# Introducción

En esta sección se va a proceder lo que se ha realizado en el proyecto desde que se heredó el código hasta que se ha finalizado el proyecto de forma escueta.

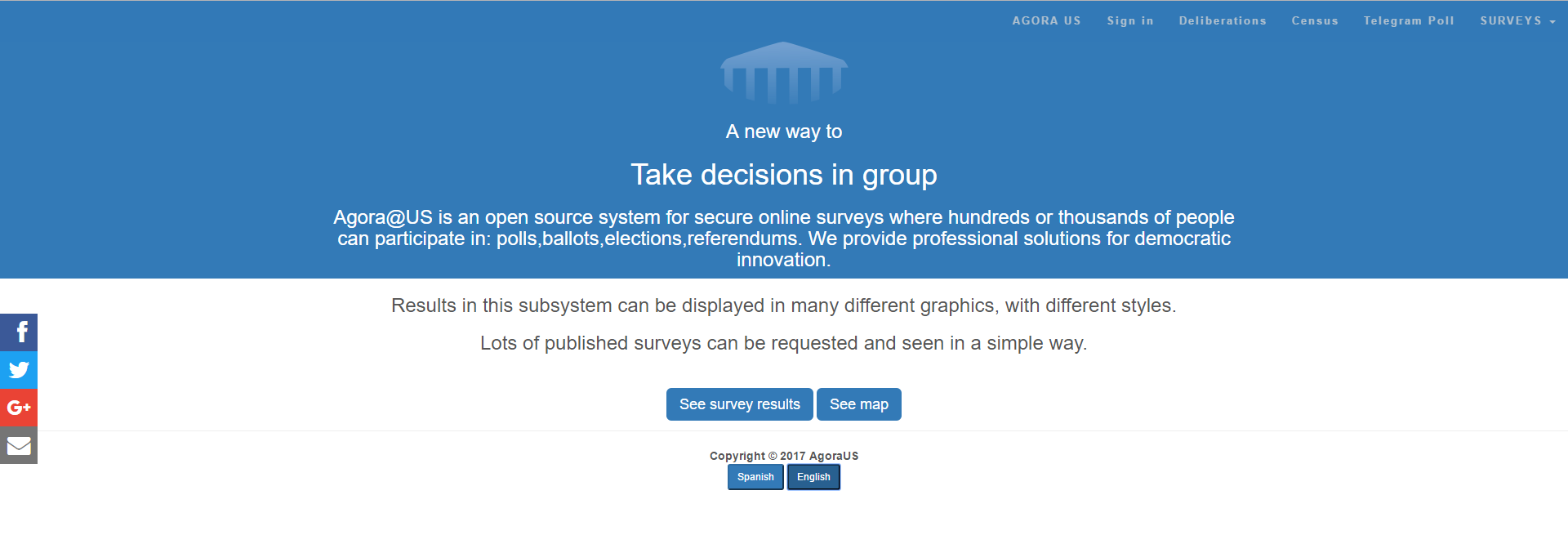
Agora US es un sistema de votaciones online creado por alumnos de la escuela de informática de la Universidad de Sevilla. Este proyecto está dividido en numerosos módulos, una de las cuales es Frontend y visualización de resultados, de la cual se encarga de mejorar este grupo.



Cuando se heredó dicho módulo, el código estaba compuesto por la parte de lógica, hecha en Spring y la parte de vistas, hecha en Angular. Lo primero que hizo nuestro grupo con respecto a lo heredado fue pasar toda la parte de Spring a NodeJS, y hacerla compatible con la parte de Angular que habíamos heredado.

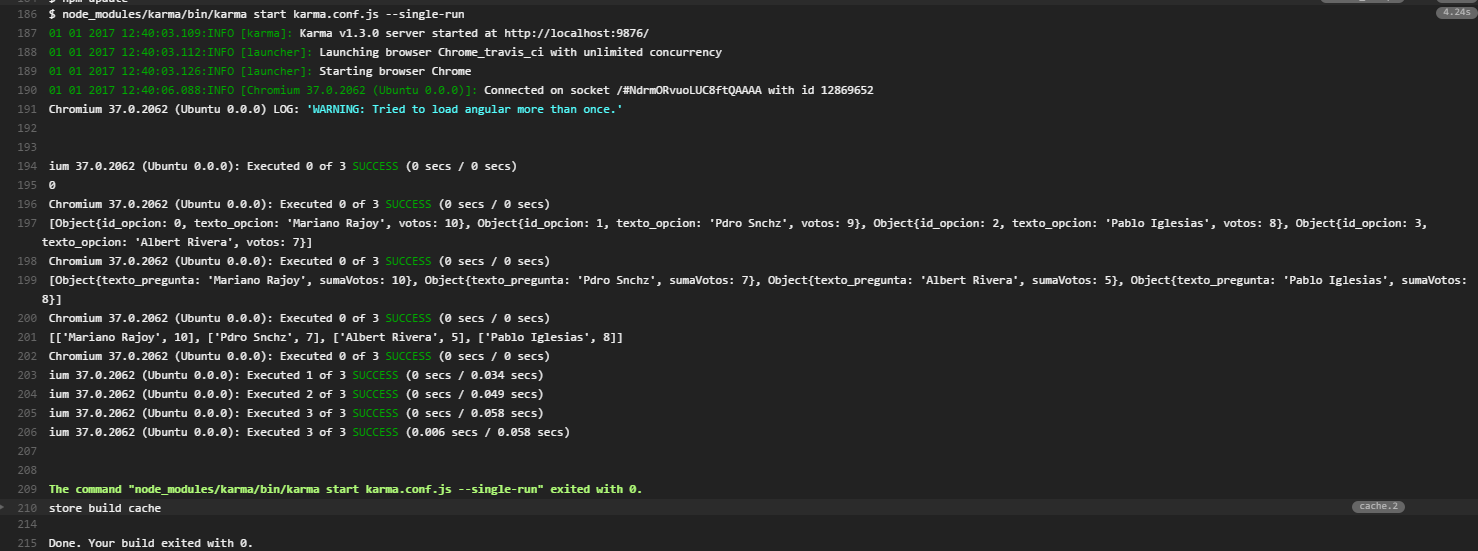
Además de eso, realizamos la integración con el módulo de recuento, el cual nos provee de los datos que nuestro módulo muestra y con el cual hemos estado en constante comunicación para que la integración se llevase a cabo con éxito. Esta integración se basa en llamadas a la API del módulo de recuento, tras la cual recibe un JSON con la información requerida.

También se ha añadido soporte a añadir idiomas al sistema (de momento español e inglés) mediante Angular Translate, funcionalidad que podía ser aportada con Spring (aunque no había sido llevada a cabo por el grupo del que heredamos el proyecto).



Por último, se han añadido tests al sistema para probar su buen funcionamiento, utilizando Karma y Jasmine, los cuales son entornos que facilitan la realización y ejecución de los tests.

Además de todo lo añadido al módulo en sí, también se ha logrado la integración continua del mismo utilizando Jenkins para tener el módulo siempre actualizado en la versión desplegada, y utilizando Travis para comprobar que se pasan los tests y el sistema sigue funcionando.



# 6 Entorno de Desarrollo

A continuación, se van a detallar las herramientas que se han utilizado para desarrollar en el grupo de Visualización y Frontend:

## Máquina virtual

Como programa de virtualización hemos utilizado Virtual Box, ya que es uno de los más conocidos, utilizado previamente por el grupo y bastante fiable. El hecho de utilizar una máquina virtual ayuda en la instalación del entorno tanto a nuestro grupo, que trabajará con entornos exactamente iguales, como al grupo que utilice nuestro entregable como heredado, ya que no necesitará investigar en exceso como instalar todos los componentes. El equipo ha trabajado con diferentes versiones de Virtual Box al tenerlo instalado de asignaturas anteriores sin problemas.

<https://www.virtualbox.org/wiki/Downloads>



En cuanto a la máquina virtual en sí, se ha utilizado Ubuntu en su versión 16.04 LTS, a la cual se le otorgaron 2 procesadores y 2GB de RAM, con lo que, en principio, se ha obtenido un rendimiento bastante aceptable, además de ser una configuración asequible para prácticamente cualquier equipo.

<https://www.ubuntu.com/download/desktop>

## NodeJS

Para NodeJS hemos utilizado la versión 6.9.2 LTS, ya que es la última versión estable de Node. Para instalar paquetes de Node se ha utilizado npm, el cual es un instalador bastante sencillo de utilizar. Algunos paquetes utilizados son nodemon (para que el módulo se ejecute en fase de pruebas directamente después de realizar cambios sin reiniciar a mano el servidor), mongoose (para modelar los datos almacenados en mongoDB en el servidor de pruebas), Jasmine y Karma para los tests y Angular para las vistas.



## Atom

Hemos utilizado Atom como editor para desarrollar el módulo. Es un editor bastante simple y a la vez muy personalizable gracias a la cantidad de plugins que se pueden añadir. Además, como está desarrollado por Github cuenta con funciones muy interesantes, como un indicador de las líneas que han sido editadas, o marcas con colores de los ficheros que han sido modificados o añadidos con colores, de forma que se pueda ver fácilmente.

<https://nodejs.org/en/download/>

## Git

Como hemos utilizado GitHub para compartir el código, es necesario el uso de Git para mantener el control de versiones y el código compartido entre todos los miembros del grupo, además de tener siempre el sistema desplegado lo más actualizado posible.

<https://git-scm.com/downloads>