



UNIVERSIDAD DE GRANADA

GESTIÓN DE LA CONFIGURACIÓN Y SEGUIMIENTO DEL PROYECTO

Autores

Juan Manuel Castillo Nievas
Víctor Torres de la Torre



MÁSTER PROFESIONAL EN INGENIERÍA INFORMÁTICA

Granada, 25 de enero de 2021

1. Identificación de los CSCI del proyecto

- **Código fuente de la aplicación:**
 - **Nombre:** Código_Fuente
 - **Versión:** 1.0
 - **Estado:** En desarrollo
 - **Localización:** UCIX/src/Código_Fuente
- **Imágenes de la interfaz:**
 - **Nombre:** IMG
 - **Versión:** 1.0
 - **Estado:** Terminado
 - **Localización:** UCIX/IMG
- **Diagramas sobre el proyecto de diferentes tipos:**
 - **Nombre:** Diagramas
 - **Versión:** 2.0
 - **Estado:** Terminado
 - **Localización:** UCIX/DOC/Diagramas
- **Consultas a BBDD:**
 - **Nombre:** BBDD
 - **Versión:** 2.0
 - **Estado:** En desarrollo
 - **Localización:** UCIX/src/BBDD
- **Planificación/estimación del proyecto:**
 - **Nombre:** Planificación_Estimación
 - **Versión:** 2.2
 - **Estado:** Terminado
 - **Localización:** UCIX/DOC/Planificación_Estimación
- **Presupuestos del proyecto:**
 - **Nombre:** Presupuestos
 - **Versión:** 1.2
 - **Estado:** Terminado
 - **Localización:** UCIX/Presupuestos
- **Documentos relacionados con la selección de personal:**

- **Nombre:** Selección_Personal
- **Versión:** 1.3
- **Estado:** Terminado
- **Localización:** UCIX/Selección_Personal
- **Cálculos de red temporal:**
 - **Nombre:** Red.Temporal
 - **Versión:** 2.5
 - **Estado:** Terminado (puede necesitar modificaciones)
 - **Localización:** UCIX/DOC/Red.Temporal
- **Documentos de la gestión de riesgos:**
 - **Nombre:** Gestión_Riesgos
 - **Versión:** 1.2
 - **Estado:** Terminado (puede necesitar modificaciones)
 - **Localización:** UCIX/DOC/Gestión_Riesgos
- **Documentos de la gestión de calidad:**
 - **Nombre:** Gestión_Calidad
 - **Versión:** 1.5
 - **Estado:** Terminado (puede necesitar modificaciones)
 - **Localización:** UCIX/DOC/Gestión_Calidad

2. Decidir el tipo de sistema de control de versiones que se desea utilizar

Se utilizará GIT como sistema de control de versiones con su estructura en árbol, no solo porque todos los integrantes del equipo están familiarizados con este sistema de control de versiones si no porque creemos que es el más adecuado para nuestro proyecto.

Además, el repositorio se encontrará almacenado en Github para que sea visible al cliente y tener total transparencia en nuestras decisiones.

3. Generación de la estructura de directorios del proyecto

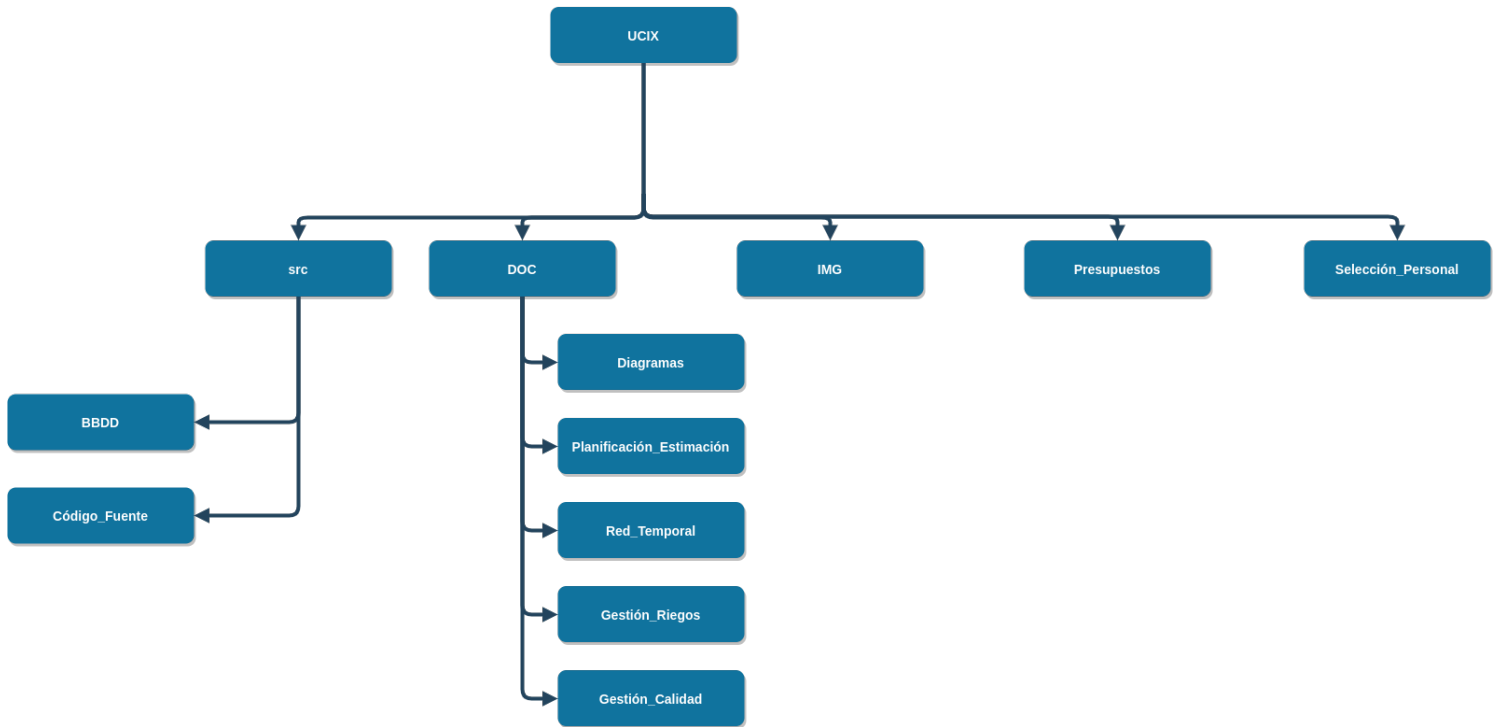


Figura 1: Estructura de directorios

4. Git Workflow para desarrollar el software

Para recordar un poco, este proyecto consistía en realizar la interfaz de usuario del producto **UCICX**. Se van a crear 5 ramas que representan cada interfaz de usuario creada. Cada rama incluirá commits que representarán las distintas tareas que hay que añadir dentro de cada interfaz. El Git Workflow se ve en la Figura 2.

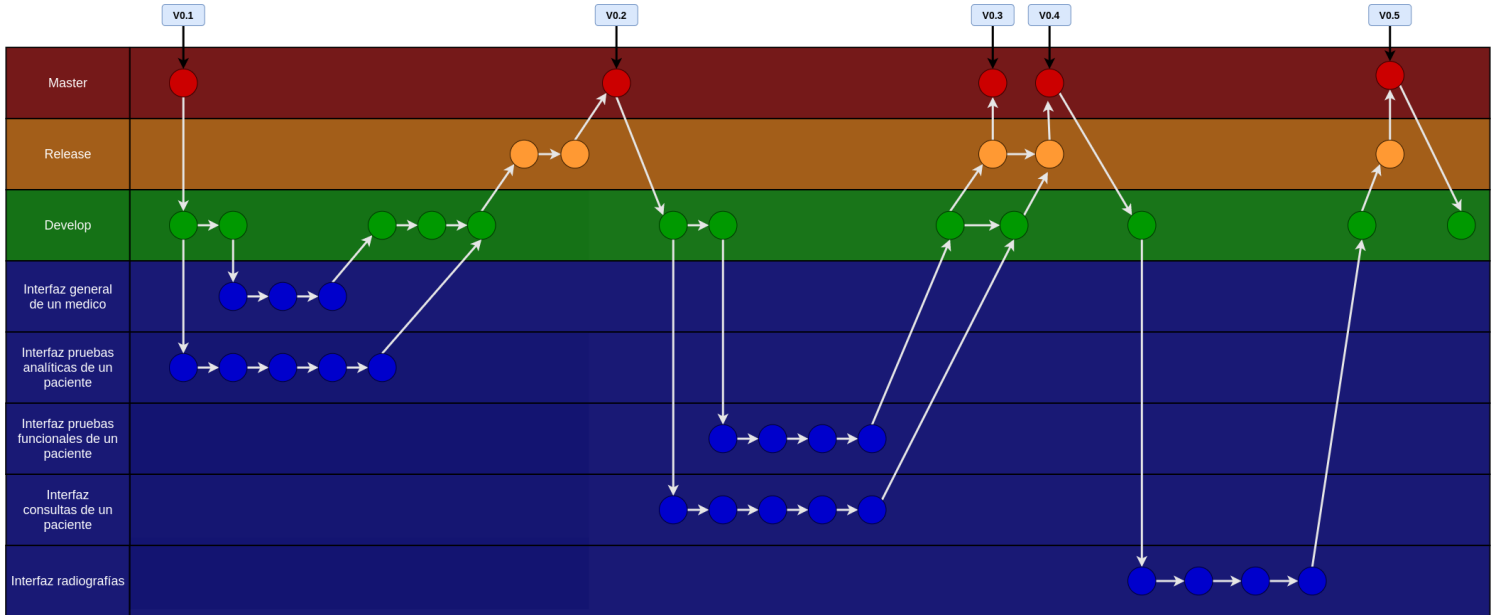


Figura 2: Git Workflow

5. Política de reuniones

Al iniciar el proyecto, se realizará **una reunión de aproximadamente 2 horas** en la que se hablará de las distintas tareas a realizar y posibles inconvenientes que pueden aparecer y cómo solucionarlos en caso de que ocurran.

Una vez empezado el proyecto, se realizarán **reuniones diarias SCRUM de sprint diario de no más de 15 minutos** en las que se hablará de los siguientes temas:

- Qué hizo el día anterior cada empleado
- Problemas encontrados en alguna tarea y cómo se podría solucionar

- Qué se va a hacer durante el día de hoy
- Reorganización de tareas, sólo en caso de que sea necesario
- Programación de una futura reunión más larga con el equipo, sólo en caso de que sea necesario
- Planteamiento de una reunión con el cliente cuando se encuentren problemas graves que hayan surgido por un mal entendido en los requisitos y que no hayan quedado claros, sólo en el caso de que sea necesario. Estas reuniones no durarán más de una hora y deberán establecerse en la misma semana en la que ocurre el conflicto.

Cada vez que una rama (interfaz) esté finalizada, se hará una reunión de **aproximadamente una hora** en la que la persona encargada de la rama presentará la interfaz realizada y, por lo tanto, lo que se va a *mergear* en la rama *release* y posteriormente *master*. En el caso de que todo se encuentre bien, se procederá a *mergear* y se asignará una nueva rama al empleado. En el caso de que algo no esté del todo bien, la reunión **se podrá alargar como máximo otra hora más** en la que se propondrán nuevos cambios y se reorganizarán las tareas entre los empleados en caso de que sea necesario.

6. Lista de aspectos que se analizarán una vez desarrollado el software

- Estructura de directorios usada
- El Git Workflow final que ha quedado tras entregar el proyecto
- Número de ramas y commits creado
- Utilidad de las distintas reuniones realizadas haciendo también una comparación calidad-duración
- Conocimientos que no se sabían por no tener la formación previa necesaria y que hicieron perder tiempo (hablamos de problemas que implican perder tiempo en días, no aquellos pequeños problemas que se solucionan buscando en menos de media hora)
- Estructura que se pensó para cada reunión con el cliente
- Estructura y diseño de la documentación generada
- Código generado:
 - Nombres de las variables y métodos
 - Documentación de cada método
 - Limpieza de código
 - Si ha habido mezcla de español-inglés a lo largo del código