**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**ESCUELA PROFESIONAL DE INGENIERÍA DE SOFTWARE**

****

**Equipo 6**

***“Plan de Gestión de la configuración del software”***

- Artadi Ruiz, Gonzalo Joaquin 17200259

-Caballero Meza Omar Vicente 17200322

-Marcos Vásquez Eduardo Manuel 17200286

-Ortiz Quispe, Akcel Eduardo 20200278

-Santa Cruz Pachas, Edward Grover 21200270

-Rojas Huamaní, Percy Ares 21200240

Asignatura: Gestión de Configuración y Mantenimiento

Docente: Lenis Rossi Wong Portillo

**LIMA-PERÚ**

**2023**

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

**1. Introducción**

**a) Situación de la empresa**

La consultora de tecnología BERTELL ha estado operando en el mercado por más de 10 años, brindando soluciones de software personalizadas a una variedad de clientes en diferentes sectores, incluyendo finanzas, salud y comercio electrónico. Durante este tiempo, hemos desarrollado una amplia gama de sistemas y soluciones innovadoras que nos han permitido consolidar nuestra posición en el mercado y adquirir la experiencia necesaria para abordar proyectos de gran envergadura.

Entre los proyectos más destacados que hemos llevado a cabo se encuentran aquellos que desarrollamos en colaboración con Caterpillar, el Ministerio de Salud y varias empresas líderes en distintas industrias. Estos proyectos abarcaron diferentes áreas, desde la implementación de sistemas avanzados de gestión de inventarios hasta el desarrollo de soluciones de software personalizadas para mejorar la eficiencia y productividad de los procesos de negocio.

En BERTELL, estamos comprometidos con la excelencia y la satisfacción del cliente. Por ello, nos esforzamos constantemente por mejorar nuestros procesos, optimizar nuestros recursos y mantenernos al día con las últimas tendencias y tecnologías del mercado. Nuestro enfoque centrado en el cliente nos permite entender las necesidades específicas de cada organización y desarrollar soluciones de software personalizadas que se ajusten a sus requerimientos y objetivos y gracias a la capacidad de nuestros empleados que se encuentran capacitados y tienen experiencia en el desarrollo de sistemas pudimos cumplir con los diversos retos de cada proyecto

**Problemática de la empresa ;**

* Durante el proyecto Caterpillar, se presentó una problemática relacionada con el cálculo del tiempo necesario para completar cada uno de los requerimientos. Este problema tuvo como consecuencia que la entrega del proyecto se retrasara, lo que a su vez generó un aumento en los gastos para la empresa.
* Durante el proyecto ISSALUD, nos encontramos con un sistema basado en Java, pero se trataba de una versión muy antigua. A pesar de que recomendamos actualizar el sistema para mejorar su eficiencia y seguridad, el beneficiario se negó a realizar la actualización. Por lo tanto, nos vimos en la necesidad de trabajar con las mismas versiones que ya existían. Además, debido a la antigüedad del sistema y a la dificultad de no poder usar la documentación actual, enfrentamos varios desafíos durante la ejecución del proyecto. En particular, encontramos vulnerabilidades en la seguridad del sistema que ponían en riesgo la integridad de los datos.
* Durante la ejecución del proyecto SuperShop, se presentó una problemática relacionada con la formulación de los requerimientos. En particular, se detectó una mala formulación de los mismos, lo que generó que, al momento de presentar la primera versión del producto, el cliente no estuviera satisfecho con los resultados obtenidos. Esta situación generó una serie de consecuencias negativas para ambas partes, ya que se tuvieron que realizar cambios adicionales para satisfacer las necesidades del cliente, lo que a su vez provocó un aumento en los gastos para la empresa y el cliente, y un retraso en el plazo de entrega.

**b) Propósito del plan:**

Nuestro propósito sería el de asegurar que nuestro software sea confiable, eficiente y esté siempre disponible para todos los requerimientos de nuestros clientes. Al implementar la gestión de la configuración del software, podemos lograr los siguientes objetivos:

Controlar los cambios en el software: La gestión de la configuración del software nos permite controlar los cambios que se realizan en nuestro programas, esto nos permite mantener la estabilidad y la calidad siempre en estado óptimo.

Identificar los componentes del software: En este aspecto la gestión nos permite identificar los diferentes componentes del software y su relación unos con otros componentes de este. Esto permite entender cómo funciona el software y hacer mejoras en el mismo.

Reducir errores: La gestión de la configuración del software permite detectar errores de gestión de forma temprana, lo que ayuda a poder resolverlos en un tiempo reducido ayudando así a la productividad y la eficiencia del software

En resumen, la implementación de la gestión de la configuración del software en nuestra empresa nos permite tener un mayor control sobre nuestro software, mejorando la calidad del mismo, reduciendo errores y mejorando la eficiencia en el desarrollo y mantenimiento.

**c) Benchmarking de herramientas**

Evaluación de criterios para las herramientas respecto a soporte multiplataforma, flexibilidad en la gestión de ramas, escalabilidad, aprendizaje intuitivo, integración de herramientas de terceros y costo.

| **Criterios** | **Git**  **[ 1]** | **GitKraken**  **[ 2]** | **TFS**  **[3 ]** | **Bazaar**  **[4]** |
| --- | --- | --- | --- | --- |
| Soporte Multiplataforma | Si | Si | No | Si |
| Flexibilidad en la gestión de ramas | Si | Si | No | Si |
| Escalabilidad | Si | Si | Si | Si |
| Intuitivo | No | Sí | No | No |
| Integración con herramientas de terceros | Sí | Sí | Sí | Sí |
| Costo | No | Sí | Sí | No |

**d) Herramienta elegida : Bazaar**

**Descripción:**

Bazaar es un sistema de control de versiones distribuido, de origen open source, desarrollado por Canonical Ltd. [1]

Esta herramienta permite a los desarrolladores trabajar en equipo de manera colaborativa en el mismo proyecto, registrando todos los cambios realizados en el código fuente y manteniendo un historial completo de revisiones. También facilita la integración y fusión de cambios realizados por diferentes personas en el mismo proyecto.[1]

Puede ser instalado en varias plataformas mediante sus respectivos gestores de paquetes [1]:

* Debian, Ubuntu
* RHEL, CentOS, Fedora
* -Windows
* Mac OS X

**Ventajas [1]**

* **Adaptable:** Orientado a la mayoría de los trabajos de desarrollo de software (local y/o remoto).
* **Amigable:** Orientado principalmente a personas que recién empiezan a hacer uso de este tipo de herramientas, enfocado principalmente
* **Rápido:** Permite obtener rápidamente sus ventajas y características sin afectar al proceso de desarrollo.
* **Eficiente:** Su almacenaje utiliza un formato altamente eficiente y limpio.
* **Ligero:** No precisa tener un servidor dedicado.
* **Extensible:** Posee una arquitectura basada en plugins, aspecto que permite dotar de mayores características específicas y acorde a cada necesidad.
* **Embebido:** Característica que le permite estar presente en una serie de aplicaciones y servicios libres y/o comerciales.
* **Seguro:** Al margen de las características que posee por sí mismo, cuenta con el patrocinio de Canonical tanto para su desarrollo como para su soporte.
* **Libre:** Disponible bajo licencia GPL.

**Desventajas [2]**

* Se trabaja siempre con acceso al servidor remoto.
* No es muy popular.
* Consume ancho de banda.
* Estadísticamente es lento.
* El Launchpad es nuevo.

**2. Identificación.**

**2.1 Clasificación del ítem.**

A continuación, se presentarán los distintos ítems de configuración que se trabajarán durante todo el tiempo de desarrollo del proyecto, los cuales se clasificarán según los tipos que en este caso son:

* E o Evolución. Documentos sujetos a revisiones periódicas.
* F o Fuente. Códigos fuente o archivos utilizados en el desarrollo.
* S o Soporte. Sistema operativo o software base.

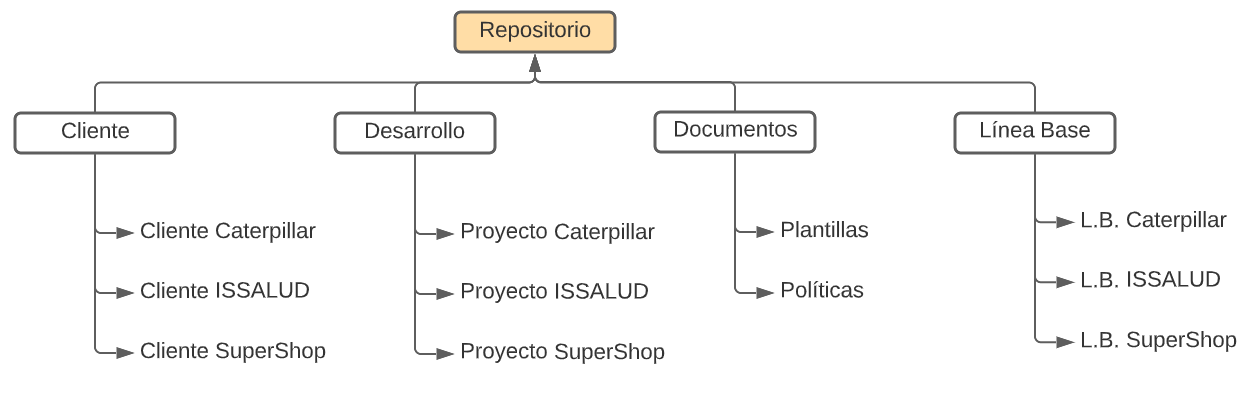
| Tipo | Nombre | Nomenclatura | Extensión | Proyecto |
| --- | --- | --- | --- | --- |
| E | Plan de Proyecto | SS-PC.DOCX | Docx | SuperShop |
| E | Documento de Especificación de Requisito 1 | SS-SDL.DOCX | Docx | SuperShop |
| E | Documento de Especificación de Requisito 2 | SS-ANA.DOCX | Docx | SuperShop |
| E | Documento de Especificación de Requisito 3 | SS-ELI.DOCX | Docx | SuperShop |
| E | Documento de Especificación de Requisito 4 | SS-MOD.DOCX | Docx | SuperShop |
| E | Documento de Especificación de Requisito 5 | SS-GRPI..DOCX | Docx | SuperShop |
| E | Documento de Arquitectura del Software | SS-DAI.DOCX | Docx | SuperShop |
| E | Documento del Diseño de Sistema de Inventario | SS-DEUI.PDF | Pdf | SuperShop |
| E | Documento de Interfaz utilizada en el Sistema de Inventario | SS-GE.PDF | Pdf | SuperShop |
| E | Documento de la Especificación de la Base de Datos | SS-DEBD.DOCX | Docx | SuperShop |
| E | Reporte del Estado Actual del Software | SS-RDS.DOCX | Docx | SuperShop |
| E | Sprint Retrospective | SS-RPS.DOCX | Docx | SuperShop |
| F | Inventario de la Base de Datos | Producto.db | Db | SuperShop |
| F | Sistema de Logeo | Login.py | Python | SuperShop |
| F | Requisito 3 | Requisito3.py | Python | SuperShop |
| F | Requisito 4 | Requisito4.py | Python | SuperShop |
| F | Requisito 5 | Requisito5.py | Python | SuperShop |
| F | Requisito 6 | Requisito6.py | Python | SuperShop |
| F | Funciones de Base de Datos | FuncionesDB.py | Python | SuperShop |
| S | Librería SQLite3 | - | - | SuperShop |
| S | Librería PyQT5 | - | - | SuperShop |
| S | PyQT5 Designer | - | - | SuperShop |

**2.2. Diseño de nomenclatura para documentos.**Se presentarán los diseños y casos de las nomenclaturas de los documentos del proyecto:

* Respecto a la nomenclatura del proyecto, tomamos como acrónimo la
* s letras “SS” dado que el nombre es “SuperShop”.A continuación, se presentarán los casos presentes en el proyecto:

| Descripción del caso | Ejemplo |
| --- | --- |
| Para los documentos relacionados con el desarrollo del proyecto, se utiliza la siguiente fórmula:  PY ACRÓNIMO + - + ACRÓNIMO DEL DOCUMENTO  O también, se podrá recurrir a la siguiente fórmula:  PY ACRÓNIMO + - + LETRAS INICIALES DEL DOCUMENTO | SS-SDL o SS-ANA |
| Para aquellos documentos que no pertenecen a un proyecto único, se identifican únicamente por el acrónimo. | PGCS |
| Para aquellos documentos donde se tiene elementos con diferentes versiones referentes a un proyecto, la fórmula es la siguiente:  PY ACRÓNIMO + - + ACRÓNIMO DEL DOCUMENTO + Número | SS-DAI2 |
| Para aquellos ítems de configuración fuente o código, simplemente tendrá el nombre del archivo referente a su función y la extensión: | Login.py |

**2.3 Diseño de la estructura del repositorio.**

Como diseño principal de nuestro repositorio, su estructura puede ser explicada fácilmente con este esquema.

Como vemos, nuestro repositorio bifurca en 4 bibliotecas, a continuación se dará una explicación de cada una:

* **Cliente:**

- Responsable: El líder del proyecto Artadí Ruiz, Gonzalo.

- Actividades: Gestionar los requerimientos y cualidades de los clientes con los que trabajaremos, jerarquizando prioridades para el avance de las líneas base.

- Contenidos: En esta biblioteca se almacenará a todos nuestros clientes juntos con los documentos donde se especifica los requerimientos que necesitan para su aplicación.

- Accesos: El acceso total a esta biblioteca lo tienen el Líder del proyecto y los desarrolladores solo acceso de lectura.

* **Desarrollo:**

- Responsable: Desarrolladores de software y base de datos.

- Actividades: Mantener actualizado la versión de nuestro software junto con la documentación de este, además de incorporar nuevas versiones de estos.

- Contenidos: En estas bibliotecas se encuentran ya las aplicaciones diseñadas para cada cliente, en esta no solo pueden estar archivos de la aplicación, sino también manuales sobre cada programa en donde se explicaría las funcionalidades de estas.

- Accesos:El acceso total a esta biblioteca lo tienen el Líder del proyecto y los desarrolladores.

* **Documentos:**

- Responsable: El analista de Datos Caballero, Omar

- Actividades: Dar a conocer a nuestros clientes nuestro método y entorno de trabajo para cada cliente y en cada etapa de trabajo.

- Contenidos: En esta biblioteca se almacenan todos los documentos sobre nuestra empresa y su funcionamiento, desde plantillas o software usado, hasta las políticas con las que trabajamos.

- Accesos: El acceso total de esta biblioteca lo tiene el lider del proyecto, en cambio un acceso solo de lectura solo se dará a nuestros clientes

* **Línea Base:**

- Responsable: El arquitecto de software Artadí Ruiz Gonzalo.

- Actividades: Obtener una versión de nuestro programa funcional listo para testeo de nuestro equipo o del mismo cliente.

- Contenidos: En esta biblioteca se encuentran las copias de versiones de nuestra aplicación completamente funcionales con los ítems designados por el cronograma.

- Accesos: El acceso de esta biblioteca lo tienen tanto los propios clientes como nuestros testers, además del arquitecto de software.

**2.4 Definición de Línea Base.**

A continuación, una explicación sobre las Líneas Bases de nuestro proyecto Supershop divididas por 3 sprints:

* Línea Base 1(SuperShop v0.4): Para este Sprint ya tendremos el diseño y la codificación de las principales funcionalidades de nuestra aplicación, así como también la declaración de las bases de datos que usaremos
* Línea Base 2(SuperShop v0.7): Este sprint se centra en la documentación de verificación junto a los testeos de las funcionalidades correspondientes
* Línea Base 3(SuperShop v1.0): En el último sprint se actualizará los documentos de testeo y verificación, y se establecerá un manual de usuario para el software Además de dar los retoques finales al programa y sacar una copia de la versión final de este listo para la distribución.

| Línea Base | Hito | Items |
| --- | --- | --- |
| **LB1** | 1 | SS-GRPI..DOCX  SS-SDL.DOCX  SS-ANA.DOCX  SS-ELI.DOCX  SS-MOD.DOCX  SS-GRPI..DOCX  SS-DAS.DOCX  SS-DDSI.PDF  SS-GE.PDF  SS-DEBD.DOCX  SS-RDS.DOCX  SS-RPS.DOCX |
| **LB2** | 2 | SS-DER.DOCX  SS-DVSI.PDF  SS-TEF.DOCX  SS-DVBD.DOCX  SS-PU.DOCX  SS-RDS2.DOCX  SS-RSS.DOCX |
| **LB3** | 3 | SS-DER.DOCX  SS-DVSI2.PDF  SS-DVBD2.DOCX  SS-DAS2.DOCX  SS-DPS.DOCX  SS-RDS3.DOCX  SS-RTS.DOCX  SS-MU.DOCX  SS-ACP.DOCX |

**6. Referencias**

[1] *Bazaar la herramienta para el control de versiones de forma distribuida*. (s. f.). Scribd. <https://es.scribd.com/document/15018566/Bazaar-la-herramienta-para-el-control-de-versiones-de-forma-distribuida#>

[2] *Sistemas de Control de Versiones*. (2013, 1 noviembre). Gabriel Fernando Chiriboga Rogel. <https://portfoliogabrielfcr.wordpress.com/2013/10/25/62/>