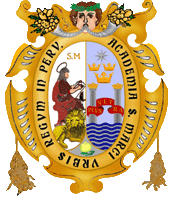
07/06/2023









**Macro System**

Plan de Gestión de la Configuración del Software

Lima, Perú.

**PLAN DE GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE**

1. **Introducción**
2. Situación de la empresa

Macro System es una consultora de software especializada en el desarrollo de soluciones personalizadas para empresas de diversos sectores. La empresa cuenta con un equipo multidisciplinario de expertos en tecnología y una amplia experie noncia en el diseño, desarrollo y mantenimiento de software de alta calidad. Posee gran presencia en el mercado contando con 8 años de antigüedad. asimismo amplia destreza en el desarrollo de software para aplicaciones empresariales, aplicaciones móviles y soluciones de automatización de procesos.

Macro System tiene una cartera variada de clientes, que incluyen empresas pequeñas, medianas y grandes corporaciones de prestigio de diferentes sectores (banca, finanzas, salud, turismo, entre otros). La empresa se enfoca en trabajar con clientes que tienen necesidades específicas de software y que requieren soluciones personalizadas para la resolución y automatización de procesos empresariales.

La empresa se dedica al desarrollo de proyectos de software de diferentes tamaños y niveles de complejidad. Los proyectos pueden incluir el desarrollo de aplicaciones web, sistemas de gestión de base de datos, entre otros. Además, todos y cada uno de estos cumplen con las necesidades específicas de los clientes logrando su satisfacción. Dentro de la totalidad de proyectos en curso por Macro System se presentan dos.

1. Desarrollo de una aplicación móvil de gestión de inventarios para una cadena de tiendas de retail: Consiste en el desarrollo de una aplicación móvil para los sistemas Android y iOS que permita a los empleados de la cadena de tiendas llevar a cabo el seguimiento a tiempo real del inventario disponible de cada uno de sus establecimientos. Asimismo, debe contar con funciones principales como realizar pedidos de productos, actualizar inventario y generar reportes de venta.
2. Sistema de Trackeo de Tickets en tiempo real del Comedor de UNMSM: Consiste en el desarrollo de una página web que permita visualizar a los estudiantes, administrativos y afines la cantidad actual de tickets del comedor. Dentro de las principales funcionalidades se resalta la del conteo de tickets con y sin numeración, tiempo promedio en culminar con los tickets e historial por día.

De acuerdo con los proyectos finalizados se presenta el más resaltante, popular y actual de la empresa.

1. Creación de un sistema de gestión de reservas para una empresa de turismo: Consiste en el desarrollo de un sistema web que permita a los clientes externos realizar reservas en líneas de diferentes paquetes de viaje y excursiones. Además, debe contar con funciones afines como creación y modificación de paquetes, asignación de guías turísticos, gestión de pagos y generación de reportes de ventas.

Actualmente, los proyectos de Macro System se encuentran en diferentes estados de proceso referente al cronograma individual de cada uno de estos para asegurar que cumplan con los plazos y requisitos establecidos por sus clientes. En el caso de “Desarrollo de una aplicación móvil de gestión de inventarios para una cadena de tiendas de retail” se encuentra en la fase de diseño para obtener como objetivos la arquitectura de la aplicación e interfaz del usuario. El proyecto “Sistema de Trackeo de Tickets en tiempo real del Comedor de UNMSM” se encuentra en la fase de planificación para la correcta definición de los requisitos y próximas tareas a realizar. El proyecto de “Creación de un sistema de gestión de reservas para una empresa de turismo” se encuentra finalizado cumpliendo con todo el ciclo de desarrollo y entrega del sistema al cliente.

Ampliando el concepto de sistemas de producción en la empresa, se logra identificar a tres proyectos, de los cuales dos han sido descritos anteriormente. Macro System gestiona sus proyectos finalizados para evitar complicaciones a futuro con los proyectos en curso, dado que de esta manera el trabajo simultáneo de los proyectos se realizan óptimamente. Además, es importante tener en cuenta que cada sistema en producción debe ser tratado como un objeto independiente y único en la gestión de la configuración de software.

Macro System ha experimentado algunos problemas relacionados con la gestión de la configuración de software en sus proyectos. Estos son el control de versiones (pérdida y sobreescritura de código) e integración de diferentes componentes de software (correcciones de implementación tardías). Este causante provocó deficiencias en la documentación, incompatibilidad de versiones, recrear el trabajo realizado anteriormente y aumento de costos. En específico, el proyecto “Creación de un sistema de gestión de reservas para una empresa de turismo” desarrolló deficiencias en la fase de diseño con respecto al control de versiones por una mala gestión por parte de un personal nuevo, esto provocó retrasos en fechas de entrega y recreación de todo lo avanzado hasta aquel momento. Asimismo, al final del proyecto se encontraron fallas en la implementación, con ello bajó la calidad del producto y se tuvo que realizar correcciones de última hora. De acuerdo a los proyectos actuales y en curso, presentan pocas deficiencias en el control de versiones, pero aún así es significante las consecuencias de este fallo.

A continuación, se describen las principales funciones y tareas de cada miembro que compone el equipo.

* Desarrollador Backend: Responsable de la programación del lado del servidor de la aplicación, lo que implica trabajar con bases de datos, servidores y la lógica del negocio.
* Administrador de base de datos (DBA): Responsable de diseñar, implementar y mantener las bases de datos necesarias para el correcto funcionamiento de la aplicación.
* Scrum Master: Encargado de liderar el equipo de desarrollo de software, asegurando que se siga la metodología Agile y se cumplan los objetivos establecidos en el proyecto.
* Diseñador UX: Responsable de diseñar la experiencia de usuario de la aplicación, trabajando en la interfaz gráfica y la usabilidad.
* Desarrollador Frontend: Responsable de programar el lado del cliente de la aplicación, lo que implica trabajar con HTML, CSS y JavaScript para crear la interfaz de usuario.
* Jefe de proyecto: Responsable de liderar el proyecto en su totalidad, estableciendo los objetivos, coordinando el equipo y asegurando que se cumplan los plazos.
* Arquitecto de software: Encargado de diseñar la estructura del software, estableciendo las bases para el desarrollo del mismo y asegurando que sea escalable y sostenible en el tiempo.
* Analista QA / tester: Responsable de probar la aplicación, identificando y reportando errores antes de su implementación en producción.

1. Propósito de Implementar la Gestión de la Configuración del Software

Macro System ha decidido implementar la Gestión de Configuración del Software (SCM) con el objetivo de evitar errores presentes en anteriores proyectos donde por la falta de un proceso de control y que se estableciera un orden casi se pierde el avance, de esta manera lo que se busca es:

•**Control y gestión documentación, código y registro de cambios**: De esta manera se evitará la sobreescritura del código de programación así como la pérdida de código que se dio en anteriores proyectos debido a despiste de los trabajadores internos dentro del proyecto

•**Asegurar el mantenimiento y la escalabilidad**: Gracias al SCM la empresa podrá realizar de manera más eficiente el manteniendo sus aplicaciones puesto que contará con un registro ordenado de la documentación realizada lo largo de toda la vida del proyecto de modo que se tenga una base para el posterior análisis del sistema cuando el tiempo haya transcurrido y no se recuerde información que pueda resultar clave. De esta manera evitamos la situación actual de la empresa, donde el mantenimiento de algún software desarrollado previamente requiere un análisis exhaustivo solo para volver a comprender el código fuente

•**Facilitar el trabajo en equipo**: Volviendo al ejemplo de “Creación de un sistema de gestión de reservas para una empresa de turismo” observamos como el tener una carencia de herramientas que permitan una mejor cohesión del equipo a la otra de trabajar causó retrasos en la fecha de entrega así como fallas en la implementación. Gracias a la naturaleza del SCM este problema se verá solucionado puesto que las personas del proyecto pese a trabajar en diferentes módulos podrán compartir archivos, hacer seguimiento de tareas y resolver conflictos de manera eficiente

•**Sirve como base para adaptar estándares internacionales**: A la hora de negociar con clientes potenciales de carácter internacional Macro System ha quedado mal parada, puesto que no cuenta con un marco de trabajo reconocido y fiable, trabajándose en gran parte de los proyectos de manera caótica y sin rumbo, lo cual ha conducido a que no se entreguen productos de buena calidad. De esta forma la aplicación del SCM permitirá sentar las bases para que la empresa se enrumbe a la práctica de mejores prácticas teniendo como objetivo lograr la implementación del estándar ISO 12207, el cual tiene como un proceso clavo el SCM

1. Benchmarking de herramientas

Para un claro entendimiento de las herramientas de desarrollo de software para la posterior elección de una de éstas; se realiza un *benchmarking* de ellas, el cual se muestra en la **Tabla 1,** donde se muestran escalas del 1 al 5 en los aspectos incluidos**.**

**Tabla 1.**

*Benchmarking de herramientas de control de versiones con criterio de escala 1-5.*

| **Criterios** | **Git** | **GitKraken** | **TFS** | **Bazaar** | **SVN** |
| --- | --- | --- | --- | --- | --- |
| **Facilidad de uso**: ¿Cómo de fácil es para los usuarios utilizar la herramienta? | 4 | 4 | 2 | 5 | 4 |
| **Funcionalidad**: ¿Cuál es el grado de funciones útiles para el desarrollo del proyecto? | 5 | 5 | 5 | 3 | 4 |
| **Integración:** ¿Con qué facilidad se integra la herramienta con otros sistemas y herramientas? | 5 | 5 | 4 | 3 | 5 |
| **Escalabilidad:** ¿Cómo de bien escala la herramienta con grandes equipos y proyectos? | 5 | 5 | 5 | 3 | 2 |
| **Costo:** ¿Cuál es el costo de la herramienta, si lo hay? | De código abierto y gratuito. | Modelo Freemium. | Costo varía de acuerdo al plan. | Código abierto y gratuita. | De código abierto y gratuito. |
| **Soporte y documentación:** ¿Cómo de buena es la documentación de la herramienta y qué tan bueno es el soporte que se ofrece? | 5 | 5 | 4 | 3 | 5 |

Nota. Tabla de elaboración propia en base a contenidos de [1], [2], [3], [4], [5].

Breve análisis del contenido dela **Tabla 1**

Se observa, adicionalmente que dentro de los datos mostrados en la tabla, se aprecia que en el caso de SVN, éste tiene la peor valoración, por lo cual se descarta fácilmente su uso, TFS se descarta además por su poca valoración en cuanto a facilidad de uso y Bazaar no destaca especialmente en casi ninguna categoría. Los mejores candidatos serían Gist y Git Kraken, no obstante, se decide el segundo debido a que cuenta con mejor interfaz gráfica.

1. Herramienta elegida: GitKraken [1]
2. Descripción:

Git Kraken es una herramienta de gestión de versiones de software y colaboración que ofrece una interfaz visual intuitiva para trabajar con Git, el sistema de control de versiones más popular. Con Git Kraken, los desarrolladores pueden gestionar sus proyectos de Git de manera más eficiente y con menos errores.

1. Arquitectura de la Herramienta:

Se basa en la tecnología Electrón, lo que significa que es una aplicación de escritorio multiplataforma que funciona en Windows, Mac y Linux. Está diseñada para ser fácil de usar, incluso para los principiantes, y ofrece una interfaz de usuario gráfica que muestra de manera visual la estructura de un proyecto de Git, lo que facilita la comprensión y el control del código.

1. Interfaz Visual:

Es una gran interfaz gráfica multiplataforma para Git desarrollada con Electrón. Su interfaz de usuario visual e intuitiva permite una fácil navegación y el uso del Git.

1. Integración con GitHub:

Git Kraken se integra perfectamente con GitHub, GitLab, Bitbucket y otras plataformas de Git, lo que permite a los usuarios administrar fácilmente sus repositorios y proyectos. Permite alojar los repositorios y ayuda a incorporar usuarios a Git con éxito de una forma fácil y segura que permita contribuir desde el primer día de avance, gracias a estas capacidades se puede optimizar el flujo de trabajo del equipo.

1. Herramientas colaborativas:

Git Kraken ofrece herramientas de colaboración integradas que permiten a los usuarios trabajar juntos de manera efectiva en proyectos de Git, lo que incluye comentarios, menciones en el código, colaborar en cambios y ver cambios existentes.

1. Comando integrado de Git:

Git Kraken cuenta con una serie de comandos de Git integrados que permiten a los usuarios realizar tareas de Git directamente desde la herramienta.

1. Historial de Cambios:

Git Kraken además, da la facilidad de mostrar el historial de cambios que se han hecho en un repositorio de manera sencilla.

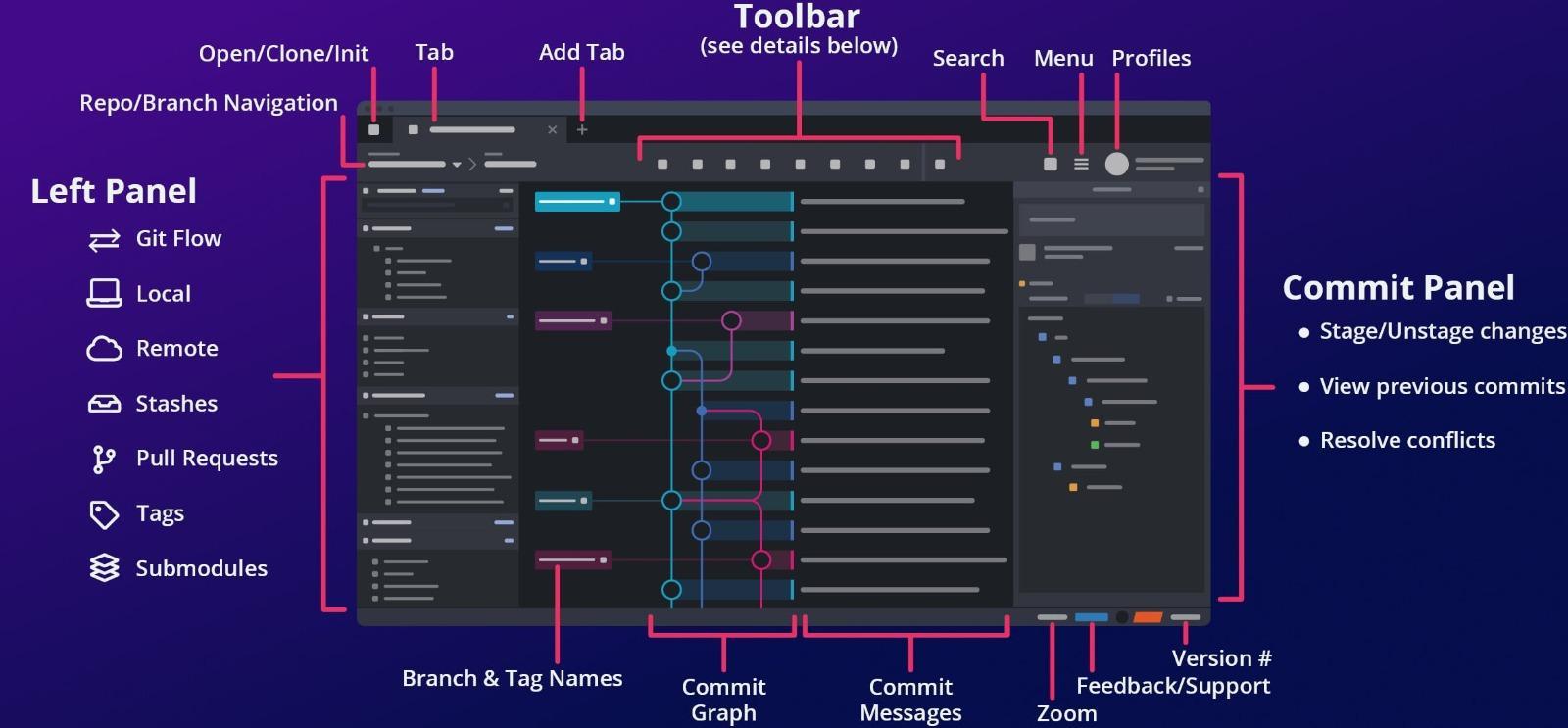
1. Integración con Herramientas de Terceros:

Cuenta con compatibilidad con herramientas como Jira, Trello y Slack.

En resumen, GitKraken es una herramienta de versiones de software que ofrece una interfaz de usuario intuitiva, atractiva y fácil de usar además de tener una gran compatibilidad de plataforma con herramientas de colaboración entre el equipo y seguimiento de versiones, aunque éste requiera más recursos de Hardware que las otras opciones.

**Figura 1.**

*Arquitectura de la herramienta*



Nota. Se muestran por secciones la arquitectura de la herramienta. Fuente: <https://pythones.net/gitkraken/>

1. **Identificación**
   1. **Clasificación de Ítem**

Es el proceso de etiquetar cada elemento de configuración unívocamente. Para realizarlo se utiliza un sistema de etiquetado jerárquico según su función y relación con otros elementos. La clasificación es esencial para garantizar la gestión adecuada de los ítems relevantes durante todo el ciclo de vida del software y simplificar la gestión de la configuración. Se dividen principalmente en tres categorías.

1. **Ítems por evolución:** Relacionado con documentos y archivos ejecutables.
2. **Ítems fuente:** Relacionado con el código fuente y archivos para compilar una aplicación.
3. **Ítems de soporte:** Relacionado con los sistemas operativos y software base.

A continuación, se presenta un cuadro comparativo teniendo en cuenta los conceptos vistos anteriormente junto a los documentos y archivos que presenta el proyecto “Sistema de Trackeo de Tickets en tiempo real del Comedor de UNMSM”.

**Tabla 2.**

*Clasificación de los Ítems de la Configuración (CI).*

| **Tipo**  **(E= Evolución**  **F=Fuente**  **S=Soporte)** | **Nombre del Item (CI)** | **Nomenclatura** | **Extensión** | **Proyecto** |
| --- | --- | --- | --- | --- |
| E | *Plan de Proyecto* | GT-PP.DOCX | DOCX | GT |
| E | *Cronograma del Proyecto* | GT-CP.XLSX | XLSX | GT |
| E | *Documento de Especificación de Requisito 1* | GT-DER\_01.DOCX | DOCX | GT |
| E | *Documento de Especificación de Requisito 2* | GT-DER\_02.DOCX | DOCX | GT |
| E | *Documento de Especificación de Requisito 3* | GT-DER\_03.DOCX | DOCX | GT |
| E | *Documento de Especificación de Requisito 4* | GT-DER\_04.DOCX | DOCX | GT |
| E | *Documento de Especificación de Requisito 5* | GT-DER\_05.DOCX | DOCX | GT |
| E | *Documento de Especificación de Requisito 6* | GT-DER\_06.DOCX | DOCX | GT |
| E | *Documento de Guía de Estilos* | GT-DGE.DOCX | DOCX | GT |
| E | *Documento de Especificación de la Base de Datos* | GT-DEBD.DOCX | DOCX | GT |
| E | *Documento de Arquitectura del Software* | GT-DAS.DOCX | DOCX | GT |
| E | *Reporte del Primer Sprint* | GT-RS\_01.DOCX | DOCX | GT |
| F | *Estructura de la base de datos* | GT-EBD.SQL | SQL | GT |
| F | *Codificar Requisito 1 y 2* | Carpeta [Codificación] | \* | GT |
| E | *Documento de prueba unitaria requisito 1* | GT-DPUR\_01.DOCX | DOCX | GT |
| E | *Documento de prueba unitaria requisito 2* | GT-DPUR\_02.DOCX | DOCX | GT |
| E | *Reporte del Desarrollo del Software 1* | GT-RDS\_01.DOCX | DOCX | GT |
| E | *Reporte del Sprint 2* | GT-RS\_02.DOCX | DOCX | GT |
| F | *Codificar Requisito 3 y 4* | Carpeta [Codificación] | \* | GT |
| E | *Documento de prueba unitaria requisito 3* | GT-DPUR\_03.DOCX | DOCX | GT |
| E | *Documento de prueba unitaria requisito 4* | GT-DPUR\_04.DOCX | DOCX | GT |
| E | *Documento acceso a la base de datos* | GT-DABD.DOCX | DOCX | GT |
| E | *Documento de Pruebas del Software* | GT-DPS.DOCX | DOCX | GT |
| E | *Documento de Manual de usuario* | GT-DMU.DOCX | DOCX | GT |
| E | *Reporte del Desarrollo del Software 2* | GT-RDS\_02.DOCX | DOCX | GT |
| E | *Reporte del Sprint 3* | GT-RS\_03.DOCX | DOCX | GT |
| E | *Acta de cierre del proyecto* | GT-ACP.DOCX | DOCX | GT |
| F | *Codificación del sistema 1* | SGRET-CS\_01.JAVA | JAVA | SGRET |
| F | *Codificación del sistema 2* | SGRET-CS\_02.JAVA | JAVA | SGRET |
| S | *Java Setup Version 8* | JAVASETUP8U371.EXE | EXE | SGRET |
| F | *Codificación del Requerimiento 3* | AMGICTR-CR\_03.CSPROJ | CSPROJ | AMGICTR |
| F | *Codificación del Requerimiento 4* | AMGICTR-CR\_04.CSPROJ | CSPROJ | AMGICTR |
| E | *Ejecutable de la codificación del requerimiento 3* | AMGICTR-ECR\_03.EXE | EXE | AMGICTR |
| E | *Ejecutable de la codificación del requerimiento 4* | AMGICTR-ECR\_04.EXE | EXE | AMGICTR |
| S | *Visual C Sharp 2005* | VISUALCSHARP2005.EXE | EXE | AMGICTR |

*Nota.* Elaboración Propia.

La clasificación por tipos de ítems para los 3 proyectos que cuenta la consultora (“Sistema de Trackeo de Tickets en tiempo real del Comedor de UNMSM”-GT, “Sistema de Gestión de reservas para una Empresa de Turismo”-SGRET, “Aplicación Móvil de Gestión de Inventarios para una Cadena de Tiendas de Retail”- AMGICTR) presenta gran cantidad de ítems de evolución, siendo estos fundamentales porque contienen información crítica para el proceso de desarrollo y gestión adecuada mejorando la calidad del software. Por otro lado, se presentan 16 ítems de fuente, los cuales presentan gran importancia en el proyecto por garantizar la trazabilidad de los cambios en el software. Finalmente, se ejemplifican 2 ítems de soporte, esto se resalta puesto que puede evitar problemas con la compatibilidad, seguridad y rendimiento del software.

* 1. **Definición de Nomenclaturas**

Para dar un mayor entendimiento acerca de los documentos presentados, en cuanto a su estructura y jerarquía, se presentan los escenarios en la **Tabla 3**.

**Tabla 3.**

*Casos para la nomenclatura de los documentos.*

| **Casos** | **Descripción** | **Ejemplo** |
| --- | --- | --- |
| 1 | Para ítems que no son específicos a un proyecto único, tales como en este caso, el título del proyecto, solo se usa el  **ACRÓNIMO DEL PROYECTO** | GT (Gusano Tracker) |
| 2 | Para ítems que se encuentran en el proyecto, se usa la siguiente fórmula para nombrarlos.  **ACRÓNIMO DEL PROYECTO + “-” + ACRÓNIMO DEL ELEMENTO** | GT-DAS (Documento de Arquitectura de Software) |
| 3 | Para ítems que hacen referencia al mismo tipo de elemento, con distinta numeración.  **ACRÓNIMO DEL PROYECTO + “-” + ACRÓNIMO DEL ELEMENTO + “\_” + “NUMERACIÓN CORRELATIVA AL DOCUMENTO”** | GT-DER\_02 y GT-DER\_03 (Documentos de especificaciones de requisitos para el requisito 2 y 3 respectivamente) |
| 4 | Para ítems cuyas siglas coincidan en con uno anterior.  **ACRÓNIMO DEL PROYECTO + "-" + ACRÓNIMO DEL ELEMENTO + “-” + TRES PRIMERAS LETRAS DE LA SEGUNDA PALABRA DEL ELEMENTO** | GT-DAS-SOF  (Documento de Análisis de Software)  GT-DAS-SIS  (Documento de Arquitectura de Sistemas) |

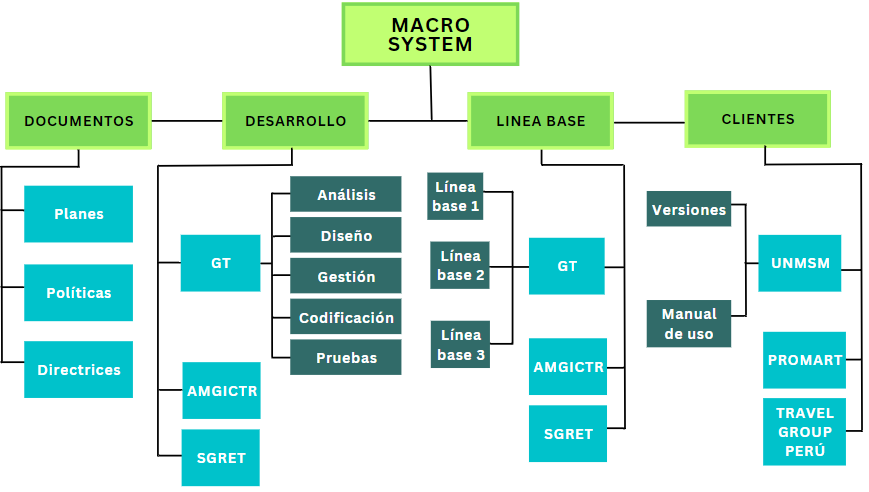
Nota. Tabla de elaboración propia

* 1. **Diseño de la estructura del repositorio (Descripción de las librerías)**

Mediante el siguiente diagrama se muestra como nuestra empresa organiza la estructura de su repositorio, esto resulta una parte fundamental ya que una buena estructuración permite la organización, mantenimiento, escalabilidad y colaboración en el desarrollo de software. Estos factores son sumamente importantes ya que permiten un manejo claro y sistemático de todos los proyectos que hayan sido desarrollados o se encuentren en proceso de desarrollo por parte de Macro System.

**Figura 2.**

*Estructura del repositorio de MacroSystem.*



*Nota.* Elaboración propia.

**Descripción de las carpetas/librerías del repositorio**

1. **Documentos**
   1. **Responsable**

* Jefe de proyecto
  1. **Actividades**
* Establecimiento y actualización pertinente de aspectos generales relacionados con los lineamientos de la empresa, como el establecimiento de directrices a seguir, políticas de la empresa, planes, etc.
  1. **Contenido**
* Planes acerca de cómo se deben llevar a cabo el proceso de desarrollo
* Políticas establecidas para regular y guiar el uso del software en el entorno de trabajo
* Directrices establecidas para estandarizar y guiar el desarrollo de software de forma que se establezca un modelo consistente y eficiente
  1. **Accesos**

| **Rol** | **Tipo de acceso** |
| --- | --- |
| **Jefe de proyecto** | * **Leer** * **Escribir** * **Modificar** |
| **Desarrolladores y Testers** | * **Leer** |

1. **Desarrollo**
   1. **Responsable**

* Arquitecto de software
  1. **Actividades**
* Análisis de los requerimientos de modo que se puedan establecer las especificaciones y funcionalidades del software a desarrollar
* Codificación y prueba de los requerimientos
* Gestión de los avances de modo que se haga una autoevaluación de los avances desarrollados
  1. **Contenido**

Conformado por bibliotecas(carpetas) de proyectos, donde cada proyecto contiene las siguientes subcarpetas:

* Gestión
* Análisis
* Diseño
* Codificación
* Pruebas
  1. **Accesos**

| **Rol** | **Tipo de acceso** |
| --- | --- |
| **Arquitecto de software** | * **Leer** * **Escribir** * **Ejecutar** * **Modificar** * **Eliminar** |
| **Jefe de proyecto** | * **Leer** |
| **Desarrolladores** | * **Leer** * **Escribir** * **Ejecutar** |
| **Analista QA/tester** | * **Leer** * **Ejecutar** |

1. **Línea base**
   1. **Responsable**

* Gerente de la Configuración
  1. **Actividades**
* Llevar una copia de una versión específica del software que se ha aprobado y se ha establecido como punto de referencia para el desarrollo posterior
* Establecer nuevas versiones de las líneas base si llegara a darse un cambio
  1. **Contenido**

Conformado por bibliotecas(carpetas) de proyectos, donde cada proyecto contiene:

* Líneas base correlativas a la iteración
  1. **Accesos**

| **Rol** | **Tipo de acceso** |
| --- | --- |
| **Gerente de la Configuración** | * **Leer** * **Escribir** * **Ejecutar** |
| **Jefe de proyecto** | * **Leer** * **Ejecutar** |
| **Desarrolladores y Testers** |

1. **Clientes**
   1. **Responsable**

* Gerente de la Configuración
  1. **Actividades**
* Mantener un registro completo y actualizado de todas las versiones del software, documentación y otros artefactos entregados al cliente.
  1. **Contenido**

Conformado por bibliotecas(carpetas) de proyectos, donde cada proyecto contiene:

* Versiones del software entregadas al cliente
* Manuales de usuario
  1. **Accesos**

| **Rol** | **Tipo de acceso** |
| --- | --- |
| **Gerente de la Configuración** | * **Leer** * **Escribir** * **Ejecutar** |
| **Jefe de proyecto** | * **Leer** * **Ejecutar** |
| **Desarrolladores y Testers** |

* 1. **Definición de Líneas Base (Ejemplo)**

A continuación líneas base del proyecto GT …

**Línea Base 1:**

* Esta comprende los ítems necesarios para poder realizar la especificación de los requisitos, como también los documentos del diseño, bases de datos y arquitectura necesarias para la realización del proyecto de manera concisa y estructurada.

**Línea Base 2:**

* Aquí se cuenta con los puntos de la codificación de los 2 primeros requisitos y sus respectivas evaluaciones unitarias para terminar con el documento de reporte del estado del software que se cuente en ese momento.

**Línea Base 3:**

* Aquí se cuenta con los puntos de la codificación de los requisitos 3 y 4 y sus respectivas evaluaciones unitarias para terminar con los documentos aplicativos del diseño, reglas de acceso necesarias para la estructura de la base de datos y terminando con el reporte final, con su acta correspondiente.

| **Línea Base** | **Hito** | **Item** |
| --- | --- | --- |
| Línea Base 1 | 1 | Macro-System\_G2  GT-PP.DOCX  GT-CP.XLSX  GT-ER\_01.DOCX  GT-ER\_02.DOCX  GT-ER\_03.DOCX  GT-ER\_04.DOCX  GT-ER\_05.DOCX  GT-ER\_06.DOCX  GT-GE.DOCX  GT-DBD.DOCX  GT-AS.DOCX  GT-SPRT\_01.DOCX |
| Línea Base 2 | 2 | GT-EBD.DOCX  GT-COD\_01.DOCX  GT-PU\_01.DOCX  GT-PU\_02.DOCX  GT-RDS\_01.DOCX  GT-SPRT\_02.DOCX |
| Línea Base 3 | 3 | GT-COD\_02.DOCX  GT-PU\_03.DOCX  GT-PU\_04.DOCX  GT-DG.DOCX  GT-ABD.DOCX  GT.PF.DOCX  GT-RDS\_02.DOCX  GT-SPRT\_03.DOCX  GT-ACP.DOCX |

1. **Control**
   1. **Control de vida de los ítems**

Clasificación de ítems del proyecto “Sistema de Trackeo de Tickets en tiempo real del Comedor de UNMSM ” que se encuentren en las etapas “Borrador” y “Aceptado”, siendo hoy 7 de junio del 2023.

**Tabla 4.**

*Clasificación de ciclo de vida de los ítems.*

| **ITEM** | **ETAPA** | **FECHA INICIO** | **FECHA FIN** |
| --- | --- | --- | --- |
| *Repositorio GitHub* | *Aceptado* | 09/04/2023 | 15/04/2023 |
| *Plan de Proyecto* | *Aceptado* | 09/04/2023 | 16/04/2023 |
| *Cronograma del Proyecto* | *Aceptado* | 09/04/2023 | 16/04/2023 |
| *Documento de Especificación de Requisito 1* | *Aceptado* | 16/04/2023 | 23/04/2023 |
| *Documento de Especificación de Requisito 2* | *Aceptado* | 16/04/2023 | 23/04/2023 |
| *Documento de Especificación de Requisito 3* | *Aceptado* | 16/04/2023 | 23/04/2023 |
| *Documento de Especificación de Requisito 4* | *Aceptado* | 16/04/2023 | 23/04/2023 |
| *Documento de Especificación de Requisito 5* | *Aceptado* | 16/04/2023 | 23/04/2023 |
| *Documento de Especificación de Requisito 6* | *Aceptado* | 16/04/2023 | 23/04/2023 |
| *Documento de Guía de Estilos* | *Aceptado* | 16/04/2023 | 23/04/2023 |
| *Documento de Especificación de la Base de Datos* | *Aceptado* | 16/04/2023 | 23/04/2023 |
| *Documento de Arquitectura del Software* | *Aceptado* | 23/04/2023 | 29/04/2023 |
| *Reporte del Sprint 1* | *Aceptado* | 23/04/2023 | 29/04/2023 |
| *Estructura de la Base de Datos* | *Aceptado* | 01/05/2023 | 08/05/2023 |
| *Codificar Requisito 1 y 2* | *Aceptado* | 01/05/2023 | 15/05/2023 |
| *Documento de Prueba Unitaria Requisito 1* | *Aceptado* | 15/05/2023 | 19/05/2023 |
| *Documento de Prueba Unitaria Requisito 2* | *Aceptado* | 15/05/2023 | 19/05/2023 |
| *Reporte del Desarrollo del Software 1* | *Aceptado* | 19/05/2023 | 23/05/2023 |
| *Reporte del Sprint 2* | *Aceptado* | 19/05/2023 | 23/05/2023 |
| *Codificar Requisito 3 y 4* | *Aceptado* | 25/05/2023 | 07/06/2023 |
| *Documento de Prueba Unitaria Requisito 3* | *Borrador* | 07/06/2023 | 10/06/2023 |
| *Documento de Prueba Unitaria Requisito 4* | *Borrador* | 07/06/2023 | 10/06/2023 |
| *Documento Acceso a la Base de Datos* | *Borrador* | 07/06/2023 | 10/06/2023 |

*Nota.* Elaboración propia.

Es crucial mantener un seguimiento adecuado y cumplir con los plazos establecidos para asegurar un avance constante en el proyecto. Al analizar la tabla, se observa que algunos ítems se encuentran en la etapa "Aceptado", lo que indica que se han completado dentro de los plazos previstos. Por otro lado, hay otros ítems que están en la etapa "Borrador" y aún están en proceso de desarrollo. Esto resalta la importancia de dar continuidad a estos ítems pendientes para mantener el flujo del proyecto y lograr su conclusión exitosa.

* 1. **Formato de Solicitud de cambio**
  2. **Proceso de Gestión de cambios**

1. **Estado**
2. **Entrega y Gestión de Realease**

Referencias Bibliográficas

[1] GitKraken vs GitHub Desktop [OMS]. (2023). GitKraken <https://www.gitkraken.com/compare/gitkraken-vs-github-desktop>

[2] Sitio web de GitKraken. Gitkraken <https://www.gitkraken.com>

[3] Azure DevOps server <https://azure.microsoft.com/en-us/services/devops/server/>

[4] Bazaar <https://bazaar.canonical.com/en/>

[5] Git. <https://git-scm.com/>