**GESTIÓN DE LA CONFIGURACIÓN**

1. Estándares de Código

Respecto a los estándares de código, se priorizará la creación de código claro y limpio, evitando anotaciones innecesarias y dejando pequeños comnentarios que ilustren lo que hace cada función. Se procurará mantener una coherencia tanto en el idioma empleado para los nombres de las clases, como con los nombres en cuestión, evitando llamar a un mismo elemento por diferentes nombres en diferentes funciones.

1. Políticas de mensajes de commit

Los commits se harán siguiendo un formato que sigue la siguiente estructura:

Título: tipo(rama): “titulo”

Descripción: breve descripción indicando que se ha modificado o añadido y por qué.

Siendo:   tipo = {init, docs, code, test, fix, refact, deploy, config...}

* 1. rama = rama a la que va dirigida dicho commit
  2. titulo = cambios importantes realizados

De esta forma, un ejemplo de mensaje de commit sería:

Docs(Tarea-A2.7): “Documento de gestión de configuración” “Se ha añadido a la carpeta de documentos el archivo que refleja la gestión de la configuración del proyecto”

1. Estructura de repositorio y ramas

Para la gestión del código fuente se ha utilizado la estrategia de ramas de GitFlow, en la cual encontramos la rama main, la rama development y una rama para cada desarrollo de una nueva funcionalidad.

1. Estrategia de ramas GitFlow
   1. Desarrollo de Futuras ramas

Para cada tarea o funcionalidad nueva a implementar, se creará una rama particular. Los commits se realizarán a estas ramas de funcionalidades, cuya nomenclatura será Tarea-AX.Y-z, siendo la X y la Y los números de la tarea, y z el apartado de la misma.

* 1. Preparación de entregas

Para las entregas, se revisará todo el contenido de la rama development para asegurar su correcto funcionamiento. Después, se hará un merge de esa rama a la rama main. Por último, se procederá al despliegue de la versión de la rama main de la aplicación en App Engine.

* 1. Arreglo de bugs en producción

Para reportar errores se creará una plantilla. En la plantilla de errores, se especificará, al menos, un título corto y descriptivo, los pasos a seguir para obtener el error y el sistema en el que se ha ejecutado. En todas las incidencias se deberá añadir a uno o varios encargados de solucionarla. También se debería añadir al menos tres etiquetas que especifiquen el tipo, la prioridad y el estado.

Como etiquetas para asignarle a dichas incidencias tenemos: urgent, medium priority y low priority (para la prioridad de la incidencia), new, accepted, started, fixed y verified (para el estado de la incidencia), help wanted...

1. Política de versionado

Para el versionado se empleará un versionado semántico en el que la nomenclatura tendrá la estructura X.Y.Z, donde X será la versión a gran escala, Y será la versión menor dentro de la X, y Z será el número del parche de dicha versión. Un incremento en la X reseteará la Y y la Z, y un incremento en la Y reseteará el valor de Z.