**QUANTUM ANTS SOFTWARE S.A.C**

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

# Introducción

## a)Situación de la empresa

Quantum Ants Software S.A.C. es una empresa de desarrollo de software con un equipo experimentado de profesionales de TI. Entre las personas más importantes de la empresa se encuentra Max Bruno Saavedra Monterrey, CEO, jefe de proyecto y arquitecto de software. Max es una persona altamente capacitada en tecnología de la información y cuenta con una amplia experiencia en liderazgo de proyectos y desarrollo de software empresarial.

Jatziry Fernanda Sanchez Wong es una desarrolladora back end, analista y administradora de base de datos con amplia experiencia en el desarrollo de aplicaciones web personalizadas y software empresarial. Geomar Willy Fernandez Camacho es otro desarrollador back end que se destaca por su experiencia en el diseño de bases de datos y la creación de soluciones personalizadas para los clientes.

Entre los desarrolladores front end de Quantum Ants Software S.A.C. se encuentran Fabricio Vidal Chuquispuma Merino, Jose Alessandro Quispe Cabello y Renzo Jesus Leon Bautista. Fabricio es un diseñador UX con experiencia en el desarrollo de aplicaciones web personalizadas y aplicaciones móviles. Jose Alessandro es un tester y diseñador UI que se enfoca en la calidad y la experiencia de usuario. Renzo Jesus es un diseñador UI con habilidades en la creación de interfaces de usuario atractivas y funcionales. En conjunto, el equipo de Quantum Ants Software S.A.C. trabaja para proporcionar soluciones innovadoras de alta calidad a sus clientes en diferentes sectores.

Quantum Ants Software S.A.C. es una empresa de desarrollo de software con sede en Lima Perú fundada en 2010 Quantum Ants Software S.A.C. Tiene más de una década de experiencia en el mercado de desarrollo de software.

La empresa se especializa en el desarrollo de aplicaciones web personalizadas, aplicaciones móviles y software empresarial para una amplia gama de clientes en diferentes sectores. Con un equipo experimentado de profesionales de TI, Quantum Ants Software S.A.C. se dedica a proporcionar soluciones innovadoras de alta calidad a sus clientes.

La empresa ha completado con éxito varios proyectos de desarrollo de software, tanto en producción como en desarrollo, para clientes nacionales e internacionales en los últimos años. Los proyectos en producción incluyen aplicaciones web personalizadas, software empresarial y aplicaciones móviles.

Nuestra empresa ha tenido la oportunidad de desarrollar dos proyectos exitosos y actualmente está trabajando en un tercero. El primer proyecto, un sistema de reservación de hoteles para el cliente HOTEL HORMIGA S.A.C, se completó hace 13 años. Durante la fase de desarrollo, nuestro equipo experimentó dificultades en la gestión de la configuración de software, incluida la falta de control de versiones del código fuente y la entrega errónea de versiones al cliente debido a la falta de gestión de control de cambios. A medida que avanzamos en nuestro segundo proyecto, un sistema de trackeo de aviones para el aeropuerto Hormiga Plane S.A.C, enfrentamos el desafío de documentar adecuadamente los cambios realizados en el código fuente. Además, cuando contratamos nuevos desarrolladores temporales, no se adaptaron al ritmo del proyecto, lo que resultó en una menor eficiencia y mayores tiempos de desarrollo. Actualmente, estamos trabajando en el desarrollo de un sistema de registro de operaciones contables para el cliente BCPOLAR S.A.C, y estamos implementando las lecciones aprendidas de nuestros proyectos anteriores para asegurarnos de una gestión eficaz de la configuración de software.

El equipo de trabajo de Quantum Ants Software S.A.C. incluye principalmente un jefe de proyecto, tres desarrolladores front-end, dos desarrolladores back-end y una administradora y analista de base de datos. Todos los miembros del equipo tienen una sólida formación de informática y experiencia en el desarrollo de software. El jefe de proyectos se encarga de liderar y gestionar el proyecto, mientras que los desarrolladores front-end y back-end se encargan de la programación y diseño de la interfaz de usuario; además de la estructura de la aplicación. Por otro lado, quien analiza la base de datos se encarga de la gestión de la base de datos y el análisis de datos.

Quantum Ants Software S.A.C. Enfrenta varios desafíos en la gestión de la configuración y el control de cambios en el desarrollo de software. El equipo de gestión de configuración de Quantum Ants Software S.A.C se asegura de que todos los elementos del Software se controlen y documenten adecuadamente en cada etapa del proceso de desarrollo. La empresa utiliza herramientas de control de cambios y gestión de la configuración para realizar un seguimiento de los cambios realizados en el código fuente, la documentación y otros elementos relacionados con el proyecto.

**Visión:** Ser la empresa de referencia en términos de calidad, cumplimiento, profesionalismo y compromiso; haciendo de nuestros bajos costos la mejor y más confiable inversión de dinero de nuestros clientes.

**Misión:** Proporcionar nuevas alternativas para la simplificación de los procesos más complejos de los clientes, creando expectativas de mejora contínua a través de las herramientas puestas a su alcance.

**Nosotros:** Quantum Ants Software S.A.C. es una empresa prestadora de servicios que a través de un equipo caracterizado por su visión innovadora y su experiencia, determina, analiza y desarrolla programas que permiten a sus clientes optimizar y automatizar los procesos que la rigen.

**Valores:** Nuestra empresa se basa en la innovación y con valores tales como:

Dedicación y Pasión por la Excelencia.

Integración y honestidad.

Creatividad e Innovación.

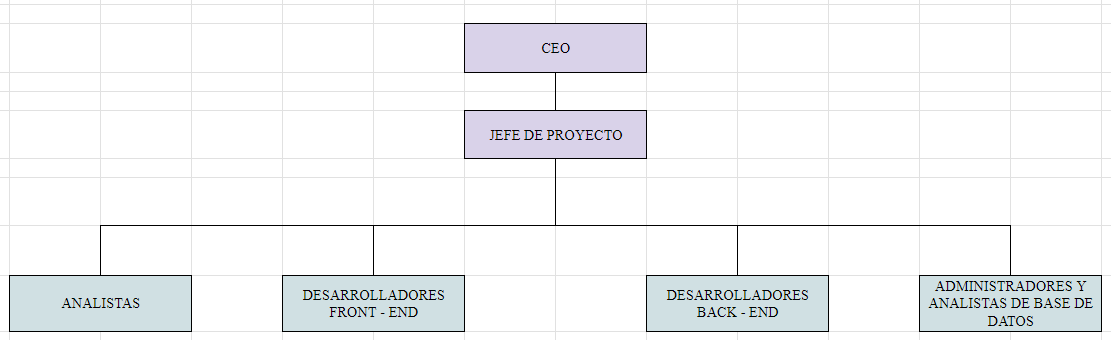
Colaboración y respeto.

Diversidad.

**Organigrama de la empresa Quantum Ants Software S.A.C.**

**Figura 01**

*Organigrama de la empresa Quantum Ants S.A.C.*



*NOTA: Organigrama general de la empresa.*

* CEO: encargado de la toma de decisiones estratégicas y la dirección general de la empresa.
* JEFE DE PROYECTO: responsable de liderar y gestionar los proyectos de desarrollo de software.
* ANALISTAS: Son responsables de analizar los requerimientos y necesidades del cliente para poder diseñar una solución de software adecuada.
* DESARROLLADORES FRONT - END: encargados de la programación y diseño de la interfaz de usuario.
* DESARROLLADORES BACK - END: encargados de la programación y diseño de la interfaz de usuario.
* ADMINISTRADORES Y ANALISTAS DE BASE DE DATOS: encargados de la programación y diseño de la interfaz de usuario.

Algunos problemas que Quantum Ants Software pudo haber enfrentado inicialmente con sus clientes podrían haber sido:

1. Malentendidos en cuanto a las expectativas del proyecto: A menudo, los clientes tienen expectativas poco realistas sobre lo que puede lograrse con el software. Esto puede deberse a una falta de comprensión de los procesos de desarrollo de software o a la falta de claridad en la comunicación entre el equipo de desarrollo y el cliente. Como resultado, los clientes pueden sentirse decepcionados con el producto final y pueden surgir problemas de retroalimentación.
2. Cambios constantes en los requisitos del proyecto: Los clientes pueden cambiar los requisitos del proyecto a medida que avanzan las etapas de desarrollo, lo que puede aumentar el costo y el tiempo de entrega. La falta de un proceso formal de gestión de cambios puede llevar a problemas de control de versiones, cambios no autorizados y errores en la configuración del software.
3. Problemas de compatibilidad y escalabilidad: Los clientes pueden tener requisitos específicos de compatibilidad y escalabilidad para el software que Quantum Ants Software está desarrollando. Si estos requisitos no se tienen en cuenta desde el principio, puede llevar a problemas de integración y dificultades para ampliar el software en el futuro. Además, puede haber problemas de configuración del software si el equipo de desarrollo no tiene en cuenta estas consideraciones desde el principio.

## b)Propósito de Implementar la Gestión de la Configuración del Software

La gestión de la configuración del software es importante de implementar en nuestros proyectos ya que nos permite controlar los cambios en el código fuente, la documentación y otros artefactos del software durante su ciclo de vida. Nos ayudará a gestionar de manera más efectiva los cambios y a mantener un registro de versiones de software.

Al aplicar la gestión de la configuración, se disminuye la incidencia de este tipo de problemas con la ayuda de sus características principales:

1. Registro de cambios: Permite rastrear quién realizó un cambio, cuándo se realizó y qué se cambió.
2. Control de versiones: Cada versión del software se almacena en el repositorio de control de versiones, lo que te permite acceder a versiones anteriores del software en caso de que necesites hacer cambios o revertir a una versión anterior.
3. Control de acceso: Definir quién tiene acceso a qué partes del código y quién puede realizar cambios en el mismo.
4. Trabajo en equipo: Cada miembro del equipo puede trabajar en su propia rama y fusionar sus cambios en la rama principal cuando estén listos.
5. Identificación de problemas: Utilizar herramientas de seguimiento de problemas permite identificar problemas y asociarlos con cambios específicos en el código fuente.
6. Implementación de cambios: Utilizar herramientas de integración permite automatizar la implementación de cambios en el software.

En resumen, la gestión de configuración de software es esencial en el control de cambios en el desarrollo de software y contribuye directamente con el éxito y los resultados de los proyectos que desarrolla la empresa Quantum Ants Software S.A.C.

## c)Benchmarking de herramientas

A continuación, se presenta una tabla que facilitará el análisis de las herramientas de control de versiones según criterios que favorezcan al desarrollo de proyectos de calidad.

**Tabla 1**

*Evaluación de herramientas de control de versiones según criterios de funcionalidad, integración, seguridad, facilidad de uso, escalabilidad, soporte, costo,almacenamiento máximo, máximo de colaboradores en versión gratuita y funciones en dicha versión y requerimiento de software adicional*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Criterios** | **GitHub**  **[ 1]** | **GitKraken**  **[ 2]** | **Team Foundation Server**  **[3 ]** | **Bazaar**  **[4]** | **AWS CodeCommit [5]** | **Subversion [6]** |
| Amplias funcionalidades | SÍ | SÍ | SÍ | SÍ | SÍ | SÍ |
| Capacidad de Integración | SÍ | SÍ | SÍ | NO | SÍ | NO |
| Seguridad | SÍ | SÍ | SÍ | SÍ | SÍ | SÍ |
| Facilidad de uso | SÍ | SÍ | SÍ | SÍ | SÍ | SÍ |
| Escalabilidad | SÍ | SÍ | SÍ | SÍ | SÍ | SÍ |
| Soporte | SÍ | SÍ | SÍ | NO | SÍ | SÍ |
| Costo | GRATIS | PAGO | PAGO | GRATIS | PAGO | GRATIS |
| Almacenamiento máximo | Ilimitado | Ilimitado | Ilimitado | Ilimitado | Ilimitado | Ilimitado |
| Máximo colaboradores versión gratis | Ilimitado | 0 | 5 | Ilimitado | 0 | Ilimitado |
| Funcionalidades limitadas en versión gratuita | SÍ | SÍ | NO | SÍ | SÍ | SÍ |
| Requiere instalar software adicional | NO | SÍ | SÍ | SÍ | NO | NO |

*Nota.* Los criterios se evaluaron con una escala de SÍ y NO. Las herramientas se clasificaron según la presencia de criterios cumplidos.

*Fuente.* Elaboración propia a partir de información recopilada de las páginas web oficiales de cada herramienta evaluada.

**Análisis de las herramientas de control de versiones:**

Después de revisar la Tabla 1 y, de esta manera, los criterios de funcionalidad, integración, seguridad, facilidad de uso, escalabilidad, soporte, costo,almacenamiento máximo, máximo de colaboradores en versión gratuita y funciones en dicha versión y requerimiento de software adicional , se concluye que:

* **AWS Code Commit:** Ofrece una buena integración con otras herramientas de Amazon Web Services, pero su costo es elevado comparado con otras herramientas.
* **Bazaar:** Es una herramienta que ha sido utilizada por la comunidad de software libre, pero su soporte es limitado. Además, no cuenta con muchas funcionalidades avanzadas.
* **GitKraken:** Fácil de usar y cuenta con una interfaz gráfica intuitiva, lo que la hace una buena opción para equipos que no tienen experiencia previa con control de versiones. Sin embargo, su costo es elevado.
* **GitHub:** Ofrece una amplia variedad de funcionalidades y herramientas de integración, incluyendo una buena integración con Visual Studio Code, lo que la hace una buena opción para equipos que ya trabajan con esta herramienta. Además, es gratuita para proyectos de código abierto y cuenta con planes de pago asequibles para empresas.
* **Subversion:** Ofrece una buena integración con otras herramientas de desarrollo y cuenta con una buena funcionalidad, pero resulta complicada para equipos que no tienen experiencia previa con control de versiones.
* **Team Foundation Server:** Es una herramienta desarrollada por Microsoft que ofrece una buena integración con otras herramientas de desarrollo de la compañía. Sin embargo, su costo es elevado y puede ser complicado de usar para equipos que no tienen experiencia previa con control de versiones.

En base a estos criterios, podemos concluir que GitHub es la mejor opción para nuestro equipo de trabajo en Quantum Ant Software S.A.C. Esto se debe a que cuenta con todas las funcionalidades que necesitamos, incluyendo una buena integración con el lenguaje de programación Python y Visual Studio Code, que es la herramienta que utilizamos para programar. Además, es gratuita para proyectos de código abierto y cuenta con planes de pago asequibles para empresas.

## d)Herramienta elegida: GitHub

**Descripción de la herramienta:**

GitHub es una plataforma de control de versiones de software y alojamiento de repositorios en línea que se ha convertido en una de las herramientas de desarrollo de software más populares y utilizadas en todo el mundo. Fue adquirida por Microsoft en 2018 y es compatible con una variedad de sistemas operativos y lenguajes de programación.

**Arquitectura:**

GitHub se basa en la arquitectura de Git, un sistema de control de versiones distribuido que permite a los usuarios colaborar en el desarrollo de software y rastrear cambios en el código fuente. El sistema utiliza una estructura de árbol de objetos para almacenar y recuperar versiones de archivos y se basa en la creación de ramas y fusiones para permitir que los desarrolladores trabajen en paralelo en diferentes características y correcciones de errores.

**Figura 02**

*Flujo de trabajo en GitHub mediante el uso de ramas y fusiones*

*Fuente.* INGENIERÍA DE SOFTWARE. *Procesos de desarrollo con GitHub.*

**Beneficios:**

* Facilita la colaboración entre desarrolladores y equipos, permitiendo que varios desarrolladores trabajen en un mismo proyecto y fusionen sus cambios de forma sencilla y segura.
* Proporciona un control de versiones completo y transparente, lo que permite a los desarrolladores seguir los cambios realizados en el código fuente y revertir cualquier cambio no deseado.
* Se integra fácilmente con una amplia variedad de herramientas de desarrollo, como IDEs, servicios de integración continua y sistemas de seguimiento de errores.
* Ofrece almacenamiento gratuito y seguro en la nube para los repositorios de código, lo que permite a los desarrolladores acceder a su trabajo desde cualquier lugar con una conexión a Internet.
* Cuenta con una gran comunidad de desarrolladores y proyectos de código abierto, lo que permite a los usuarios descubrir nuevos proyectos y colaborar en ellos.

**Limitaciones:**

A pesar de que GitHub ofrece medidas de seguridad como autenticación y autorización, cifrado de datos y auditorías de cambios, algunos usuarios pueden sentirse incómodos al almacenar su propiedad intelectual en la nube y prefieren alojar su repositorio en su propio servidor privado para mayor seguridad y privacidad.

Aunque GitHub ofrece almacenamiento gratuito y un plan de suscripción asequible para usuarios individuales, los planes empresariales pueden ser bastante costosos y no siempre son viables para pequeñas empresas o equipos de desarrollo.

Los usuarios necesitan acceso a Internet para poder trabajar en sus proyectos y acceder a los repositorios. Si hay problemas de conectividad, puede haber dificultades para trabajar en su código fuente.

GitHub tiene un límite en el tamaño de los archivos que se pueden almacenar en un repositorio, lo que puede ser un problema para aquellos que trabajan con grandes archivos de datos o multimedia.

El plan gratuito de GitHub limita el número de colaboradores que pueden trabajar en un repositorio, lo que puede ser un problema para equipos más grandes que necesitan trabajar en un proyecto juntos.

# Identificación

## Objetivos del plan

El objetivo principal del plan es establecer los procedimientos y estándares para la identificación de los elementos de la configuración del software, para asegurar que se gestionen adecuadamente durante todo el ciclo de vida del software.

## Alcance del plan

El alcance del plan incluye todos los elementos de la configuración del software, como el código fuente, la documentación, las especificaciones y los artefactos generados durante el proceso de desarrollo.

## Procedimientos

Se establecerán los procedimientos para la identificación de los elementos de la configuración del software, incluyendo la nomenclatura de los nombres de los archivos, el etiquetado de versiones, la estructura de la jerarquía de carpetas, etc.

## Identificadores únicos

Se utilizarán identificadores únicos para los elementos de la configuración del software, para evitar confusiones entre diferentes versiones o ramas del código fuente.

## Responsabilidades

Se asignan las responsabilidades para la identificación de los elementos de la configuración del software, y se definirán los roles y responsabilidades de los diferentes miembros del equipo de desarrollo en el proceso.

## Herramientas de apoyo

Se identificarán y seleccionarán las herramientas de software necesarias para el soporte de la identificación y gestión de la configuración del software, como Git, Subversion, o AWS CodeCommit.

## Control de cambios

Se establecerá un proceso de control de cambios para la identificación y gestión de los cambios a los elementos de la configuración del software, asegurando que se documenten y aprueben adecuadamente.

## Auditoría de la configuración

Se establecerá un proceso de auditoría periódica de la configuración del software, para garantizar que se están aplicando adecuadamente los procedimientos de identificación y gestión de la configuración del software.

## Entregables

Se especificarán los entregables del plan de gestión de la configuración del software, como los informes de auditoría, la documentación del proceso de identificación, los procedimientos de control de cambios, etc.

## Mantenimiento

El plan de gestión de la configuración del software se mantendrá actualizado a lo largo del ciclo de vida del software, para reflejar cualquier cambio en los procedimientos o herramientas utilizadas para la identificación y gestión de la configuración del software.

## Clasificación de Ítems

En tipos de ítem están clasificados por evolución = E , fuente = F y soporte = S.

**Tabla 2**

*Clasificación de los ítems de la configuración.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tipo | Nombre del item | Nomenclatura | Extensión | Proyecto |
| E | Plan de Proyecto (PROJECT CHARTER) | SGDS - PC | DOCX | SDGS |
| E | *Cronograma del Proyecto* | SGDS - CP | XLS | SDGS |
| E | *Repositorio GitHub* | - | - | SDGS |
| E | *Documento de Especificación de Requisitos 01* | SGDS - RS 01 | DOCX | SDGS |
| E | *Documento de Especificación de Requisitos 02* | SGDS - RS 02 | DOCX | SDGS |
| E | *Documento de Especificación de Requisitos 03* | SGDS - RS 03 | DOCX | SDGS |
| E | *Documento de Especificación de Requisitos 04* | SGDS - RS 04 | DOCX | SDGS |
| E | *Documento de Especificación de Requisitos 05* | SGDS - RS 05 | DOCX | SDGS |
| E | *Documento de Especificación de Requisitos 06* | SGDS - RS 06 | DOCX | SDGS |
| E | *Documento de Especificación de Requisitos 07* | SGDS - RS 07 | DOCX | SDGS |
| E | *Documento de Especificación de Requisitos 08* | SGDS - RS 08 | DOCX | SDGS |
| E | *Documento de Especificación de Requisitos 09* | SGDS - RS 09 | DOCX | SDGS |
| E | *Documento de Arquitectura del Software* | SGDS - DAS | DOCX | SDGS |
| E | *Documento de Especificación de la BD* | SGDS - DEBD | DOCX | SDGS |
| E | *Documento de Especificación de UI* | SGDS - DEUI | DOCX | SDGS |
| E | *Documento de Guía de Estilos* | SGDS - GE | DOCX | SDGS |
| F | *Reporte del Desarrollo del Software* | SGDS - RDS | PY | SDGS |
| E | *Reporte del Primer Sprint* | SGDS - RPS | DOCX | SDGS |
| F | *Documento de implementación y Verificación de Requisito 01* | SGDS - IVR 01 | PY | SDGS |
| F | *Documento de implementación y Verificación de Requisito 02* | SGDS - IVR 02 | PY | SDGS |
| F | *Documento de implementación y Verificación de Requisito 03* | SGDS - IVR 03 | PY | SDGS |
| F | *Documento de implementación y Verificación de Requisito 04* | SGDS - IVR 04 | PY | SDGS |
| F | *Documento de Especificación de la BD* | SGDS - DEBD | PY | SDGS |
| F | *Documento de Implementación de UI* | SGDS - DEUI | JS | SDGS |
| F | *Reporte del Desarrollo del Software* | SGDS - RDS | PY | SDGS |
| E | *Reporte del Segundo Sprint* | SGDS - RSS | DOCX | SDGS |
| F | *Documento de implementación y Verificación de Requisito 05* | SGDS - IVR 05 | PY | SDGS |
| F | *Documento de implementación y Verificación de Requisito 06* | SGDS - IVR 06 | PY | SDGS |
| F | *Documento de implementación y Verificación de Requisito 07* | SGDS - IVR 07 | PY | SGDS |
| F | *Documento de implementación y Verificación de Requisito 08* | SGDS - IVR 08 | PY | SGDS |
| F | *Documento de implementación y Verificación de Requisito 09* | SGDS - IVR 09 | PY | SGDS |
| E | *Documento de Pruebas del Software* | SGDS - DPS | DOCX | SGDS |
| E | *Reporte del Desarrollo del Software* | SGDS - RDS | DOCX | SGDS |
| E | *Reporte del Tercer Sprint* | SGDS - RTS | DOCX | SGDS |
| E | *Acta de cierre del proyecto* | SGDS - ACP | DOCX | SGDS |
| S | *Visual Code Studio* | VisualStudioCode | EXE | SGDS |
| E | *Documento de especificación de base de datos* | SRH-EBD | DOCX | SRH |
| E | *Documento de configuración de servidores y redes* | SRH-CSR | DOCX | SRH |
| E | *Documento de especificación de control de versiones 01* | SRH-ECV 01 | DOCX | SRH |
| E | *Documento de especificación de control de versiones 02* | SRH-ECV 02 | DOCX | SRH |
| E | *Documento de especificación de control de versiones 03* | SRH-ECV 03 | DOCX | SRH |
| E | *Documento de Especificación de Requisitos 01* | STA - RS 01 | DOCX | STA |
| E | *Documento de Especificación de Requisitos 02* | STA - RS 02 | DOCX | STA |
| E | *Documento de Especificación de Requisitos 03* | STA - RS 03 | DOCX | STA |
| F | *Código fuente del proyecto* | STA - CFP | CPP | STA |
| E | *Ejecutable del proyecto* | STA - EP | EXE | STA |
| S | *Microsoft Visual C++ 2022* | MicrosoftVisualC++2022 | EXE | STA |

*Fuente.* Elaboración propia

## Definición de Nomenclaturas

La tabla muestra una serie de casos y su correspondiente nomenclatura para la identificación de documentos utilizados en un proyecto. Los casos representan diferentes tipos de documentos, como políticas, planes, requisitos, diseños, reportes, retrospectives, implementación, verificación y actualización, y pruebas. La nomenclatura para cada caso se compone de una sigla del proyecto, siglas específicas para cada tipo de documento, y una correlación única que identifica cada documento de forma individual. Esta estructura de nomenclatura permite una fácil identificación y organización de los diferentes documentos utilizados en el proyecto.

*Tabla: Tabla de nomenclaturas*

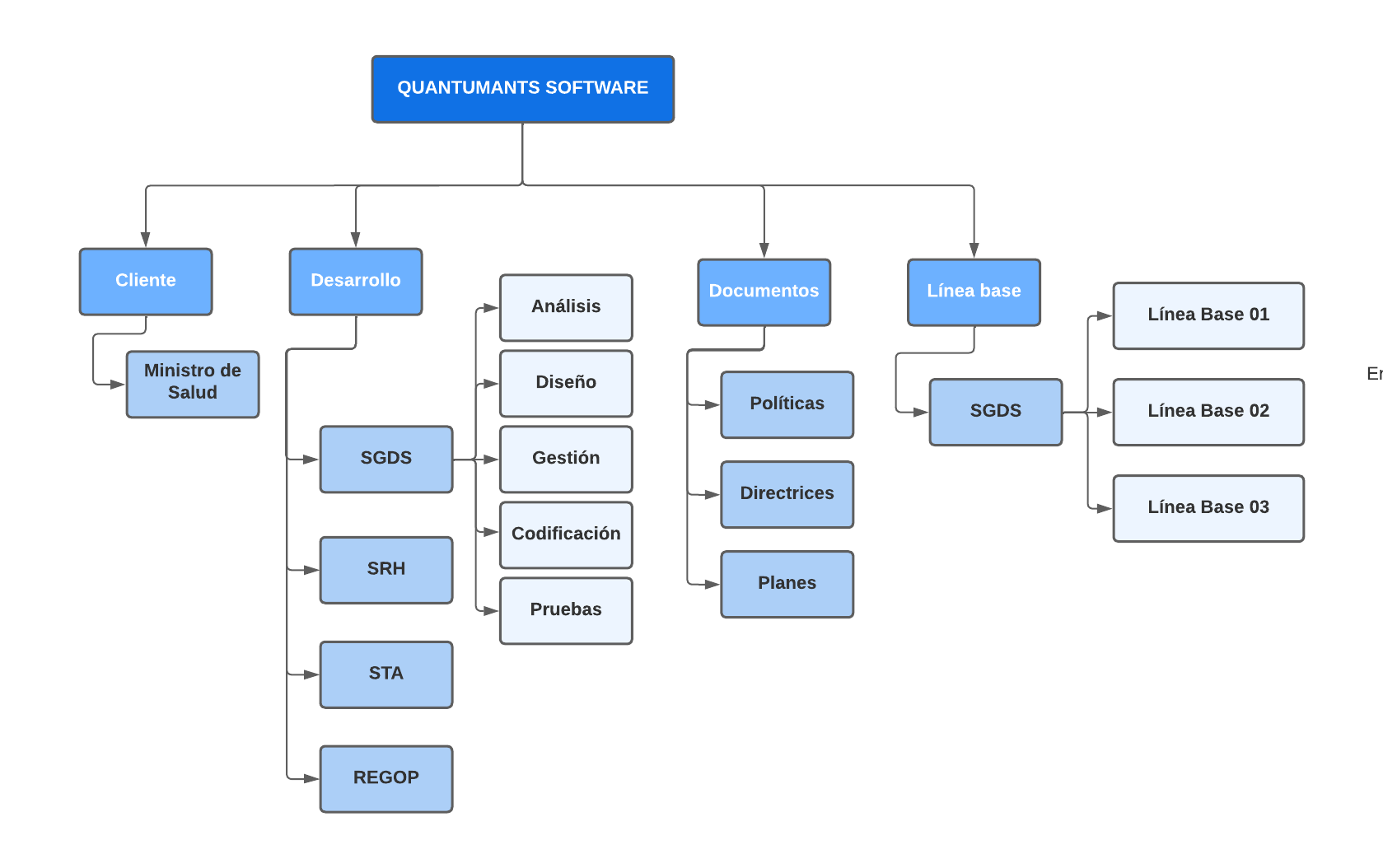
|  |  |  |
| --- | --- | --- |
| Casos | Nomenclatura | Descripción |
| caso 1 | Acrónimo del documento | Políticas, planes y otros documentos generales de la empresa |
| Caso 2 | Sigla del proyecto - Siglas del documento - Correlación | Requisitos del proyecto |
| Caso 3 | Sigla del proyecto - Siglas del documento - Correlación | Diseños del proyecto |
| Caso 4 | Sigla del proyecto . Siglas del documento - Número del sprint | Reportes de cada sprint del proyecto |
| Caso 5 | Sigla del proyecto - Siglas del documento - Número del sprint | Documento de retrospectiva del sprint del proyecto |
| Caso 6 | Sigla del proyecto - Siglas del documento - Correlación | Documentos de implementación, verificación y actualización del proyecto |
| Caso 7 | Sigla del proyecto - Siglas del documento - Correlación | Plan de pruebas del proyecto |

## Diseño de la Estructura del Repositorio

El repositorio de la empresa "QuantumAnts Software" cuenta con cuatro carpetas o librerías:

* **Cliente.-** Contiene los entregables actualizados que se les proporciona a los clientes, permitiendo llevar un registro detallado y facilitando la gestión de las relaciones comerciales.
* **Desarrollo.-** Alberga los proyectos de la empresa, incluyendo documentos, códigos fuente, diagramas, pruebas e informes, permitiendo mantener un registro detallado y gestionar eficientemente los elementos de cada proyecto.
* **Documentos.-** Contiene los documentos no relacionados directamente con los proyectos, como políticas, planes y directivas.
* **Línea base.-** Utilizada como punto de referencia para futuras revisiones y cambios del software, contiene elementos probados y verificados, garantizando la calidad y estabilidad del software a lo largo del tiempo.

**Figura 03**

*Diagrama de la estructura del repositorio de la empresa QuantumAnts Software*

*Fuente.* Elaboración propia

A continuación, se detallan las librerías del repositorio de la empresa "QuantumAnts Software" en función de su responsable, las actividades asociadas, los contenidos que albergan y los niveles de acceso correspondientes:

* **Librería Cliente:**
  + **Responsable**

Jefe de Proyecto: Saavedra Monterrey, Max

* + **Actividades**
    - Tener actualizados los entregables en su última versión
  + **Contenido**

Conformado por bibliotecas por proyecto: Entregables, Reportes, Documentos

* + - Ministro de Salud
  + **Accesos**

En cuanto a los accesos a esta librería de la empresa, el Jefe de Proyecto tiene acceso completo, lo que le permite leer, escribir y eliminar elementos.

Por otro lado, los desarrolladores tienen acceso limitado, ya que solo pueden leer elementos de la librería.

**Tabla 3**

*Tipo de acceso según rol para la librería Cliente*

|  |  |
| --- | --- |
| **ROL** | **TIPO DE ACCESO** |
| Jefe de Proyecto | * Leer * Escribir * Eliminar |
| Desarrolladores | * Leer |

*Fuente.* Elaboración propia

* **Librería Desarrollo:**
  + **Responsable**

Arquitecto de Software: Saavedra Monterrey, Max

* + **Actividades**
    - Mantener actualizadas la versión del software y su documentación actual
    - Incorporar las nuevas versiones aprobadas
  + **Contenido**

Conformado por bibliotecas por proyecto: Análisis, codificación, diseño, gestión y pruebas

* + - SGDS (Sistema de gestión de donantes de sangre)
    - SRH (Sistema de registro para hoteles)
    - STA (Sistema de tracking de aviones)
    - REGOP (Sistema de registro de operaciones contables)
  + **Accesos**

En cuanto a los accesos a esta librería de la empresa, el Arquitecto de Software tiene acceso completo, lo que le permite leer, escribir, ejecutar y eliminar elementos.

Por otro lado, los desarrolladores tienen acceso limitado, ya que solo pueden leer, escribir y ejecutar elementos de la librería, lo que les permite trabajar en los proyectos en los que están involucrados sin modificar otros elementos del repositorio.

**Tabla 4**

*Tipo de acceso según rol para la librería Desarrollo*

|  |  |
| --- | --- |
| **ROL** | **TIPO DE ACCESO** |
| Arquitecto de software | * Leer * Escribir * Ejecutar * Eliminar |
| Desarrolladores | * Leer * Escribir * Ejecutar |

*Fuente.* Elaboración propia

* **Librería Documentos:**
  + **Responsable**

Gerente de la Configuración: Quispe Cabello, Jose

* + **Actividades**

Check in y Check out de ítems pertenecientes a la biblioteca

* + **Contenido**
    - Políticas
    - Directrices
    - Planes (Como el Plan de la gestión de la configuración)
  + **Accesos**

En cuanto a los accesos a esta librería de la empresa, el Gerente de la Configuración tiene acceso completo, lo que le permite leer, escribir, ejecutar y eliminar elementos.

Por otro lado, los desarrolladores tienen acceso limitado, ya que solo pueden leer y ejecutar elementos de la librería.

**Tabla 5**

*Tipo de acceso según rol para la librería Documento*

|  |  |
| --- | --- |
| **ROL** | **TIPO DE ACCESO** |
| Gerente de la Configuración | * Leer * Escribir * Eliminar |
| Desarrolladores | * Leer |

*Fuente.* Elaboración propia

* **Librería Línea Base:**
  + **Responsable**

Gerente de la Configuración: Quispe Cabello, Jose.

* + **Actividades**

Mantener actualizadas las líneas base establecidas durante el transcurso del proyecto.

* + **Contenido**

Conformado por bibliotecas por proyecto: Hitos, Reportes

* + - Línea Base 01
    - Línea Base 02
    - Línea Base 03
  + **Accesos**

En cuanto a los accesos a esta librería de la empresa, el Gerente de la Configuración tiene acceso completo, lo que le permite leer, escribir, ejecutar y eliminar elementos.

Por otro lado, el jefe del proyecto también dispone de acceso completo; sin embargo, solo podrá eliminar elementos bajo autorización del gerente de configuración.

Por último, los desarrolladores tienen acceso limitado, ya que solo pueden leer y ejecutar elementos de la librería.

**Tabla 6**

*Tipo de acceso según rol para la librería Línea Base*

|  |  |
| --- | --- |
| **ROL** | **TIPO DE ACCESO** |
| Gerente de Configuración | * Leer * Escribir * Ejecutar * Eliminar |
| Jefe de proyecto | * Leer * Escribir * Ejecutar * Eliminar (con autorización del gerente de configuración) |
| Desarrolladores | * Leer * Ejecutar |

*Fuente.* Elaboración propia

## Definición de Líneas Base

**PROYECTO:** Sistema de Gestión de Donantes de Sangre (SGDS)

|  |  |  |
| --- | --- | --- |
| **LÍNEA BASE** | **HITO** | **ÍTEM (Nomenclatura)** |
| LB1 | 1 | * SGDS-PC.DOCX * SGDS-CP.XLS * SGDS-RS01.DOCX * SGDS-RS02.DOCX * SGDS-RS03.DOCX * SGDS-RS04.DOCX * SGDS-RS05.DOCX * SGDS-RS06.DOCX * SGDS-RS07.DOCX * SGDS-RS08.DOCX * SGDS-RS09.DOCX * SGDS-DAS01.DOCX * SGDS-DBD01.DOCX * SGDS-DEUI.DOCX * SGDS-GE.DOCX * SGDS.RDS-01.py * SGDS-RPS.01.DOCX |
| LB2 | 2 | * SGDS-IVR01.py * SGDS-IVR02.py * SGDS-IVR03.py * SGDS-IVBD01.py * SGDS-DEUI.js * SGDS-DEUI.DOCX * SGDS.RDS-02.py * SGDS-RPS.02.DOCX |
| LB3 | 3 | * SGDS-IVR04.py * SGDS-IVR05.py * SGDS-IVR06.py * SGDS-IVR07.py * SGDS-IVR08.py * SGDS-IVR09.py * SGDS-VABD01.db * SGDS-DEUI.js * SGDS-DEUI.DOCX * SGDS-DPS.DOCX * SGDS.RDS-03.py * SGDS-RPS.03.DOCX * SGDS-ACP.DOCX |

# Control

* 1. **Procedimientos de control:** Se establecerán los procedimientos para el control de los elementos de la configuración del software, incluyendo el seguimiento de cambios, la gestión de versiones, la gestión de ramas y la integración continua.
  2. **Repositorio:** Se establecerá un repositorio de control de versiones para el almacenamiento y seguimiento de los cambios en los elementos de la configuración del software. Se podrían utilizar herramientas de control de versiones como Git, Subversion, o AWS CodeCommit.
  3. **Gestión de versiones:** Se establecerá un proceso para la gestión de versiones, para controlar los cambios y las versiones de los elementos de la configuración del software, incluyendo la identificación de versiones, la creación de ramas y la gestión de ramas.
  4. **Integración continua:** Se establecerá un proceso de integración continua para el control de la configuración del software, para automatizar la compilación, las pruebas y la generación de versiones del software.
  5. **Gestión de cambios:** Se establecerá un proceso de gestión de cambios para el control de la configuración del software, para registrar y gestionar los cambios en los elementos de la configuración del software.
  6. **Auditoría de la configuración**: Se establecerá un proceso de auditoría periódica de la configuración del software, para garantizar que se están aplicando adecuadamente los procedimientos de control de la configuración del software.

# Estado

* **Identificación de los elementos de configuración (EC):** Se identificarán los diferentes componentes del sistema, como el código fuente, la base de datos, los documentos de diseño y requisitos, los manuales de usuario, entre otros. Cada uno de estos componentes se consideraría un EC.
* **Control de versiones:** Se utilizará un sistema de control de versiones para gestionar el código fuente y cualquier otro componente que cambie con el tiempo. Esto permitiría rastrear los cambios realizados a lo largo del tiempo y permitiría volver a versiones anteriores si fuera necesario.
* **Control de cambios:** Se establecerá un proceso formal para controlar los cambios en los EC. Este proceso incluiría la identificación de la necesidad de cambio, la evaluación del impacto del cambio en el sistema, la aprobación del cambio y su implementación.
* **Trazabilidad:** Se mantendrá un registro de todas las versiones de los EC, así como de todos los cambios realizados en el sistema. Esto permitiría realizar un seguimiento del historial de cambios y proporcionará una visión general de la evolución del sistema.
* **Gestión de la documentación:** Se establecerá un proceso para gestionar la documentación relacionada con el sistema, como los requisitos, los diseños y los manuales de usuario. Esta documentación se consideraría también como EC y se mantendría actualizada a medida que cambie el sistema.
* **Auditorías:** Se realizarán auditorías periódicas del sistema para asegurar que se estén siguiendo los procesos establecidos y que se estén cumpliendo los requisitos del sistema.

# Entrega y Gestión de Release

* **Planificación:** En esta fase, se definirán los objetivos de la entrega, el alcance y los requisitos de los usuarios, así como se identificarán los recursos necesarios y se programarán las actividades. Se establecerá un calendario de lanzamiento para asegurarse de que el software esté listo para su entrega en la fecha acordada.
* **Diseño y Desarrollo:** En esta fase, se realizará el diseño y desarrollo del software.
* **Pruebas y Validación:** En esta fase, se realizarán pruebas de calidad adicionales y se validará la funcionalidad del software para garantizar que cumpla con los requisitos del usuario y los estándares de calidad.
* **Preparación y Empaquetado:** En esta fase, se preparará el software para su distribución, lo que incluirá la creación de paquetes de instalación, la documentación y la creación de imágenes de disco. También se creará un manual de usuario que detalle el proceso de uso del software para los usuarios finales.
* **Distribución y Soporte:** En esta fase, se distribuirá el software a los usuarios finales y se proporcionará soporte técnico para la instalación, configuración y uso del software.

# Referencias bibliográficas

AWS Code Commit: Amazon Web Services, Inc. (s.f.). AWS CodeCommit. Recuperado el 19 de abril del 2023, de <https://aws.amazon.com/codecommit/>

Bazaar: Canonical Ltd. (s.f.). Bazaar. Recuperado el 19 de abril del 2023, de <https://bazaar.canonical.com/en/>

GitKraken: Axosoft LLC. (s.f.). GitKraken. Recuperado el 19 de abril del 2023, de <https://www.gitkraken.com/>

GitHub: GitHub. (s.f.). Recuperado el 19 de abril del 2023, de <https://github.com/>

La importancia de la Gestión de la Configuración del Software, en una Empresa

de Desarrollo. (s.f.)

<https://journal.espe.edu.ec/ojs/index.php/geeks/article/view/257>

Subversion: The Apache Software Foundation. (s.f.). Subversion. Recuperado el 19 de abril del 2023 de, <https://subversion.apache.org/>

Team Foundation Server: Microsoft Corporation. (s.f.). Team Foundation Server. Recuperado el 19 de abril del 2023 de, <https://visualstudio.microsoft.com/tfs/>