

Plan de Gestión de Configuraciones

StockManager

Collareda Agustín y Frey Hugo

Índice

Introducción.....	3
Entorno de gestión de versiones.....	4
Herramientas a utilizar.....	4
Estructura del repositorio.....	4
Detalles a tener en cuenta.....	4
Identificación de configuración.....	6
Nomenclatura de elementos.....	6
Elementos de configuración.....	6
Elementos de configuración relacionados.....	6
Líneas Base.....	6
Procedimientos.....	8
Registro de las desviaciones.....	8
Seguimiento de las desviaciones.....	10
Informes.....	12
Manejo del ambiente controlado.....	13

Plan de Gestión de Configuraciones

Introducción

El presente Plan de Gestión de Configuraciones tiene como objetivo establecer los lineamientos, procesos y responsabilidades necesarios para identificar, controlar, registrar y auditar los elementos de configuración a lo largo del ciclo de vida del proyecto. La gestión adecuada de la configuración permite asegurar la integridad y trazabilidad de los productos, facilitando el control de versiones, la gestión de cambios y la consistencia entre los distintos entregables.

Entorno de gestión de versiones

Herramientas a utilizar

Se utilizarán las herramienta Git, GitLab y Google Docs como herramienta de control de versiones y repositorios.

Estructura del repositorio

/

- /Documentación
 - /Documentación de Usuario
 - Manual de Usuario
 - Manual de Instalación
 - /Especificación de Requerimientos
 - Especificación de Requerimientos
 - Diagramas asociados
 - /Plan de Gestión de Configuraciones
 - Plan de Gestión de Configuraciones
 - /Plan SQA
 - Plan SQA
 - /Planificación
 - Informe de Cambios
 - <fecha> - StockManager
 - /Pruebas
 - **A definir**
- /Implementación
 - /ABM
- README.md

Detalles a tener en cuenta

Los integrantes del equipo trabajarán con un repositorio local, el cual obtendrán mediante la clonación del repositorio principal alojado en GitLab. Antes de realizar cualquier modificación, deberán asegurarse de tener su repositorio local actualizado. Los commits se realizarán siguiendo las convenciones definidas por [Conventional Commits](#).

El código de la aplicación será modificado desde cada repositorio local. En cuanto a la documentación, esta se elaborará en formato .doc a través de Google Drive, lo que permitirá la edición colaborativa en tiempo real. Una vez finalizada, la documentación se convertirá a formato .pdf y se subirá al repositorio de GitLab.

Identificación de configuración

Nomenclatura de elementos

La nomenclatura de los elementos de configuración será:

<Nombre del documento>_<Miembros del equipo>

Esta forma de nombrar a los archivos permite que los miembros del equipo puedan identificar, buscar y modificar los documentos que correspondan de manera más sencilla.

Elementos de configuración

Los elementos de configuración que se identificaron son:

- Planificación.
- Especificación de requerimientos.
- Prototipo de la aplicación.
- Plan de aseguramiento de calidad de software (SQA).
- Plan de gestión de configuraciones.
- Código Fuente.
- Pruebas realizadas.

Elementos de configuración relacionados

Los elementos de configuración que se encuentran relacionados son:

- El plan SQA está relacionado con: La planificación, el plan de gestión de configuraciones y la especificación de requerimientos.
- La especificación de requerimientos está relacionada con el prototipo de la aplicación, código fuente y las pruebas realizadas.
- El prototipo de la aplicación tiene relación con el código fuente.

Líneas Base

Las líneas base definidas en este momento son:

- LB01: Especificación de requerimiento, Plan SQA, Prototipo de la aplicación y Código fuente. Esta tiene como fecha límite el 13/05/2025.
- LB02: Plan SQA y Plan de Gestión de configuraciones. Esta tiene como fecha límite el 20/05/2025

- LB03: Plan de Gestión de configuraciones y el Plan SQA. Esta tiene como fecha límite el 27/05/2025
- LB04: Plan de Gestión de configuraciones, el Plan SQA y el código fuente. Esta tiene como fecha límite el 03/06/2025

Por el momento, no se tiene más líneas bases definidas. Por otra parte, al seguir un proceso evolutivo, los documentos se pueden repetir en estas líneas bases y la planificación debe incluirse en cada línea base.

Procedimientos


Registro de las desviaciones

Para registrar una desviación se utilizara la herramienta github issues. Este registro tiene una serie de pasos determinadas:

- La persona que encontró la desviación tendrá que crear el issue en la herramienta dicha anteriormente.
- Elegirá de qué tema trata según la etiqueta. Contamos con las siguientes etiquetas: ayuda, bug, consulta, documentación y mejora. Estas ya traen su plantilla por lo que al crear una issue se podrá seleccionar una de estas categorías.
- Completará la planilla para la issue que esté generando.
- Se creará la issue sin ninguna persona asignada.
- Los miembros del equipo se tendrán que comunicar por whatsapp para establecer quién realizará esa issue.
- Se asignará la issue a la persona decida.
- Se creará una tarjeta en el trello que represente la issue.

Un detalle a tener en cuenta es que los issues deben ser una instancia sobre la plantilla presentada en el github issues. A continuación se muestran ejemplos:

- **Solicitud de ayuda**
- **Reporte de Bug**
- **Consulta General**
- **Sugerencia de documentación**

 **fxex** opened 1 minute ago · edited by fxex Edits ▾ ...

Descripción

El documento plan de gestión de configuraciones en la sección "Identificación de configuración" en la subsección "líneas bases" falta reflejar dos líneas bases (la anterior y la actual). Por otra parte en la sección "Procedimientos" en la subsección "Registro de las desviaciones" falta añadir ejemplos concretos sobre las plantillas generadas.

Tipo de cambio

☐ Corrección ortográfica o gramatical ...
☐ Aclaración de conceptos ...
☒ Agregado de contenido faltante ...
☐ Actualización técnica ...

Archivo o sección involucrada

El archivo se denomina "Plan de Gestión de Configuraciones_Collareda Agustín y Frey Hugo" y las secciones referidas en la descripción se encuentran en la pagina 6 (al final) y en la pagina 8 (al principio).


Propuesta

En la actualización de las líneas bases, se debe actualizar según la planificación realizada y la entrega.


En la agregación de ejemplos de la plantilla, se puede tomar de ejemplo esta issue para la etiqueta documentación.

Información adicional

N/A

Create sub-issue ▾ 

- **Solicitud de mejora**

 **Ardkrav** opened 3 days ago ...

Descripción

Existen archivos del código del sistema que incumplen algunas de las recomendaciones del estándar PSR-12. Se debe modificar dichos archivos para cumplir con el estándar.

Problema que resuelve

Esta mejora corrige problemas de legibilidad del código, permitiendo una mayor comprensión de este por parte de los desarrolladores.

Impacto esperado

Mediante el cumplimiento del estándar PSR-12, se espera generar un código mas mantenible y confiable, al evitar errores y fallos del sistema derivados de la mala comprensión del código.

Información adicional

- *Implementación/Mueble/listarMuebles.php*: Incumple recomendaciones de Files, Lines (línea 1, 9 y 56) y ControlStructures (línea 46).

```
C:\Users\hugoa\Desktop\StockManager\Implementacion\Mueble>phpcs --standard=PSR12 listarMueble.php
FILE: C:\Users\hugoa\Desktop\StockManager\Implementacion\Mueble\listarMueble.php
-----
FOUND 4 ERRORS AND 1 WARNING AFFECTING 4 LINES
-----
 1 | ERROR   | [x] End of line character is invalid; expected "\n" but found
   |         | "\r\n"
 9 | ERROR   | [x] Whitespace found at end of line
 9 | ERROR   | [x] Whitespace found at end of line
46 | ERROR   | [x] Expected 1 space after closing parenthesis; found 0
56 | WARNING | [ ] Line exceeds 120 characters; contains 143 characters
-----
PHPCBF CAN FIX THE 4 MARKED SNIFF VIOLATIONS AUTOMATICALLY
-----

Time: 349ms; Memory: 6MB
```

- *Implementación/Mueble/crearMuebles.php*: Incumple recomendaciones de Files, Declare Statements, Namespace, and Import Statements (línea 1).

```
C:\Users\hugoa\Desktop\StockManager\Implementacion\Mueble>phpcs --standard=PSR12 crearMueble.php
FILE: C:\Users\hugoa\Desktop\StockManager\Implementacion\Mueble\crearMueble.php
-----
FOUND 2 ERRORS AFFECTING 1 LINE
-----
 1 | ERROR | [x] Header blocks must be separated by a single blank line
 1 | ERROR | [x] End of line character is invalid; expected "\n" but found
   |       | "\r\n"
-----
PHPCBF CAN FIX THE 2 MARKED SNIFF VIOLATIONS AUTOMATICALLY
-----

Time: 100ms; Memory: 6MB
```

En caso de no seguir contener la información mínima, se etiquetara como invalido y el creador deberá modificarlo y notificarlo para que sea válido.

Seguimiento de las desviaciones

Para el seguimiento de una desviación se utilizara la herramienta Trello.

Este seguimiento tiene el siguiente procedimiento:

- Una vez que se genera la tarjeta en el trello, se debe revisar con mayor detenimiento el issue generada.
- El encargado deberá dar respuesta del análisis. Es decir, deberá responder a la issue con un confirmado o recibido y con una justificación de lo que se hará.
- Se deberá generar una lista de tareas en el trello con el fin de conseguir el objetivo para el fin de la iteración.

- Una vez completada se comenta el cierre con el número de la tarjeta para tener una mayor trazabilidad.
- Se deberá subir el commit siguiendo la normativa planteada en las secciones anteriores y añadiendo la instrucción `Resolve #<número de issue>`

Gestión de ramas

El uso de ramas en git y github permite una mayor trazabilidad si se usa correctamente. El propósito de esta sección es establecer un procedimiento adecuado para la gestión de ramas.

Se debe tener en cuenta que la rama Main es la rama principal y la que cuenta con la versión más estable del proyecto. La gestión de las ramas es la siguiente:

- En primera instancia se debe tener la última versión de la rama principal. En este caso se usa el comando **git pull**.
- Se generará una etiqueta con el comando **git tag -a vX.0 -m "Mensaje representativo"**. Donde X es un número natural y se va incrementando a medida que se hace una Pull Request (PR).

Esta etiqueta se debe subir al repositorio remoto con la instrucción **git push origin v<número de versión>**

- Una vez que se inicia la iteración, se generará una rama desde la rama principal, la cual se denominará **Iteración_dd-mm-yyyy**. Esta tendrá la fecha de creación asociada a su nombre.

Para la persona creadora de la rama deberá utilizar los siguientes comandos:

git checkout -b <nombre de la rama> (Creación de rama local)

git push origin <nombre de la rama> (Creación de rama remota)

Para la otra persona deberá usar los siguientes comandos:

git fetch (obtener ramas remotas)

git switch -c <nombre de la rama local> origin/<nombre de la rama remoto>

- A medida que se van completando las tareas y las issues a resolver en esa iteración. Se realizará un tag con el segundo número modificado, es decir, vX.Y, donde Y es un número natural que se va incrementando.

- Una vez finalizada la iteración se realizará una PR a la rama principal con el fin de establecer el fin de la iteración.
- Las PR deben tener:
 - Una descripción clara.
 - Referencia a issues solucionadas y generadas.
 - Se deberá revisar y aprobar los cambios a la rama principal.
- Se deberá eliminar la rama de manera local con un **git branch -d <nombre de rama>** y de manera remota **git push origin --delete <nombre de la rama>**

Informes

Se deberá realizar una evaluación al final de cada una de las líneas base, con el fin de tener registro de los documentos que fueron utilizados y modificados. Esta evaluación tendrá en cuenta la planificación prevista y revisará si fue cumplida o no, además de hacer un análisis retrospectivo sobre lo ocurrido.

Esta evaluación se incorporará en un informe denominado “Informe de Cambios”, el cual deberá incluir la fecha de emisión y un resumen de los cambios. Este informe estará ordenado según la fecha de la más actual a la antigua.

Manejo del ambiente controlado

Un ambiente controlado es un entorno donde se gestionan, almacenan y manipulan los elementos de configuración de manera segura, estructurada y bajo control de cambios. Su propósito es asegurar la integridad, trazabilidad y consistencia de los productos o componentes del proyecto.

Para mantener un ambiente controlado se va a utilizar una máquina virtual con VirtualBox versión 7.1.8 con Ubuntu 22.04 LTS en la cual se compartirá el disco virtual, a través de una memoria usb, con el fin de utilizar las mismas herramientas y garantizar el espacio seguro.

Las herramientas que estarán incorporada en este disco virtual son:

- Xampp 8.2.12-0 y sus versiones de apache y mariaDB.
- Composer 2.8.9.
- Visual Studio Code 1.100 y las extensiones prettier, git extension pack, autoclose tag, autocomplete tag, php extension pack, spanish version y vscode-pdf.
- Git 2.34.1.
- Navegador de internet Firefox 130.0.
- PHP_CodeSniffer 3.13.0.