**Subsistema de**

**Autentificacion**



Javier Montilla Miranda

EGC

10/09/16Indice

Control de versiones**1.Resumen**

En este trabajo realizado para la asignatura de Evolucion y Gestion de la Configuracion (EGC) intentaremos reforzar y añadir funcionalidad al proyecto denominado Agora Voting el cual despues de años realizandose por otros compañeros esta bastante avanzado.

Este sistema esta diseñado para agilizar tramites y poder realizar votaciones online por parte de la comunidad universitaria, concretamente esta destinado a los estudiantes y para que expresen su opinion/voto mediante estas herrramienta, ademas todo esto los resultados podran ser analizados y sacar conclusiones mas facilmente que si tuvieramos que votar mediante urnas u otro sistema de votacion.

El sistema se divide en varios subsistema , concretamente este trabajo se va a encargar de la parte del subsistema de autentificacion la cual es indispensable para poder introducirse en el sistema y realizar la votacion mas tarde.

Como hemos podido observar de otros años este subsistema tiene varias mejoras introducidas y algunos compañeros de años anteriores y estos años han facilitado y han puesto sus conocimientos para poder incluir en ellos sistema de autentificacion como UVUS (Usuario Virtual US) o incluso mediante cuenta Google.

Despues de una reflexion sobre las mejores oportunas en este subsistema vamos a intentar incluir en el subsitema la opciones de poder identificarse tanto con Linkedlin como con Facebook por razones que mas tarde desarrolllaremos.

En la presente memoria describimos como se ha desarrollado el proyecto de la asignatura de Evolución y Gestión de la Configuración (EGC), que consiste principalmente en integrar nuestro proyecto que corresponde a un subsistema, con otros subsistemas para ofrecer una herramienta de voto online.

Al estar realizando este trabajo de manera individual por diversos motivos , la toma opiniones en numerosas ocasiones han tenido que salir de personal fuera del ambito estudiantil, cosa excepcional , pero esto no significa que sea malo, ya que con estas opiniones me he podido acercar mas al funcionamiento basico que quieren las personas de a pie y con ello intentar dotar al subsistema de las maximas facilidades de uso posibles para introducirse en el , ya que cabe recordar que este sistema lo usaran tanto usuarios habituales como usuarios inexpertos los cuales les vendra muy bien una herramienta facil e intuitiva (por ejemplo los universitarios de primer año).

Ademas de todo esto se ha seguido el guion de trabajo de esta asignatura correctamente y por ello dejaremos constancia en este trabajo de las diferentes herramientas, procesos , tiempos, politicas de trabajo y pautas que han sido necesarias para realizar completamenta la integracion y mejoras de estas nuevas funcionalidades.

# 2. Introducción

Como hemos indicado anteriormente vamos a trabajar sobre un proyecto desarrollado poco a poco durante años anteriores por universitarios de la escuela, el proyecto se llama Agora Voting y tiene numerosos subsistemas los cuales se han ido modificando y añadiendo nueva funcionalidades a lo largo de los años.

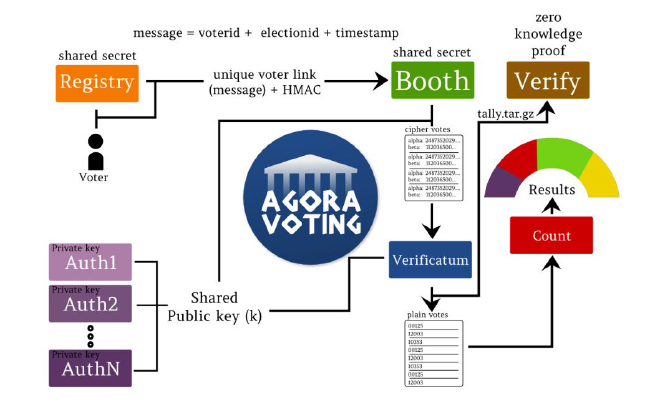


Figura 1. Arquitectura de Agora Voting

Referencia: <https://1984.lsi.us.es/wiki-egc/index.php/Lista_de_proyectos_a_realizar_14-15>

Como podemos apreciar en la figura que se muestra a continuacion el sistema esta compuesto por varios subsistemas :

* Verificación.
* Artefactos comunes.
* Cabina de votación.
* Autenticación.
* Recuento y modificación de resultados.
* Almacenamiento de votos.
* Creación y administración de censos.
* Deliberaciones.
* Creación y administración de votaciones
* Front-end y visualización de resultados.

**3.Subsitema Autentificacion**

Nuestro subsitema a desarrollar sera el de Autentificacion se encargará de realizar la autenticación del usuario antes de que este acceda al resto de la aplicación. Con esto quedara constancia de su estado como autentificado , ademas podemos indicar que este subsitema es uno de los mas importante ya que sin el no se podra acceder a la plataforma y poder realizar una votacion. Aunque todos tenga importancia este subsitema en esta plataforma concretamente es de vital importancial ya que gracias a el tendremos seguridad e identificacion del usuario que vota.

Este subsistema se encargara tambien de controla quien puede votar en cada tipo de votacion y tambien quien ha realizado su voto para evitar que se realicen multiples votos de una misma persona. Ademas de todo esto el sistema debe ser facil e intuitivo tanto en el manejo de usuario como en la codificacion ya que muchos otros subsistemas se van ayudar de este para realizar comprobaciones o mirar algunos datos .

A continuacion haremos un breve resumen de los diferentes puntos que nuestro trabajo debe contemplar para esta asignatura , estos puntos nos han sido facilitados en una guia la cual debemos de seguir para realizar este trabajo .

* **Elementos de control:** se enumeran los elementos de gestión de la configuración y cómo se va a gestionar la configuración de cada uno de ellos.
* **Entorno de desarrollo:** se describe cuál es el entorno de desarrollo que se ha usado y los pasos para instalar nuestro sistema como los subsistemas relacionados para hacer funcionar el sistema completo.
* **Gestión del código fuente**   Se explicarán los procesos , técnicas y herramientas para la gestión de código del proyecto. En las actualidad todo el codigo se puede encontrar en google drive ya que antiguamente se encontraba en Githun pero este repositorio no se encuentra en la actualidad.
* **Gestión de la construcción y de la integración:** similar al apartado anterior se definirán los procesos que se usan a la hora de contruir el proyecto ademas se describen las técnicas de integración continua, principalmente usaremos Jenkins (recomendación de la asignatura) para automatizar el trabajo de las tareas cuando se produzcan cambios en el repositorio, se especificará con más detalle en el apartado correspondiente.
* **Gestión del cambio, incidencias y depuración:** se describen las herramientas y los procesos que se usan y como se usan.
* **Gestión de liberaciones, despliegue y entrega:** se describe el procedimiento usado para desplegar en una máquina los subsistemas de forma que estén integrados. (No consta ya que el trabajo se hace documental ya que soy una sola persona)
* **Mapa de herramientas:** se describe un esquema de cómo se conectan las las herramientas usadas en el nuestro proyecto, que relaciones tienen o que relaciones propondría añadir.
* **Conclusiones**: En este apartado indicaremos las conclusiones que hemos sacado del trabajo y de nuestras modificaciones del subsitema elegido. Ademas podremos incluir las mejoras posibles en un futuro.

3.1.Descripcion funcional del nuevo sistema a implantar

Para comenzar con esta descripcion funcional tenemos que sitiarnos donde nos encontraremos antes de realizar una votacion, un usuario desea acceder al portal AgoraVoting para realizar una votacion por diversos motivos esto no seria posible sin una identificacion la cual es sumamente importante debido a que solo se podria contabilizar un voto por persona.

Pero como sabemos en muchas ocasiones no recordamos nuestras credenciales UVUS o no nos apetece hacer un registro nuevo en esta pagina por tanto mediante esta nueva funcionalidad facilitamos que el usuario acceda mediante dos de las redes sociales mas importante hasta el momento.

Por un lado tendremos Facebook, red social que cuenta con mas de 1600 millones de usuarios y la cual esta en plena expansion ademas esta red social puede presumir de tener a un gran numero de universitarios y profesores en su lista de usuarios .

Por otro lado y totalmente complementaria una red social denominada LinkedIn la cual ella misma se denomina como la mayor red de profesionales del mundo.

La eleccion de estas dos redes sociales se ha realizado basandonoes en las personas que pueden utilizar el sistema AgoraVoting los cuales seran en su mayoria estudiantes y profesores. Para estos dos grupos le sera muy facil acceder con las credenciales de estas redes sociales a AgoraVoting y poder dejar su voto facilmente .

Pero debemos indicar que el proceso de acceso se debe realizar de manera simple porque aunque estas redes sociales sean muy usuales no podriamos adaptar AgoraVoting bien si no nos facilitaran una api con la que poder interactuar facilmente.

Para ellos se ha realizado un estudia de las diferentes Apis que facilitan estas dos redes sociales y hay que indicar que las dos tienen algo en comun, son tanto faciles de usar por un usuario normal (no tiene porque ser usuario experto) ,como faciles de implantar en nuestro sistema AgoraVoting.

Pero no solo vamos a basar nuestra descripcion funcional en estas dos redes sociales , ademas de incluirlas vamos a intentar hacer una renovacion de nuestro Index (pagina de comienzo) la cual sea mucho mas moderna y su navegacion y usabilidad sea mas facil.

Index:

En nuestra pagina de inicio totalmente renovada aparacerá de forma muy vistosa y formal los tres principales pilares con lo que podemos introducirnos en nuestro sistema AgoraVoting: UVUS, FACEBOOK,LINKEDIN.

En las siguientes imágenes os dejamos un prototipo de cómo se vera el incio al intentar acceder tanto con Facebook como con LinkedIn , el cual nos pedira las credenciales correspondientes de estas redes sociales:

Facebook

linkedIn



3.2. Descripcion Tecnica

3.2.1) Descripcion tecnica login Facebook

La red social Facebook tiene reservado un espacio en su pagina web totalmente dedicado a desarrolladores y programadores, con manuales de ayuda, codigo reutilizable y demas funciones que nos sirven de gran ayuda para poder realizar este proyecto. (<https://developers.facebook.com/>)

Concretamente nosotros nos vamos a dedicar al desarrollo del login y autentificacion en portales Web ya que nuestro sistema AgoraVoting estara disponible en estos momentos solo para un portal web y no para otras plataformas aunque si que disponemos de informacion y documentacion para en un futuro acceder al portal a traves de esta red social desde distintas plataformas.

Para realizar el login a traves de Facebook vamos a utilizar SDK de Facebook para JavaScript con el cual resultara muy facil y sencillo de implementar esta funcionalidad, nos vamos a basar en 5 puntos principales para el desarrollo que seran:

1**) Comprobar el estado del inicio de sesion**.- esto se utiliza para comprobar si alguien ya tiene una sesion inicada en el navegador y poder continuar sin introducir las credenciales siempre que el usuario acepte.

Este proceso comienza con una llamada a [**FB.getLoginStatus**](https://developers.facebook.com/docs/reference/javascript/FB.getLoginStatus). Esta función activa una llamada a Facebook para obtener el estado de inicio de sesión y llamar a tu función de devolución de llamada con el resultado.

Esto es parte del código que se ejecuta durante la carga de la página para comprobar el estado de inicio de sesión de una persona, tomado del código de ejemplo anterior:

FB.getLoginStatus(function(response) { statusChangeCallback(response); });

El objeto response que se proporciona a la devolución de llamada puede contener uno o varios campos:

status: especifica el estado de inicio de sesión de la persona que usa la aplicación. El estado puede ser uno de los siguientes:

* connected: la persona inició sesión en Facebook y en tu aplicación.
* not\_authorized: la persona inició sesión en Facebook, pero no en tu aplicación.
* unknown: la persona no inició sesión en Facebook, de modo que no sabes si la inició en tu aplicación. O bien, se llamó a [FB.logout()](https://developers.facebook.com/docs/facebook-login/web#logout) con anterioridad y no se pudo conectar con Facebook.
* authResponse: se incluye si el estado es connected, y consta de los siguientes elementos:
* accessToken: contiene un token de acceso para la persona que usa la aplicación.
* expiresIn: indica la hora en formato UNIX en que el token caduca y se debe renovar.

2) **Peticion de Usuario y Contraseña**.- si el usuario no esta inciado se creara un cuadro de dialogo de inicio de sesion y la aceptacion de una serie de permisos.

Hay dos modos para permitir que una persona inicie sesión:

* **Utilizar el**[**botón "Iniciar sesión"**](https://developers.facebook.com/docs/plugins/login-button)**.**
* **Utilizar la llamada a FB.login() del SDK para JavaScript.**

Nosotros nos vamos a basar en la segunda modo para que lance un cuadro de dialogo en una ventana emergente en cual podamos introducir nuestras credenciales para ellos tan solo debemos debemos hacer referencia a la llamada FB.login()  en el boton HTML que indiquemos en la pagina.

3) **Verificacion de su identidad**.- Se comprobara que los datos obtenidos son corrrectos.

4) **Almacena el token de acceso resultante**

5) **Realiza llamadas a la API**

6) **Cierre sesion**

3.2.2) Descripcion tecnica login LinkedIn

Para la red social LinkedIn tambien nos vamos a surtir de la documentacion y informacion que tiene su pagina web para desarrolladores y su api, aunque está en este momento no se encuentra en español es muy facil y entendible. (<https://developer.linkedin.com/>)

Como hemos realizado en la anterior descripcion tecnica de Facebook en esta tambien nos vamos a surtir del JavaScript SDK de LinkedIn, para ello vamos a resumir en tres pasos en proceso para incluirlo en nuestra plataforma:

1. **Incializar SDK** .- Para utilizar cualquiera de la funcionalidad proporconada por el SDK de JavaScript, debe incluirse en la seccion <head> de nuestra pagina web (en nuestro caso Index)

2) **Crear botón “Inicio Sesion LikendIn”.-** Coloque el siguiente bloque <script> especial en su formato HTML:

javascript

<script type="in/Login"></script>

Esto generará un botón que se parece a la imagen de abajo. Tenga en cuenta que el texto en el botón se presenta en Inglés por defecto, pero se puede localizar a otros idiomas utilizando el argumento *lang*.

3) **Controlar la autenticación**

Cuando un usuario hace clic en el botón Iniciar sesión, se presentará una ventana emergente que contiene el diálogo de autenticación de LinkedIn. Una vez que el usuario ha aceptado la solicitud y han facilitado sus credenciales de cuenta de LinkedIn, será despedido de la ventana y el SDK realizará la llamada de back-end a LinkedIn para completar el proceso de autenticación en silencio para ti.

4. Gestión de código fuente

Para la gestión del código fuente hemos utilizado un repositorio de código, en este caso GitHub ha sido la plataforma utilizada para realizar esta tarea.

4.1. Roles en la gestión del código

En cuanto a los roles, el único miembro del grupo ha tenido que hacer la funciones de desarrollador del módulo principalmente aunque también a tomado los roles de:

●  **Gestor de incidencias**: es la persona encargada de abrir y cerrar las incidencias internas y cerrar las incidencias una vez se han corregido. Además, antes de cerrar debe velar de que la incidencia se ha resuelto correctamente.

●  **Gestor de la construcción**: es la persona encargada de administrar el archivo pom.xml encargado de la construcción automática del proyecto.

● **Gestor de código**: es la persona encargada de gestionar el repositorio donde esté el subsistema. Esto incluye: crear y eliminar ramas, crear tags y unificar ramas con la rama master.

4.2. Aprobar cambio

Cuando se ha realizado un cambio y se ha subido al repositorio, es el gestor de código quien repasa el código subido al repositorio. Si el código es correcto lo comunica a la persona que lo ha subido, si hay algún error, lo comunica al gestor de incidencias para que abra una nueva incidencia.

4.3. Aplicar un cambio

La persona que aplica un cambio en la parte del código que sea necesario es también quien ha realizado la codificación de esa parte del código..

4.4. Gestión de ramas

Para iniciar algún desarrollo en una nueva rama se debe crear una nueva. Una vez finalizada y probada la funcionalidad se une la rama con master.

5. Gestión de Construcción e Integración Continua

5.1. Integración continua en nuestro subsistema

En lo relativo a la integración continua del subsistema, se ha utilizado la herramienta Jenkins junto con Maven para automatizar la ejecución de los test del proyecto, de modo que si se realiza algun cambio que produzcan errores en los test, se detecte rápidamente, posibilitando una corrección rápida del código.

Con la herramienta Jenkins hemos creado dos tareas que se encargan de automatizar la integración:

1. **Comprobación de tests**: esta tarea se ha programado para ejecutarse automaticamente. Cada vez que se ejecuta, se limpia la zona de trabajo gracias al comando de Maven “clean” evitando así que se desvirtúen los resultados de los test debido a fallos en la actualización de los archivos. Después se conecta al repositorio de verificación descargando los archivos que contiene la rama “origin/master”. Además, cada minuto comprueba si se ha realizado algún cambio en el repositorio, si es así realiza la tarea de forma autónoma. Una vez tiene los archivos ejecuta los comandos de Maven “compile” y “test”. En caso de que la ejecución falle, se envía un correo ya que si se realiza un cambio erróneo debe ser notificado de este error.
2. **Generación de .jar**: esta tarea es de ejecución manual. Cada vez que se ejecuta, al igual que la anterior limpia la zona de trabajo con el comando “clean” de Maven. Después descarga el contenido del repositorio de la rama “origin/master”. Una vez descargado ejecuta el comando “package” de Maven y guarda el jar generado en la carpeta target/\*.jar. Si la ejecución falla, ocurre como en la tarea anterior, avisa del error por correo electronico.
3. **El proceso es el siguiente**: cada vez que se realice un nuevo aporte al repositorio en la rama master, Jenkins descargará la última versión subida al repositorio Git del subsistema y procederá a la ejecución de los tests unitarios que incluya el proyecto, usando para ello la herramienta Maven.

De este modo, cualquier cambio que impida que los test sean ejecutados correctamente, podrá ser detectado rápidamente, ya que Jenkins nos informaría de la situación.

6. Gestión del Cambio, Incidencias y Depuración

6.1. Gestión de cambio

(documentación de manera teórica ya que no he podido tener un grupo de trabajo)e manera teorica ya que no he podido tener un grupo de trabajo)sola persona)Script, debe incluirse en la seccion <head> de nues

Cuando un miembro del equipo quiere realizar un cambio en el código, contacta con el jefe de proyecto quien valora la petición y da su visto bueno o rechazo a la petición. En el caso de que esté de acuerdo con el cambio, evalúa su importancia y decide si se ejecuta inmediatamente o se puede posponer. Si por el contrario no cree conveniente el cambio, contacta con el miembro que quería hacer el cambio para llegar a un acuerdo sobre este y así poder rectificar o aclarar al miembro el motivo del cambio.

6.2. Gestión de incidencias

Hay que dividir las incidencias internas de las externas.

En cuanto a las incidencias internas, si un miembro detecta un error en el código crea un “issue” en el repositorio de GitHub con el siguiente formato:

-  Título: Resumen de la incidencia

-  Descripción:

* 1. ●  Descripción detallada.
  2. ●  Pasos para reproducir la incidencia.
  3. ●  Qué error da.
  4. ●  Otros comentarios.

Las incidencias externas sólo las pueden abrir otros grupos que se conectan con nuestro subsistema. Si encuentra algún error, crea un “issue” en nuestro repositorio de GitHub con el siguiente formato:

* -  Título: Resumen de la incidencia
* -  Descripción:
  + ●  Nombre del subsistema que reporta la incidencia.
  + ●  Descripción detallada.
  + ●  Pasos para reproducir la incidencia.
  + ●  Qué error da.

● Otros comentarios.

Cuando se han resuelto las incidencias, el gestor de incidencias tras revisar el proceso, cierra la incidencia en cuestión.

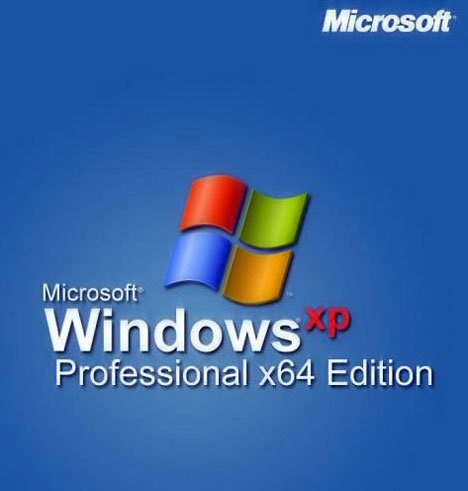
6.3. Gestión de depuración

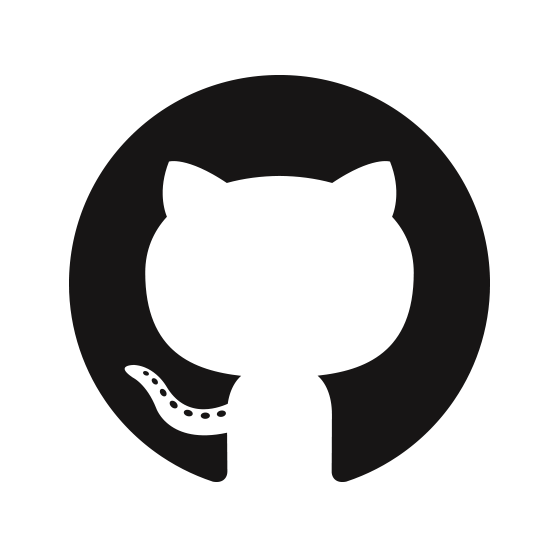
A partir de una incidencia abierta, es el encargado de la programación de esa parte del código el encargado de su corrección por su mejor conocimiento del código. Una vez se ha asegurado de que el código funciona bien, lo sube al repositorio y avisa al resto de miembros de que ha solucionado el error.

7.Mapa de Herramientas

Para el desarrollo del trabajo, han sido necesarias las siguientes herramientas y tecnologías:

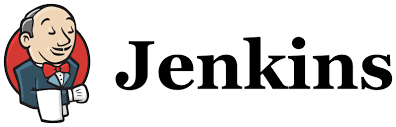
* Virtual Box: es un gestor de maquinas virtuales el cual tiene gran reputación y es fácil su uso y manejo, se tomo la decisión de usar una maquina virtual para intentar conseguir la máxima homogeneidad entre todos los grupos y conseguir que sea mas fácil la integración de los mismos. Dentro de esta maquina virtual usaremos Window Xp 64 bits con el cual se está mas familiarizado en la escuela.



* GitHub: plataforma de desarrollo escogida donde se creó el repositorio para alojar el proyecto. Fue escogido debido a que parecía el más adecuado por su facilidad de uso, y porque AGORA@US también estaba alojada en la plataforma GitHub.
* Git: software de control de versiones seleccionado para gestionar el código. Debido a su fácil integración en todo tipo de entornos, sin necesidad de entornos de desarrollo, y a que el equipo de desarrollo estaba familiarizado con el mismo, pareció la decisión más lógica.



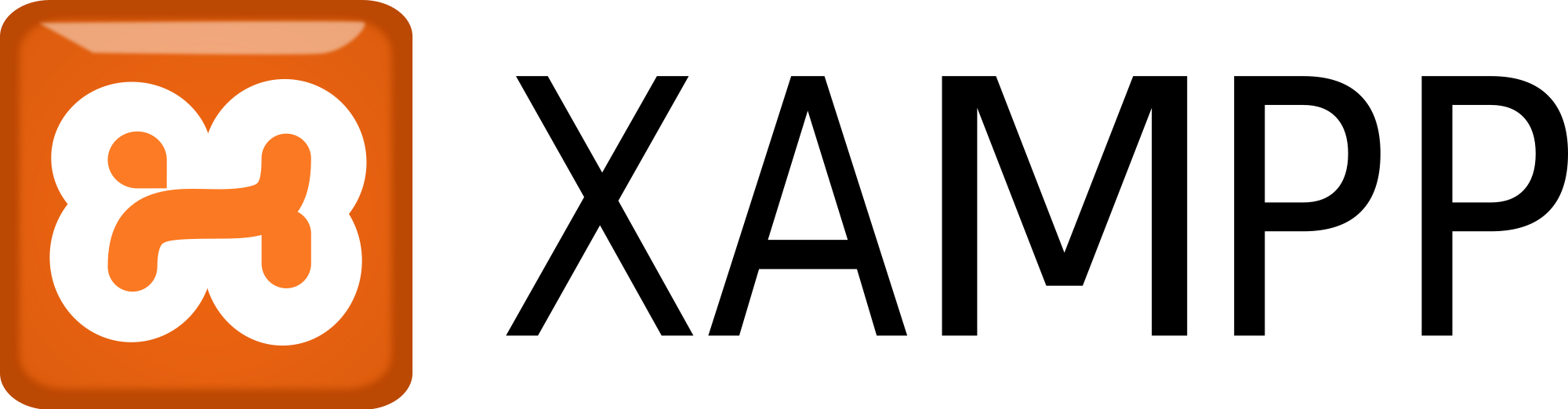
* Jenkins: servidor de integración continua elegido para llevar la tarea de construcción e integración automática. El equipo es la herramienta con la que estaba más familiarizado para este tipo de tareas.



* Aptana: entorno de desarrollo integrado de software libre basado en eclipse y provee soporte para lenguajes que utilizaremos en el proyecto. El equipo está más familiarizado con este entorno y por ello su elección.



* FileZilla: es un [cliente FTP](https://es.wikipedia.org/wiki/Cliente_FTP) [multiplataforma](https://es.wikipedia.org/wiki/Multiplataforma) de [código abierto](https://es.wikipedia.org/wiki/C%C3%B3digo_abierto) de fácil manejo .
* XAMPP: es un sevidor independiente que consiste principalemnete en el sistema de gestión de bases de datos (MySQL), servidor Apache y interpretes PHP



* Google Drive: herramienta desarrollada por google para tener , editar y crear archivos en la nube.



* Trello: herramienta para la organización y sincronización del equipo usada por su facilidad y por lo fácil que es ver lo que le queda al proyecto con un solo vistazo (metodologías agiles).

