



MEJORA DE ACCESIBILIDAD

CABINA DE AUTENTICACIÓN

GRUPO 2 EGC CURSO 16-17

TABLA DE CONTENIDOS

Contenido

Documentación del proyecto	1
Equipo de desarrollo	1
Roles	1
Resumen	1
Introducción	2
Cambios desarrollados	4
Planificación del trabajo	5
Comunicación	6
Comunicación externa	6
Comunicación interna	6
Entorno de desarrollo	7
Elementos de control	8
Gestión de las incidencias	9
incidencias internas	9
incidencias externas	10
Gestión del código	11
Gestión de ramas	11
Aprobación de cambios	11
Gestión de la construcción	12
Gestión de los entregables	16
Conclusiones	17
Enlaces de importancia	18

AGORA VOTING

DOCUMENTACIÓN DEL PROYECTO

Equipo de desarrollo

Aurora Gómez Medina aurgommed@alum.us.es

Eduardo Ger Rodríguez edugerrod@alum.us.es

Jorge Ramos Rivas jorramriv@alum.us.es

José Luis Maya Quirós josmayqui@alum.us.es

Sara Chávez García sarchagar1@alum.us.es

Roles

Eduardo Ger Rodríguez (Coordinador/Desarrollador)

Sara Chávez García (Subcoordinador/Desarrollador)

Aurora Gómez Medina (Desarrollador)

José Luis Maya Quirós (Desarrollador)

Jorge Ramos Rivas (Desarrollador)

Resumen

Nuestro objetivo se centrará en la parte de autenticación y registro de la aplicación AgoraVoting. Para ello, nuestro principal desarrollo será, previsiblemente, en el repositorio de agora-gui-common, el cual está desarrollado principalmente en Javascript con AngularJS.

Usaremos ARIA, la cual define maneras de hacer el contenido Web y las aplicaciones Web (especialmente las desarrolladas con Ajax y JavaScript) para que sean más accesibles para las personas con discapacidad.

Esta solución pensamos que será de gran ayuda para personas invidentes, ya que cada campo podrá ser interpretado por la máquina mediante audio.

Introducción

El proyecto sobre el que trabajamos es AgoraVoting, un sistema de votación electrónica por Internet para votar de manera segura y confidencial. Es un sistema de software libre. Sobre este sistema se realizarán las mejoras pertinentes usando ARIA.

Según la página de Mozilla, esta es la definición de ARIA:

*“Accessible Rich Internet Applications (**ARIA**) define cómo realizar contenido Web y aplicaciones Web (especialmente las desarrolladas con Ajax y JavaScript) más accesibles a personas con discapacidades. Por ejemplo, ARIA posibilita puntos de navegación accesibles, widgets JavaScript, sugerencias en formularios y mensajes de error, actualizaciones en directo, y más.*

ARIA es un conjunto de atributos especiales para accesibilidad que pueden añadirse a cualquier etiqueta, pero especialmente adaptado a HTML. El atributo role define el tipo general del objeto (como un artículo, una alerta o un deslizador). Atributos adicionales de ARIA proporcionan otras propiedades útiles, como una descripción para un formulario o el valor actual de una barra de progreso.

ARIA está implementado en la mayoría de los navegadores y lectores de pantalla más populares. De todas formas, las implementaciones varían y las tecnologías antiguas no lo soportan bien (si es que lo soportan). Use "safe" en ARIA, que reduce elegantemente la funcionalidad en caso necesario o pida a los usuarios actualizarse a una tecnología más actual.”

Fuente: <https://developer.mozilla.org/es/docs/Web/Accessibility/ARIA>

Descripción del sistema

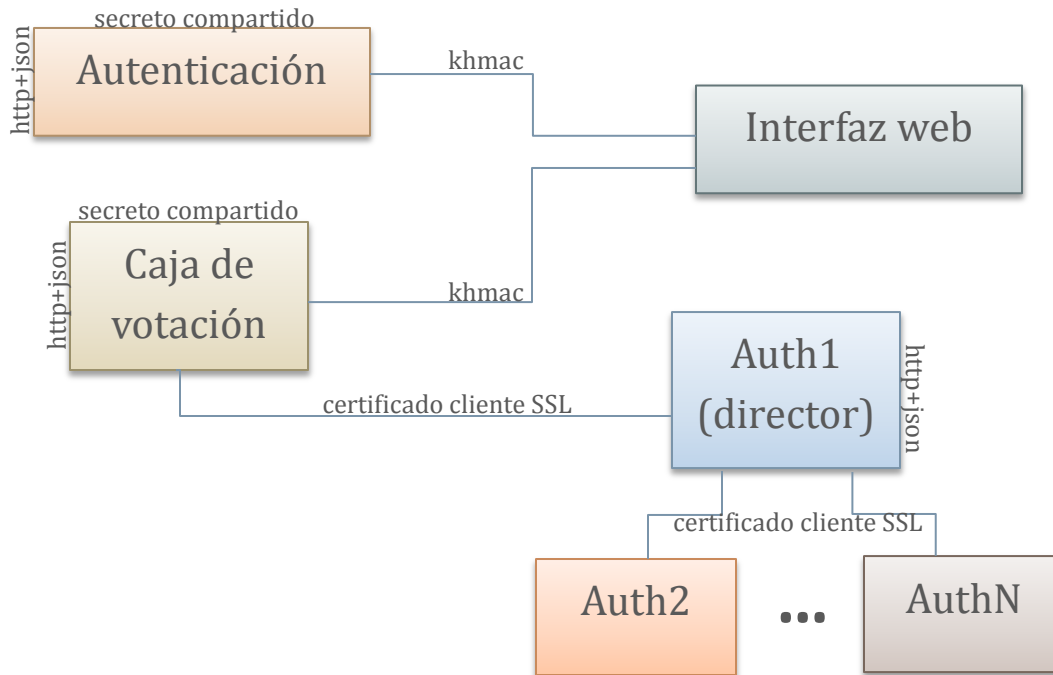
El sistema de AgoraVoting funciona de la siguiente manera:

“

- *El voto se manda cifrado desde el cliente, no pasa por ningún servidor, por lo que todo este proceso ocurre en el ordenador del votante.*
- *El voto ya cifrado entra en una caja A1 que sólo puede abrir la autoridad A1. Esta caja entra cifrada en otra caja A2 que sólo puede abrir la autoridad A2.*
- *El votante se queda con un hash de su voto, que servirá para verificar que su voto se ha tenido en cuenta y que no ha sido alterado. El hash lo genera el propio cliente, por lo que es totalmente confiable.*
- *La autoridad A2 descifra las cajas A2, con lo que obtiene las cajas A1. La autoridad baraja el resultado antes de pasárselo a la siguiente autoridad.*
- *La autoridad A1 descifra las cajas A1, y obtiene el voto en claro. La autoridad baraja el resultado antes de darlo.*
- *Las autoridades publican las pruebas de que el proceso lo han realizado correctamente para que cualquiera pueda comprobar que no ha habido ninguna alteración en la votación.*
- *El votante verifica con su hash que su voto se ha tenido en cuenta y no ha sido alterado.”*

AGORA VOTING

Arquitectura de AgoraVoting:



Fuentes:

<https://1984.lsi.us.es/wiki-egc/images/c/c3/00-Agora1516.pdf>

Eduardo Robles, <https://agoravoting.com>

CAMBIOS DESARROLLADOS

Los cambios que hemos realizado en la aplicación son los siguientes:

- Con ARIA, hemos modificado las etiquetas correspondientes a los campos de autenticación, es decir, los campos de email y de contraseña, de manera que puedan ser leídos en voz alta por la máquina cuando se pone el cursor en ellos, para que sea más accesible para las personas invidentes.

AGORA VOTING

PLANIFICACIÓN DEL TRABAJO

A continuación se enunciarán las tareas realizadas y el miembro del equipo que ha llevado a cabo cada tarea.

Miembro del equipo	Tareas realizadas
Aurora Gómez Medina	Preparación GitHub
	Despliegue del entorno de desarrollo
	Implementar ARIA
	Configurar y automatizar construcción
	Reuniones
	Clases de revisión
	Reuniones con AgoraVoting
	Documentación
Eduardo Ger Rodríguez	Preparación entregables
	Preparación GitHub
	Despliegue del entorno de desarrollo
	Implementar ARIA
	Configurar y automatizar construcción
	Reuniones
	Clases de revisión
	Reuniones con AgoraVoting
Jorge Ramos Rivas	Documentación
	Preparación entregables
	Preparación GitHub
	Despliegue del entorno de desarrollo
	Implementar ARIA
	Configurar y automatizar construcción
	Reuniones
	Clases de revisión
Jorge Ramos Rivas	Reuniones con AgoraVoting

AGORA VOTING

José Luis Maya Quirós

Documentación
Preparación entregables
Preparación GitHub
Despliegue del entorno de desarrollo
Implementar ARIA
Configurar y automatizar construcción
Despliegue online
Reuniones
Clases de revisión
Preparación entregables

Sara Chávez García

Preparación GitHub
Despliegue del entorno de desarrollo
Implementar ARIA
Configurar y automatizar construcción
Reuniones
Clases de revisión
Reuniones con AgoraVoting
Documentación
Preparación entregables

Comunicación

COMUNICACIÓN EXTERNA

Las reuniones con el equipo de Ágora Voting han tenido lugar usando el editor de textos colaborativo TitanPad: <https://titanpad.com/KxlHuhjhcZ>

Y mediante el uso de Mumble, gestor de salas de conferencias online.

COMUNICACIÓN INTERNA

Usaremos WhatsApp como herramienta principal y Teamviewer, además de las reuniones presenciales.

Entorno de desarrollo

En nuestro entorno de desarrollo estamos usando una máquina virtual basada en Ubuntu, versión 16.04 LTS, que es gestionada por el gestor de máquinas virtuales VMware. La máquina virtual ha sido creada en VMware a partir de una imagen del sistema operativo mencionado en formato .iso, el cual hemos descargado desde aquí: <https://atlas.hashicorp.com/boxes/search>.

Con la máquina creada, se ha procedido a desplegar la aplicación AgoraVoting, lo cual es necesario para realizar los cambios. Para ello, hemos seguido los pasos que indica el equipo de AgoraVoting aquí: https://github.com/agoravoting/agora-dev-box/blob/next/doc/devel_deployment.md, aunque con versiones diferentes que las indicadas por ellos. Concretamente, hemos usado las siguientes versiones de cada componente:

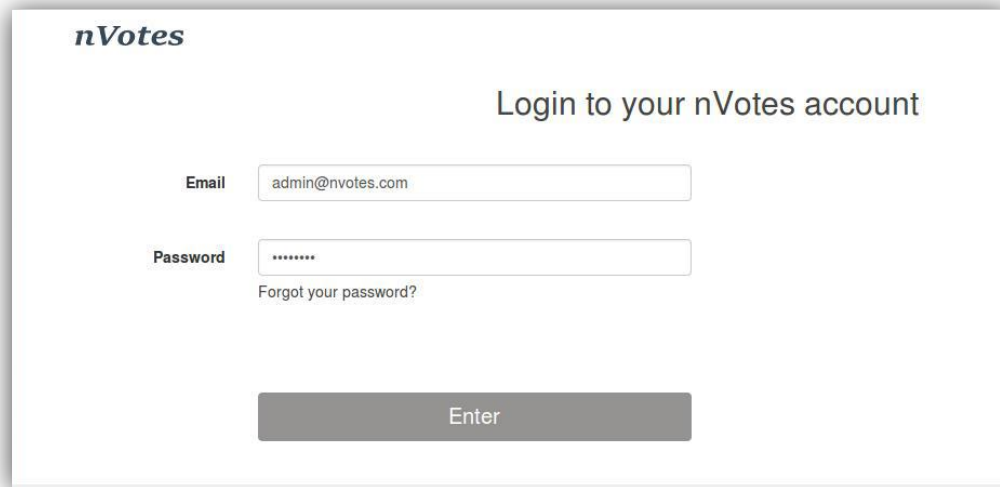
- Oracle VirtualBox v5.1.10
- Vagrant v1.9.0
- Ansible v2.1.3.0

Después de seguir el proceso indicado anteriormente, dentro de VirtualBox se crean tres máquinas virtuales Ubuntu 14.03 que actúan como servidores, necesarias para el despliegue de la aplicación. El resultado es entonces una máquina virtual Ubuntu en VMware, y dentro de ésta tres máquinas virtuales más (auth1, auth2 y agora) de Ubuntu en VirtualBox.

Una vez que hemos hecho los pasos que se indican en la página mencionada, ya tenemos todo lo necesario para desplegar el sistema. Para hacerlo, arrancamos la máquina de agora, que es la única que hace falta para nuestros propósitos, con el comando `vagrant up`.

El aspecto una vez puesta en marcha debe ser el siguiente:

AGORA VOTING



Las credenciales para entrar son: el email admin@nvotes.com y la contraseña QWERTY33.

Elementos de control

En nuestro proyecto, los elementos que están bajo la gestión de la configuración son los siguientes:

- Repositorios de GitHub, gestionados mediante el propio sistema de versionado de Git.
- Documento del proyecto, realizado con Microsoft Word y gestionado con Git y Google Drive.
- Código fuente de la aplicación, gestionado por Git y realizado a través de nuestra máquina virtual, de manera que se van actualizando los repositorios de GitHub.
- Diario de grupo, realizado en Microsoft Excel y gestionado con Google Drive.
- Wiki de la página wiki de la asignatura, gestionada a través de la misma.
- Nuestros propios equipos, a los que les hemos instalado GitHub para el control de versiones con Git, y el gestor de máquinas virtuales VMware, para utilizar nuestra máquina virtual.

Gestión de las incidencias

Para la gestión de incidencias nuestro grupo ha decidido gestionarlas a través de la creación de issues con GitHub.

INCIDENCIAS INTERNAS

Estas incidencias corresponden a las incidencias del propio grupo. A cada incidencia se le asignará una prioridad, que podrá ser alta, neutra o baja. El proceso para la creación de una incidencia es el siguiente:

- Un miembro del grupo crea una incidencia con el estado inicial "Pendiente". Dicho miembro debe subir adjunto el archivo más adecuado para describir la incidencia. Podrán usarse imágenes, texto, enlaces a código GitHub, etc.
- El miembro que ha creado la incidencia comunica al grupo mediante Whatsapp que acaba de crear una incidencia.
- Otro miembro del grupo, cuando ha sido notificado de la incidencia, cambia el estado de la misma a "En proceso" y se pone a trabajar sobre ella.
- Una vez que el miembro ha resuelto los problemas que definían la incidencia, cambia de nuevo su estado a "Resuelta".
- Antes de cambiar finalmente el estado a "Cerrada", al menos un miembro del grupo debe verificar que la incidencia se ha resuelto correctamente.
- Eventualmente, si una incidencia ha pasado a "Cerrada" y aún quedan problemas de esa incidencia por solventar, podrá pasar a un estado intermedio "Reabierto" para luego volver a pasar al estado "Pendiente".
- Una incidencia que esté en estado "Pendiente" puede pasar directamente al estado "Cerrada", bien porque se haya abierto erróneamente, bien porque el problema que ha ocasionado la incidencia no sea suficientemente grave como para crear una incidencia, o bien porque el problema es imposible de resolver.
- De igual manera, una incidencia que está "En proceso" puede pasar al estado "Cerrada" también directamente, por ejemplo porque una vez que alguien del grupo

AGORA VOTING

se ha puesto a trabajar sobre esa incidencia, no ha podido conseguir solucionar los problemas de esa incidencia.

INCIDENCIAS EXTERNAS

Las incidencias externas son las incidencias que se generan con el equipo de AgoraVoting. Hemos acordado con otro grupo abrir una única incidencia a AgoraVoting, ya que nuestra incidencia era la misma, para que ellos no tuvieran que gestionar varias veces la misma incidencia.

El motivo de dicha incidencia, que ha sido la única incidencia externa, ha sido la imposibilidad de desplegar la aplicación siguiendo los pasos descritos por el grupo de AgoraVoting. La causa fue porque AgoraVoting tiene muchas dependencias de distintas aplicaciones de terceros, y no han tenido en cuenta que esas aplicaciones se van actualizando. Uno de estos componentes, Celery, se actualizó a la versión 4 el día 4 de noviembre de 2016, versión que no es compatible con la versión que se instalaba de Django, por tanto, al aprovisionar la máquina servidora de Agora dentro de nuestra máquina virtual de Ubuntu, se estaban instalando componentes que eran incompatibles entre sí. Esta incidencia se solucionó ya que el equipo de Agora actualizó las versiones de los componentes, de manera que ya pudimos desplegar la aplicación con normalidad.

Gestión del código

GESTIÓN DE RAMAS

Se ha realizado un fork de cada uno de los repositorios de AgoraVoting en GitHub a nuestra organización. Cada miembro del grupo debe hacer un git clone de los repositorios a su máquina, de manera que cada uno trabaja en su máquina en local, en una rama que se ha creado llamada development, haciendo commit cada vez que sea necesario. Todos los miembros del grupo trabajaremos sobre esa rama.

APROBACIÓN DE CAMBIOS

Si el miembro logra obtener cambios estables, notificará al grupo mediante Whatsapp que va a realizar un git push, pero antes de hacerlo, al menos un miembro del grupo debe revisar que el cambio funciona correctamente. Una vez que se ha hecho git push con el cambio, hay que volver a desplegar AgoraVoting con dicho cambio. Cada vez que se haga un push, se deben documentar y describir los nuevos cambios. Estos documentos generados de los cambios se subirán tanto a la wiki del grupo como a nuestro Google Drive.

Gestión de la construcción

Para la gestión de la construcción se usan combinadamente la herramienta Bower como gestor de dependencias y Grunt para automatizar la generación y despliegue de la aplicación.

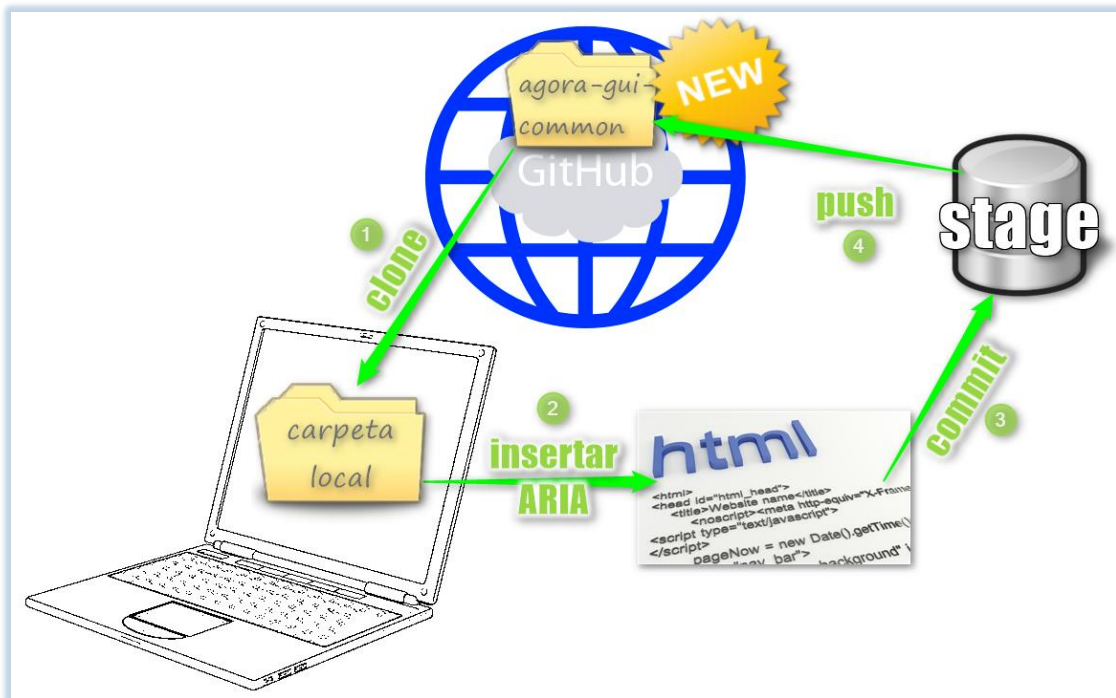
Dado que Bower obtiene los archivos de código desde GitHub, una de las problemáticas encontradas es que el flujo de trabajo se complica ligeramente, ya que para poder desplegar modificaciones deben haberse subido primero a GitHub. Esto implica que a veces puedes subir código no comprobado a GitHub para posteriormente, una vez desplegado y pudiendo comprobar el comportamiento, encontrar un error y tener que acometer el proceso completo de nuevo.

Básicamente el proceso completo se centra en el script de construcción “build.sh” contenido dentro del sistema de la máquina virtual de Agora. Este script se encarga de gestionar las dependencias usando Bower y efectuar la construcción con Grunt, además coloca todos los archivos finales en la ruta correcta lista para ser consultada externamente desde el navegador.

Previo a la ejecución del constructor, hay que configurar Bower en cada uno de los subproyectos de interfaz de usuario (agora-gui-admin, agora-gui-booth y agora-gui-elections) para especificar que obtengan el módulo común (agora-gui-common) desde nuestro repositorio remoto, que es donde están los cambios y modificaciones con respecto al sistema original. Tras ello, se ejecuta el script de construcción “build.sh” que irá llamando a cada uno de esos subproyectos y ejecutando Bower (que ya apunta a nuestro repositorio de código) para gestionar y descargar las dependencias, incluyendo el contenido de nuestro repositorio remoto, y Grunt para construir la estructura de la web.

A continuación se muestra un esquema del proceso a seguir, esperando conseguir una mayor comprensión del flujo de trabajo en la construcción:

AGORA VOTING



1. Se debe tener el repositorio agora-gui- common del repositorio remoto en local, en principio mediante un “git clone”.
2. Una vez con el sistema de interfaz común en local, acceder a los archivos de código fuente e insertar los cambios deseados. En nuestro caso inyectar etiquetas ARIA en los distintos elementos web para mejorar la accesibilidad.
3. Realizar el commit de los cambios a la zona intermedia.
4. Subir los cambios a nuestro GitHub remoto mediante un “git push”.

AGORA VOTING



5. Dentro de cada repositorio de interfaz de la máquina virtual de AgoraVoting, editar el archivo “bower,json” y especificar la dependencia de la variable “avCommon” hacia nuestro repositorio.
6. Una vez todos los repositorios de interfaz tienen referenciado nuestro repositorio común con los cambios aplicados, referenciar correctamente los subproyectos de interfaz de usuario en el script de construcción “build.sh” y las rutas donde debe acabar el sistema construido.
7. Una vez configurados los proyectos y el script constructor, ejecutar este último. Comenzará a gestionar las dependencias y a realizar la construcción del sistema en las rutas especificadas en el paso anterior.
8. Si todo ha ido bien, deberías poder acceder al servidor web para comprobar que se ha construido la web con los cambios efectuados.

Si se desea implementar otro cambio, se debe comenzar el proceso desde el paso 2, o sea, omitiendo el paso de volver a clonar el repositorio remoto.

AGORA VOTING

Como ventaja principal, al usar Bower y Grunt, una vez realizada una modificación en el sistema se puede construir y hacer un despliegue con bastante facilidad, tan solo ejecutando el script de construcción.

Entre los problemas encontrados, Bower no se puede ejecutar como superusuario, pero sin embargo a veces requiere la creación de alguna nueva carpeta en el sistema, por lo que tras varios problemas debido a ello, se optó por añadir al sistema una instrucción en la configuración de Bower que permitiera la ejecución como superusuario, y finalmente efectuar la construcción siempre como superusuario.

Otro de los principales inconvenientes fue la localización exacta de los archivos de código fuente, ya que dentro del tremendo enmarañado de la estructura heredada de AgoraVoting, no existe una documentación que concrete la estructura o arquitectura específica, y esto complica en gran medida la implementación de nuevo código o modificaciones ya que se pierde mucho tiempo en localizar el archivo específico que debe ser modificado.

Fuente de imágenes:

-Icono 'Directorio': MiRincon-Misaani.blogspot.com.es

-Icono 'Archivo de texto': Iconfinder.com

-Icono 'Base de datos': Iconfinder.com

-Icono 'Portátil': Dibujos.net

-Imagen 'Código HTML': Ciudadano2cero.com

Gestión de los entregables

Los entregables del proyecto son los milestone propuestos por el profesorado de la asignatura. Dichos milestone son los siguientes, para el grupo al que pertenecemos:

- *M1: Ecosistemas preparado*
 - *V 11 Nov*
- *M2: Sistema funcionando con incremento*
 - *V 25 Nov*
- *M3: Taller de automatización*
 - *V 16 Dic*
- *M4: Entrega y defensa de trabajos*
 - *X 11 Ene / V 13 Ene*
- *M5: Mejoras*
 - *2 Feb en horario oficial*

Fuente:

https://1984.lsi.us.es/wiki-egc/index.php/Trabajo_-_16/17

La entrega final consistirá en un solo entregable en formato .rar, que contendrá la máquina virtual utilizada para desplegar el proyecto y para realizar las mejoras. La entrega se hará a través de la plataforma Opera: <http://opera.eii.us.es/>

El entregable también contendrá el diario de grupo, que consiste en un informe de las horas que cada miembro del equipo ha invertido en cada tarea del proyecto, y el presente documento del proyecto, que se trata de un documento de texto.

Conclusiones

Referente a la planificación inicial del proyecto cometimos los errores de ser muy optimistas, no mirar el calendario y confiar en que no tendríamos interrupciones de otras materias ni problemas externos a nuestra organización, tales como no poder desplegar la aplicación.

Si hubiéramos previsto todo eso, que lo podríamos haber hecho porque lo teníamos en el calendario escolar, en el calendario de cada asignatura y en el calendario de exámenes, seguramente nos hubiéramos ajustado más a la planificación.

Como resumen señalamos los siguientes puntos importantes:

1. Hacer una planificación realista, y tener en cuenta que siempre se presentarán imprevistos.
2. Prometer menos, dar más.
3. Algo que hemos hecho bien es la comunicación que hemos tenido los miembros del grupo y las reuniones semanales para ver qué habíamos hecho y qué nos quedaba por hacer.

Durante el desarrollo del trabajo hemos podido ir desgranando poco a poco Agora Voting, gracias a ello hemos conocido su complejo funcionamiento, la dificultad de acceder y modificar un código heredado y sobre todo hemos perdido un tiempo valioso en aprender a utilizar un entorno como es Ubuntu, al que no estamos acostumbrados a usar, y acceder a código a través de máquinas virtuales.

Unas de las mejoras que nos hubiera gustado hacer y la dejamos pendiente para el futuro implementar un teclado virtual que sea accesible mediante el ratón.

En definitiva, para nosotros ha sido un todo un reto enfrentarnos a una aplicación real y poder haber realizado pequeñas modificaciones.

Enlaces de importancia

Organización nuestra en GitHub (con los repositorios de AgoraVoting forkeados):

<https://github.com/EGCAgoraVotingAccG2>

Organización de AgoraVoting en GitHub:

<https://github.com/agoravoting>

URL de acceso a servidor de aplicación:

<http://egcagoravotingaccg2.hopto.org/>

Wiki en la página de la asignatura:

[https://1984.lsi.us.es/wiki-egc/index.php/Agora Voting - Mejora de la accesibilidad2 - Autenticaci%C3%B3n](https://1984.lsi.us.es/wiki-egc/index.php/Agora_Voting_-_Mejora_de_la_accesibilidad2_-_Autenticaci%C3%B3n)