

Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos



Agora_US: Verificación

Asignatura: Evolución y Gestión de la Configuración

Grupo: 1 (Mañana)

Curso: 2016/2017

Hernán González Peñalver.

Manuel de la Fuente Carmona.

Álvaro Sánchez López.

Alejandro Sanabria Cruz.

Lidia Sánchez Vaz.

Índice

Enlaces.....	3
Resumen.....	3
Introducción y contexto	4
Descripción del sistema.....	5
Elementos de control	5
Entorno de desarrollo	5
Gestión de código fuente	6
Gestión de la construcción e integración continua.....	6
Gestión de incidencias	6
Gestión de despliegue.....	7
Mapa de herramientas.....	7
Conclusión	8

Enlaces

- Repositorio de código: <https://github.com/AgoraUS-G1-1617/Verification>
- Repositorio compartido: <https://github.com/AgoraUS-G1-1617>
- Repositorio de incidencias: <https://github.com/AgoraUS-G1-1617/Verification/issues>
- Wiki: <https://1984.lsi.us.es/wiki-egc/index.php/Verificaci%C3%B3n>
- Máquina virtual:
<https://drive.google.com/open?id=0B3lfSTvPNdKILUpVWFpTallRYzA>

Resumen

En la asignatura de Evolución y Gestión de la Configuración se propone a los alumnos la realización de una versión nueva llamada Ágora US del proyecto proporcionado por AgoraVoting.

El proyecto está dividido en una serie de subsistemas en los que cada grupo de alumnos de la asignatura van a participar en el desarrollo, actualización y mejora de cada una de las partes que componen el proyecto propuesto por la asignatura.

En nuestro proyecto vamos a presentar una breve introducción sobre el subsistema que hemos elegido para realizar las mejoras; se plantean todos los problemas encontrados desde el inicio hasta el resultado final del proyecto entregado al igual que las posibles soluciones encontradas y llevadas a cabo.

También se indican las diferentes políticas de gestión que hemos utilizado para el desarrollo del proyecto que más adelante explicaremos detalladamente, se hará una visión de las herramientas utilizadas y del uso que se le ha dado en nuestro subsistema y para finalizar una conclusión global de lo aprendido y de lo que hemos conseguido realizar en nuestro proyecto.

Las políticas de gestión utilizadas son las siguientes:

- Gestión de código fuente: explicar cómo se gestiona el código, las políticas seguidas y cómo hemos resuelto los conflictos.
- Gestión de la construcción e integración: las herramientas usadas para la construcción y automatización con los demás subsistemas para lograr la integración.
- Gestión de incidencias: cómo informamos y nos organizamos los miembros del equipo ante un problema, si se necesitan cambios o algún que otro ajuste.
- Gestión de la calidad: asegurar que el código cumpla con los requisitos establecidos para mejorar la calidad tanto del código como del proyecto.
- Gestión de despliegue: explicaremos como hemos desplegado nuestro proyecto para poder ser usado por los demás subsistemas.

Introducción y contexto

Para entrar en contexto con el subsistema que hemos desarrollado por parte de nuestro equipo de trabajo, trabajamos sobre Agora Voting, basado en un sistema de votaciones con sus correspondientes subsistemas que procedemos a indicar:

- Autenticación
- Creación y Administración de Votaciones
- Sistema de modificación de Resultados
- Almacenamiento de Votos
- Deliberaciones
- Recuento
- Creación/Administración de Censos
- Frontend de Resultados
- Visualización de Resultados
- Verificación
- Cabina de Votación

Cada uno de estos subsistemas será desarrollado por cada uno de los grupos de alumnos establecidos. Nuestro grupo de proyecto está basado en Verificación en Ágora US.

Las funcionalidades que presenta el subsistema de Verificación elegida por nuestro grupo son las siguientes:

- Crear una clave pública y privada dada una votación
- Comprobar si un voto ha sido cifrado o no
- Comprobar si un voto ha sufrido alguna alteración

Con respecto a la integración, nuestro subsistema necesita dependencias e integrar con Creación/Administración de Votaciones, Cabina de Votación y Recuento, a los que vamos a proporcionarle el despliegue de nuestro proyecto para que ellos puedan usarlo.

Descripción del sistema

Nuestro sistema como hemos enunciado antes se basa en verificar, proporcionando unas claves públicas y privadas, que el voto ha sido cifrado correctamente para el funcionamiento del subsistema. Para ello, nuestro subsistema será encargado de cifrar y descifrar los votos que nos pidan desde los subsistemas con los que hacemos la integración. Para empezar a implementar nuestras nuevas funcionalidades, se eliminan sistemas de encriptado (DES) que no van a usarse, se empieza a trabajar sobre el sistema de encriptado *RSA* que será el que utilizaremos en nuestro proyecto, se eliminan algunas dependencias con bases de datos que no usaremos para el despliegue de nuestro subsistema y se empieza a realizar todos los cambios propuestos para llevar a cabo el proyecto con éxito.

En cuanto a la planificación, nos planteamos la opción de trabajar todos a la vez hasta que pudiéramos dividir el trabajo en pequeñas partes a desarrollar pese a la mala gestión del proyecto que heredamos del año anterior. Nos encontramos con un subsistema que apenas había podido desarrollar todo lo que el subsistema debería y cuya parte realizaban otros subsistemas para que el proyecto acabase lo mejor posible. Una vez planteado el proyecto, cada miembro del equipo se encarga de realizar una parte del proyecto para que cada uno pudiese trabajar en local. Cada miembro del equipo podrá unirse a las *issues* que se vayan creando en el repositorio.

Elementos de control

Para el control y gestión de la configuración, nos basaremos en la utilización de la herramienta estudiada en la asignatura GitHub que será mediante la cual vamos a identificar las diferentes tareas que cada miembro del equipo vaya utilizar e ir observando constantemente todos los posibles cambios de forma que aparezca en detalles el código cambiado de cada miembro del equipo que vaya realizando cambios y mejoras en el proyecto.

Entorno de desarrollo

Nuestro elemento principal de desarrollo para el subsistema de Verificación es la aplicación de Eclipse para el uso del código en entorno JAVA. Ya que el proyecto tiene que ser integrado con el resto de subsistemas, se usa la compilación del proyecto en formato *.JAR* para que pueda ser accesible a través de líneas de comando. En nuestro subsistema no ha sido necesario la instalación de una máquina virtual puesto que todos los miembros del equipo hemos desarrollado el código del proyecto en nuestro Eclipse de forma local y no hemos requerido la instalación de ningún otro componente.

Gestión de código fuente

En un principio, cada vez que queramos realizar un cambio en el código o seguir trabajando en nuestro proyecto, un miembro del equipo realizará una *issue* para que pueda ser asignada por otro miembro y ponerse a trabajar en ella. Una vez que la *issue* quede asignada, el encargado podrá empezar a trabajar en los cambios. Conforme vayan haciéndose cambios, el código que esté en desarrollo se le hará *commit* a la rama *developer*. Cuando el cambio se considere finalizado, se harán las pruebas y *tests* necesarios para que el jefe de Proyecto pueda comprobar el cambio, y decidir si se sube a la rama 'master' para incorporar de manera definitiva el nuevo cambio a nuestro proyecto. Excepcionalmente se podrán abrir ramas adicionales con propósitos concretos, como probar ideas, implementaciones alternativas etc.

Gestión de la construcción e integración continua

La integración y construcción de nuestro subsistema con el resto de subsistemas ha sido una compilación del proyecto en Eclipse en un formato .JAR. Éste será consumido por el subsistema de Recuento de votación, que será accesible a través de nuestro repositorio en GitHub. Una vez que ellos estén dispuestos a consumir de nuestros recursos, podrán hacerlo con nuestro .JAR a través de la línea de comandos:

```
java -jar verification.jar <comando> [parametros]
```

Con esto, ya pueden crear los pares de claves y realizar los cifrados y descifrados de las claves públicas y privadas que sean necesarias.

Gestión de incidencias

Cualquier miembro del equipo puede publicar y asignar incidencias, y será responsabilidad del jefe de proyecto, Hernán González, darlas por válidas, innecesarias o concluidas. En caso de que se genere un número elevado de incidencias, se utilizará el sistema de etiquetado para organizarlas.

Todas las incidencias que informen de un error en el comportamiento del sistema deberán incluir pasos para reproducir el error, salida esperada y salida obtenida.

Cuando una incidencia sea marcada como resuelta, un colaborador diferente al que la resolvió revisará que la solución sea correcta.

GitHub nos proporciona un amplio abanico de etiquetas para ayudar a clasificar las incidencias que reportamos, sin embargo, no las utilizamos todas. Según el objetivo de la *issue*, añadimos las siguientes etiquetas:

- Utilizamos la etiqueta de *bug* cuando se trata de un fallo o error detectado.
- Utilizamos la etiqueta *enhancement* cuando se trata de una mejora que se pretende hacer.

Gestión de despliegue

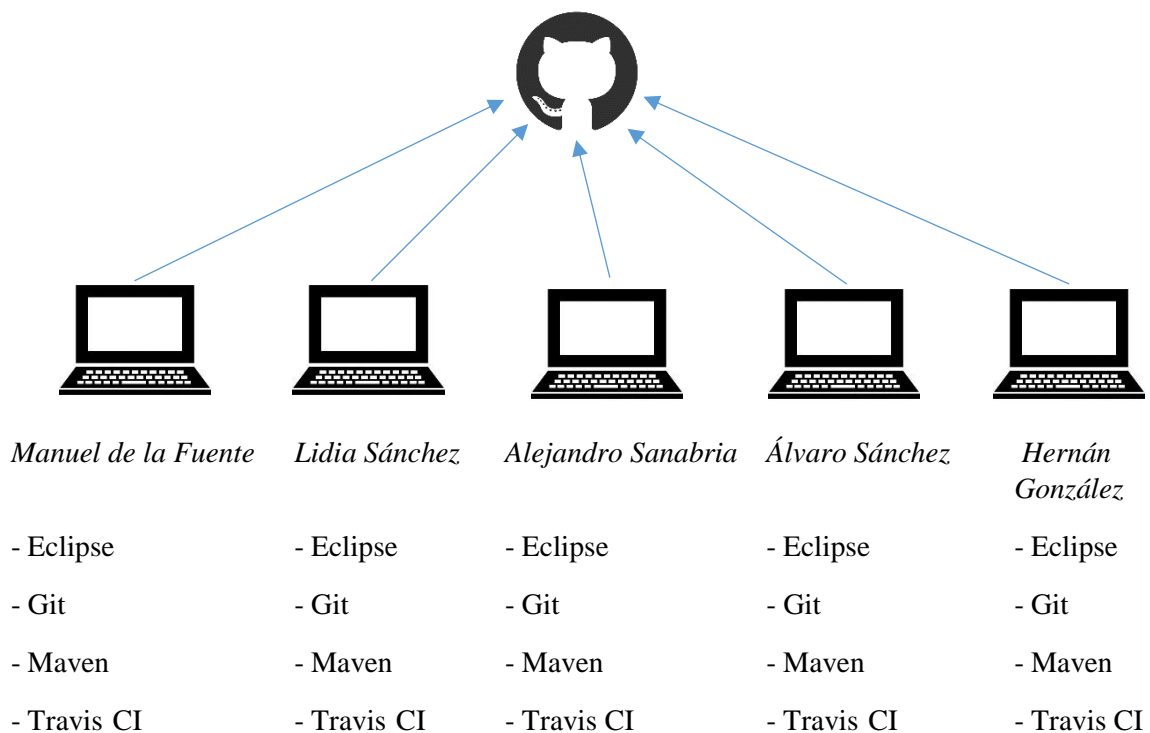
Para desplegar nuestro proyecto una vez que tengamos una versión consolidada de lo que hemos realizado en el código, podemos realizarlo de dos maneras posibles; a través de línea de comandos usando *Maven* o a través de la misma plataforma de Eclipse. Para hacerlo con *Maven*, solo tenemos que introducir un comando en líneas de comando de Windows que es la siguiente:

```
mvn clean compile package assembly:assembly
```

Para realizar la exportación en Eclipse tenemos que seguir los siguientes pasos:

- Clic derecho en el proyecto en Eclipse
- Clic en Export
- En el selector de export wizard, Java > JAR file
- Select the export destination: <camino>/verification.jar
- Clic Next hasta la última página: Select the class of the application entry point:Browse y se marca esta clase.
- Finish

Mapa de herramientas



Conclusión

El proyecto que nos encontramos en el repositorio del año anterior era un subsistema cuya utilidad era nula, ya que no era funcional y no hemos podido sacar nada que nos sirviera de ayuda para nuestro desarrollo. Por eso ha sido fundamental realizar las mejoras en nuestro proyecto, para que fuese funcional y pudiera ser consumido por los demás subsistemas.

A pesar de los problemas encontrados para la gestión del código que había del repositorio anterior, hemos sabido hacer de un subsistema no funcional a cumplir todas las funcionalidades que nos piden para el desarrollo del subsistema de Verificación.

No nos ha sido posible la integración de forma automática con el resto de subsistemas para que puedan consumir de las funciones que ofrecemos, por lo que este punto sería una mejora para el futuro de este proyecto, y que puedan consumirse nuestros servicios de forma más fácil y directa, sin tener que ir descargando nuestro archivo .JAR para poder hacer la integración con los demás subsistemas.