**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS.**

(Universidad del Perú**,** DECANA DE AMÉRICA).

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**Curso:** GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE

**Plan de Gestión de la Configuración**

**Grupo 1**

Integrantes:

* Cano Vasquez, Juan José
* Huaman Uriarte, Cesar Alberto
* Mallco Laurante, Frans Andre
* Munayco Sirlopu, Ronald
* Rojas Rojas, Darwin Jostein
* Sandoval Falcon, David Brian

**Empresa: FISIDEV\_SOLUTIONS**

**2025-1**

**Indice**

[**1. Introducción 3**](#_heading=h.o73bcuyu7kki)

[1.1 Situación de la empresa 3](#_heading=h.c0jew3ymsahf)

[1.1.1 Proyectos en desarrollo: 3](#_heading=h.dbgpipjsxv6p)

[1.1.3 Clientes: 4](#_heading=h.gg8w83yrodri)

[1.1.4 Problema relacionado a control de versiones: 4](#_heading=h.97kydf3pct8q)

[1.2. Objetivo 5](#_heading=h.3ehn28vzmbli)

[**2. Gestion 6**](#_heading=h.c2is3ihf07zx)

[2.2 Benchmarking de Herramientas, Gráfica de la infraestructura (de la herramienta elegida) mínimo 4 6](#_heading=h.jj0wbvx9t95)

[**3. Actividades de la GCS 8**](#_heading=h.iikzlcifbfz1)

[3.1 Identificación de la GCS 9](#_heading=h.jo8g5j4lall)

[3.1.1 Definir estructura de la librerías del repositorio(todo grafico se describe) 9](#_heading=h.a9ochoeayatg)

[3.1.2 Definición de Nomenclatura. 9](#_heading=h.lbbjeunzcl4z)

[3.1.4 Definicion de Linea Base 11](#_heading=h.ry4r4roo7ss0)

# Introducción

## 1.1 Situación de la empresa

**FISIDEV SOLUTIONS** es una empresa emergente especializada en el desarrollo de soluciones de software a medida que actualmente atraviesa una fase estratégica de expansión. Con un equipo en crecimiento y una cartera de clientes en aumento, la compañía ha logrado posicionar tres proyectos exitosos en producción mientras continúa fortaleciendo sus procesos internos.

**Tamaño del equipo:** Pequeño, en proceso de ampliación.

**Infraestructura:** Mixta (trabajo local + repositorios remotos).

**Retos actuales:**

* Formalización de procesos de desarrollo y entrega.
* Mejora de la comunicación entre desarrolladores.
* Implementación completa de herramientas de control de versiones y gestión de proyectos.

### 1.1.1 Proyectos en desarrollo:

|  |  |  |  |
| --- | --- | --- | --- |
| Proyecto | Cliente | Codigo | Estado |
| Plataforma de Gestión de Pedidos | Cevichería "Julia Fishr" | SGC | En desarrollo |
| Sistema de Gestión Interna | Consultoría Group JC SAC | MGD | En desarrollo |
| Sistema de Control de Inventario | Farmacia EDP Salud | EDP | En desarrollo |

1.1.2 Proyectos en producción:

|  |  |  |  |
| --- | --- | --- | --- |
| Proyecto | Cliente | Codigo | Estado |
| Sistema de Reservas | Salón de Eventos "Luz y Fiesta" | SDE | Completa |
| Plataforma de E-commerce | Tienda Virtual "Moda YA" | TV | Completa |
| Aplicación de Gestión de Citas | Clínica Médica NovaSalud | CMN | Completa |

### 1.1.3 Clientes:

* **Sector Gastronómico:** Cevichería "Julia Fish"
* **Sector Consultoría:** MGD Group
* **Sector Salud:** Farmacia EDP Salud, Clínica Médica NovaSalud
* **Sector Corporativo:** Corporación WQ Business
* **Sector Eventos:** Salón de Eventos "Luz y Fiesta"
* **Sector Retail:** Tienda Virtual "Moda YA"

### 1.1.4 Problema relacionado a control de versiones:

**Problemática Identificada:**

* Almacenamiento inicial de código en dispositivos USB, generando riesgos de seguridad y pérdida de información
* Inconsistencias en nomenclatura de archivos que provocaban errores durante la sincronización
* Ausencia de control de versiones estructurado, resultando en pérdida de avances y duplicación de esfuerzos

**Solución Implementada:**

* Migración a sistema Git con arquitectura de ramificación organizada
* Centralización de repositorio en plataforma GitHub
* Programa de capacitación técnica para el equipo en gestión de control de versiones
* Establecimiento de protocolos de revisión de código y aprobación de cambios

## 1.2. Objetivo

El Plan de Gestión de Configuración para el sistema de gestión integral de pedidos de JuliaFish tiene como propósito fundamental establecer un marco de control sistemático, estructurado y documentado sobre todos los artefactos generados durante el ciclo de vida del proyecto. Esta iniciativa responde a la carencia identificada de procesos formalizados de gestión de configuración en proyectos anteriores de la organización.

### **Alcance y Beneficios**

Este plan establece los procedimientos y herramientas necesarios para administrar metódicamente los elementos constitutivos del sistema, incluyendo:

* Código fuente y componentes ejecutables
* Documentación técnica y de usuario
* Diagramas y modelos arquitectónicos
* Archivos de configuración y parametrización
* Control de versiones y líneas base

**¿Porque proponen implementar el plan de la Gestión de configuración ?**

Dado que la empresa no contaba previamente con procesos formales de gestión de configuración, se identificó la necesidad de conservar, documentar y organizar adecuadamente los elementos del desarrollo. El presente plan se propone como una solución estructurada que permitirá manejar de forma más eficiente los procesos de desarrollo, reduciendo errores, mejorando la trazabilidad y facilitando la colaboración dentro del equipo.

# Gestion

* 1. Roles y Responsabilidades

### **1.- Comité de Control de Cambios (CCB)**

**Miembros asignados:**

* Valeria Mendoza Ríos
* Gabriel Torres Salvatierra
* Lucía Ramírez Calderón

**Responsabilidades:**

* Evaluar y registrar eventos y solicitudes de cambio.
* Gestionar y dar seguimiento a las solicitudes de cambio a lo largo de su ciclo de vida.
* Brindar soporte a los responsables de reportar cambios o incidentes.
* Coordinar con otros comités de control de cambios si aplica.
* Asegurar que las decisiones de cambio estén alineadas con los objetivos del proyecto.

**2.- Bibliotecario**

**Responsable asignado:**

* Tomás Alarcón Velarde

**Responsabilidades:**

* Definir y mantener las bibliotecas utilizadas en la gestión de configuración.
* Organizar y custodiar los artefactos del proyecto (código, documentación, configuraciones, etc.).
* Asegurar el control de acceso, integridad y versiones de los componentes del sistema

### **3.- Gestor de la Gestión de Configuración**

**Responsable asignado:**

* Isabel Núñez Farfán

**Responsabilidades:**

* Implementar, mantener y mejorar el plan de gestión de configuración.
* Garantizar el uso correcto de las herramientas de control de versiones.
* Supervisar que todos los cambios y configuraciones estén documentados y trazables.
* Coordinar con los demás roles para asegurar la integridad del producto en cada entrega.

## Benchmarking de Herramientas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criterios** | **Github** | **PlasticSCM** | **AWSCodeCommit** | **Mercurial** |
| **Paga** |  |  |  |  |
| **Distribuido** |  |  |  |  |
| **Centralizado** |  |  |  |  |
| **Integrable** |  |  |  | Parcial |
| **Facilidad de uso** | Alta | Media | Media | Media |
| **Comunidad/soporte** | Muy amplia | Moderada | Amplia | Reducida |
| |  | | --- | | **Plataformas**  **soportadas** | | Multiplataforma | Multiplataforma | AWS/LINUX | Multiplataforma |
| **Escalabilidad** | Alta | Alta | Alta | Media |

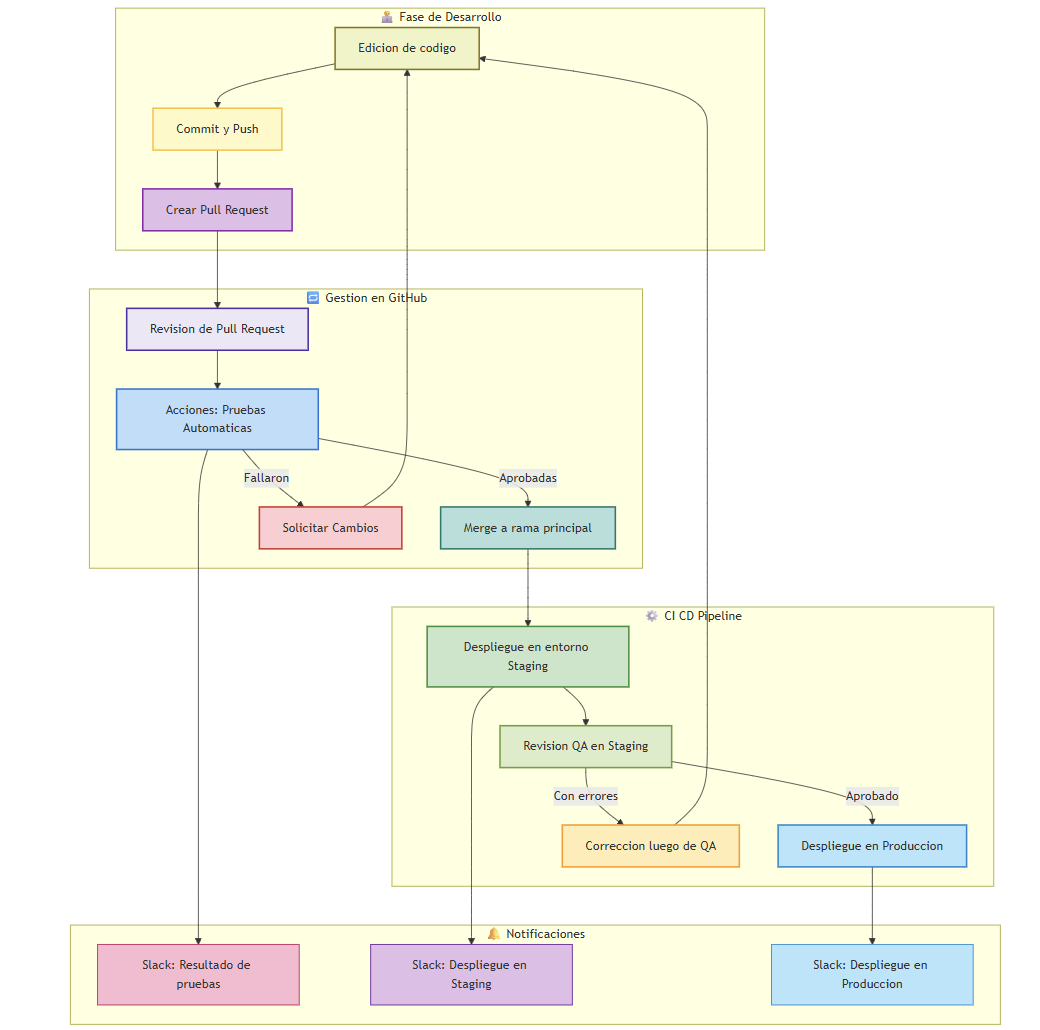
### **Propósito de la tabla comparativa**

La tabla tiene como objetivo evaluar diferentes sistemas de control de versiones con el fin de identificar cuál resulta más conveniente para el proyecto, considerando aspectos clave como el tipo de repositorio (centralizado o distribuido), facilidad de uso, control de versiones, manejo de ramas, historial de cambios, soporte para trabajo colaborativo y compatibilidad con herramientas externas.

### **Justificación del uso de Git**

Git se presenta como la opción más adecuada debido a su naturaleza distribuida, lo que permite que cada desarrollador trabaje de forma independiente sin depender de una conexión constante al repositorio central. Esto brinda mayor flexibilidad, especialmente en entornos con múltiples colaboradores.

Adicionalmente, Git facilita el desarrollo de aplicaciones modernas como las construidas en React, ya que permite gestionar múltiples ramas para distintas funcionalidades, integrarse fácilmente con servicios de despliegue automático, y conectarse con herramientas de control de calidad y prueba.

**2.2 Gráfica de la infraestructura**

La arquitectura de Git presentada optimiza el desarrollo web mediante un sistema que permite trabajo simultáneo entre desarrolladores con entornos locales conectados a un repositorio central. Implementa dos ambientes de despliegue: producción y pre-producción, permitiendo validar cambios antes de ser publicados al entorno final.

La estructura se basa en **tres ramas principales**:

* **'development'** para el desarrollo activo y colaboración diaria,
* **'test'** para validar nuevas funcionalidades en un entorno seguro,
* **'master'** para versiones estables que están en producción.

El flujo de trabajo incluye:

* El uso de **git commit** para registrar los cambios localmente,
* **git push** para enviar esos cambios al repositorio compartido,
* y fusiones controladas (**merge**) entre ramas para integrar nuevas funcionalidades.

Finalmente, el despliegue se puede automatizar mediante hooks de Git, que permiten actualizar automáticamente el servidor cuando se realiza un push, reduciendo errores manuales y facilitando actualizaciones continuas.

# Actividades de la GCS

## Identificación de la GCS

### Definir estructura de la librerías del repositorio (los graficos deben describirse)

### 

SGC

Directrices

JuliaFish

FisiDev**-**Solutions

Línea base

Clientes

Desarrollo

Documento

WQ

MGD

Planes

PlazaVea

EDP

EDD

Políticas

LopezFarma

SGC

Gestion

Requisitos

Pruebas

Despliegue

Fuentes

Modelo de Negocio

SGC-DR-01

Cronograma

SGC-DN

SGC-DR-02

SGC-DG

SGC-DR-03

SGC-GSC

SGC-DR-04

SCG-PC

SGC-DR-06

SGC-DR-05

El mapa conceptual de FisiDev Solutions organiza sus procesos clave en torno a clientes, desarrollo, documentación y control de calidad. Refleja una estructura en expansión, con énfasis en la formalización de entregables y trazabilidad. Destaca la implementación de un sistema de gestión que centraliza cronogramas, pruebas y requisitos técnicos. La jerarquía muestra un enfoque progresivo hacia la estandarización y eficiencia operativa. Cada componente apunta a mejorar la colaboración y la calidad del software. Es una representación clara de su avance hacia una gestión más profesional y estructurada.

### Definición de Nomenclatura.

La nomenclatura propuesta para los ítems del proyecto clasificado por tipos se basa en el siguiente formato:

* Documentos

ACRÓNIMO DEL PROYECTO + “-” + “ACRÓNIMO DE ELEMENTO” + “-” + NÚMERO O EXTENSIÓN

Ejemplo:

* **SGC-DR01-GP:**  
  + SGC → Sistema de Gestión de Configuración
  + DR01 → Documento de Requisitos, versión 01
  + GP → Gestión de Pedidos
* Módulos resultado del proyecto

ACRÓNIMO DEL PROYECTO + “-” + “Nombre del artefacto” + “-v” + versión + ”.” +revisión + actualización

Ejemplo:

* **SGC-UserAuth-v1.1a:**  
  + - SGC → Sistema de Gestión de Configuración
    - UserAuth → Módulo autentificación de usuarios
    - v1→ Versión 1
    - .1 → El módulo ha tenido una revisión, posiblemente nueva funcionalidad o cambio mínimamente notable
    - a → Primer parche de errores
* Items de soporte

ACRÓNIMO DEL PROYECTO + “-” + “Nombre del item de soporte” + “-” + versión

Ejemplo:

* **SGE-JDK-8:**  
  + SGE → Sistema de Gestión Empresarial
  + JDK → Java Development Kit
  + 8 → versión 8

Es definido de esta manera puesto a que algunos programas corren con una versión de Java 8 pero no posteriores. Esto es un item del que depende el programa y por lo tanto un ejemplo de ítem de soporte

Este esquema permite identificar de forma clara y ordenada los componentes del proyecto, facilitando el control de versiones y la trazabilidad dentro del proceso de desarrollo.

* + 1. Lista de Items (Clasificación de Ítems)

Los ítems son artefactos seleccionados por su recurrencia en el desarrollo y que pueden ser de tres tipos en este plan:

* Evolución: Para aquellos artefactos de software que documenten información como parte del ciclo de vida del software y que pueden presentar modificaciones de acuerdo al cliente o evaluación de requisitos
* Fuente: Conjunto del código fuente que componen al proyecto
* Soporte: Software del que requiere un proyecto para su funcionamiento tanto en desarrollo como en producción

A continuación se enlista una serie de artefactos necesarios de reconocer como parte del

proyecto SGC en desarrollo para ejemplificar la identificación del plan de gestión de configuración del software como parte del control, seguimiento y manipulación de las versiones.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tipo** | **Item** | **Descripción** | **Nomenclatura** | **Proyecto** |
| Evolución | Documento de Gestión del proyecto | Detalla los objetivos, planificación y roles | SGC-DG.docs | SGC |
| Evolucion | Documento de Negocio | Describe los procesos del negocio involucrado | SGC-DN.docx | SGC |
| Evolucion | Documento de Requisitos | Lista y explica los requerimientos funcionales y no funcionales | SGC-DR.docx | SGC |
| Evolución | Documento de diseño | Define la arquitectura general del sistema | SGC-DD.docx | SGC |
| Evolución | Feedback de diseño | Registro de validaciones y observaciones de clientes | SGC-FD.docx | SGC |
| Evolución | Diagrama de base de datos | Representación visual de la estructura de datos | SGC-DBD.docx | SGC |
| Evolución | Modelo de base de datos | Archivo editable con el diseño estructural de la BD | SGC-BD.mwb | SGC |
| Evolucion | Documento del plan de gestión de la configuración | Establece el control de artefactos relacionados a los procesos de desarrollo de la empresa | SGC-PGC.docx | SGC |
| Evolucion | Project Charter | Establece los acuerdos entre la empresa desarrolladora y el cliente | SGC-PC.pdf | SGC |
| Evolución | Documento de evidencia de pruebas | Documento con evidencias de pruebas de integración entre frontend y backend | SGC-DEP.docx | SGC |
| Evolución | Reporte de pruebas | Resultados de pruebas unitarias e integradas | SGC-RP.docx | SGC |
| Evolución | Manual de despliegue | Guía para publicar el sistema en un entorno de producción | SGC-MD.docx | SGC |
| Fuente | Código fuente frontend relacionado al proyecto SGC | Archivos comprimidos del desarrollo del lado del cliente | SGC-FF.zip | SGC |
| Fuente | Código fuente backend relacionado al proyecto SGC | Archivos comprimidos del desarrollo del lado del servidor | SGC-BE.zip | SGC |
| Soporte  (Librerías/Extensiones) | Node.js | Entorno de ejecución. Usado para la construcción de la aplicación y obtención de módulos | SGC - Node-v23.10.0 | SGC |
| Soporte  (Librerías/Extensiones) | React | Biblioteca principalmente usada para construir las interfaces gráficas de los programas | SGC - React-v19.0.10 | SGC |

El reconocimiento de estos ítems como parte del plan permite reconocer las versiones sobre las que está trabajando el equipo, evitando errores en el desarrollo y facilitando su corrección.

### Definicion de Linea Base

Describe el Sistema de gestión de pedidos para una cevicheria (SGC) en base a

los siguientes hitos:

|  |  |  |  |
| --- | --- | --- | --- |
| Lineas Base | Descripción | Elementos incluidos  (poner nomenclatura) | Fecha Estimada |
| Linea Base 1 - Hito 1: Analisis | Documentos de requisito y bases para el desarrollo del sistema. | * Documento de gestion SGC-DG * Documento de negocios SGC-DN * Documentos de requisitos SGC-DR-XX * Documento de Análisis SGC-DA | 8/05/2025 |
| Línea Base 2 - Hito 2: Diseño | Documentos relacionados al diseño del sistema. | * Documento de diseño SGC-DD * Diagrama de base de datos SGC-DBD * Modelo de base de datos SGC-BD * Feedback de diseño SGC-FD | 15/05/2025 |
| Línea Base 3 - Hito 3: Desarrollo y Despliegue | Versión funcional validada para ser liberada. | * Documentación de Entorno SGC-DEVSETUP * Código fuente frontend SGC-FE * Código fuente backend SGC-BE * Pruebas de integración SGC-INTEGRATION * Reporte de pruebas SGC-PRUEBAS * Manual de Despliegue SGC-DEPLOY | 12/06/2025 |

### **Línea Base 1 - Hito 1: Análisis**

**Descripción:** Esta fase corresponde al análisis del sistema, donde se definieron los requerimientos funcionales y no funcionales, así como los objetivos del proyecto. Esta línea base constituye la base documental sobre la cual se diseñará el sistema.

**Elementos incluidos (nomenclatura):**

* Documento de Gestión del Proyecto: SGC-DG
* Documento de Negocio: SGC-DN
* Documento de Requisitos: SGC-DR-XX (donde XX representa la versión)
* Documento de Análisis: SGC-DA

### **Línea Base 2 - Hito 2: Diseño**

**Descripción:** En esta etapa se definió la estructura técnica del sistema, incluyendo la arquitectura del software, la base de datos y las interfaces. Esta línea base marca el cierre de la fase de diseño y el inicio del desarrollo.

**Elementos incluidos (nomenclatura):**

* Documento de Diseño: SGC-DD
* Diagrama de Base de Datos: SGC-DBD
* Modelo de Base de Datos: SGC-BD
* Feedback de Diseño (revisiones): SGC-FD

### **Línea Base 3 - Hito 3: Desarrollo y Despliegue**

**Descripción:** Esta línea base incluye los componentes funcionales del sistema listos para ser desplegados, con sus respectivas pruebas, configuraciones y manuales de instalación. Representa el cierre del ciclo de desarrollo técnico.

**Elementos incluidos (nomenclatura):**

* Documentación del Entorno de Desarrollo: SGC-DEVSETUP
* Código Fuente Frontend: SGC-FE
* Código Fuente Backend: SGC-BE
* Pruebas de Integración: SGC-INTEGRATION
* Reporte de Pruebas: SGC-PRUEBAS
* Manual de Despliegue: SGC-DEPLOY
  1. **Control de la GCS**
     1. **Ejemplos de Solicitudes de cambio (1 por alumno)**