Miguel Angel .

Miembros

Cordero Vela, Francisco Manuel  
Lázaro Domínguez, José Manuel

Mogrovejo Campero, Miguel Ángel

Benítez Rodríguez, Daniel

Cabina telegram

Proyecto para EGC

Índice

[1. Control de versiones 2](#_Toc501131404)

[2. Resumen 3](#_Toc501131405)

[3. Gestión del código fuente 3](#_Toc501131406)

[3.1. Herramientas 3](#_Toc501131407)

[3.2. Gestión de ramas 3](#_Toc501131408)

[3.3. Procedimientos de gestión de código 4](#_Toc501131409)

# Control de versiones

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Autor | Fecha | Cambios |
| 1.0.0 | Miguel Ángel Mogrovejo Campero | 10/12/2017 | Esquema del documento, apartados principales hasta la fecha. |
|  |  |  |  |
|  |  |  |  |

# Resumen

El objetivo principal es dar una solución a la problemática de realizar encuestas por el móvil, esto dentro del proyecto sobre votación de la asignatura Egc (Evaluación y Gestión de la Configuración), para ello se ha decidido utilizar la aplicación telegram que es gratuita y utilizar el servicio de bots que te da esta plataforma, estos bots se pueden programar con una api abierta que te ofrece telegram.

Para llevar a cabo esta tarea, hemos decidido usar las siguientes herramientas:

* Para la gestión del código se utiliza Git.
* Para la gestión de las incidencias se utiliza GitHub.
* Para la comunicación entre el grupo usamos Discord y Telegram.
* Para la modificación del código usamos Sublime Text 3.

//TODO añadir mas contenido cuando se termine el proyecto

# Gestión del código fuente

## Herramientas

Para la gestión del código se usará la herramienta de control de versiones Git, así como la plataforma GitHub para la coordinación de todos los miembros del equipo. las operaciones de gestión de código se hacen a partir de la consola de comandos , ya que así se nos enseñó en las prácticas de la signatura Egc.

## Gestión de ramas

La gestión de ramas en el repositorio remoto y sobre el que se articulará el trabajo del equipo son dos: “master” y “dev”.

* Rama master: aquí se mantendrán las versiones estables del sistema. La última versión de esta rama será nuestra línea base a la hora de desarrollar el código. Esta rama solo se conecta con la rama dev, es decir, solo se pueden obtener modificaciones de esta rama, y no de cualquier forma, sino siguiendo un procedimiento de control de cambios.
* Rama dev: En esta rama es donde se trabajará y se subirá las modificaciones antes de pasar a la rama master. Una de las reglas que sea decidido es que no se deberá subir código que contenga errores de compilación o errores que no permitan el funcionamiento del proyecto.

## Procedimientos de gestión de código

De forma local, se puede trabajar directamente en la rama dev para cambios menores, pero para el resto de cambios, se define el siguiente procedimiento de gestión de código:

* Cuando un trabajador quiera implementar algún cambio, debe crear una rama nueva a partir de la última versión en dev.
* Con cada commit se debe implementar algún cambio, o se debe hacer referencia a la incidencia que se ha tratado.
* En las modificaciones realizadas que no han supuesto la implementación completa de un cambio, no es necesario que sean subidas al repositorio remoto si es el caso de que se continuará trabajando en esta modificación mas adelante, en caso contrario como por ejemplo que no se haya terminado la modificación por encontrar incidencia q no se han sabido resolver, si se deberá subir los cambios al repositorio remoto de esa rama.
* En las modificaciones realizadas que, si suponen la total implementación del cambio, se debe realizar un commit que haga referencia a la resolución de la incidencia o del cambio en cuestión, todos los miembros asignados deben comprobar la validez del cambio. Una vez aceptado, se pueden unir los cambios con la rama dev.