# QuarenTEAM

# 

*Nombre del grupo:* ***QuarenTEAM***

*Nombre del proyecto:* ***Collateral***

*Nombre del documento:* ***Plan de Administración de Configuración***

*Integrantes:*

***Don Enzo Martin***

***Quinteros Tomás***

***Coschica Francisco Nicolás***

***Hidalgo Juan Nahuel***

*Fecha:* ***19/04/2020***

*Versión del documento:* ***V2.0***

## Historial de Cambios

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Fecha** | **Cambio** | **Autor** |
| 1.0 | 19/04/2020 | Planificación | Grupo |
| 2.0 | 20/05/2020 | Modificaciones | Grupo |
|  |  |  |  |

## Acrónimos/Glosario

|  |  |
| --- | --- |
| **Acrónimo** | **Descripción** |
| ISS | Del Inglés ISSUE. Problemas. |
| REPO | Repositorios. |
| Tag | a.k.a etiquetas. Identificadores para los ítems de configuración. |
| CCB | Change Control Board. Tabla de Control de Cambios. |

## 

## 

## 

## Índice:

1. [**Introducción:**](#_eb0hzget5nsq) 
   1. [Propósito y alcance del plan………………………………………………………..3](#_hkifxrnm5rgo)
   2. [Propósito del plan de gestión de configuraciones…………………………….....3](#_2ecwm1gikvxb)
   3. [Herramientas para la administración de configuraciones……………………....3](#_o49cg4phtzwj)
      1. [Forma de acceso……………………………...…………………………...4](#_o49cg4phtzwj)
   4. [Roles y responsabilidades ……………………………...……………………..5](#_p948vv2a2fzv)
2. [**Esquema de directorios:**](#_yzxmk58hswz4)
   1. [Estructura de directorios y su propósito………………………………………....6](#_cni0let0pfjk)
   2. [Normas de etiquetado y nombramiento de los archivos……………………....6](#_g0xwbe6b7qr5)
3. [**Gestión de la configuración del código:**](#_afcec7k2eu69)

3.1 [Esquema de ramas……………………………...……………………………...……...7](#_j51fk0fc5b34)

3.2 [Política de etiquetado de las ramas……………………………...…………………..8](#_kf1hqraduwjm)

3.3 [Política de fusión de archivos……………………………...…………………………9](#_vulld1osfgtn)

**4.** [**Gestión de cambios - Change Control Board (CCB):**](#_wkxc74z5ch7)

4.1 [Introducción y objetivos……………………………...………………………………..10](#_ev934mmfq7dc)

4.2 [Miembros……………………………...………………………………………………..10](#_yugcrhqwnybh)

4.3 [Frecuencia de reunión de trabajo……………………………...…………………….10](#_t68kkrlyhhjv)

4.4 [Proceso de control de cambios……………………………...……………………….11](#_vo87mzanm3i1)

**5.** [**Gestión de entregas**](#_lufuzblquzaj)

5.1 [Formato de entrega de releases……………………………...……………………...12](#_2svn942kgzyd)

5.2 [Formato de entrega del instalador…………………………………………………...12](#_lstixgs2j0rp)

5.3 [Instrucciones mínimas de instalación………………………………………………..12](#_extrqr9ap5g5)

## 

## 

## 1- INTRODUCCIÓN

### 1.1 Propósito y alcance del plan

Este documento cubre el Plan de gestión de configuración para Collateral. La intención del Plan CM es controlar la configuración de los requisitos, documentos, software y herramientas utilizados para este proyecto. Así como registrar y procesar los cambios propuestos al sistema.

### 1.2 Propósito del plan de gestión de configuraciones

• Asegurar consistencia en la práctica durante el desarrollo del Software .

• Mantener la integridad del producto durante todo el ciclo de vida.

• Informar a quien lo requiera sobre el estado del proyecto.

• Crear un historial verificable de los estados actuales y anteriores de productos de trabajo

• La mejora de procesos utilizados para llevar a cabo el proyecto

### 1.3 Herramientas para la administración de configuraciones

* *Control de versiones. GitHub*
* *Gestión de tareas. Projects - Herramienta de Github*
* *Gestión de defectos. Github - Issues*
* *Integración continua. Jenkins*
* *Construcción automática. Jenkins*

***1.3.1 Forma de acceso:***

### 

|  |  |  |  |
| --- | --- | --- | --- |
| Herramienta | Propósito | Control de Herramienta | forma de Acceso |
| GitHub | Servidor de integración continua y herramienta de control de cambios. | Las compilaciones se activan automáticamente en función de los cambios en los repositorios. Solicitudes de cambio. | <https://github.com/juannahuelhidalgo/QuarenTeam> |
| Projects de Github | Gestión del flujo de trabajo. | Asignado de Tareas, seguimiento de procesos y estatus del trabajo. | <https://github.com/juannahuelhidalgo/QuarenTeam/projects/1> |
| Issues | Control de cambio | Herramienta disponible en servidor GitHub para el control y administración de todos los requerimientos de cambio/ mejoras. | <https://github.com/juannahuelhidalgo/QuarenTeam/issues> |
| Jenkins | Integración continua/ Construcción automática | La herramienta se encarga de revisar una rama determinada, construir el proyecto,realizar las pruebas automatizadas y informar de los resultados a los integrantes del proyecto | <https://www.jenkins.io/> |
| Diseño de diagramas UML | Google dibujos y Visual Paradigm | Genera los diagramas que ayudan al entendimiento del proyecto. | N/A |

### 

### 

### 

### 

### 

### 

### 1.4 Roles y Responsabilidades

|  |  |
| --- | --- |
| **Rol** | **Responsabilidad** |
| **GLOBAL PROJECT CONFIGURATION MANAGER**  **Responsable: Hidalgo, Juan Nahuel**  **Suplente: Coschica, Francisco Nicolás** | **Poseer la responsabilidad general de todos los elementos de configuración y asegurar la integridad del proyecto.**  **Coordinar y asistir a los integrantes dentro del proyecto.**  **Asegurar la ejecución correcta de este documento.**  **Asistir en las decisiones de desarrollo del software.** |
| **PROJECT CONFIGURATION MANAGER**  **Responsable: Quinteros Tomas** | **Encargarse de asistir y controlar el desarrollo del software. Así como evaluar arreglos de bugs y decidir sobre la implementación de nuevas funcionalidades.** |
| **EQUIPO** | **Crear ramas y etiquetas acordes a la planificación.**  **Garantizar la integridad del producto y trabajar con la mayor eficacia/eficiencia para lograr los objetivos.**  **Debe participar en las reuniones y comunicar la información necesaria para el desarrollo.** |

## 

## 

## 

## 

## 2- ESQUEMA DE DIRECTORIOS

### 2.1 Estructura de directorios y su propósito

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Propósito** | **Link** |
| Documentación del proyecto | Tiene todos los documentos que se han utilizado para realizar el desarrollo del proyecto. | <https://github.com/juannahuelhidalgo/QuarenTEAM/tree/master/Documentaci%C3%B3n%20del%20proyecto> |
| Documentación del usuario | Posee todas las guías e información necesaria para el usuario. | <https://github.com/juannahuelhidalgo/QuarenTEAM/tree/master/Releases/Documentaci%C3%B3n%20del%20usuario> |
| Releases | Contiene el ejecutable del juego. | <https://github.com/juannahuelhidalgo/QuarenTEAM/tree/master/Releases> |
| Collateral | Contiene el proyecto del juego. | <https://github.com/juannahuelhidalgo/QuarenTEAM/tree/master/Collateral> |

### 2.2 Normas de etiquetado y nombramiento de los archivos.

*Se enlistan las normas sobre el nombramiento/etiquetado de distintos ítems como los documentos y las releases del proyecto.*

|  |  |  |  |
| --- | --- | --- | --- |
| *Significado* | *Nivel* | *Regla* | *Ejemplo* |
| *Extensión de la primer release* | *Nuevo producto* | *Empieza en 1.0.0* | *1.0.0* |
| *Se solucionan problemas* | *Release de parche* | *Incrementa el tercer dígito* | *1.0.1* |
| *Nuevas implementaciones* | *Release menor* | *Incrementa el segundo dígito y el último dígito se pone en 0.* | *1.1.0* |
| *Implementación de grandes cambios y funcionalidades* | *Release mayor* | *Incrementa el primer dígito y los demás toman valor cero* | *2.0.0* |

### 

|  |  |
| --- | --- |
| Plan de gestión de las configuraciones | *CM\_<version>* |
| Documento de requerimientos | DR\_<version> |

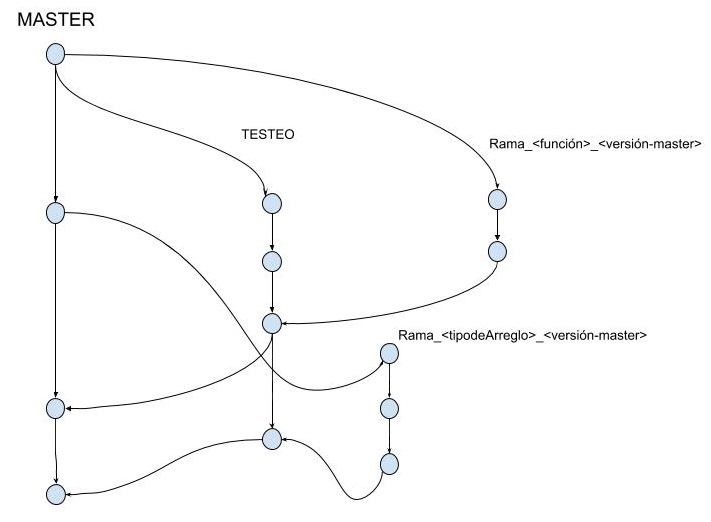
### 

### 

### 

## 3- GESTIÓN DE LA CONFIGURACIÓN DEL CÓDIGO

### 3.1 Esquema de ramas

****

*Los tipos de ramas que se utilizarán se definen de la siguiente manera:*

*• Rama de integración: la MASTER se define como la rama de integración principal donde se guardarán todas las funciones publicadas.*

*• Ramas de desarrollo: son aquellas ramas donde se codifica el desarrollo de nuevas características / funcionalidades.*

*• Rama de Testeo: Es aquella rama donde se prueba que las funcionalidades básicas para que el juego corra.*

*• Rama de Arreglos: Es aquella rama donde se solucionan problemas que traspasan las pruebas de unidad realizadas.*

### 3.2 Política de etiquetado de las ramas

Las ramas que se utilizarán serán etiquetadas de la siguiente manera:

*Rama\_<funcion>\_<version-master>*

*Rama\_<tipodeArreglo>\_<version-master>*

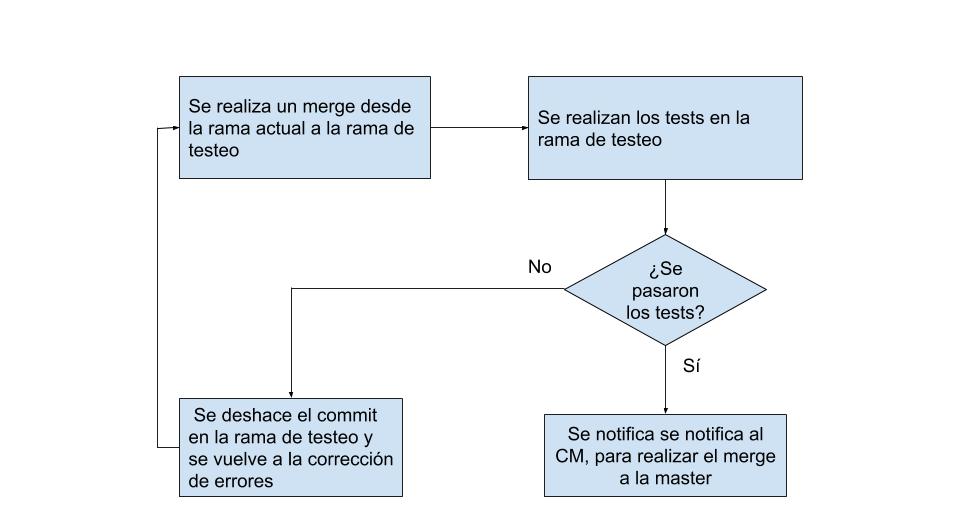
La rama MASTER y la de testeo no siguen esta norma de etiquetado.

### 

### 

### 3.3 Política de fusión de archivos

*Describir la estrategia de merge dado el esquema de ramas utilizado.*

**

Se realizará un merge a la rama de testeo la cual irá acompañando la evolución de la master constantemente, si la nueva funcionalidad funciona adecuadamente en la rama de testeo se le notificará al CM para que haga el correspondiente merge a la master. En caso de que los tests en la rama de Testeo no se pasen, se cancelara el commit y se volverá a la rama original de la funcionalidad para hacer las correspondientes correcciones

## 

## 

## 4-GESTIÓN DE CAMBIOS - Change Control Board (CCB)

### 4.1 Introducción y objetivos

*El CCB es un comité que se encarga de garantizar que los cambios sean lo adecuados en base a los objetivos y autoriza su implementación. El CBB debe monitorear, aprobar y controlar las solicitudes de cambio de documentos y código, garantizando la calidad del producto.*

### 4.2 Miembros

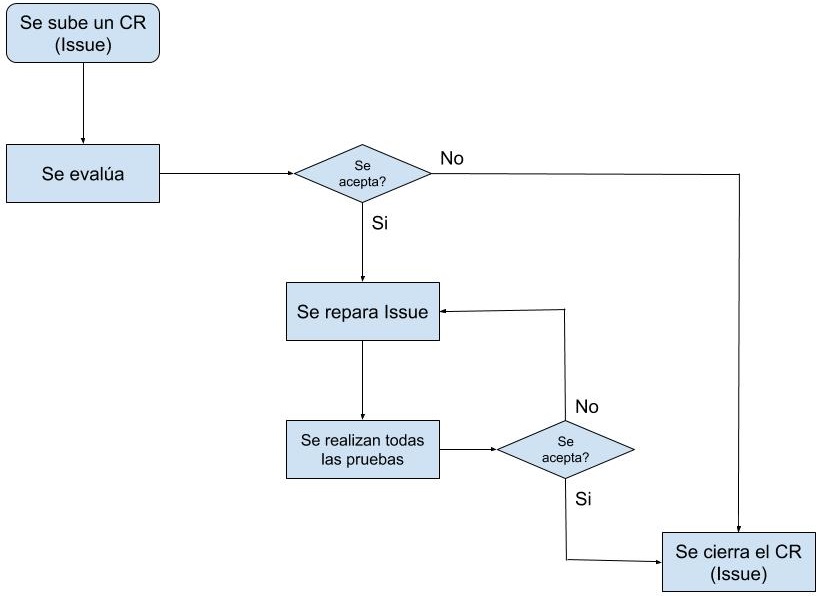
|  |  |  |
| --- | --- | --- |
| **Roles CCB** | **Nombre** | **Actividad/Responsabilidad** |
| Monitor | Quinteros Tomás | Monitorear la correcta implementación del proceso de CCB y asistir en caso de ser necesario a los otros miembros |
| Coordinador Issues | Don Enzo Martin | Evaluar las solicitudes de cambio para determinar si son válidas o no |
| Aprobación | Coschica Francisco  Hidalgo Juan Nahuel | Definir o no la implementación de los cambios al producto. |

### 4.3 Frecuencia de reunión de trabajo

|  |  |
| --- | --- |
| **CCB Reuniones** | **Frecuencia** |
| **<Collateral> CCB** | **Reunión realizada cada 3 días o según demanda.** |

### 4.4 Proceso de control de cambios

*El flujo de control de cambios se refleja en el siguiente gráfico:*



### 

## 

## 5- GESTIÓN DE ENTREGAS

### 5.1 Formato de entrega de releases

Se entregará el release en una carpeta dentro del repositorio de nombre, Releases, correspondiente a la versión que se halla seleccionado del proyecto.

### 5.2 Formato de entrega del instalador

*El instalador será un único archivo ejecutable (****“.exe”****).  
Se incluye la guía de juego que además contiene el método de instalación y requisitos del Juego.*

### 5.3 Instrucciones mínimas de instalación

*Se proveerá dentro de la guia de juego un pequeño instructivo.*