Decide-single-salares-1

Miembro	Implicación
García Quijada, José Mª	10
Gaviro Martínez, Miguel	7
Gravan Bru, Victor	0
Moreno Domínguez, Eloy	10
Stefan, Bogdan Marian	1
Zuleta de Reales Toro, Santiago	4

- Grupo 2
- Curso escolar:2022/2023
- Asignatura: Evolución y gestión de la configuración

Enlace de interés:
https://github.com/decide-single-salares-1/decide-single-salares-1.git

Documento del proyecto

Indicadores del proyecto

Miembro del equipo	Horas	Commits	LoC	Test	Issues	Incremento
GARCÍA QUIJADA, JOSÉ MARÍA	66h 23 min	26	497	6	9	3
GAVIRO MARTÍNEZ, MIGUEL	50h 49 min	20	1254	6	8	1.5
GRAVAN BRU, VÍCTOR	-	-	-	-	-	-
MORENO DOMÍNGUEZ, ELOY	61h 42 min	40	729	6	12	2
STEFAN, BOGDAN MARIAN	-	-	-	-	-	-
ZULETA DE REALES TORO, SANTIAGO	-	-	-	-	-	-

Esta tabla es un breve resumen, en la parte inferior se detalla con profundidad estos datos.

Integración con otros equipos

En nuestro caso no aplica ya que nuestro proyecto es de tipo "*single"*, sin integración externa con otros equipos.

Resumen ejecutivo

Debido a la necesidad de mostrar el trabajo realizado y el esfuerzo en horas del equipo durante el progreso de esta entrega, se desarrolla el presente documento.

Para ello se sintetizará todo aquello que se ha realizado durante el desarrollo de la aplicación *Decide* en diferentes apartados de este documento para así tener claras las ideas que este contiene, siendo estos; descripción del sistema, visión global del proceso de desarrollo, entorno de desarrollo, ejercicio de propuesta de cambio y conclusiones y trabajo a futuro.

Para finalizar podemos afirmar que, tras crear este documento, hemos sido capaces de poner en práctica un reparto equitativo de tareas al verlas reflejadas en carga de trabajo.

Descripción del sistema

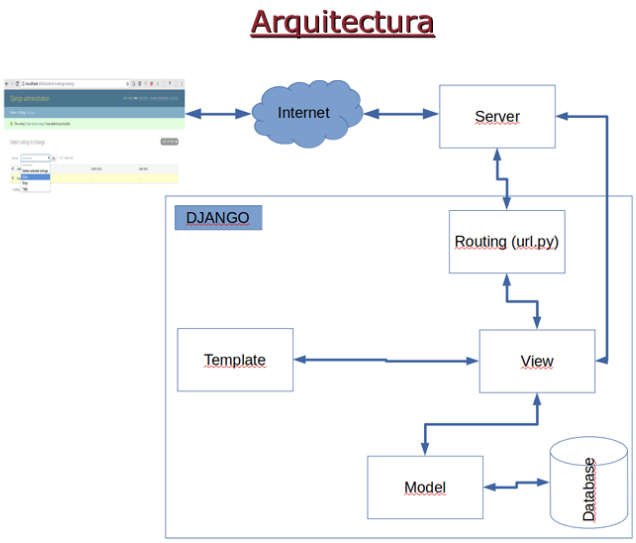
En primer lugar, comenzaremos hablando y poniendo en contexto sobre qué es Decide, cómo funciona y cuál es su objetivo principal.
Decide es un proyecto educativo, pensado para el estudio de sistemas de votación, por lo que prima la simplicidad por encima de la eficiencia cuando sea posible. Por lo tanto, se asumen algunas carencias para permitir que sea entendible y extensible.

Puesto que el objetivo principal es que este sea extensible, nuestro trabajo con esta aplicación consiste en ampliarla con diferentes subsistemas disponibles. Entre ellos, destacan los siguientes; autenticación, censo, votaciones, cabina de votación, almacenamiento de votos cifrado, recuento, post procesado o visualización de datos.

Así pues, el objetivo de este proyecto es implementar una plataforma de voto electrónico seguro, que cumpla una serie de garantías básicas, como la anonimidad y el secreto del voto.

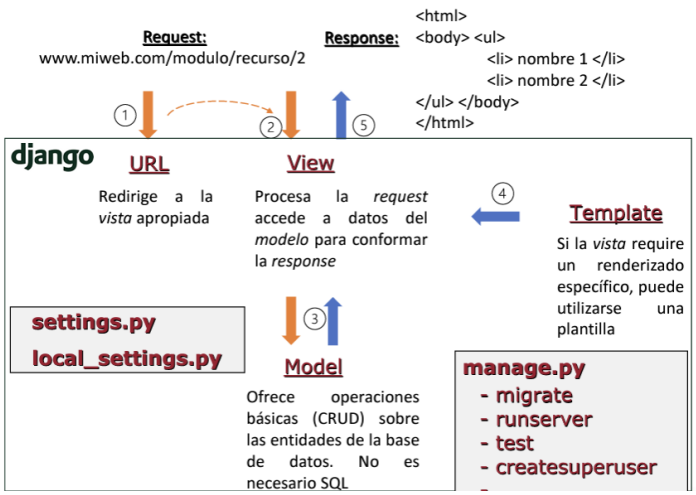
Para que todo esto funcione, el sistema debe estar formado por diferentes componentes funcionando correctamente cada uno por separado, así como unidos.

En este caso la arquitectura de *Decide* está formada por:



Donde podemos observar que está basada en el framework *Django* el cuál hace la mayor parte del trabajo en este caso, ya que este trabaja con un sistema de modelos que funcionan de enlace entre las vistas y la base datos. Además, dichas vistas se suben a un servidor y por ende a Internet gracias a un archivo url.py que permite el enlace entre las vistas y el servidor.

Así pues, *Django* esta administrado por dos archivos, siendo estos manage.py y local.setting.py siendo estos el motor principal de *Django* los cuales hacen que este funcione correctamente, además de cerrar el vínculo con las dependencias y archivos estáticos que componen al proyecto. Todo esto queda mejor reflejado y entendible en la imagen adjuntada en la próxima página.



Tras explicar cómo funciona *Decide*, es conveniente explicar qué subsistemas vamos a integrar en él, además de explicar cómo vamos a integrarlos.

En nuestro caso vamos a integrar los subsistemas:

Traducciones: En él se encuentran las diferentes tareas a realizar.

- Hacer la interfaz traducible
- Traducir la interfaz a español
- Traducir la interfaz a otros idiomas

Para ello se crea un documento html a partir del que ya tenemos y se traduce manualmente al idioma correspondiente, en nuestro caso inglés. Posteriormente se crea un botón en el documento html principal que tenga la lógica suficiente como para cada vez que lo pulses, te redirija al html del otro idioma. Una vez hecho esto en una vista realizamos el mismo procedimiento en la otra vista.

Visualización de resultados: En él se encuentran las diferentes tareas a realizar:

- Pintado de gráficas y estudio de datos.
- Mostrar información relevante a tiempo real
- Implementar visualizaciones para diferentes plataformas como *Telegram*.

Para realizar el pintado de gráficas; sabiendo el id de la votación, extraemos la información y la metemos en listas, con ellas utilizamos los recursos de la libreria *matplotlib* para hacer representar las diferentes gráficas.
Una vez que sabemos representar gráficas. Vamos a enlazarlas con *Telegram*. Para ello, hemos integrado un bot de *Telegram*, empleando una librería llamada *Telegram* la cuál suministra un bot previamente configurado al que se le pasa un token y este subirá al canal aquellas fotos que reciba.

Diseño y usabilidad: En él se encuentran las diferentes tareas a realizar:

- Estudiar la usabilidad y definir una nueva interfaz usable, con componentes css.
- Hacer la interfaz responsive, para que funcione en móviles.
- Estudiar la accesibilidad de la interfaz y hacerla accesible.

Para poder abordar este subsistema, hemos estudiado la interfaz con la que contaba por defecto, llegando a la conclusión de que se debía mejor la UI (User interface) y UX (User experience). Para ello se ha redefinido una interfaz más centrada en la pantalla con colores más vivos y actualizados para obtener una mejora del aspecto de la página.
Además, después de estudiar la accesibilidad, se decidió implementar diferentes funciones que alterar el aspecto de la pantalla permitiendo acomodarlo al gusto del usuario. Siendo estas funcionalidades, modo oscuro, modo escala de grises, modo de alto contraste, aumento del tamaño de fuente y aumento del interlineado.

Todo ello se ha creado haciendo uso de *Css* y *Javascript*.

Una vez explicados que subsistemas hemos realizado e integrado con Decide a lo largo de este cuatrimestre, vamos a pasar a ver qué desempeño ha tenido cada uno de los integrantes del grupo a la hora de realizar dichos subsistemas, además de los correspondientes documentos necesarios.

- **Traducciones:** Para la realización de este subsistema, así como sus respectivos incrementos funcionales, **Miguel Gaviro y Santiago Zuleta de Reales** fueron los responsables de realizarlo. Ambos se pusieron de acuerdo y realizaron de forma mutua dicho subsistema de forma completa y sin ningún problema.

- **Visualización de resultados:** Para este subsistema, los integrantes **Eloy Moreno, Bogdan Marian Stefan y Víctor Gravan** se pusieron de acuerdo para dividirse sus incrementos; Eloy realizó los siguientes de forma individual ("*Pintado de gráficas y estudio de datos*" e "*Implementar visualizaciones para diferentes plataformas como Telegram" (*bot de Telegram donde se pueden visualizar gráficas de las diferentes votaciones*)*) y Bogdan y Víctor realizaron de forma conjunta **"***Mostrar información relevante a tiempo real*".

- **Diseño y usabilidad:** Este subsistema al completo fue desarrollado por **José Mª García** implementando sin ningún problema los diferentes incrementos funcionales de los que este se compone.

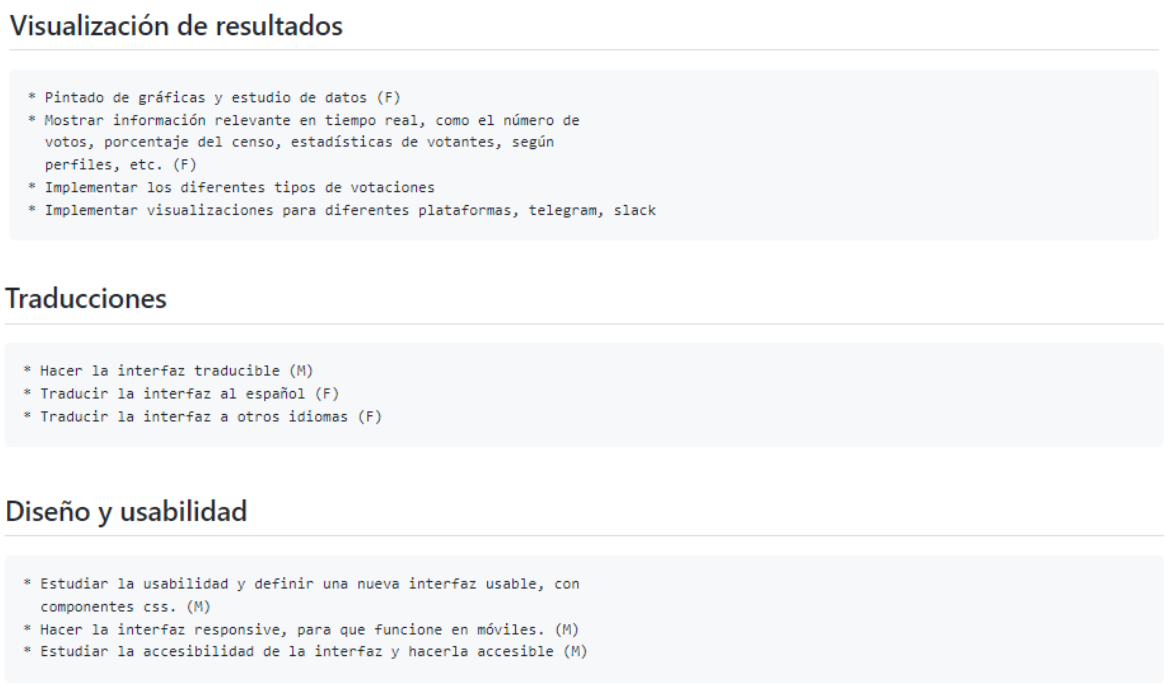
Cuando hablamos de documentación realizada, podemos afirmar que, el *Documento del proyecto* ha sido realizado por **Jose Mª García**, *el Diario de equipo* por **Bogdan Marian y Víctor Gravan**, el *Patrón de commits* por **Santi Zuleta de Reales**, el *Proceso de Gestión de incidencias* y la *Plantilla de incidencias* por **Miguel Gaviro** y la *Gestión de conflictos* y el *Acta Fundacional* por **Eloy Moreno.**

Así pues, a modo de breve resumen podemos afirmar que tras esta explicación tenemos que:

Miguel Gaviro cuenta con **1.5** incrementos funcionales y **2** documentos.
Santiago Zuleta de Reales cuenta con **1.5** incrementos funcionales y **1** documento.
Eloy Moreno cuenta con **2** incrementos funcionales y **2** documentos.
Bogdan Marian cuenta con **0.5** incrementos funcionales y **0.5** documentos.
Víctor Gravan cuenta con **0.5** incrementos funcionales y **0.5** documentos.
José Mª García cuenta con **3** incrementos funcionales y **1** documento.

Visión global del proceso de desarrollo

Para dar una visión global del proceso de desarrollo, debemos explicar como ha ido evolucionando el proyecto a lo largo de las semanas, explicando y argumentando cada una de las decisiones tomadas en el tiempo.
En primer lugar, tomamos la decisión de realizar únicamente el subsistema **de Visualización de resultados**. Para ello nos dividimos las diferentes tareas referidas en el apartado anterior y nos pusimos manos a la obra. No obstante, vimos que quizás se iba a quedar corto por lo que decidimos que la mejor idea sería elegir nuevos subsistemas. Finalmente, ante esta decisión, los subsistemas elegidos fueron el de **Diseño y usabilidad** y **Traducciones**.

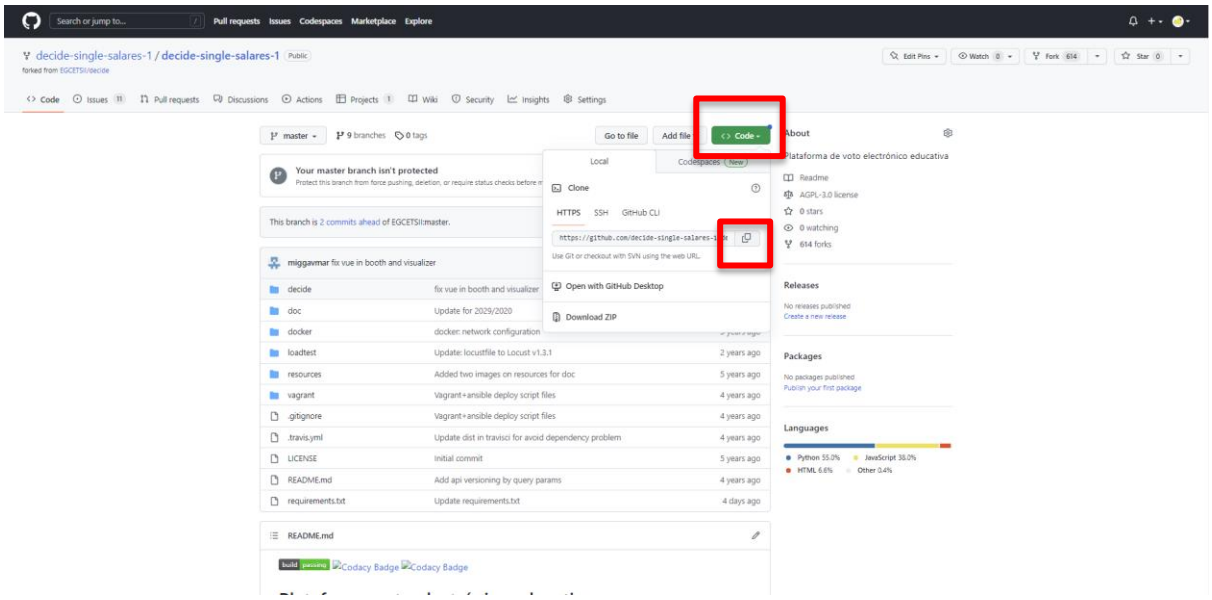


El proceso seguido, en nuestro proyecto, para realizar el subsistema de **Traducciones** ha sido el siguiente:
En primer lugar, todos los integrantes del grupo se reúnen, independientemente del medio, para negociar como poder abordar el nuevo subsistema de **Traducciones**. Durante la llamada se propusieron una serie de pros y contras sobre la decisión tomada y cómo esta va a repercutir en el proyecto. Una vez que se han expuesto y abordados estos temas pasamos

a ver la opinión de cada integrante del grupo para ver su opinión al respecto. Tras exponer cada uno su opinión decidimos que finalmente si se abordaría el subsistema. Como la respuesta ha sido de índole positiva entonces no pusimos a rellenar la plantilla de gestión de cambios/incidencias que se encuentra creada con dicho objetivo.
Una vez que este proceso finalizó, aquellos que se encargan de desarrollar dicha tarea, en este caso Miguel y Santiago, se ponen manos a la obra en proceso de producción hasta lograr su finalización de forma adecuada.

Entorno de desarrollo

El entorno de desarrollo empleado para la correcta elaboración del proyecto por parte de todos los integrantes del grupo es **Visual Studio Code**, ya que este nos permite agilizar el proceso de creación y subida de archivos a la plataforma de cambio de versiones, siendo en nuestro **GitHub**. El hecho de que todos los integrantes del grupo cuenten con el mismo entorno de desarrollo no es casualidad, puesto que ha sido previamente acordado por parte de todos con el objetivo de normalizar el entorno y evitar futuros errores debido al uso de diversos entornos.
Para desplegar la aplicación con sus subsistemas, será necesario seguir los siguientes pasos. En primer lugar, se deberá dirigir al siguiente enlace https://github.com/decide-single-salares-1/decide-single-salares-1 y copiar la url en el botón que se le indica.



Posteriormente debe ir al terminal de su dispositivo y escribir el comando:
git clone <url>
Siendo <url> la url previamente copiada.
Una vez que tenga clonado el repositorio en local deberá crear un entorno virtual con el siguiente comando(Teniendo previamente python3 instalado):
Python3 -m venv<env>
Siendo <env> el nombre que queramos ponerle al entorno virtual

Tras crear el entorno virtual entramos en él:
Source <env>/bin/activate

Una vez activado será necesario instalar las dependencias del proyecto, las cuales se encuentran el archivo requirements.txt
Pip install -r requirements.txt

Tras esto deberemos crea una base de datos local con postgres de la siguiente forma:
Sudo su -postgres
Psql -c "créate user decide with password 'decide'"
Psql -c "créate database decide owner decide"
Exit

Cuando salimos postgres debemos movernos hasta la carpeta de decide dentro de nuestro proyecto y hacer una migración y crear un super usuario, para finalmente arrancar el servidor.

Cd decide-single-salares-1
Cd decide
./manage.py migrate
./manage.py createsuperuser
./manage.py runserver

Además, debemos destacar que las versiones usadas de los diferentes componentes del proyecto son:

Vs code versión 2022
Django==2.0
pycryptodome==3.6.6
djangorestframework==3.7.7
django-cors-headers==2.1.0
requests==2.18.4
django-filter==1.1.0
psycopg2==2.8.4
django-rest-swagger==2.2.0
coverage==4.5.2
django-nose==1.4.6
jsonnet==0.12.1
…

Siguiendo las instrucciones no debería tener problemas para arrancar la aplicación de forma local en su dispositivo.

Ejercicio de propuesta de cambio

El proceso seguido a la hora de determinar cualquiera de estos cambios es el siguiente:
En primer lugar, todos los integrantes del grupo se reúnen, independientemente del medio, para negociar como poder abordar un nuevo subsistema. Durante la llamada se proponen una serie de pros y contras sobre la decisión tomada y cómo esta va a repercutir en el proyecto. Una vez que se han expuesto y abordados estos temas pasamos a ver la opinión

de cada integrante del grupo para ver su opinión al respecto. Tras exponer cada uno su opinión se decide finalmente si se aborda o no el subsistema. Si la respuesta final es negativa, el subsistema no sigue adelante y todo continúa como antes. En cambio, si la respuesta es de índole positiva entonces será necesario rellenar la plantilla de gestión de cambios/incidencias que se encuentra para dicho objetivo.
Una vez que este proceso finaliza, aquellos que se encargan de desarrollar dicha tarea se ponen manos a la obra hasta lograr su finalización de forma adecuada.

Conclusión y trabajo futuro

Para finalizar este documento será necesario sacar diversas conclusiones sobre el proyecto, así como su desarrollo y prácticas a mejorar en un futuro.
Como conclusiones principales podemos sacar que para un correcto y cómo desarrollo del proyecto a lo largo del curso, es necesaria una buena planificación, una comunicación sobresaliente entre todos los integrantes del grupo e interés por igual en el desarrollo, ya que si alguno de estos aspectos no se cumple, como ha sido nuestro caso, el desarrollo del proyecto va a ser complicado y exhaustivo.
Como aprendizaje para mejorar en un futuro debemos quedarnos en mejorar desde primera hora la configuración de Django ya que esta es muy importante y el hecho de no tener experiencia previa con dicho framework no ayudó.