

# DECIDE

Plataforma de voto electrónico  
educativa

<https://github.com/wadobo/decide/>



Víctor Ramírez de la Corte <[virako@wadobo.com](mailto:virako@wadobo.com)>  
Daniel García Moreno <[danigm@wadobo.com](mailto:danigm@wadobo.com)>



# Índice

- 1 - Sistemas de voto electrónico
- 2 - Decide
- 3 - Arquitectura
- 4 - Introducción a Django
- 5 - Código y despliegue
- 6 - Extensiones proyecto



# Voto electrónico

- Sistemas de voto surgen a partir de la necesidad de toma de decisiones
- Existen diferentes formas de votar, mano alzada, papel en urna, etc.
- También existen diferentes tipos de votaciones, voto público, voto privado, etc



# Voto electrónico

¿Qué es el voto electrónico?

- Uso de ordenadores en una votación
- A través de internet o presencial



# Voto electrónico

Ventajas del voto electrónico:

- Es más barato realizar votaciones
- Es mucho más accesible para personas con movilidad reducida, votantes en diferentes zonas del mundo



# Voto electrónico

## Inconvenientes del voto electrónico

- Más \*complejo\* de auditar
- Más fácil de atacar (servidores o clientes)
- Puede facilitar la compra de votos o la extorsión a la hora de votar



# Voto electrónico

## Tipos de voto electrónico

- Voto público (fácil de implementar)
- Voto secreto sin cifrado (confianza en el servidor y administradores)
- Voto secreto criptográficamente seguro (confianza en una serie de autoridades)



# Voto electrónico

## Diferentes tipos de votaciones

- Todos los votos valen lo mismo
- Votos con pesos
- Votación directa, 1 voto suma 1 y gana la opción más votada
- Votación por preferencia, ordenación de opciones (Borda, Condorcet, etc)





# Decide

¿Qué es Decide?

- Un sistema de voto electrónico educativo
- Implementa un sistema básico de votación electrónica con voto secreto y criptográficamente seguro punto a punto
- Modular y extensible



# Decide

¿Qué aporta Decide?

- Se resuelve el problema de forma efectiva, sin ser un sistema de uso real, por lo que se pueden simplificar los problemas
- Aporta una implementación de referencia y permite extensión en cualquier tecnología cumpliendo una API web básica



# Decide

## ¿Qué aporta Decide? (2)

- Proyecto software libre, público con el código en github
- Una serie de puntos de extensión para que los estudiantes puedan trabajar
- Basado en agora-voting
  - <https://github.com/agoravoting>



# Arquitectura

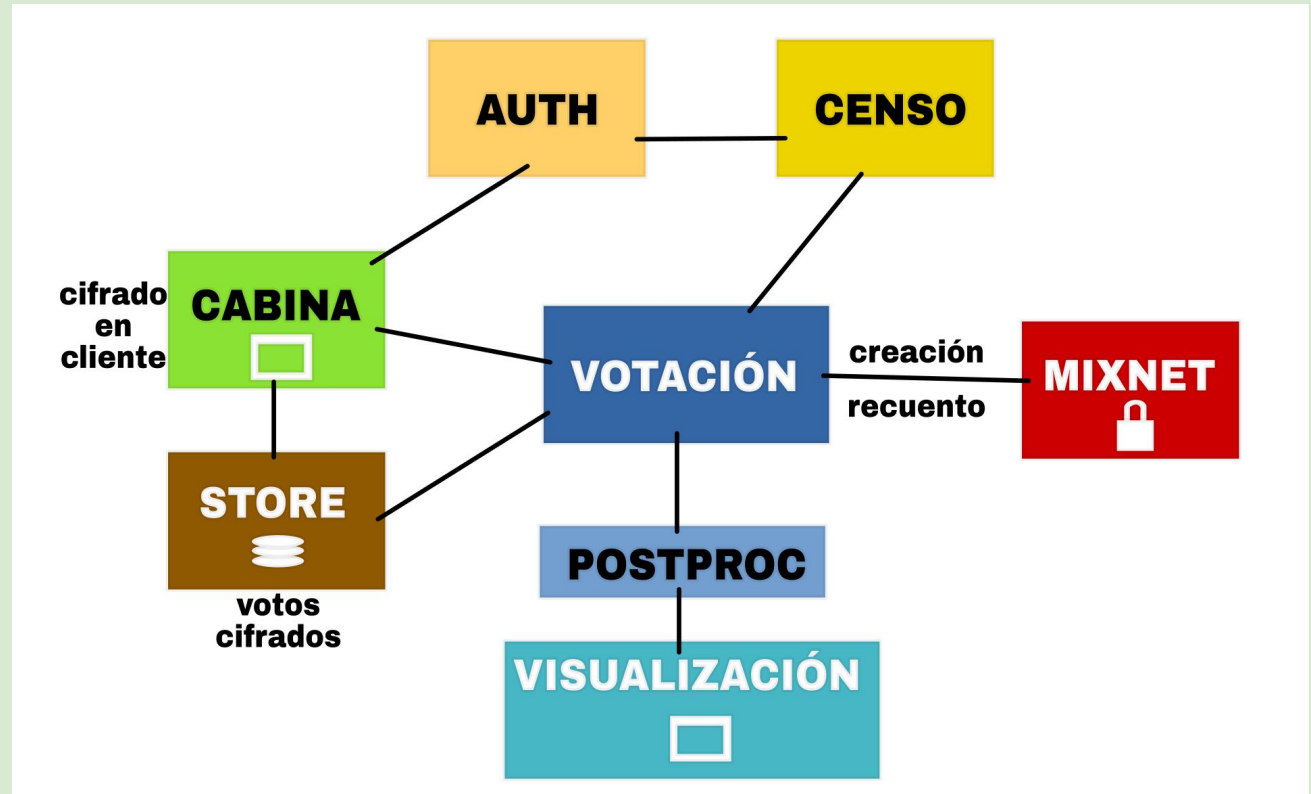
- Módulos independientes
  - No dependemos de un lenguaje
  - Podemos distribuir la carga en diferentes máquinas
  - Más sencillo a la hora de abordar una tarea
- Conectados mediante API (requests)



# Arquitectura

Módulos:

- base
- auth
- census
- voting
- booth
- mixnet
- store
- postproc
- visualizer



# Arquitectura

- Base:
  - Funcionalidades comunes
  - Similar a una librería
- Authentication:
  - Autenticación de votantes
  - Actualmente funciona con usuario/contraseña
  - Futuro: podremos añadir otros modos de autenticación, enlace único por sms o correo, certificado FNMT, redes sociales, etc.



# Arquitectura

- **Census:**
  - Fuertemente relacionado con el módulo authentication
  - Guardamos quién puede votar y quién ha votado
- **Voting:**
  - Definición de las votaciones:
    - Preguntas
    - Opciones
    - Tipo de voto (simple, elección múltiple, por peso).
- **Booth**
  - Cabina de votación, donde entrará el votante para emitir su voto



# Arquitectura

- Mixnet:
  - Encargada de la parte criptográfica
  - La mixnet sería una red de ordenadores (1 a n), los cuales generan una clave compartida para cifrar votos, por lo que solo se podrán descifrar los votos, una vez todas se pongan de acuerdo
  - En el recuento, el votante se desliga del voto, ya que cada autoridad realiza un barajado y se lo pasa a la siguiente. Anonimización del voto
  - Faltarían las pruebas de cero-conocimiento, que por unas propiedades matemáticas, el orden de los factores no altera el producto, y también podríamos comprobar que las autoridades no han modificado nada





# Arquitectura

Encrypted votes

=>

mixnet

=>

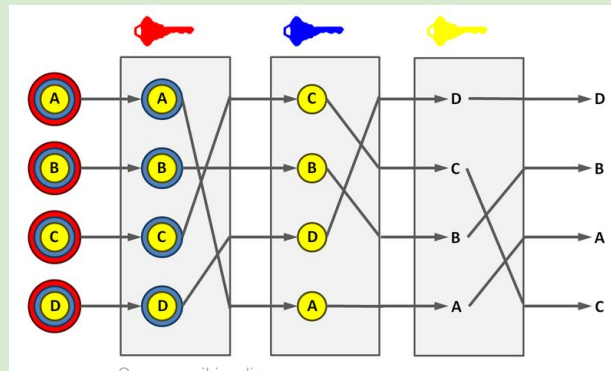
clear text votes and anonymous

John Vote

"alpha:"42589531826772237145388040926874905138  
9633699899139097103333560979961144187706452613  
6245284657244122497804586518859294565187778554  
1874859765385259678770773327674493904355475862  
3230798244900422118578808183914623385583455279  
2403587041865476454151642574286811110595709606  
3059435853688983226705444108194553893898313678  
0995873018329959498913007653851094568763566368  
3098296210633619365132089120745366382408142036  
5227472962308730

Jane Vote

"alpha:"42589531826772237145388040926874905138  
9633699899139097103333560979961144187706452613  
6245284657244122497804586518859294565187778554  
1874859765385259678770773327674493904355475862  
3230798244900422118578808183914623385583455279  
2403587041865476454151642574286811110595709606  
3059435853688983226705444108194553893898313678  
0995873018329959498913007653851094568763566368  
3098296210633619365132089120745366382408142036  
5227472962308730



option A

option B

option A

option C

option A

option A

option C

.. and more votes ..

.. and more votes ..

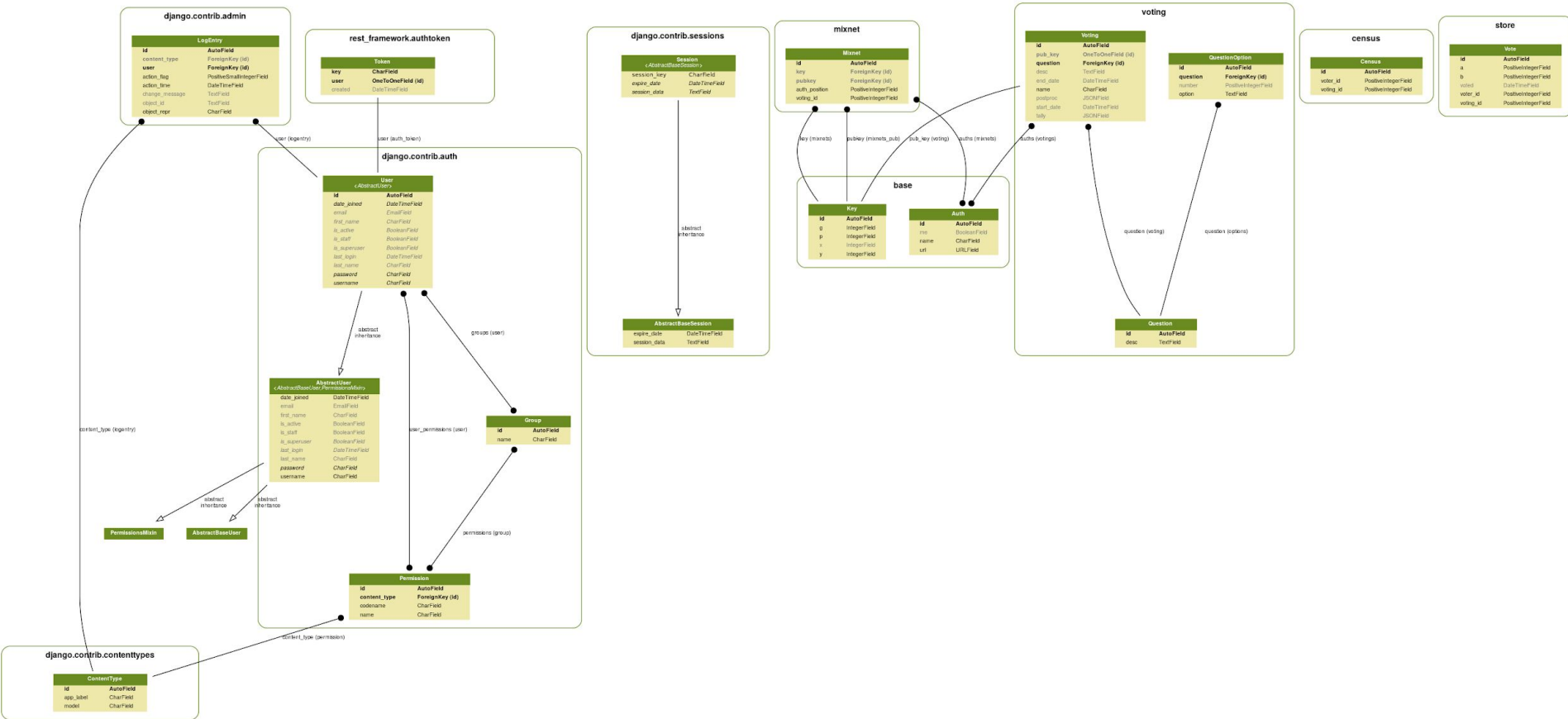


# Arquitectura

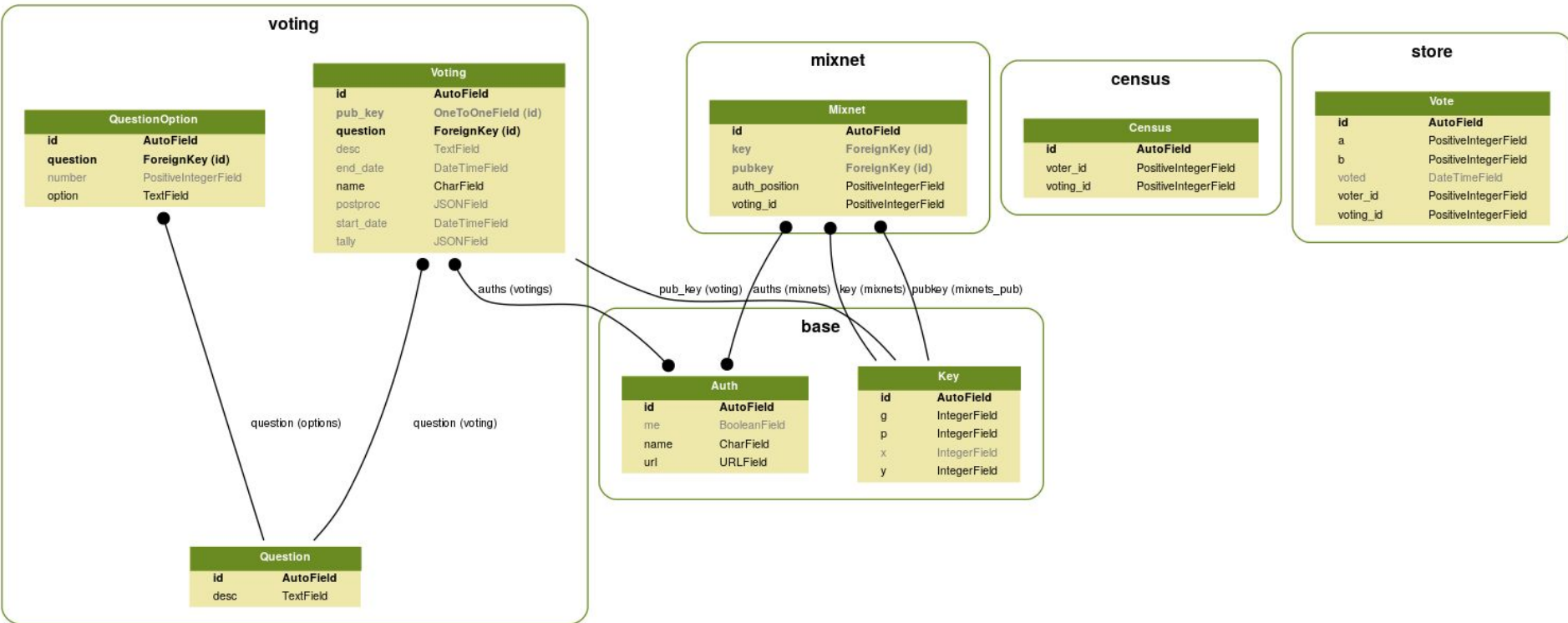
- **Postproc:**
  - Postprocesado del voto, donde con los números obtenidos en el recuento, generamos un resultado coherente al tipo de votación
  - Aquí aplicamos las reglas necesarias, dependiendo del tipo de votación: simple, ley de paridad o ley de d'Hondt
- **Visualizer:**
  - visualizador de datos, simplemente sería un frontend, donde con los resultados obtenidos, los pintamos de forma bonita (gráficas o tablas)



# Arquitectura



# Arquitectura



# Introducción a Django

- Framework web python
- Documentación maravillosa  
<https://www.djangoproject.com/>



# Introducción a Django

- Python: virtualenv y dependencias
  - `python -m venv env`
  - `source env/bin/activate`
  - `pip install -r requirements.txt`
  - `deactivate`
- Proyecto django:
  - `django-admin startproject decide .`
  - `./manage.py startapp`
  - `./manage.py runserver`
  - `./manage.py makemigrations`
  - `./manage.py migrate`
  - `./manage.py test`



# Introducción a Django

- Estructura proyecto de Django:
  - manage.py
  - project/urls.py
  - project/settings.py
  - project/wsgi.py
- Estructura app de Django:
  - app/urls.py
  - app/models.py
  - app/views.py
  - app/tests.py
  - app/admin.py
  - app/serializers.py
  - app/templates/



# Introducción a Django

- Django permite utilizar cualquier BD:
  - mysql, postgres, sqlite, oracle, mongo, ...
- Elección de postgres:
  - `sudo su - postgres`
  - `psql -c "create user decide with password 'decide' "`
  - `psql -c "create database decide owner decide"`
- Django Rest Framework:
  - Para crear API Rest sencillas





# Código y despliegue

<https://github.com/wadobo/decide/>



# Extensiones proyecto

<https://github.com/wadobo/decide/blob/master/doc/issues.md>



# Preguntas

?

