Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 07/09/2016 | 1.0 | Creación del plan de Gestión de la Configuración | Fredy Sebastián Lopera |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, Acronyms, and Abbreviations 3

1.4 References 3

1.5 Overview 3

2. Software Configuration Management 3

2.1 Organization, Responsibilities, and Interfaces 3

2.2 Tools, Environment, and Infrastructure 3

3. The Configuration Management Program 3

3.1 Configuration Identification 3

3.1.1 Identification Methods 3

3.1.2 Project Baselines 3

3.2 Configuration and Change Control 3

3.2.1 Change Request Processing and Approval 3

3.2.2 Change Control Board (CCB) 3

3.3 Configuration Status Accounting 3

3.3.1 Project Media Storage and Release Process 3

3.3.2 Reports and Audits 3

# Introduction

En este documento se evidenciarán los procesos que se tendrán en cuenta para la gestión de la configuración para el proyecto “HC Responsable”.

## Purpose

El propósito de este documento es informar detalladamente sobre la gestión de la configuración que se tendrá para el proyecto.

## Scope

Para el proyecto se tendrán unos ítems de la gestión de la configuración en los cuales podemos encontrar las políticas de versiones, el manejo de las versiones, Baseline y las solicitudes de cambio. En este documento se demostrará cual es el trato que se le dará a estos ítems de la gestión de la configuración.

## Definitions, Acronyms, and Abbreviations

**Baseline:** Una línea base es un concepto de [gestión de la configuración](https://es.wikipedia.org/wiki/Gesti%C3%B3n_de_la_configuraci%C3%B3n) que lleva a controlar los cambios sin impedir seriamente los cambios justificados.

**Bussiness process model:** Un modelo estándar de procesos de negocio y la notación (BPMN) proporcionará a las empresas la capacidad de entender sus procedimientos internos de negocios en una notación gráfica y darán a las organizaciones la capacidad de comunicar estos procedimientos de manera estándar. Además, la notación gráfica facilitará la comprensión de las colaboraciones de rendimiento y las transacciones comerciales entre las organizaciones. Esto asegurará que las empresas entiendan a sí mismos ya los participantes en sus negocios y permitirá a las organizaciones a adaptarse a las nuevas circunstancias de negocio internos y B2B rápidamente.

**Github:** Es una potente herramienta de colaboración, revisión de código, y la gestión de código.

## References

http://www.bpmn.org/

https://es.wikipedia.org/wiki/L%C3%ADnea\_base

https://www.apachefriends.org/es/index.html

## Overview

[This subsection describes what the rest of the **Configuration Management Plan** contains and explains how the document is organized.]

# Software Configuration Management

## Organization, Responsibilities, and Interfaces

* **Identificar:** En esta disciplina de la gestión de la configuración, se construirá una línea base en la cual se identificará la base para el inicio del proyecto.

La base para el inicio del proyecto comprenderá los siguientes artefactos:

* Business Vision
* Vision
* Stakeholders Request
* Software Architecture
* Risk List

La elección de este Baseline se enfocó en la importancia de los objetivos que se desean alcanzar (Business Vision, Vision), las necesidades e ideas del cliente (Stakeholders Request), la búsqueda un modelo que abarque todos los requisitos del software (Software Architecture Document) y los posibles riesgos con el producto a desarrollar (Risk List).

* **Control:** En el control se tendrá las políticas de versión, como se manejara las versiones en un repositorio, el repositorio elegido para el proyecto y el control del cambio.

**Política 1:**

Cada vez que se incorpore alguna clase referente a algún indicador de responsabilidad social, puede entenderse como un avance significativo, y en su versión se reflejará por la adición de una unidad. Por ejemplo: Versión 3.0 – Después del cambio: Versión 4.0

**Política 2:**

Cada vez que se incorpore un método nuevo, puede entenderse como la incorporación de una funcionalidad nueva, y en su versión se reflejará por la adición de una unidad en el segundo dígito de la versión. Por ejemplo: Versión 3.0 – Después del cambio: Versión 3.1

**Política 3:**

Cada vez que se modifique un método, puede entenderse como la mejora de una funcionalidad existente, y en su versión se reflejará por la adición de una unidad en el tercer dígito de la versión. Por ejemplo: Versión 5.2 – Después del cambio: Versión 5.2.1

**Política 4:**

La creación, modificación o corrección de un artefacto o documento significativo para el desarrollo del proyecto, el cambio en su versión se representará por la adición de 1 unidad. Por ejemplo: Versión 1 – Después de la modificación: Versión 2

***REPOSITORIO***

El repositorio se creará en la herramienta GITHUB para hacer control de las versiones que se tengan con los artefactos del Baseline. También se hará control al código que se modificará la versión según las políticas de versión anteriormente mencionadas.

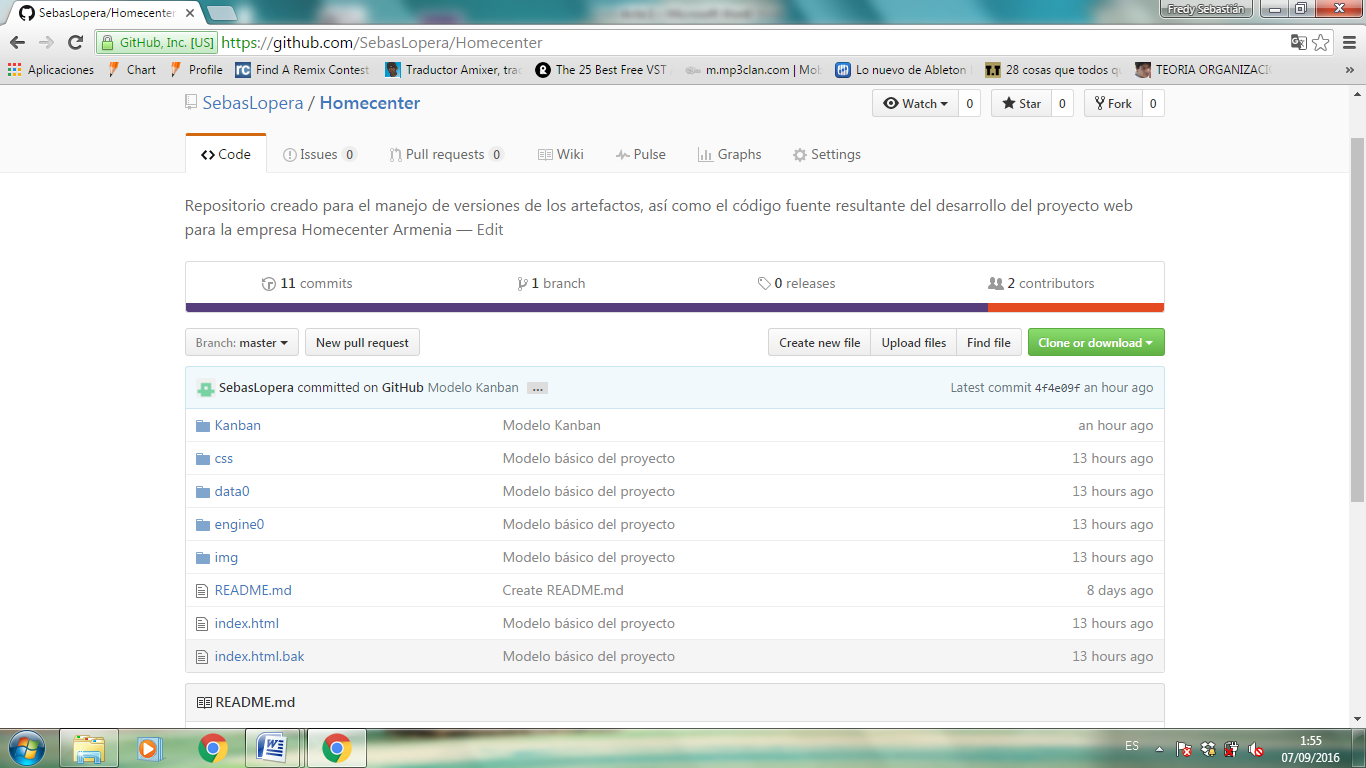


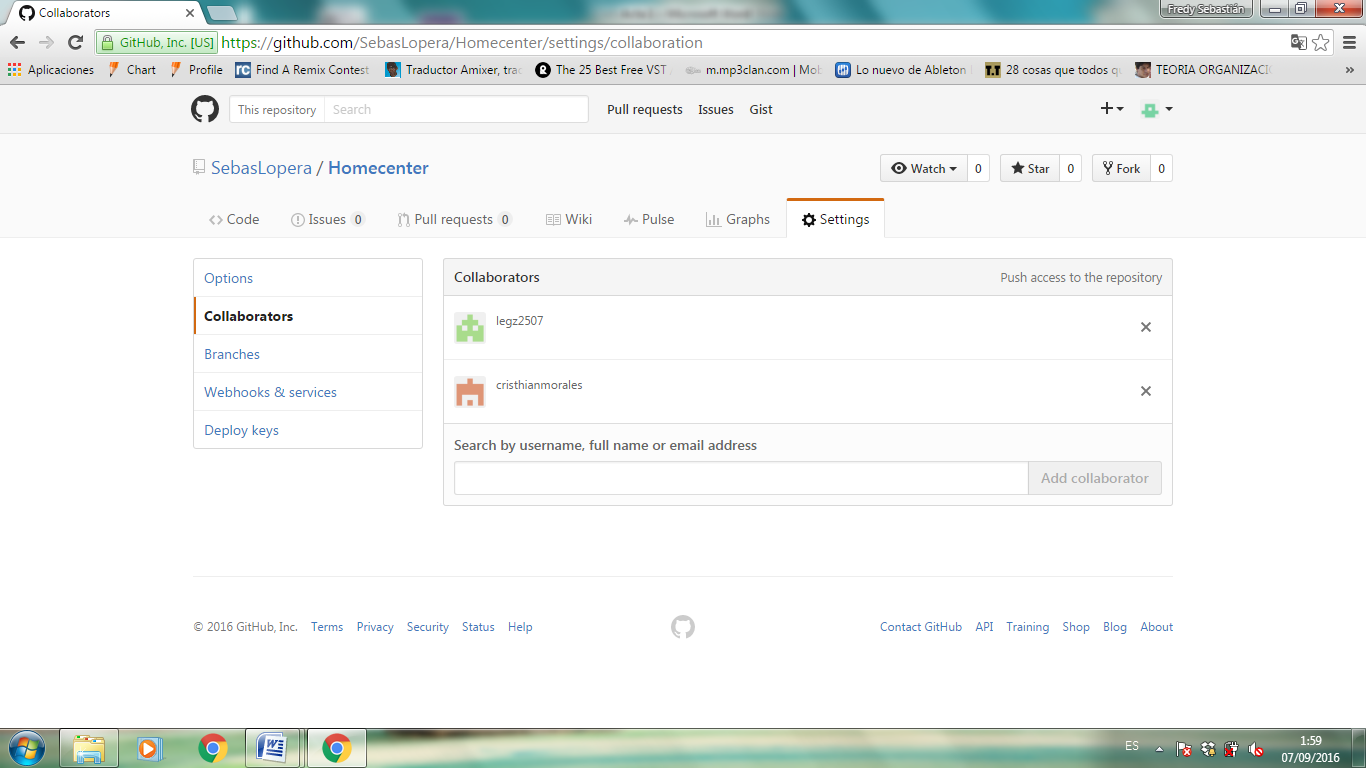
Ilustración 1 Repositorio Homecenter

En la **ilustración 1** podemos observar el repositorio creado, en el cual se manejará previamente las versiones del proyecto y los artefactos según las políticas definidas.

Todos los artefactos se subirán a una carpeta en el repositorio llamada “Artefactos”.

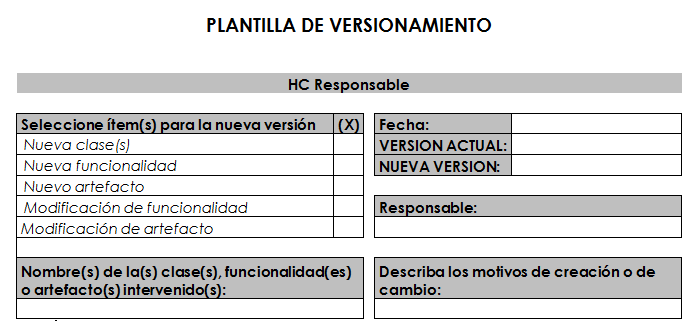
En la carpeta Desarrollo se incluirán las versiones y cambios del desarrollo de la página web.

En la figura 2 se evidencian las cuentas del equipo de trabajo:



*Ilustración 2. Colaboradores del repositorio*

Por último, y como soporte de cada nueva versión o modificación de un artefacto, se diseñará un formato que permita llevar un control donde se evidencie el responsable, el cambio y el tipo de cambio del artefacto como se evidencia en la **ilustración 3.**



*Ilustración 3. Plantilla de Control de Versiones*

**Gestión del control del cambio:** Para la Gestión del Control del Cambio es necesario definir o crear un modelo que rija los pasos a seguir para la realización de un cambio ya sea en el código o en algún requisito del cliente.

En caso de que el cliente requiera un cambio o tenga alguna idea sobre el desarrollo y que puede implementarse, el equipo de desarrollo le facilitará un formato de solicitud de cambio en el cual se especifique de manera clara y concisa la modificación que se desea realizar.

Este formato entonces, llegará a manos de analistas en el equipo de trabajo que estudiarán la viabilidad del cambio sugerido. En caso de que en el contrato celebrado esté estipulado en algún inciso que se puede realizar cambios, el equipo debe hacer un análisis detallado a nivel de costos, tiempo y esfuerzo.

En caso de que el contrato no especifique de ninguna manera que puede hacerse cambios, puede entrarse en negociación con el cliente a fin de llegar a un acuerdo en el que las responsabilidades financieras recaigan sobre él.

Luego de aprobado el cambio, el desarrollador se dispondrá a implementarlo y luego realizará su respectivo testing.

## Tools, Environment, and Infrastructure

Las herramientas que se van a utilizar para el desarrollo de la gestión de la configuración son:

* **Bussiness process model:**Un modelo estándar de procesos de negocio y la notación (BPMN)proporcionará a las empresas la capacidad de entender sus procedimientos internos de negocios en una notación gráfica y darán a las organizaciones la capacidad de comunicar estos procedimientos de manera estándar. Además, la notación gráfica facilitará la comprensión de las colaboraciones de rendimiento y las transacciones comerciales entre las organizaciones. Esto asegurará que las empresas entiendan a sí mismos ya los participantes en sus negocios y permitirá a las organizaciones a adaptarse a las nuevas circunstancias de negocio internos y B2B rápidamente.

En el BPMN se trabajara el modelado de la gestión del cambio, donde se diseñara de manera gráfica como se manejaran los cambios propuestos para el proyecto.

* **Framework:** [Sin definir]
* **Github:** Es una potente herramienta de colaboración, revisión de código, y la gestión de código.

En esta herramienta se manejara el repositorio donde se subirá el código y los artefactos, en la cual también se garantiza el control de las versiones con el repositorio creado en esta herramienta.

# The Configuration Management Program

## Configuration Identification

### Identification Methods

[Describe how project or product work products are to be named, marked, and numbered. The identification scheme needs to cover hardware, system software, Commercial-Off-The-Shelf (COTS) products, and all application development work products listed in the product directory structure; for example, plans, models, components, test software, results and data, executables, and so on.]

### Project Baselines

La base para el inicio del proyecto comprenderá los siguientes artefactos:

* Business Vision
* Vision
* Stakeholders Request
* Software Architecture
* Risk List

La elección de este Baseline se enfocó en la importancia de los objetivos que se desean alcanzar (Business Vision, Vision), las necesidades e ideas del cliente (Stakeholders Request), la búsqueda un modelo que abarque todos los requisitos del software (Software Architecture Document) y los posibles riesgos con el producto a desarrollar (Risk List).

## Configuration and Change Control

### Change Request Processing and Approval

### Change Control Board (CCB)

## Configuration Status Accounting

### Project Media Storage and Release Process

En el control se tendrá las políticas de versión, como se manejara las versiones en un repositorio, el repositorio elegido para el proyecto y el control del cambio.

**Política 1:**

Cada vez que se incorpore alguna clase referente a algún indicador de responsabilidad social, puede entenderse como un avance significativo, y en su versión se reflejará por la adición de una unidad. Por ejemplo: Versión 3.0 – Después del cambio: Versión 4.0

**Política 2:**

Cada vez que se incorpore un método nuevo, puede entenderse como la incorporación de una funcionalidad nueva, y en su versión se reflejará por la adición de una unidad en el segundo dígito de la versión. Por ejemplo: Versión 3.0 – Después del cambio: Versión 3.1

**Política 3:**

Cada vez que se modifique un método, puede entenderse como la mejora de una funcionalidad existente, y en su versión se reflejará por la adición de una unidad en el tercer dígito de la versión. Por ejemplo: Versión 5.2 – Después del cambio: Versión 5.2.1

**Política 4:**

La creación, modificación o corrección de un artefacto o documento significativo para el desarrollo del proyecto, el cambio en su versión se representará por la adición de 1 unidad. Por ejemplo: Versión 1 – Después de la modificación: Versión 2

### Reports and Audits

Se realizara un informe sobre la auditoria en las cuales se evaluara los ítems más importantes de la gestión de la configuración entre estos están:

* Políticas de versión
* Manejo de versión
* Control del cambio.
* Línea base