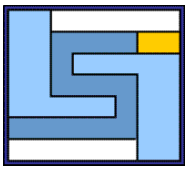


Escuela Técnica Superior de Ingeniería Informática

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

- 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 formativa del sistema de votación online AgoraVoting llamada ÁgoraUS.

El proyecto global está dividido en una serie de subsistemas que serán asignados a un determinado grupo de alumnos de la asignatura que participarán en el desarrollo, actualización y mejora del subsistema seleccionado. Tratando cada subsistema como un proyecto exclusivo que posteriormente será integrado con el resto de proyectos (o subsistemas).

En este documento, vamos a presentar una breve introducción sobre el subsistema que hemos elegido para realizar las mejoras; se plantearán 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 (más adelante las 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 abarcadas en el proyecto incluyen:

- Gestión de código fuente: explicar cómo se gestiona el código, las políticas seguidas y cómo hemos resuelto los conflictos de código.
- 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 liberaciones: explicaremos como liberamos nuestro proyecto para que pueda ser consumido por los subsistemas relacionados.

Introducción y contexto

Para entrar en el contexto de los proyectos a realizar en la asignatura, y basándonos en Agora Voting, los correspondientes subsistemas a desarrollar serían:

- 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 un grupo de alumnos. Nuestro grupo será el encargado de desarrollar y mejorar el subsistema de Verificación en Ágora US.

Las funcionalidades que presenta nuestro subsistema de Verificación son las siguientes:

- Crear un par de claves clave pública y privada para cifrar y descifrar votaciones.
- Cifrar un voto dado.
- Descifrar un voto dado.

Con respecto a la integración, nuestro subsistema necesita integrarse en los siguientes subsistemas:

- Recuento y modificación: necesitan de nuestro subsistema para obtener el par de claves (de cifrado y descifrado), así como la posibilidad de descifrar el texto que nos indiquen a través de nuestra funcionalidad.
- Cabina de votaciones y cabina de Telegram: se encargarán de cifrar datos a través de nuestros métodos y haciendo uso de la clave pública (generada por nuestro subsistema) que Recuento ofrecerá a través de su API.
- Creación y administración de votaciones.

Descripción del sistema

Nuestro sistema de Verificación será el encargado de generar el par de claves (pública y privada) necesarias para el cifrado y descifrado de votos. Similarmente ofreceremos la capacidad 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 simétrico (DES) que no va a usarse; se empieza a trabajar sobre el sistema de encriptado asimétrico *RSA* que será el que utilizaremos en nuestro proyecto; se eliminará toda gestión con base de datos pues no seremos los encargados de almacenar ningún tipo de datos 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 identificáramos paquetes de trabajo que nos permitiesen dividir el proyecto en tareas a desarrollar individualmente, debido a la mala gestión del proyecto que heredamos del año anterior.

En un primer momento nos encontramos con un subsistema corrupto, sin utilidad para el resto de subsistemas que ya habían desarrollado las tareas necesarias para el funcionamiento global del sistema y que deberían haber sido desarrolladas por Verificación. Una vez planteado el proyecto y definir las funcionalidades que podemos ofrecer al resto de subsistemas, cada miembro del equipo se encarga de realizar una parte del mismo de manera local para su posterior integración en el repositorio del proyecto.

En términos de gestión de incidencias, se utilizan las issues de GitHub, de manera que cada miembro del equipo podrá crear y unirse a las incidencias que se vayan en el repositorio del equipo.

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

Nuestra herramienta de desarrollo principal para el subsistema de Verificación es la aplicación de Eclipse para código JAVA. Ya que el proyecto tiene que ser integrado con el resto de subsistemas, se usa la compilación del proyecto en formato .JAR y se crearán los métodos correspondientes para que este sea accesible a través de línea de comandos. En nuestro subsistema no ha sido necesaria la instalación de una máquina virtual puesto que todos los miembros del equipo hemos desarrollado el código del proyecto de forma local y no hemos requerido la instalación de ningún otro componente más allá de las usadas en las clases prácticas

y de obligatoria instalación (Maven, Travis, etc.). Aun así, se ofrece una máquina virtual con las herramientas y componentes necesarios para el desarrollo del sistema a fin de cumplir con los requisitos de entregables de la asignatura.

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 (o autoasignada) 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 verificarlo, 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 formato .JAR. Éste será consumido por los subsistemas con los que nos integramos y estará accesible a través de nuestro repositorio en GitHub bajo el nombre: *verification.jar*

Una vez que ellos dispongan de nuestro .JAR, podrán empezar a consumir de nuestros recursos a través de la línea de comandos:

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

De este modo podrán utilizar las funcionalidades ofrecidas (generación de claves, cifrado y descifrado) por nuestro subsistema.

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. Además se utilizará el sistema de etiquetado para organizar las incidencias en función de su carácter. También es posible personalizar el etiquetado para dotar estas incidencias de estados (stated, fixed, consulting, etc.).

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 resuelta, el jefe de proyecto deberá comprobar que la resolución es correcta y será su decisión añadir dicha modificación a la rama principal (master).

Entre todas las etiquetas ofrecidas por GitHub para clasificar las incidencias que reportamos, son dos las que más hemos utilizado:

- 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 liberaciones

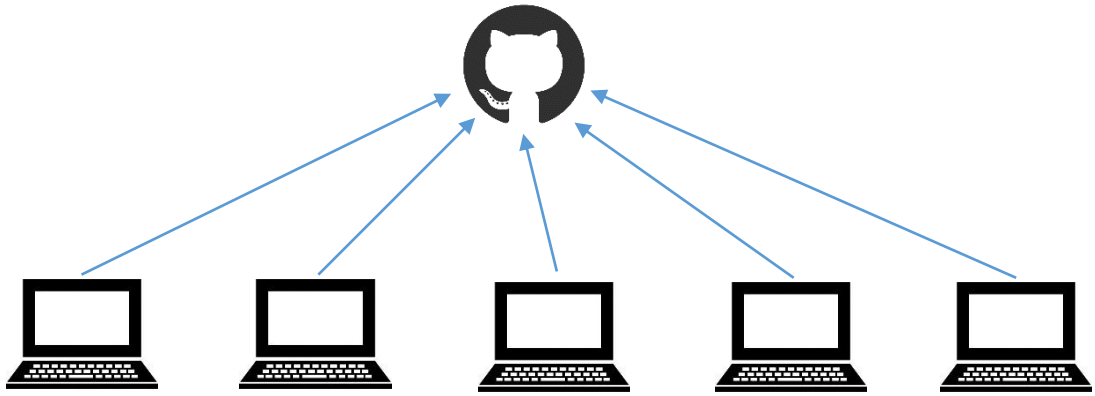
Para desplegar nuestro proyecto una vez que tengamos una versión consolidada de lo que hemos realizado en el código, deberemos generar el archivo Java (JAR). Para ellos, a través de línea de comandos usando *Maven* o a través de la misma plataforma de Eclipse. Para hacerlo con *Maven*, solo tenemos que 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

Una vez generado el archivo, se subirá al repositorio de GitHub desde donde estará accesible para el resto de subsistemas.

Mapa de herramientas



Alejandro	*Álvaro*	*Hernán*	*Manuel*	*Lidia*
- Eclipse	- Eclipse	- Eclipse	- Eclipse	- Eclipse
- Git	- Git	- Git	- Git	- Git
- Maven	- Maven	- Maven	- Maven	- Maven
- Travis CI	- Travis CI	- Travis CI	- Travis CI	- Travis CI

Conclusión

El proyecto que nos encontramos en el repositorio del año anterior era un subsistema sin utilidad alguna, ya que no era funcional y no hemos podido sacar nada que nos sirviera de ayuda para nuestro desarrollo.

Por ello, nuestra principal preocupación desde el principio fue que el subsistema fuese funcional y pudiera ser consumido por los demás subsistemas. A partir de ahí, y tras emplear más tiempo del deseado en la investigación de la funcionalidad que debía implementar Verificación en un sistema global que ya había sido desarrollado el año anterior sin nuestro subsitema, intentamos aplicar mejoras para facilitar la integración al resto de subsistemas.

A pesar de los problemas encontrados por la gestión del código que había del repositorio anterior, hemos sabido hacer de un subsistema funcional que cumple con las funcionalidades principales que debe implementar un subsistema de verificación.

No nos ha sido posible la automatización del despliegue de manera que nos vemos obligados a subir “a mano” el .JAR generado a partir de los nuevos cambios que vayamos aplicando al subsistema. Se plantea como posible mejora para el grupo del curso venidero que opte por este subsistema.