, Programador Front.

**José Benjamin Flores Gómez**: Programador Front, Analista QA.

### Miembros del equipo de desarrollo

### **Responsabilidad**: Seguir las políticas de control de versiones definidas por el Gestor de Configuración, asegurándose de actualizar siempre la rama antes de comenzar a trabajar. El equipo se encarga de desarrollar tanto el frontend como el backend, realizando las tareas asignadas y colaborando en la integración del software.

### **Miembros involucrados**:

**Joar Jesús Aranda Vega**: Programador Mobile, Diseñador UX, Diseñador Web.

**Harumi Nayeli Contreras Quispe**: Programador Front, Diseñador UX, Diseñador Web.

**José Luis Parrales Torres**: Programador Back, Tester, DBA.

## 2.2 Herramientas, entorno e Infraestructura

**Herramienta 1: Git**

Es un sistema de control de versiones distribuido ampliamente utilizado en la industria del software, conocido por su flexibilidad y eficiencia. Permite tanto trabajo local como colaboración distribuida.

**Herramienta 2: Mercurial**

Similar a Git en ser distribuido, Mercurial es conocido por su facilidad de uso y menor curva de aprendizaje, aunque no es tan flexible como Git en algunos aspectos avanzados.

**Herramienta 3: Subversion (SVN)**

A diferencia de Git y Mercurial, SVN es un sistema de control de versiones centralizado. Es más fácil de entender para los equipos que prefieren un enfoque de servidor único, pero con menos flexibilidad en cuanto al trabajo distribuido.

| **CRITERIOS** | **HERRAMIENTA 1** | **HERRAMIENTA 2** | **HERRAMIENTA 3** |
| --- | --- | --- | --- |
| Gratuito | Sí | Sí | Sí |
| Curva de aprendizaje | Moderada | Baja | Alta |
| Centralizado | No | No | Sí |
| Distribuido | Sí | Sí | No |

**Herramienta seleccionada: Git**

Los beneficios que hemos podido destacar para seleccionar esta herramienta:

* Distribución y Flexibilidad
* Gestión Eficiente de Ramas
* Fusión y Colaboración
* Desempeño
* Amplia Adopción y Soporte
* Control Fino y Registro Detallado
* Portabilidad y Compatibilidad

**Tecnologías del Proyecto:**

**Frontend**: React, Next.js

React es una biblioteca de JavaScript utilizada para construir interfaces de usuario, mientras que Next.js es un framework que facilita la creación de aplicaciones React con características como renderizado del lado del servidor (SSR) y generación de sitios estáticos.

**Backend y Base de Datos**: Supabase

Supabase es una alternativa de código abierto a Firebase. Ofrece una plataforma para crear aplicaciones con autenticación, bases de datos, almacenamiento de archivos, y funciones en el backend sin necesidad de gestionar la infraestructura.

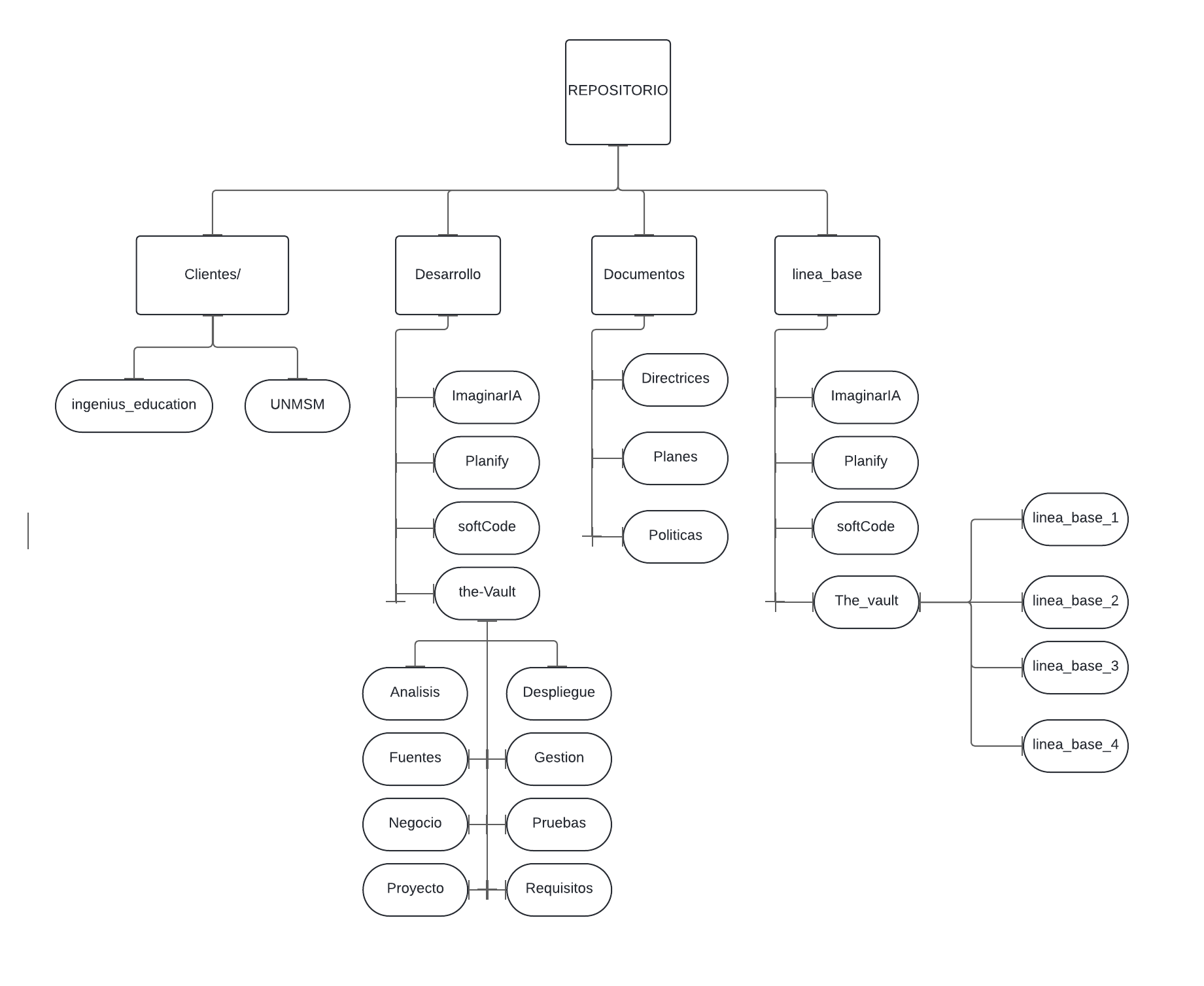
**Infraestructura:**

**Entorno de Desarrollo**: El equipo trabaja en un entorno basado en la nube para facilitar la colaboración y el acceso remoto, con servidores gestionados a través de plataformas como GitHub y Supabase para bases de datos. La infraestructura se adapta a las necesidades de escalabilidad y seguridad del proyecto.

# 3. Actividades de la GCS

## 3.1 Identificación

### 3.1.1 Definición de la estructura de las librerías



### 3.1.2 Definición de Líneas Base

* Para the\_vault

| **Hito** | **Nombre de línea base** | **Items (nomenclatura)** |
| --- | --- | --- |
| Hito 1: 26/09/24 | Línea Base 1 | Entrevista  Reunión  Historia de usuario 01 (TVT-HU01.docx)  Historia de usuario 02 (TVT-HU02.docx)  Historia de usuario 03 (TVT-HU03.docx)  Historia de usuario 04 (TVT-HU04.docx)  Historia de usuario 05 (TVT-HU05.docx)  Historia de usuario 06 (TVT-HU06.docx)  Historia de usuario 07 (TVT-HU07.docx)  Estudio de Viabilidad (TVT-EV.docx)  Documento de análisis (TVT-DA.docx)  Cronograma de sprints (TVT-SP.xlsx) |
| Hito 2: 24/10/24 | Línea Base 2 | Estructura del proyecto  Interfaz de usuario (TVT-SKETCH.pdf)  Diagrama ER (TVT-BD.pdf)  Implementación de Supabase  Desarrollo del código fuente (Carpeta /components)  Desarrollo del código fuente, update de las otras (Carpeta /app)  Archivos relacionados a los datos (types.ts)  Desarrollo del código fuente, update de las otras (Carpeta /hooks) |
| Hito 3: 14/11/24 | Línea Base 3 | Desarrollo del código fuente, update de las otras (Carpeta /api)  Datos para las tablas  Documento de la planificación para la prueba (TVT-PDP.docx) |
| Hito 4: 21/11/24 | Línea Base 4 | Testing de la página web, update de las carpetas con alguna falla (Carpeta /api /hooks /app /components)  Documento de los resultados y las modificaciones (TVT-RDP.docx)  Despliegue de la página web (thevault.com) |

* Para el proyecto “ImaginarIA”

| **Hito** | **Nombre de línea base** |
| --- | --- |
| Hito 1 - Análisis | Línea Base 1 |
| Hito 2 - Diseño | Línea Base 2 |
| Hito 3 - Desarrollo y Despliegue | Línea Base 3 |

* Para el proyecto “Planify”

| **Hito** | **Nombre de línea base** |
| --- | --- |
| Hito 1 - Análisis | Línea Base 1 |
| Hito 2 - Diseño | Línea Base 2 |
| Hito 3 - Desarrollo y Despliegue | Línea Base 3 |

* Para el proyecto “Softcode”

| **Hito** | **Nombre de línea base** |
| --- | --- |
| Hito 1 - Análisis | Línea Base 1 |
| Hito 2 - Diseño | Línea Base 2 |
| Hito 3 - Desarrollo y Despliegue | Línea Base 3 |

### 3.1.3 Lista de clasificación de Items

| **Tipo**  (E= Evolución  F= Fuente  S= Soporte) | **Nombre del Item** | **Nomenclatura** | **Extensión** | **Proyecto** |
| --- | --- | --- | --- | --- |
| E | Documento de especificación de requisitos | TVT-DER | docx | The Vault |
| F | Código fuente del Frontend | - | js | The Vault |
| S | Datos iniciales | TVT-DI | csv | The Vault |
| E | Documento de Análisis | TVT-DA | docx | The Vault |
| E | Esquema de la base de datos | TVT-EBD | pdf | The Vault |
| E | Informe de Avance Hito 1 | TVT-IAH01 | docx | The Vault |
| S | React 18.0 | - | tsx | The Vault |
| E | Historia de Usuario 01 | TVT-HU01 | docx | The Vault |
| E | Interfaz Diseñada para el Usuario | TVT-IDU | pdf | The Vault |
| E | Plan Documentado para pruebas | TVT-PDP | docx | The Vault |
| E | Estudio de Viabilidad | TVT-EV | docx | The Vault |
| E | Resultados Documentados de las pruebas | TVT-RDP | docx | The Vault |
| E | Sprints del proyecto | TVT-SP | xlsx | The Vault |
| F | Supabase (backend) | - | api | The Vault |
| S | Conexión con la base de datos | types\_db | ts | The Vault |

### 3.1.4 Definición de la Nomenclatura de ítems

**Caso 1: Ítems que pertenecen a un proyecto específico**

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

Proyecto: TheVault (TVT)

Ítem: Documento de Análisis (DA)

Nomenclatura: TVT-DA.docx

**Caso 2: Ítems con el mismo nombre con contenido diferente**

| Acrónimo del proyecto + “-” + Acrónimo del ítem + “Número de ítem” + ”.” + extensión |
| --- |

Proyecto: TheVault (TVT)

Ítem 1: Historia de usuario 01 (HU01)

Ítem 2: Historia de usuario 02 (HU02)

Nomenclatura

* TVT-HU01.docx
* TVT-HU02.docx

**Caso 3: Ítems con el mismo acrónimo pero diferente nombre**

| Acrónimo del proyecto + “-” + Acrónimo del ítem + “2° y 3° letra de la última palabra parte del nombre” + ”.” + extensión |
| --- |

Proyecto: TheVault (TVT)

Ítem 1: Documento de análisis (DA)

Ítem 2: Documentación de auditorías (DA)

Nomenclatura

* TVT-DANA.docx
* TVT-DAUD.docx

**Caso 4: Ítems pertenecientes al código fuente**

| Nombre del acrónimo + “.” + extensión del archivo |
| --- |

Proyecto: TheVault (TVT)

Ítem: Componente “sidebar”

Nomenclatura: Sidebar.tsx

### 3.1.5 Lista de ítem con la nomenclatura

| **Nombre del Item** | **Nomenclatura** | **Extensión** | **Proyecto** |
| --- | --- | --- | --- |
| Documento de especificación de requisitos | TVT-DER | docx | The Vault |
| Código fuente del Frontend | - | tsx | The Vault |
| Código fuente de las dependencias | - | js | The Vault |
| Datos iniciales | TVT-DI | csv | The Vault |
| Documento de Análisis | TVT-DA | docx | The Vault |
| Esquema de la base de datos | TVT-EBD | pdf | The Vault |
| React 18.0 | - | ts | The Vault |
| Historia de Usuario 01 | TVT-HU01 | docx | The Vault |
| Interfaz Diseñada para el Usuario | TVT-SIU | pdf | The Vault |
| Planificación Documentada de las pruebas | TVT-PP | docx | The Vault |
| Estudio de Viabilidad | TVT-EV | docx | The Vault |
| Resultados Documentados de las pruebas | TVT-RP | docx | The Vault |
| Sprints del proyecto | TVT-SP | xlsx | The Vault |
| Supabase (backend) | - | - | The Vault |
| Conexión con la base de datos | types\_db | ts | The Vault |

## 3.2 Control de la GCS

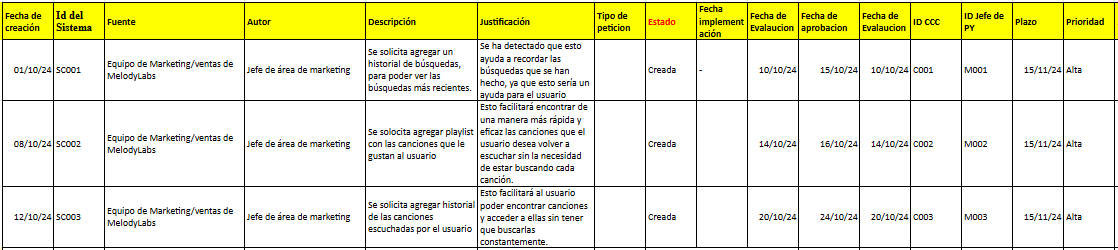
### 3.2.1 Formatos de solicitudes de Cambio

| ID Solicitud | SC001 |
| --- | --- |
| Fecha de creacion | 01 de Octubre ( antes del despliegue) |
| Sistema | The Vault |
| Fuente | Equipo de Marketing/ventas de MelodyLabs |
| Autor | Jefe de marketing |
| Descripcion | Se solicita agregar un historial de búsquedas, para poder ver las búsquedas más recientes. |
| Justificacion | Se ha detectado que esto ayuda a recordar las búsquedas que se han hecho, ya que esto sería un ayuda para el usuario |

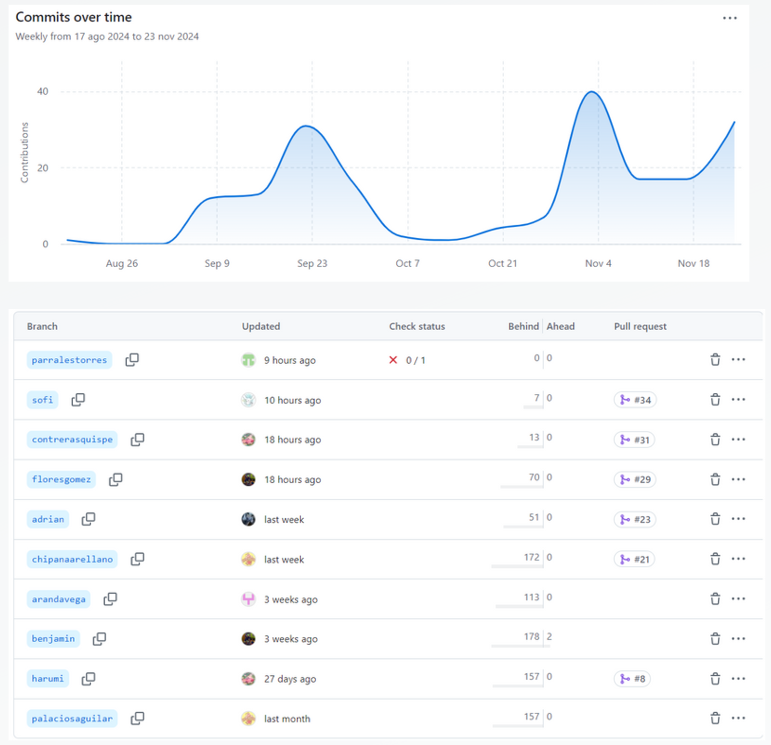
| ID Solicitud | SC002 |
| --- | --- |
| Fecha de creacion | 08 de octubre (antes del despliegue) |
| Sistema | The Vault |
| Fuente | Equipo de Marketing/ventas de MelodyLabs |
| Autor | Jefe de marketing |
| Descripcion | Se solicita agregar playlists con las canciones que le gustan al usuario. |
| Justificacion | Esto facilitará encontrar de una manera más rápida y eficaz las canciones que el usuario desea volver a escuchar sin la necesidad de estar buscando cada canción. |

| ID Solicitud | SC002 |
| --- | --- |
| Fecha de creacion | 08 de octubre (antes del despliegue) |
| Sistema | The Vault |
| Fuente | Equipo de Marketing/ventas de MelodyLabs |
| Autor | Jefe de marketing |
| Descripcion | Se solicita agregar playlists con las canciones que le gustan al usuario. |
| Justificacion | Esto facilitará encontrar de una manera más rápida y eficaz las canciones que el usuario desea volver a escuchar sin la necesidad de estar buscando cada canción. |

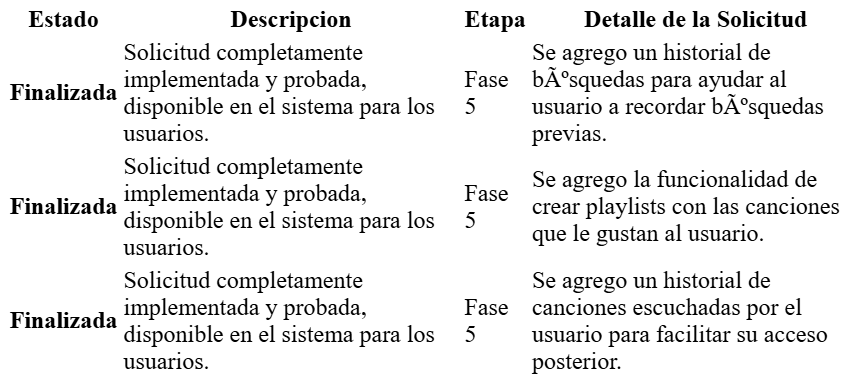
### 3.2.4. Proceso de Gestión de Cambios



## 3.3 Estado



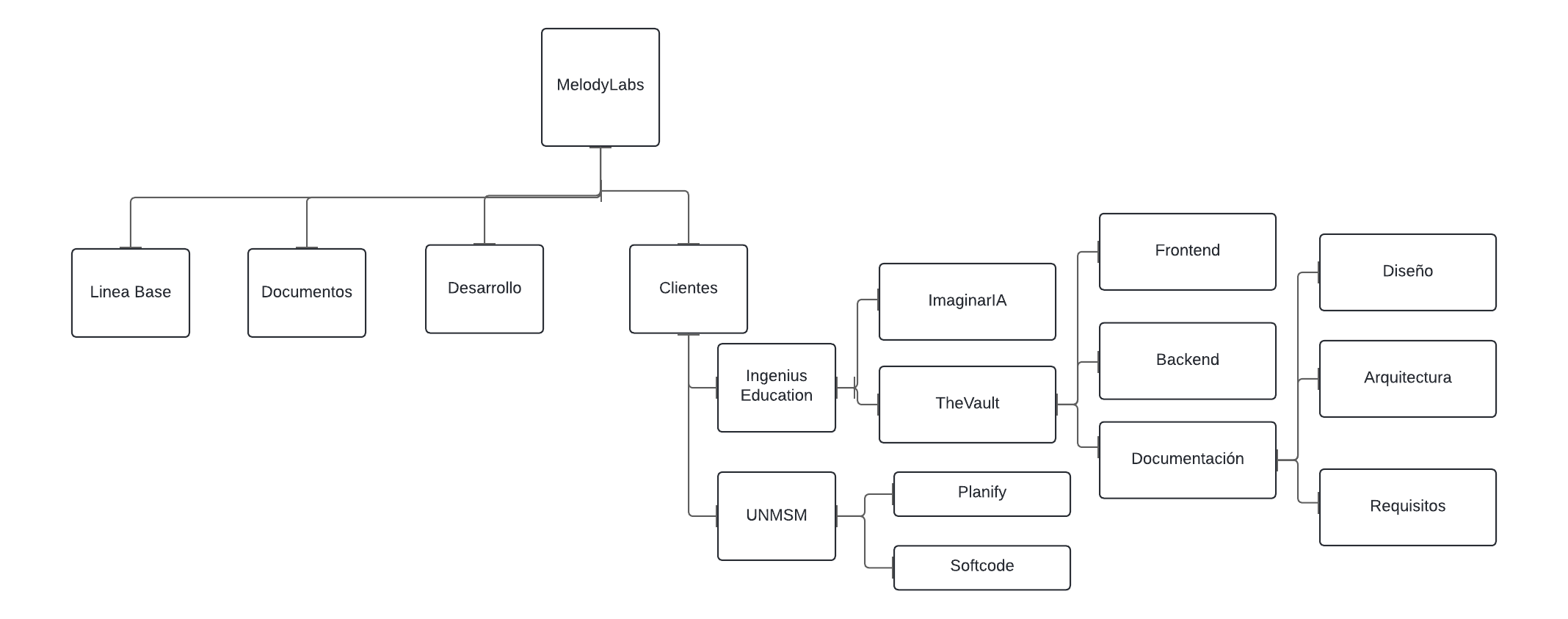
## 3.4 Auditoría



## 3.5 Entrega del software y Gestión de Release

### 3.5.1. Gestión del Release

Actualmente Melody Labs tiene como único proyecto en desarrollo a The Vault del cliente Ingenius Education, en el contrato pactado con el cliente se hizo hincapié en que al concluir el desarrollo se entregaría el código fuente tanto del Frontend y Backend, documentación y documentación técnica relacionada al software.



### 3.5.2 Entrega del software

Con respecto a la entrega del software The Vault, dentro de la carpeta clientes se encuentran la carpeta de Ingenius Education y UNMSM, dado que el proyecto es del cliente Ingenius Education, se coloca la carpeta relacionada al proyecto en esa ubicación.



Dentro de la carpeta se podrá encontrar el código fuente tanto del Frontend y Backend y la documentación técnica relacionada. Dentro de esta carpeta también se colocarán los futuros releases del proyecto.



Mencionar que dentro de las carpetas de Backend y Frontend se encuentran cada uno de los archivos necesarios para el correcto funcionamiento del software y sus dependencias. En la documentación técnica podemos encontrar una guía detallada indicando los pasos a seguir para la instalación de dependencias y software.