

2021

Testigos de Turing

PLAN DE GESTIÓN DE CONFIGURACIONES

Autores:

- Ciordia Cantarella , Francisco
- Martinez, Brenda Sofia
- Fernandez, Santiago
- Rao, Maximiliano
- Villane, Santiago

Tabla de contenido

HISTORIAL DE REVISIONES	2
REGISTRO DE APROBACIÓN	3
1 - INTRODUCCIÓN	4
1.1 PROPÓSITO Y ALCANCE DEL PLAN	4
1.2 PROPÓSITO DEL PLAN DE GESTIÓN DE CONFIGURACIONES	4
1.3 HERRAMIENTAS PARA LA ADMINISTRACIÓN DE CONFIGURACIONES	4
2 - NORMAS DE ETIQUETADO DE ARCHIVO Y DIRECTORIO	5
2.1 TAG DE DIRECTORIOS	5
2.2 TAG DE VERSIONADO DE ARCHIVOS	6
3 - CHANGE CONTROL BOARD	6
3.1 INTRODUCCIÓN Y OBJETIVOS	6
3.2 EQUIPO DE TRABAJO	7
3.3 REUNIONES DE TRABAJO	8
3.4 PROCESO DE CONTROL DE CAMBIOS	8
4 - SOURCE CODE CONFIGURATION MANAGEMENT	9
4.1 ESQUEMA DE RAMAS Y POLÍTICAS DE FUSIÓN DE ARCHIVOS	9
5 - MANTENIMIENTO DE CM PLAN	12
6 - BACKUP Y RECOVERY EN CASO DE DESASTRE	12
7 - RELEASE MANAGEMENT	12

HISTORIAL DE REVISIONES

Versión	Fecha	Resumen de cambios	Autores
0.1.0	28/04/2021	Documento inicial(Ahre)	Si no se ponen las pilas no los pongo
1.0.0	29/04/2021	Documento mejorado(casi)	Ole que aca estoy
1.1.0	29/04/2021	Incorporación Indice	
1.1.1	29/04/2021	Informe final	

REGISTRO DE APROBACIÓN

Gerente de configuración (CM)	Fecha	Firma

Subgerentes de manejo de la configuración	Fecha	Firma

1. INTRODUCCIÓN

1.1. .PROPÓSITO Y ALCANCE DEL PLAN

El presente documento expone el Plan de Gestión de Configuraciones realizado para el proyecto "LUDO MATIC". Mediante este plan se busca poner en conocimiento las políticas, estrategias, herramientas y métodos empleados para el manejo del software producido.

1.2. .PROPÓSITO DEL PLAN DE GESTIÓN DE CONFIGURACIONES

- Definir políticas y procesos de la administración de configuración; y explicitar los fundamentos de las mismas.
- Mantener la integridad el proyecto.
- Establecer cuáles son los roles en el CCB.
- Definir herramientas de gestión de versiones y entrega.
- Exponer los principios para la construcción del sistema que garanticen la calidad requerida y pretendida.
- Crear un historial del desarrollo del proyecto y mantener informados a los integrantes sobre el estado actual del mismo.

1.3. HERRAMIENTAS PARA LA ADMINISTRACIÓN DE CONFIGURACIONES

Herramienta/Proceso	Propósito
GitLab	Control de versiones. Integración continua
VisualParadigm	Herramienta gráfica para diagramas UML
java	Herramienta para programar
Google meet	Herramienta para comunicarnos
Microsoft Word	Editor de texto para la realización de los informes

2. NORMAS DE ETIQUETADO DE ARCHIVO Y DIRECTORIO

2.1. TAG DE DIRECTORIOS

Estructura de directorios y fines.

Para la documentación, código fuente y los releases se realizaron las siguientes denominaciones:

- LudoM.Documento: Directorio de planes generales
- LudoM.Releases: Directorio donde se encuentran los releases.
- LudoM.src: Directorio donde se ubica el código fuente del proyecto y sus diversos testings.

2.2. TAG DE VERSIONADO DE ARCHIVOS

Para el versionado utilizamos el tipo estándar, separando por puntos, según se trate de un cambio menor, mayor o un simple fix:

<Nombre_de_archivo>_Mayor.Menor.Fix.java

Estos siendo:

- Un fix, cuando se arregla algún error.
- Un cambio menor cuando se agrega una funcionalidad sin tener que cambiar el resto del código.
- Uno mayor cuando la funcionalidad agregada no era compatible previamente y se hicieron cambios en el código base respecto a la versión anterior. Los cambios son crecientes y tras cierto número de cambios de menor jerarquía se considera uno de mayor, es decir, cada 10 de menor se le suma 1 a mayor seteado a menor en 0 nuevamente. Además, un cambio en la jerarquía mayor, también setea a las anteriores en 0, es decir:

Pasamos de Main_2.3.4.java -> a un Main_3.0.0.java de un solo salto.

También se agregan sufijos del tipo Beta, Alpha y Pre-Release, a versiones que están por ser consideradas terminadas.

3. CHANGE CONTROL BOARD

3.1. INTRODUCCIÓN Y OBJETIVOS

El Comité de Control de Cambios es un grupo dentro del equipo que se encarga de evaluar, aprobar o rechazar los pedidos de cambios realizados por diversos agentes.

Tiene como fin recibir los pedidos de cambios y evaluar su autorización para luego planear y generar la respectiva documentación y código.

La toma de decisiones tiene como criterio el cumplimiento de los requerimientos de calidad y el correcto desenvolvimiento de cada prueba.

3.2. EQUIPO DE TRABAJO

Posición dentro del CCB	Titular	Suplente	Función
Director del CCB	Fernández, Santiago	Ciordia, Francisco	<ul style="list-style-type: none"> • Aprobación o modificación de cambios. • Organizador de reuniones. • Manejo del CCB. • Moderador.
Gerente de manejo de configuraciones (GMC)	Rao, Maximiliano	Ciordia, Francisco	<ul style="list-style-type: none"> • Responsable en general de tareas de planeamiento, diseño de las políticas de control de seguimiento y su actualización. • Veedor de cumplimiento de lo anterior

Gerente de ingeniería o de release (GI)	Villane, Santiago	Martínez, Brenda Sofía	<ul style="list-style-type: none"> • Evaluación del costo en cuanto a cambios de infraestructura del sistema, fechas de entrega, etc.
Gerente de coordinación (GI)	Martínez, Brenda Sofía	.	<ul style="list-style-type: none"> • Se encarga de las peticiones de cambio mediante change request forms y logs. • Documenta y traduce a modo formal los cambios.
Gerente de pruebas	Ciordia, Francisco		<ul style="list-style-type: none"> • Aplica el plan de requerimientos para contrastar con lo obtenido. • Organizar las pruebas • Definir las actividades a implementar como estrategia de prueba (test) • Evalúa impactos del cambio en la calidad de la entrega y el calendario de testing

3.3. REUNIONES DE TRABAJO

Las reuniones se harán de manera remota los días jueves a las 17:00 hs y fines de semana si es necesario discutir alguna cuestión extra. Además de los medios de comunicación personales habituales

3.4. PROCESO DE CONTROL DE CAMBIOS

Este proceso consta de las siguientes etapas:

- 1) Se inicia la solicitud de cambio
- 2) Se analiza la solicitud en primera instancia si es de carácter necesario y en segunda el impacto de la misma en cuanto a recursos.
- 3) Es el CCB quien toma la decisión de aceptar o rechazar el cambio después de un análisis hecho por el equipo. También se puede dar que el cambio se postergue por falta de necesidad inmediata del mismo o porque se acepta la necesidad del mismo pero aún hay que trabajar en elementos de mayor prioridad.
- 4) De aceptarse, se destinan los recursos necesarios para ello.
- 5) Una vez verificado que el cambio se hizo de manera correcta se da por finalizado el proceso de cambio.

4. SOURCE CODE CONFIGURATION MANAGEMENT

4.1. ESQUEMA DE RAMAS Y POLÍTICAS DE FUSIÓN DE ARCHIVOS

Se procederá a especificar las ramas del desarrollo del proyecto. Se encontraran 4 ramas:

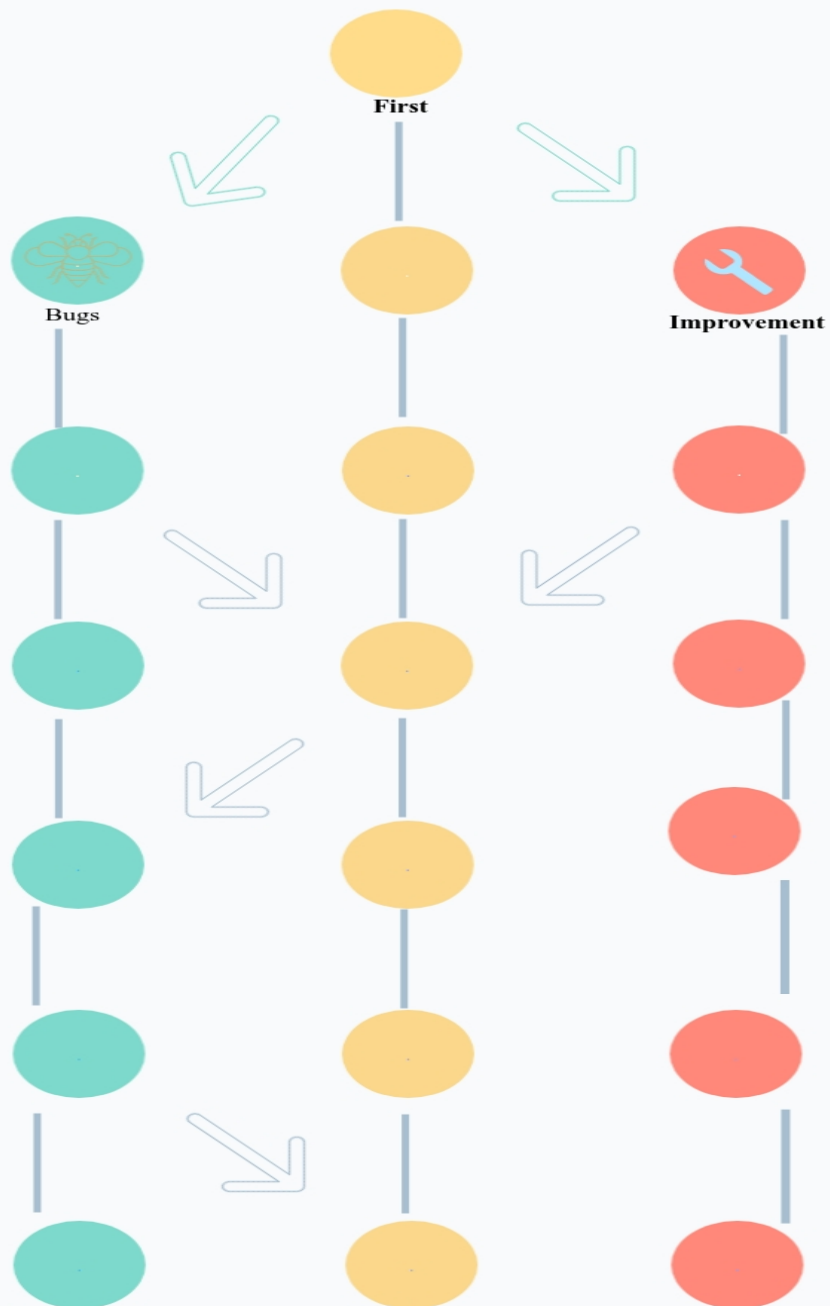
First, Improvement y Bugs.

- Las rama **First** es la rama principal del proyecto. Estará indefinidamente en el proyecto y es el punto de partida de todas las ramas. De esta rama, sale la versión final del proyecto.

- La rama **Improvement** es la rama destinada a nuevas funciones o nuevas implementaciones para el proyecto. Esta rama tiene que salir de la rama First, y luego debe mergear a la rama First. Cuando dichas funciones son implementadas, esta rama termina su vida útil.

- La rama **Bugs** es la rama que tiene su aparición cuando se detectan errores en la rama First o Improvement. Como se mencionó anteriormente, parte de la rama First, luego mergea a First y termina su vida útil una vez que los bugs han sido corregidos o eliminados

Diagrama de Ramas



5. MANTENIMIENTO DE CM PLAN

El responsable de monitorear el Plan de Configuración es el GMC. Se hará una revisión del Plan de Configuración al comienzo de cada iteración. En caso de que haya modificaciones se comunicarán por correo a todos los integrantes.

6. BACKUP Y RECOVERY EN CASO DE DESASTRE

Se tendrá una política de Backup semanal, los martes de cada semana, el GMC será el encargado de realizar la un “clone” de todo el repositorio de GitHub utilizado de esta forma tendremos una copia local del mismo. Este Backup tendrá una política de retención de 2 semanas.

Para la restauración el GMC tendrá un tiempo máximo de 1 hora para recuperar el repositorio completo.

7. RELEASE MANAGEMENT

Los “releases” o lanzamientos estarán presente en la rama principal (master branch).

Para cada release se entregará un archivo .zip donde se incluirá lo siguiente:

- El ejecutable del programa .cmd
- El programa en un archivo .jar
- Archivo .pdf de instrucciones de uso
- Archivo README.txt el cual incluye información sobre versión de release,

ayuda con la instalación del software y bugs conocidos.

TdeT