

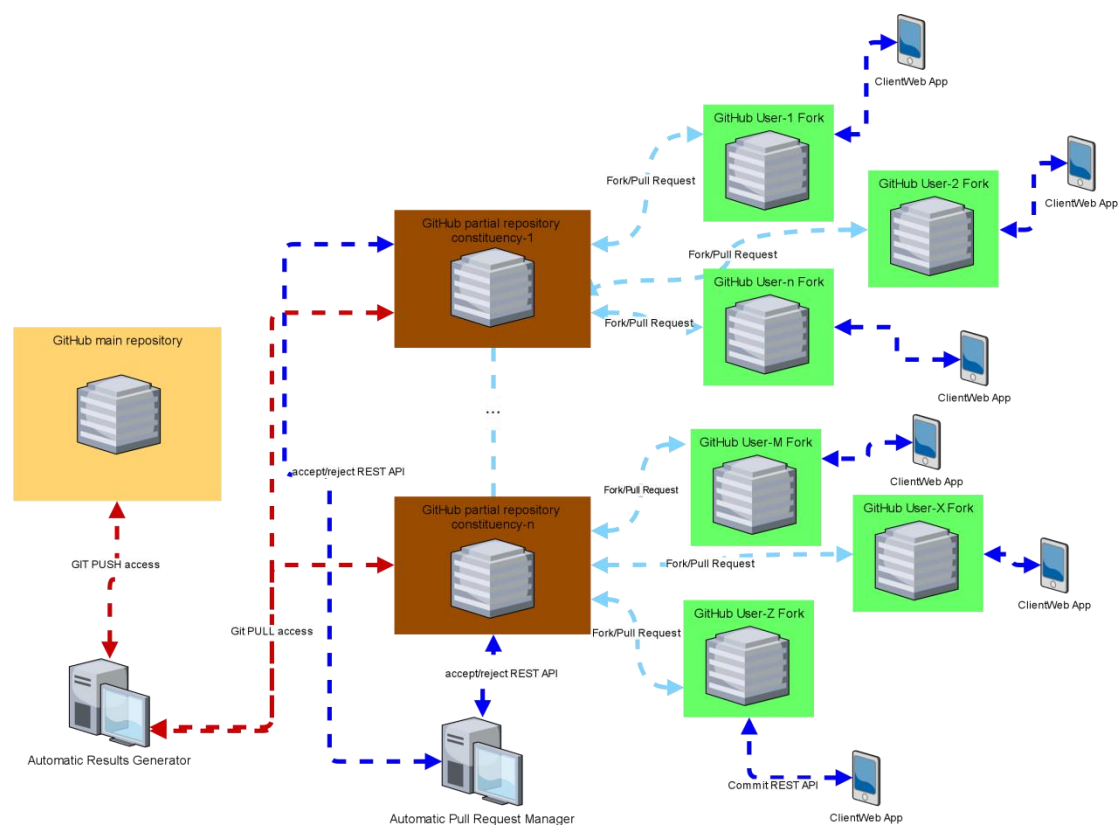
Arquitectura de la solució basada en GitHub

Aquesta solució es basa en la plataforma i infraestructura de GitHub. D'aquesta manera tenim 2 grans problemes solvents al delegar-ho directament a GitHub:

1) Gestió d'usuaris: github ja proporciona un sistema per gestionar usuaris, per donar permisos per accedir a repositoris i per gestionar pull requests cap a altres repos d'altres usuaris, cosa que ens permet delegar la gestió d'usuaris i dedicar-nos a altres aspectes més funcionals de la solució.

2) Gestió d'emmagatzematge: github es dediquen a transferir diàriament milers o milions de fitxers d'un costat al altre, esta clar que aquesta gent ja saben fer molt be la seva feina i garantir disponibilitat i performance, cosa que si ho hem de gestionar nosaltres, ens exposem a que un atac ens faci fora de la xarxa.

Diagrama de la arquitectura proposada



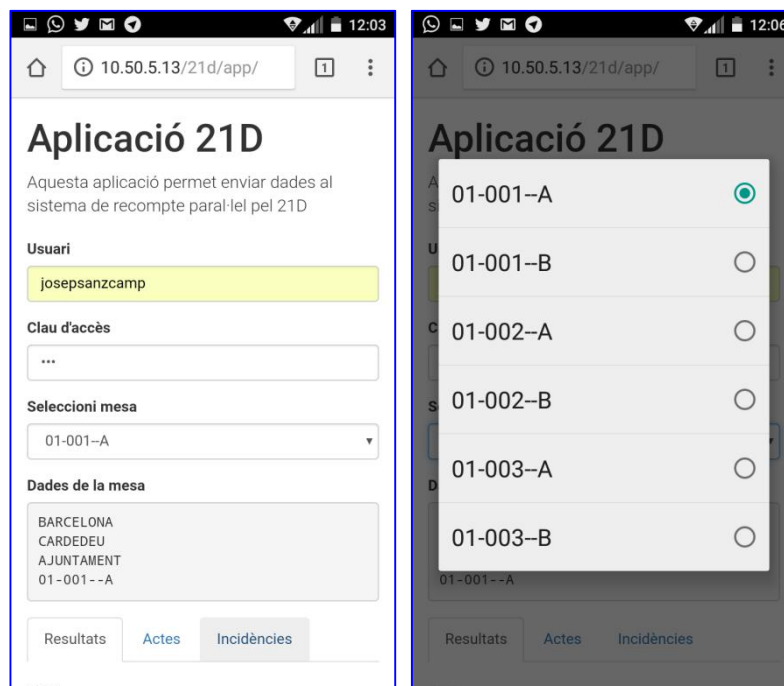
Aplicació webapp proposada:

Els usuaris que estaran a les escoles, podran enviar informació fent servir una webapp. Aquesta aplicació tindrà les següents pantalles:

1) Pantalla principal de l'aplicació: permet entrar les dades d'accés a la aplicació com son el usuari i la clau d'accés a github.



2) Un cop entrada les dades d'usuari i clau a la aplicació, aquesta carregarà del servidor les dades de les taules disponibles per l'usuari entrat a la aplicació. Aquesta prestació el que fa es cercar un fitxer al directori data del servidor amb el nom obtingut del md5 del username de github.



Per defecte, s'activa la primera taula de la llista i l'usuari pot canviar de taula en qualsevol moment.

3) Pantalla per entrar els resultats: aquesta pantalla s'activa en el moment que es selecciona una taula i depenent de la província a la que pertany la escola, mostra els partits de les candidatures per aquella província.

The image displays two side-by-side screenshots of a mobile application interface for entering election results. Both screenshots show a status bar at the top with various icons and the time 12:03.

The left screenshot shows a tabbed interface with three tabs: "Resultats" (selected), "Actes", and "Incidències". Below the tabs, there is a list of political parties with corresponding input fields for their results:

- PSC
- PP
- PACMA
- C's
- ERC-CatSí
- DIÀLEG
- CUP
- PUM+J

The right screenshot shows a list of specific vote counts with corresponding input fields:

- DN
- PFIv
- CNV
- UNIDOS-DEF-Socialistas
- VOTS EN BLANC
- VOTS NULS
- VOTS TOTALS

At the bottom of the right screenshot is a blue button labeled "Enviar".

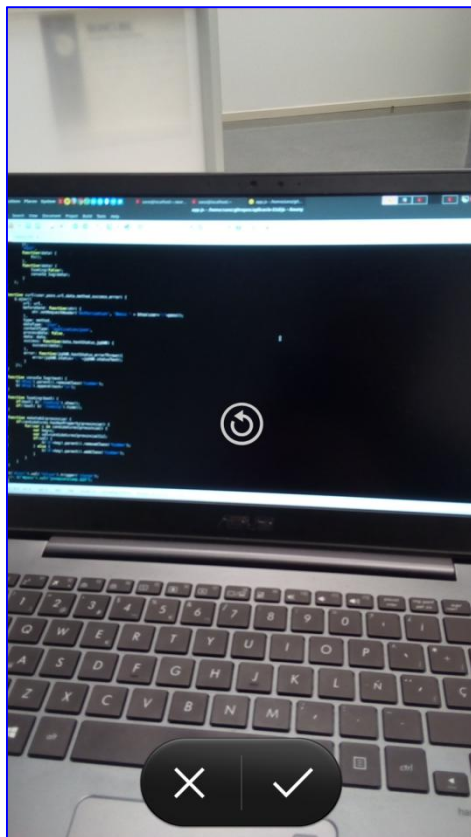
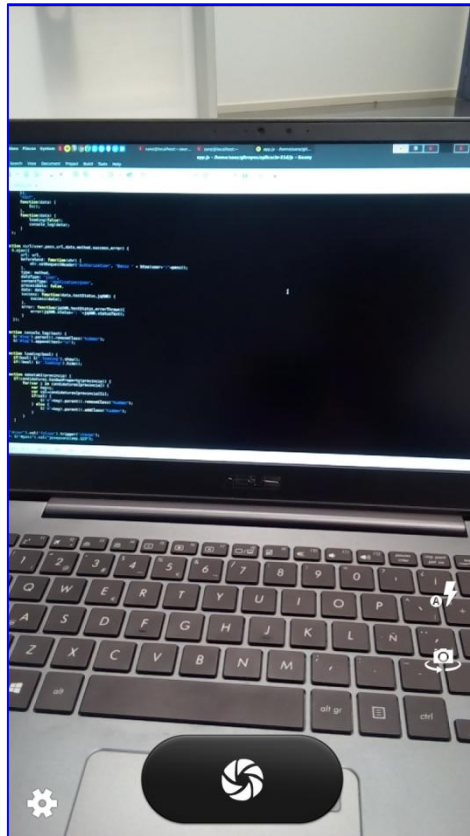
Al final de tot, l'aplicació demana 3 dades addicionals, que son els vots en blanc, els vots nuls i el sumatori de vots totals, emprat com a validació dels valors anteriors.

A més, aquesta aplicació implementa diversos controls addicionals com validar el domini de tots els nombres (enters positius entre un rang determinat)

4) Pantalla per pujar les actes: aquesta aplicació te 3 zones:

- 1) Visualització de l'acta.
- 2) Botó per accedir a la càmera.
- 3) Botor per fer enviar l'acta.

Amb el botó càmera, l'usuari podrà accedir a la càmera per fer la fotografia. Tal com es pot veure a les següents captures de pantalla, l'aplicació permet fer la fotografia, validar si es ok i com a darrer pas, enviar la fotografia de l'acta.



5) Pantalla per enviar incidències: aquesta aplicació permet informar de incidències en els punts de votació mitjançant una capça de text.

The image displays three screenshots of a mobile application interface, likely for reporting incidents. The top-left screenshot shows a screen titled "S'està desant captura de pantalla..." with a header "AJUNTAMENT 01-001--A". Below the header are three tabs: "Resultats", "Actes", and "Incidències". The "Incidències" tab is selected. The main area is labeled "Entri el text de la incidència" and contains a large text input field. At the bottom is a blue button labeled "Enviar".

The top-right screenshot shows the same interface but with a keyboard visible. The text "A la taula no hi ha aigua" is entered into the text field. The keyboard is a standard QWERTY layout with a green back arrow button.

The bottom screenshot shows the same interface with the text "A la taula no hi ha aigua" entered into the text field. The keyboard is not visible, and the "Enviar" button is at the bottom.

La idea principal es que la webapp es una aplicació que es servirà directament des de github.io, que no es mes que un servidor web però sense cap tecnologia de servidor i estarà desenvolupada només fent servir javascript. D'aquesta manera trenquem la dependència tecnològica amb el servidor.

Aquesta aplicació tindrà diversos camps:

- Nom d'usuari => es l'usuari de github
- Clau d'accés => es la clau que l'usuari ha de fer servir per accedir a github

En el moment que un usuari entri les dades del nom d'usuari i clau, per ajax es cercarà un fitxer que hi haurà al servidor, al estil de la app desenvolupada pel refloct, i que contindrà la informació de les taules a les que aquell usuari ha d'enviar dades, per tant, la aplicació carregarà les dades de les taules disponibles per aquell usuari.

A partir d'aquí, l'usuari ja esta a punt d'enviar dades i per tant, ha de escollir entre un dels 3 tabs que la aplicació li mostra:

1) El primer tab permetrà enviar dades del recompte i bàsicament serà un formulari amb una capça de text per cada valor que es vulgui entrar. L'usuari no estarà obligat a entrar tots els valors, sinó nomes els que tinguin alguna dada rellevant, es a dir, si un partit te 0 vots, no cal posar específicament 0.

2) El segon tab permetrà enviar la foto del acta. Per tal de fer-ho fàcil, la aplicació tindrà un botó amb el text "Càmera" i que permetrà fer una fotografia des del telèfon. Aquesta fotografia, si es ok i la donem per bona, es mostrarà en una capça de la aplicació a mode de preview i ja estarà l'usuari preparat per prémer el botó "Enviar"

3) El tercer i darrer tab permetrà enviar una incidència. Aquesta opció de la aplicació permetrà enviar incidències que algú observi a qualsevol dels punts de votació. Això permetrà detectar i reportar anomalies en els centres electorals, si es produeixen es clar!!!

En cada cas, s'enviarà un fitxer al sistema de recompte. La realitat tècnica es que cada vegada que s'envii alguna dada, es farà en forma de fitxer de text o imatge i comportarà un commit al fork que cada usuari farà de forma automàtica al entrar a la aplicació la primera vegada + un posterior pull request per demanar que aquelles dades siguin incloses al sistema de recompte global.

Detalls de cada transacció al sistema

1) Exemple de com enviar dades (fork + upload + pull request)

Per a les següents proves he creat un repo anomenat test-21d al usuari josepsanzcamp. També he creat uns altres repos amb les següents idees:

- josepsanzcamp1 => usuari que farà un fork + commit + pr i que serà acceptat
- josepsanzcamp2 => idem que abans però que no serà acceptat
- josepsanzcamp3 => usuari col·laborador de josepsanzcamp amb permís per acceptar i tancar pull requests del repo test-21d

Fork:

```
curl -u "josepsanzcamp1:josepsanzcamp.1" -d ''  
"https://api.github.com/repos/josepsanzcamp/test-21d/forks"
```

Upload:

```
curl -u "josepsanzcamp1:josepsanzcamp.1"  
"https://api.github.com/repos/josepsanzcamp1/test-21d/git/refs/heads/master"
```

Hem de guardar el object.sha=1221839e108ba0b5d22b3ad447a38355d428ecb3

```
curl -u "josepsanzcamp1:josepsanzcamp.1"  
"https://api.github.com/repos/josepsanzcamp1/test-21d/git/commits/1221839e108ba0b5d22b3ad447a38355d428ecb3"
```

Hem de guardar el tree.sha=cb96f808694fd3ee3274302b84b04b4332e5fb3c

```
curl -u "josepsanzcamp1:josepsanzcamp.1"  
"https://api.github.com/repos/josepsanzcamp1/test-21d/git/blobs" -X POST -d '{"content":  
"## Hello World \n\n## By Sanz\n\n", "encoding": "utf-8"}'
```

Hem de guardar el sha:85d3ac752f0c0a32c1b7135d52468e29005e3ea3

```
curl -u "josepsanzcamp1:josepsanzcamp.1"  
"https://api.github.com/repos/josepsanzcamp1/test-21d/git/trees" -X POST -d  
'{"base_tree": "cb96f808694fd3ee3274302b84b04b4332e5fb3c", "tree": [{"path": "test.md",  
"mode": "100644", "type": "blob", "sha": "85d3ac752f0c0a32c1b7135d52468e29005e3ea3"}]}'
```

Hem de guardar el sha:dc1ab654f1a8f17fe96c328a16a70b523644a1fd

```
curl -u "josepsanzcamp1:josepsanzcamp.1"
"https://api.github.com/repos/josepsanzcamp1/test-21d/git/commits" -X POST -d
'{"parents": ["1221839e108ba0b5d22b3ad447a38355d428ecb3"], "tree":
"dc1ab654f1a8f17fe96c328a16a70b523644a1fd", "message": "Update the file"}'
```

Hem de guardar el sha:272cf09531964a812f4219765e2c7f230308685d

```
curl -u "josepsanzcamp1:josepsanzcamp.1"
"https://api.github.com/repos/josepsanzcamp1/test-21d/git/refs/heads/master" -X POST -d
'{"ref": "refs/heads/master", "sha": "272cf09531964a812f4219765e2c7f230308685d"}'
```

Pull request:

```
curl -u "josepsanzcamp1:josepsanzcamp.1"
"https://api.github.com/repos/josepsanzcamp/test-21d/pulls" -X POST -d '{"title" :
"UsuariX/escolaY/TaulaZ", "head" : "josepsanzcamp1:master", "base":"master"}'
```

2) Exemple de com acceptar o denegar un pull request (merge o close)

Per obtenir la llista de pull requests:

```
curl -u "josepsanzcamp3:josepsanzcamp.3"
"https://api.github.com/repos/josepsanzcamp/test-21d/pulls"
```

Hem de comprovar el user.login que sigui d'un usuari que estigui validat, en aquest cas ha de ser josepsanzcamp1

Per acceptar:

```
curl -u "josepsanzcamp3:josepsanzcamp.3"
"https://api.github.com/repos/josepsanzcamp/test-21d/pulls/1/merge" -X PUT -d
'{"commit_title" : "first automated merge"}'
```

Per denegar:

```
curl -u "josepsanzcamp3:josepsanzcamp.3"
"https://api.github.com/repos/josepsanzcamp/test-21d/pulls/2" -X PATCH -d
'{"state":"close"}'
```