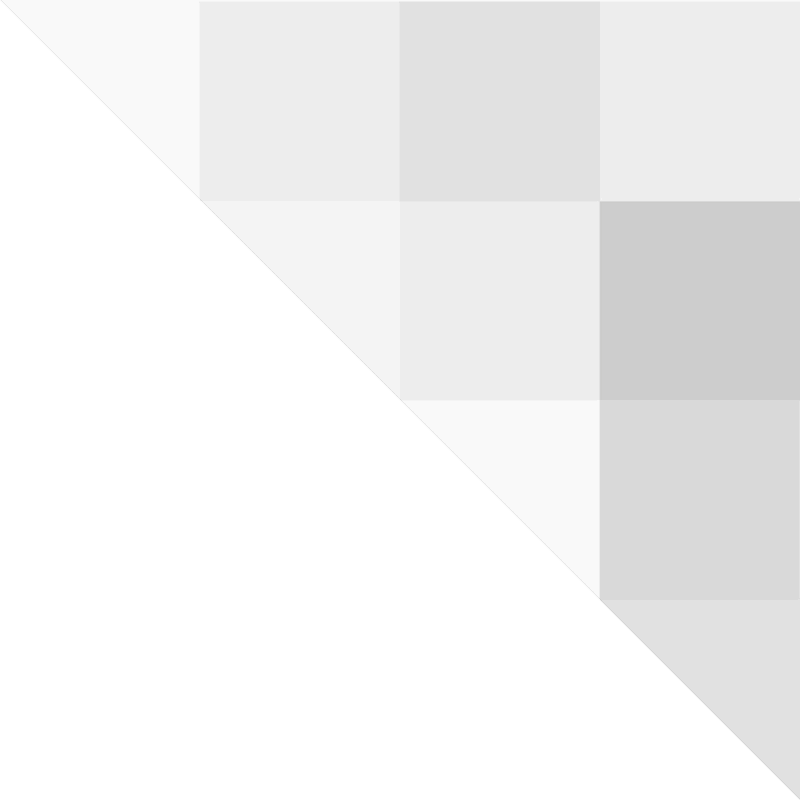
**SCM - UNMSM**

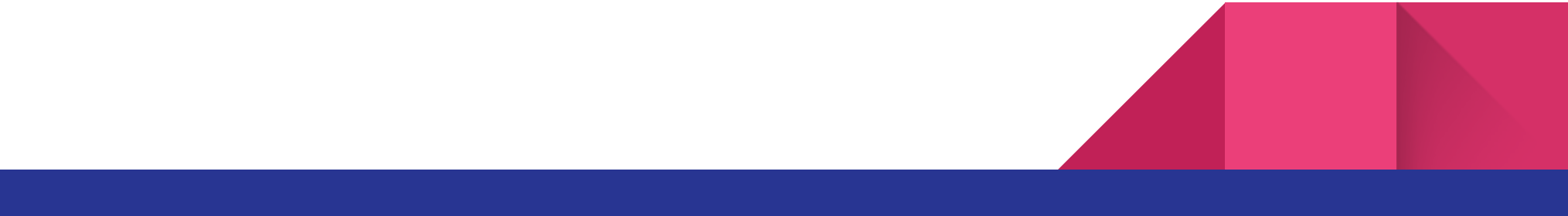
FDT-Consulting

# **Plan de Gestión de Configuración de Software**

# Integrantes (Grupo N° 5):

1. **Balarezo Ramos, Luis Jesús**
2. **Durand Caracuzma, Marlon Milko**
3. **Del Aguila Febres, Brayan Tadeo**
4. **Balceda Delgado, Mariana Alejandra**
5. **Canecillas Contreras, Juan Mariano**
6. **Soller Barnechea, Carlos Javier**
7. **Huarhua Piñas, Edson Sebastian**
8. **Justiniano Quispe, Diego André**

**Docente: Wong Portillo, Lenis Rossi**

****

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

## **Introducción**

### **a) Situación de la empresa**

FDT Consulting es una empresa dedicada al desarrollo de software a medida para empresas de diversos sectores. Desde su fundación hace más de una década, se ha destacado por su compromiso con la calidad y su capacidad para entregar proyectos de software a medida en tiempo y forma.

A lo largo de su trayectoria, FDT Consulting ha trabajado con clientes de diversos sectores, desde empresas del sector financiero hasta startups tecnológicas. Su equipo altamente capacitado y experimentado en diferentes áreas, les ha permitido enfrentar con éxito diversos desafíos tecnológicos y entregar soluciones personalizadas y de alta calidad a sus clientes. Su enfoque en la calidad, el compromiso y la excelencia les ha permitido posicionarse como una de las principales empresas de desarrollo de software en el mercado.

### **Clientes y sus proyectos:**

* **Empresa: "Distribuciones Alfalfa S.A"** (Distribuidora de productos alimenticios)

El objetivo es mejorar el proceso de distribución y entrega de productos a los clientes a través de la optimización del sistema logístico. Para lograrlo, se requiere de un sistema de gestión logística eficiente que permita el control del inventario, la planificación de rutas de entrega y el seguimiento en tiempo real del estado de los envíos.

Se propone desarrollar un software de gestión logística que gestione las rutas, entregas y seguimiento de paquetes en tiempo real. Este software incluirá un módulo de planificación de rutas y un sistema de seguimiento de paquetes en tiempo real, lo que permitirá a la empresa anticipar problemas y tomar decisiones informadas para optimizar la entrega de sus productos.

* **Empresa: "Corbata Legista S.A"** (Despacho de abogados)

Se busca mejorar la gestión de los casos y la comunicación con los clientes. Para lograrlo, se necesita un sistema de gestión de casos que permita el control eficiente de la información de los clientes, así como la gestión de documentos y tareas.

Una posible solución es la implementación de un sistema de gestión de casos que permita la carga de documentos, la gestión de tareas y la comunicación en línea con los clientes. Este software permitiría al despacho de abogados gestionar la información de sus clientes de manera más eficiente, así como comunicarse en línea con ellos en tiempo real, lo que reduciría los tiempos de respuesta y mejoraría la satisfacción del cliente. Este sistema también podría incluir recordatorios automatizados para fechas límite y citas en la corte, así como la capacidad de realizar un seguimiento de las horas facturables y generar facturas para los clientes.

* **Empresa: "Productos Verdes S.A."** (Empresa productora y distribuidora de productos orgánicos)

La empresa ha identificado la necesidad de mejorar la gestión de su inventario y la visibilidad de sus productos en línea. Para cumplir este objetivo, se requiere un sistema de gestión de inventario que permita la actualización automática de la información de los productos en la tienda en línea de la empresa.

En este sentido, se propone desarrollar un software de gestión de inventario que permita la actualización en tiempo real de los precios y las existencias en la tienda en línea. Esta solución permitirá a la empresa una mayor eficiencia en la gestión de su inventario, al permitir la actualización automática de los precios y las existencias en tiempo real. Además, este software reducirá los errores en la gestión de pedidos y mejorará la visibilidad de los productos en línea.

* **Empresa: "Mano Amiga S.A"** (Consultora de recursos humanos)

Se quiere optimizar la selección de candidatos y la gestión de sus procesos de reclutamiento. Para lograrlo, se necesitará un sistema de gestión de talentos que le permita llevar un control eficiente de la información de los candidatos y gestionar los procesos de selección de forma automatizada.

Se desarrolla un sistema de gestión de talentos que permita la carga de currículums, la realización de entrevistas en línea y la gestión de evaluaciones de desempeño. Este software permitiría a la consultora de recursos humanos una gestión más eficiente de los procesos de reclutamiento, gracias a la implementación de las funcionalidades mencionadas anteriormente, así como la automatización de las tareas administrativas para reducir el tiempo de espera y mejorar la eficiencia del proceso de selección.

* **Empresa: "Agropecuaria del Sur S.A"** (Empresa productora y distribuidora de productos agrícolas)

Se necesita mejorar la gestión de la producción y la planificación de las cosechas de Agropecuaria del Sur. Con ese propósito ,es necesario un sistema de gestión de la producción que le permita llevar un control eficiente de la producción de productos agrícolas y planificar las cosechas de forma óptima, con el objetivo de aumentar la productividad y disminuir la ineficiencia .

Con lo mencionado anteriormente, tenemos como objetivo desarrollar un sistema de gestión de la producción agrícola que permita la planificación y seguimiento de los cultivos. Este software tendrá como funcionalidad un módulo de planificación de cosechas que permitiría a la empresa optimizar la producción y minimizar los costos, así como un sistema de seguimiento de los cultivos que permitirá anticipar problemas y tomar decisiones informadas para mejorar la eficiencia del proceso productivo.

En FDT-Consulting, nos hemos encontrado con desafíos significativos en relación con la gestión de la configuración de software en nuestros proyectos, lo que ha tenido un impacto directo en nuestra relación con los clientes.

Uno de los problemas recurrentes ha sido el control inadecuado de versiones del software. Esto ha llevado a situaciones de pérdida y sobreescritura de código, lo que ha generado confusión y retrasos en la entrega de los proyectos. Como resultado, nuestros clientes han experimentado frustración y han tenido que lidiar con la falta de funcionalidad y estabilidad en sus sistemas.

Otro desafío ha sido la integración de diferentes componentes de software. En ocasiones, las correcciones de implementación se han realizado de manera tardía, lo que ha generado incompatibilidad entre los diversos elementos del sistema. Esto ha llevado a problemas de funcionamiento, falta de coherencia y una disminución en la calidad general del producto final.

Estos problemas de gestión de la configuración de software han tenido consecuencias negativas tanto para nuestros proyectos como para nuestros clientes. Hemos enfrentado deficiencias en la documentación, lo que dificulta la comprensión y el mantenimiento del software a lo largo del tiempo. Además, hemos experimentado la necesidad de rehacer el trabajo previamente realizado, lo que ha llevado a un aumento en los costos y a retrasos en los plazos de entrega.

En FDT-Consulting somos conscientes de la importancia de abordar estos desafíos de manera efectiva. Estamos trabajando en mejorar nuestros procesos de gestión de la configuración de software, implementando herramientas y prácticas más sólidas para garantizar la correcta versión, integración y documentación de nuestros proyectos. Nuestro objetivo es brindar a nuestros clientes soluciones de software confiables, estables y de alta calidad, superando los problemas que hemos enfrentado en el pasado.

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

El propósito de implementar la gestión de la configuración del software en FDT-Consulting es establecer un conjunto de prácticas y herramientas sólidas para gestionar de manera eficiente y efectiva todos los aspectos relacionados con la configuración del software en nuestros proyectos.

Nuestro objetivo principal es garantizar la integridad, la trazabilidad y la disponibilidad de los elementos de configuración del software a lo largo de su ciclo de vida. Esto implica gestionar de manera adecuada los cambios, versiones, variantes y componentes del software, así como mantener un control exhaustivo sobre la documentación y los recursos asociados.

Al implementar la gestión de la configuración del software, buscamos lograr los siguientes beneficios:

* **Control de versiones y cambios**: Establecer un sistema robusto de control de versiones que permita rastrear y gestionar todas las modificaciones realizadas en el software a lo largo del tiempo. Esto asegura que se pueda acceder a versiones anteriores, revertir cambios si es necesario y mantener un historial completo de las evoluciones del software.
* **Coherencia y consistencia**: Garantizar la coherencia y consistencia de los componentes del software, evitando incompatibilidades y conflictos en la integración de diferentes elementos. Esto mejora la estabilidad y la calidad del producto final, así como la eficiencia en el desarrollo y la resolución de problemas.
* **Seguimiento y trazabilidad:** Proporcionar una visibilidad completa de los cambios y las relaciones entre los distintos elementos del software. Esto permite un seguimiento efectivo de los problemas, errores y requisitos, facilitando la identificación y la resolución de problemas de manera rápida y precisa.
* **Eficiencia en la colaboración**: Facilitar la colaboración entre los miembros del equipo y los clientes al proporcionar un entorno estructurado para el intercambio y la gestión de los artefactos del software. Esto mejora la comunicación, evita la duplicación de esfuerzos y asegura que todos los involucrados tengan acceso a la información correcta y actualizada en todo momento.
* **Reducción de riesgos y costos**: Minimizar los riesgos asociados con la configuración del software, como la pérdida de código, la falta de documentación o la incompatibilidad de versiones. Al tener un control adecuado, podemos evitar retrasos, reprocesos y gastos innecesarios, optimizando así la eficiencia y la rentabilidad de nuestros proyectos.

### **c) Benchmarking de herramientas**

| **Criterios/Características** | **BitBucket** | **Git** | **Mercurial** | **Perforce** |
| --- | --- | --- | --- | --- |
| Gratuito | Sí | Sí | Sí(hasta 25 usuarios) | No(solo ofrece una versión de prueba gratuita) |
| Multiplataforma | Sí | Sí | Sí | Sí |
| Interfaz Web | Sí | No(depende de herramientas externas) | No(depende de herramientas externas) | Sí |
| Integration/Plugs-ins IDE | Sí(integración con IDEs como Eclipse, IntelliJ, Visual Studio) | Sí(amplia gama de plugins para IDEs) | Sí(integración con IDEs como Eclipse, NetBeans) | Sí(integración con IDEs como Visual Studio, Eclipse) |
| Soporte para diferentes lenguajes | Sí | Sí | Sí | Sí |
| Réplica de repositorio | Sí(mediante la funcionalidad "mirroring") | Sí(mediante la funcionalidad "mirroring") | Sí(mediante la funcionalidad "clone bundle") | Sí(mediante la funcionalidad "replication") |
| Unicode | Sí | Sí | Sí | Sí |
| Mantenimiento | Sí(servicio en la nube) | Sí(software libre) | Sí(software libre) | Sí(software comercial) |

**Análisis**:

* **Git:**

Es un sistema de control de versiones distribuido que fue creado por Linus Torvalds en 2005. Es ampliamente utilizado por desarrolladores de todo el mundo y se ha convertido en el estándar de facto para el control de versiones. Estas son algunas de sus características:

* **Distribuido**: Git es un sistema de control de versiones distribuido, lo que significa que cada usuario tiene una copia completa del repositorio en su máquina local. Esto permite flujos de trabajo de desarrollo más rápidos y flexibles, ya que los desarrolladores pueden trabajar sin conexión y sincronizar los cambios con otros cuando vuelven a estar en línea.
* **Velocidad**: Git está diseñado para ser rápido, incluso con repositorios grandes y flujos de trabajo de bifurcación complejos. Esto se logra a través de funciones como la compresión delta, la clonación superficial y el procesamiento en paralelo.
* **Código abierto**: Git es un proyecto de código abierto, lo que significa que el código fuente está disponible gratuitamente y cualquiera puede modificarlo y distribuirlo. Esto ha ayudado a impulsar la innovación y la adopción de la tecnología.
* **Ramificación y fusión**: Git proporciona potentes capacidades de bifurcación y fusión, lo que permite flujos de trabajo de desarrollo complejos y colaboración entre equipos. Esto es esencial para proyectos más grandes con muchos desarrolladores y colaboradores.
* **Integración**: Git se integra con muchas otras herramientas y servicios, como GitHub, GitLab y Bitbucket, así como con varios IDE y editores de texto. Esto lo hace fácil de usar y flexible para diferentes entornos de desarrollo.
* **Seguridad**: Git proporciona sólidas funciones de seguridad, como la autenticación y el cifrado, para proteger los datos confidenciales y evitar el acceso no autorizado a los repositorios.
* **Mercurial**

Es una herramienta de control de versiones distribuido que permite a los desarrolladores rastrear y administrar cambios en el código fuente de un proyecto.

Algunas de las características de Mercurial son:

* **Personalizable**: Es personalizable debido a que puede adaptarse a grandes flujos de trabajo de desarrollo y a necesidades específicas del proyecto.
* **Escalable**: Mercurial puede manejar proyectos de cualquier tamaño, desde pequeños proyectos de una sola persona hasta grandes proyectos de miles de desarrolladores.
* **Control de versiones**: Mercurial es un sistema de control de versiones que permite a los desarrolladores mantener un historial completo de los cambios realizados en el código fuente de un proyecto.
* **Soporte para grandes archivos**: Mercurial puede manejar grandes archivos y proyectos que contienen una gran cantidad de datos.
* **Extensibilidad**: Mercurial es altamente extensible y ofrece una gran cantidad de herramientas y complementos que pueden mejorar su funcionalidad y adaptarse a diferentes necesidades de desarrollo.
* **Comandos avanzados:** Mercurial ofrece una amplia gama de comandos avanzados que permiten a los desarrolladores realizar tareas específicas, como la búsqueda en el historial de cambios, la comparación de versiones y la edición de cambios.
* **Distribuido:** Mercurial es un sistema de control de versiones distribuido, lo que significa que cada desarrollador tiene una copia completa del repositorio en su propio equipo, lo que permite a los desarrolladores trabajar de forma independiente y sin conexión a Internet.
* **Perforce**

Es una herramienta de control de versiones centralizada que se ha utilizado durante décadas en una amplia variedad de aplicaciones de software, desde el desarrollo de videojuegos hasta el diseño de chips.

Algunas de las principales características de Perforce incluyen:

* **Escalabilidad:** Perforce ha sido diseñado para manejar grandes conjuntos de datos, lo que lo hace especialmente adecuado para el desarrollo de software de alta velocidad o proyectos de ingeniería de gran tamaño.
* **Control de acceso:** Perforce permite a los administradores de sistemas establecer políticas de seguridad detalladas para controlar quién puede acceder a qué partes del repositorio de código.
* **Integración**: Perforce se integra con una amplia variedad de herramientas de desarrollo de software, incluyendo IDE, sistemas de compilación, herramientas de revisión de código y sistemas de seguimiento de problemas.
* **Historial completo de cambios:** Perforce mantiene un historial completo de todos los cambios realizados en el código fuente, lo que permite a los desarrolladores rastrear cambios y volver a versiones anteriores del código en caso de ser necesario.
* **Extensibilidad:** Perforce proporciona una API que permite a los desarrolladores crear herramientas personalizadas para integrarse con su sistema de control de versiones.
* **BitBucket**

Es una plataforma de desarrollo de software basada en la nube que proporciona una herramienta de control de versiones para equipos de desarrollo.

Algunas de las principales características de Bitbucket incluyen:

* **Soporte para Git y Mercurial**: Bitbucket soporta tanto Git como Mercurial como motores de control de versiones.
* **Control de acceso**: Bitbucket permite a los administradores de sistemas establecer políticas de seguridad detalladas para controlar quién puede acceder a qué partes del repositorio de código.
* **Comentarios en línea:** Bitbucket permite a los desarrolladores hacer comentarios directamente en el código fuente, lo que facilita la colaboración y la revisión de código.

## **Identificación**

### **2.1 Clasificación de Ítems**

La clasificación de elementos de configuración es un proceso vital para asignar identificadores únicos a cada elemento. Este proceso utiliza un sistema jerárquico de etiquetas que se basa en la función y la relación de los elementos con otros. La clasificación es esencial para garantizar una gestión adecuada de los elementos relevantes a lo largo del ciclo de vida del software y simplificar la gestión de la configuración. Estos elementos se dividen en tres categorías principales.

* La primera categoría son los "ítems por evolución", los cuales están relacionados con documentos y archivos ejecutables.
* La segunda categoría son los "ítems fuente", que están relacionados con el código fuente y los archivos necesarios para compilar una aplicación.
* Por último, tenemos los "ítems de soporte", que están relacionados con los sistemas operativos y el software base.

**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 (PROJECT CHARTER)* | SGI-PC.DOCX | DOCX | SGI |
| E | *Documento de Elicitación de Requerimientos* | SGI-PER.DOCX | .DOCX | SGI |
| E | *Documento de Especificación de Requerimiento 1* | SGI-DER1.DOCX | .DOCX | SGI |
| E | *Documento de Especificación de Requerimiento 2* | SGI-DER2.DOCX | .DOCX | SGI |
| E | *Documento de Especificación de Requerimiento 3* | SGI-DER3.DOCX | .DOCX | SGI |
| E | *Documento de Análisis de Riesgos* | SGI-DAR.DOCX | .DOCX | SGI |
| E | *Lista de Historias de Usuario* | SGI-LHU.DOCX | .DOCX | SGI |
| E | *Documento de criterios de aceptación* | SGI-DCA.DOCX | .DOCX | SGI |
| E | *Documento de Especificación de UI* | SGI-DEUI1.DOCX | .DOCX | SGI |
| E | *Documento de Especificación de la BD* | SGI-DEBD1.DOCX | .DOCX | SGI |
| E | *Documento de Arquitectura del Software* | SGI-DAS1.DOCX | .DOCX | SGI |
| E | *Reporte del Primer Sprint* | SGI-RPS.DOCX | .DOCX | SGI |
| E | *Acta de Finalización Hito 1* | SGI-AFH1.DOCX | .DOCX | SGI |
| E | *Documento de Validación de Requisitos y Especificaciones* | SGI-DVRE.DOCX | .DOCX | SGI |
| E | *Documento de Especificación de Requerimiento 4* | SGI-DER4.DOCX | .DOCX | SGI |
| E | *Documento de Especificación de Requerimiento 5* | SGI-DER5.DOCX | .DOCX | SGI |
| E | *Documento de Especificación de Requerimiento 6* | SGI-DER6.DOCX | .DOCX | SGI |
| E | *Documento de Especificación de UI* | SGI-DEUI2.PDF | .PDF | SGI |
| E | *Documento de Especificación de la BD* | SGI-DEBD2.DOCX | .DOCX | SGI |
| F | *Modulo de Página Principal* | SGI-MPP.DOCX | .DOCX | SGI |
| F | *Módulo de Autenticación de Usuario* | SGI-MAU.DOCX | .DOCX | SGI |
| F | *Módulo de Registro de Incidencias* | SGI-MRI.DOCX | .DOCX | SGI |
| E | *Documento de Pruebas de validación del Software* | SGI-DPVS.DOCX | .DOCX | SGI |
| F | *Documento de Análisis de la calidad del software* | SGI-DACS2.DOCX | .DOCX | SGI |
| F | *Reporte del Desarrollo del Software* | SGI-RDS2.DOCX | .DOCX | SGI |
| F | *Reporte del Segundo Sprint* | SGI-RSS.DOCX | .DOCX | SGI |
| F | *Acta de Finalización Hito 2* | SGI-AFH2.DOCX | .DOCX | SGI |
| F | *Módulo de asignación de personal especializado* | SGI-MAPE.DOCX | .DOCX | SGI |
| E | *Documento de Especificación de Requerimiento 7* | SGI-DER7.DOCX | .DOCX | SGI |
| E | *Documento de Especificación de Requerimiento 8* | SGI-DER8.DOCX | .DOCX | SGI |
| E | *Documento de Especificación de UI* | SGI-DEUI3.PDF | .PDF | SGI |
| E | *Documento de Especificación de la BD* | SGI-DEBD3.DOCX | .DOCX | SGI |
| E | *Documento de Arquitectura del Software* | SGI-DAS.DOCX | .DOCX | SGI |
| E | *Manual de usuario* | SGI-MU.DOCX | .DOCX | SGI |
| E | *Documento de Pruebas del Software* | SGI-DPS.DOCX | .DOCX | SGI |
| E | *Documento de Análisis de Calidad del Software* | SGI-DACS3.DOCX | .DOCX | SGI |
| E | *Reporte del Desarrollo del Software* | SGI-RDS3.DOCX | .DOCX | SGI |
| E | *Reporte del Tercer Sprint* | SGI-RTS.DOCX | .DOCX | SGI |
| E | *Acta de Finalización Hito 3* | SGI-AFH3.DOCX | .DOCX | SGI |
| E | *Acta de cierre del proyecto* | SGI-ACP.DOCX | .DOCX | SGI |

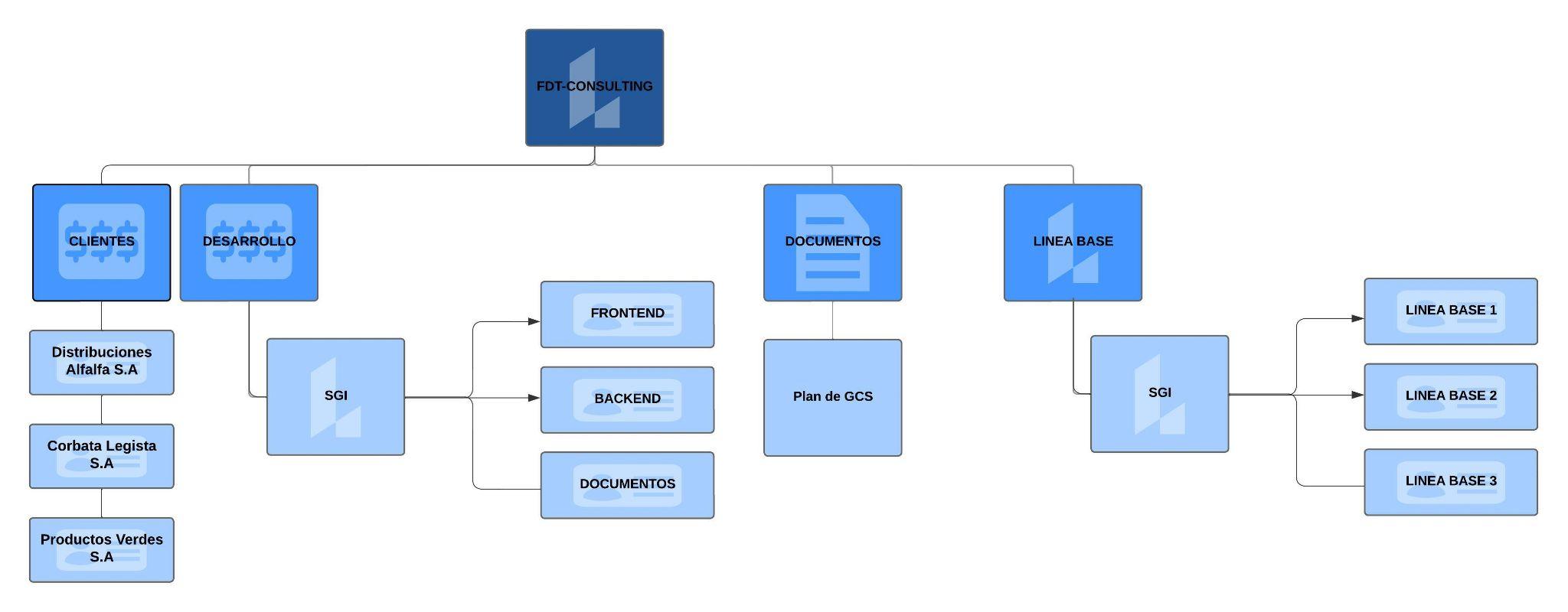
### **2.2 Definición de nomenclaturas**

| **Caso** | **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** | PGCS (Plan de Gestión de Configuración De Software) |
| 2 | Para ítems que se encuentran en el proyecto, se usa la siguiente fórmula para nombrarlos.    **ACRÓNIMO DEL PROYECTO + “-” + ACRÓNIMO DEL ELEMENTO** | SGI-PER  (Plan de Elicitación de Requerimientos) |
| 3 | Para ítems que hacen referencia al mismo tipo de elemento, con distinta numeración.    **ACRÓNIMO DEL PROYECTO + “-” + ACRÓNIMO DEL ELEMENTO + “NUMERACIÓN CORRELATIVA AL DOCUMENTO”** | SGI-DER4.DOCX  SGI-DER5.DOCX  SGI-DER6.DOCX  (Documento de Especificación de Requisito) |
| 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** | **Posibles casos:**  SGI-DAS-SOF  (Documento de Análisis de Software)  SGI-DAS-SIS  (Documento de Arquitectura de Sistemas) |

### 

### **2.3 Diseño de Estructura del Repositorio**

A través del siguiente diagrama se ilustra cómo FDT-Consulting organiza la estructura de su repositorio. Esta organización es fundamental, ya que una estructuración adecuada facilita la organización, el mantenimiento, la escalabilidad y la colaboración en el desarrollo de software. Estos factores son de gran importancia, ya que permiten gestionar de manera clara y sistemática todos los proyectos desarrollados o en proceso de desarrollo por parte de FDT-Consulting.



### 

### **2.4 Definición de Líneas Base**

**Línea Base 1:**

La Línea Base 1 comprende los entregables necesarios para la especificación de los requisitos y la realización estructurada del proyecto. Incluye una reunión con el Stakeholder, la generación de un proyecto innovador, la elaboración del Plan de Proyecto y el Cronograma del Proyecto. También implica la creación de un repositorio del proyecto y la planificación de la elicitación de requerimientos.

Además, se especifican los primeros tres requisitos del software: la creación de cuenta, la gestión de roles y la autenticación de usuarios. Se lleva a cabo un análisis de riesgos y se valida las historias de usuario, estableciendo los criterios de aceptación. Se propone un diseño inicial de la interfaz de usuario (UI) y se especifica el diseño de la base de datos, así como la arquitectura y diseño del software.

Se realiza una retrospectiva del sprint y se documenta el acta de finalización del Hito 1. Por último, se realiza la carga de la Línea Base 1, que incluye todos los documentos y entregables mencionados anteriormente.

**Línea Base 2:**

La Línea Base 2 se enfoca en la revisión, análisis y validación de los requisitos y especificaciones del software. Incluye la especificación de tres nuevos requisitos del software: el registro de incidencias, la asignación de incidencias y la priorización de incidencias. También se verifica y actualiza el documento de especificación de la interfaz de usuario (UI) y el documento de especificación de la base de datos.

Además, se realiza la codificación de la página principal, la autenticación de usuario (registro) y el registro de incidencias. Se llevan a cabo pruebas de validación del software y se realiza un análisis de la calidad del mismo. Se reporta el estado actual del software y se realiza una retrospectiva del sprint.

**Línea Base 3:**

La Línea Base 3 se centra en el desarrollo de la funcionalidad relacionada con la asignación de personal encargado. Además, se especifican dos nuevos requisitos del software: el seguimiento de incidencias y la generación de reportes.

Se verifica y finaliza el documento de especificación de la interfaz de usuario (UI) y el documento de especificación de la base de datos. También se verifica y finaliza la arquitectura y diseño del software. Se genera la documentación destinada al usuario para facilitar su comprensión y uso del sistema.

Se llevan a cabo pruebas de seguridad del software y se realiza un análisis de la calidad del mismo. Se reporta el estado actual del software y se realiza una retrospectiva del sprint.

Se documenta el acta de finalización del Hito 3 y se procede a elaborar el acta de cierre del proyecto. Finalmente, se realiza la carga de la Linea Base 3, que incluye todos los documentos y entregables mencionados anteriormente.

**2.5 Inventario de Ítems**

| **Línea Base** | **Hito** | **Ítem** | **Nomenclaturas** |
| --- | --- | --- | --- |
| **Línea Base 1** | 1 | * Plan de Proyecto (PROJECT CHARTER) * Cronograma del Proyecto * Repositorio GitHub * Documento de Elicitación de Requerimientos * Documento de Especificación de Requerimiento 1 * Documento de Especificación de Requerimiento 2 * Documento de Especificación de Requerimiento 3 * Documento de Análisis de Riesgos * Lista de Historias de Usuario * Documento de criterios de aceptación * Documento de Especificación de UI * Documento de Especificación de la BD * Documento de Arquitectura del Software * Reporte del Primer Sprint * Acta de Finalización Hito 1 | SGI-PC.DOCX  SGI-CP.XLS  SGI-PER.DOCX  SGI-DER1.DOCX  SGI-DER2.DOCX  SGI-DER3.DOCX  SGI-DAR.DOCX  SGI-LHU.DOCX  SGI-DCA.DOCX  SGI-DEUI1.DOCX  SGI-DEBD1.DOCX  SGI-DAS1.DOCX  SGI-RPS.DOCX  SGI-AFH1.DOCX |
| **Línea Base 2** | 2 | * Documento de Validación de Requisitos y Especificaciones * Documento de Especificación de Requerimiento 4 * Documento de Especificación de Requerimiento 5 * Documento de Especificación de Requerimiento 6 * Documento de Especificación de UI * Documento de Especificación de la BD * Modulo de Página Principal * Módulo de Autenticación de Usuario * Módulo de Registro de Incidencias * Documento de Pruebas de validación del Software * Documento de Análisis de la calidad del software * Reporte del Desarrollo del Software * Reporte del Segundo Sprint * Acta de Finalización Hito 2 | SGI-DVRE.DOCX  SGI-DER4.DOCX  SGI-DER5.DOCX  SGI-DER6.DOCX  SGI-DEUI2.PDF  SGI-DEBD2.DOCX  SGI-MPP.DOCX  SGI-MAU.DOCX  SGI-MRI.DOCX  SGI-DPVS.DOCX  SGI-DACS2.DOCX  SGI-RDS2.DOCX  SGI-RSS.DOCX  SGI-AFH2.DOCX |
| **Línea Base 3** | 3 | * Módulo de asignación de personal especializado * Documento de Especificación de Requerimiento 7 * Documento de Especificación de Requerimiento 8 * Documento de Especificación de UI * Documento de Especificación de la BD * Documento de Arquitectura del Software * Manual de usuario * Documento de Pruebas del Software * Documento de Análisis de Calidad del Software * Reporte del Desarrollo del Software * Reporte del Tercer Sprint * Acta de Finalización Hito 3 * Acta de cierre del proyecto | SGI-MAPE.DOCX  SGI-DER7.DOCX  SGI-DER8.DOCX  SGI-DEUI3.PDF  SGI-DEBD3.DOCX  SGI-DAS.DOCX  SGI-MU.DOCX  SGI-DPS.DOCX  SGI-DACS3.DOCX  SGI-RDS3.DOCX  SGI-RTS.DOCX  SGI-AFH3.DOCX  SGI-ACP.DOCX |

## **Control**

## **Estado**

## **Auditoría de la SCM**

## **Entrega y Gestión de Release**