**PLAN DE GESTIÓN DE LA CONFIGURACIÓN**

**Grupo: 3**

****

**Desarrolladores RedDragon**

**Versión 1.0**

# Índice

[**Índice**](#_2lwe5ilj4mu) **2**

[**1.-Introducción:**](#_bglw8d9uvww7) **3**

[1.1. Situación de la Empresa:](#_f6ger4biu1l0) 3

[1.2. Problemática](#_lssan6vidz1a) 3

[**2.- Gestión de la Configuración del Software**](#_bbwf3sovjhxv) **4**

[2.1.-Roles o responsabilidades](#_dswf347t7o8j) 4

[2.2.-Herramientas (Benchmarking)](#_kte2pmkkcu9h) 4

[2.3.-Diagrama de arquitectura de la herramienta elegida](#_y3rb998ku9jy) 8

# 1.-Introducción:

## 1.1. Situación de la Empresa:

La empresa “Desarrolladores RedDragon” es una startup con un año en el mercado de desarrollo de soluciones de software dedicada principalmente al desarrollo de software a medida y de calidad. Esto en base a las necesidades del cliente

La empresa en lo que va desde su fundación lleva dos clientes, con un total de 11 proyectos, estando en constantes reuniones para poder asegurar tres proyectos adicionales lo cual nos llevará a invertir en más desarrolladores al equipo junto a expertos o capacitadores en metodologías ágiles para poder apuntar a proyectos más interesantes.

Los proyectos actuales que lleva con los dos clientes se encuentran ya en producción y desplegados.

## 1.2. Problemática

Hasta ahora la problemática a la que más se ha enfrentado la empresa es el escalamiento de proyectos, debido al acoplamiento que poco a poco se da en los proyectos asimismo como la dificultad para hacer crecer el proyecto de forma horizontal.

Debido a eso es que se planteó la posibilidad de un Plan de Gestión de la Configuración con la intención de poder tener un mejor control de los proyectos en lo que se refactoriza los proyectos que se tienen en mantenimiento y poder dar al cliente una solución respecto al crecimiento de peticiones que está teniendo algunos de estos proyectos y así mismo poder tener un documento el cual referenciar en futuras decisiones.

# 2.- Gestión de la Configuración del Software

## 2.1.-Roles o responsabilidades

Se tendrán comités de control de la configuración, desde ahora CCC, multiniveles cada uno con responsabilidades y funciones distintas.

CCC Nivel 1: También llamado comité técnico, estará conformado por el personal con alto conocimiento técnico del proyecto en el que se esté trabajando , son los encargados de la configuración de manera pragmática.

CCC Nivel 2: Es el comité técnico más los clientes, este CCC tiene como tarea principal rescatar las necesidades de los clientes.

CCC Nivel 3: Es el comité técnico más la gerencia general, esta es la instancia máxima del CCC, cualquier decisión se puede hacer y deshacer en este nivel.

## 2.2.-Herramientas (Benchmarking)

Uno de los mayores problemas a afrontar en los proyectos de software es el control entre versiones, no solo cuando estamos desarrollando, sino también en la organización y estructuración a nivel individual. Para solventar dicho problema, se recurre a las herramientas de control de versiones, las cuales facilitan el almacenamiento de los elementos a gestionar, la recuperación de cada uno de ellos y el registro histórico e identificación de cada una de las modificaciones realizadas en las sucesivas versiones del código del proyecto.

A continuación presentaremos el Benchmarking de Baazar y Concurrent Versions System (CVS) herramientas, analizando sus características, funcionalidades, para el criterio de evaluación. A partir de dichas evaluaciones se determinará la herramienta idónea para satisfacer nuestras necesidades como equipo de proyecto.

1. Información General

La comparativa realizada sobre las características generales de las herramientas analizadas

se muestra en la Tabla 1.

Tabla 1 : Información general.

| HERRAMIENTA | PRECIO/  LICENCIA | Estado |
| --- | --- | --- |
| Bazaar | GPL | Activo |
| CVS | GPL | Activo |

A la hora de elegir una herramienta de gestión de versiones es importante tener en cuenta ya no solo el precio a pagar por licencia sino también la asidua actualización de versiones llevada a cabo por el fabricante.

En este caso evaluaremos respecto al precio o licencia asociada, se observa que tanto la herramienta Bazaar como la herramienta CVS optan una licencia GNU General Public License que permite la libre distribución, modificación y uso del software.

1. Información Técnica

La Tabla 2 presenta una comparativa respecto a parámetros técnicos, que se describen.

Tabla 2: Información técnica

| Herramienta | Tipo de Repositorio | Modelo  de Concurrencia | Historia de cambios | Alcance  del  cambio | Identificación de revisiones |
| --- | --- | --- | --- | --- | --- |
| Bazaar | Distribuido | CMM | Snapshot | Árbol | Pseudoaleatorio |
| CVS | Centralizado | CMM | Changeset | Fichero | Números |

En la tabla podemos apreciar que ambas tienen el modelo de concurrencia centrado en CMM, el cual evita los bloques innecesarios.

Cuando comparamos los historiales de cambios, por una parte el Changeset realiza ramificaciones y por otro lado el Snapshot que almacena una instantánea del

árbol de directorios antes y después del cambio.

Otro es el alcance de cambio por una parte tenemos árbol y por otro ficheros, si bien ambos tiene sus pro y sus contras, los árboles son aquellos que organizan los archivos en directorios que contienen información similar. Esta organización facilita el montaje remoto de directorios y por tanto de los cambios.

1. Opciones permitidas.

El análisis de las diferentes opciones permitidas en cada una de las herramientas se

muestra en la Tabla 3. Para cada una de ellas se ha rellenado la casilla correspondiente con check (✓) si lo contiene, una (✕) si no lo contiene y una (ᑭ) si lo contiene parcialmente

Tabla 3: Tabla comparativa

| Opciones\Herramienta | **Bazaar** | CVS |
| --- | --- | --- |
| Commits atómicos | ✓ | ✕ |
| Renombrado de ficheros y directorios | ✓ | ✕ |
| Fusión de ficheros renombrados | ✓ | ✕ |
| Trazas de funciones realizadas | ✓ | ✓ |
| Réplica de repositorio | ✓ | ✕ |
| Permisos del repositorio | ✓ | ᑭ |
| Traza de la historia del fichero o directorio | ✓ | ✓ |
| Tags | ✓ | ✓ |
| Internacionalización | ✓ | ✕ |
| Unicode | ✓ | ✕ |

Este apartado contiene las características más requeridas por nosotros los desarrolladores de software en los sistemas de gestión de versiones. Se comprueba que sistemas como Bazaar posee el 100% de las características mencionadas en la Tabla 3.

Una de las grandes desventajas que presentaban muchos sistemas de control de versiones antiguos y que hoy en día todavía poseen herramientas como CVS es la

imposibilidad de renombrar ficheros o copiarlos a otras rutas manteniendo su historial de versiones.

1. Comandos básicos

En este punto se muestra una comparativa sobre los comandos básicos soportados por cada una de las herramientas, explicados a continuación en la Tabla 4.

Tabla 4: Comandos básicos

| Comandos\Herramientas | **Bazaar** | CVS |
| --- | --- | --- |
| init | ✓ | ✓ |
| clone | ✓ | ✕ |
| pull | ✓ | ✕ |
| push | ✓ | ✕ |
| branch | ✓ | ✕ |
| checkout | ✓ | ✓ |
| update | ✓ | ✓ |
| lock | ✕ | ✓ |
| add | ✓ | ✓ |
| remove | ✓ | ✓ |
| move | ✓ | ✕ |
| copy | ✕ | ✕ |
| merge | ✓ | ✓ |
| commit | ✓ | ✓ |
| revert | ✓ | ✓ |

Inicialmente comandos básicos como init, checkout, add, remove, merge, commit, revert son soportados por ambas herramientas, sin embargo en los comandos de clone, pull, push, branch, move no lo contiene la herramienta CVS. Dado a que tienen una importancia significativa, el CVS tiene menos efectividad que el Bazaar.

Concluyendo el Benchmarking realizado, vemos que tanto la herramienta Bazaar como la CVS son usadas para para realizar la gestión de versiones en proyectos software, componente importante de la SCM, se ha analizando tanto sus características como sus

funcionalidades y carencias con el fin de comprobar cuáles de los dos serían las funcionalidades esenciales. Por el cual, las comparativas mostradas en las tablas: Tabla 1, Tabla 2, Tabla 3 y Tabla 4, se escogería a Bazaar como la herramienta idónea que satisfaga las necesidades reales de nuestros proyectos.

## 2.3.-Diagrama de arquitectura de la herramienta elegida

Bazaar presenta la siguiente arquitectura:

