

Dashboard KPI in realtime



Projecte final de màster en Enginyeria Informàtica UOC
Eduard Heredia Lubino

Introducció

KPI → Una mètrica de negoci és una part important del mateix, ja que ens permet aproximar comportaments futurs i ens alerta d'esdeveniments recents.

Un dashboard (o panell) de KPI és una eina que permet visualitzar mètriques obtingudes de diferents fonts amb l'objectiu tenir una visió general d'un conjunt de dades i veure la seva evolució en el temps.



Petició del client

Vull una pantalla on es puguin veure informació rellevant per negoci i que actualitzi el més aviat possible sense la meva intervenció.

1. Llenguatge comú (Ubiquitous Language)

<< Cal definir un llenguatge comú i rigorós entre els desenvolupadors i els usuaris >>

<http://martinfowler.com/bliki/UbiquitousLanguage.html>

Petició del client

Vull una pantalla on es puguin veure informació rellevant per negoci i que actualitzi el més aviat possible sense la meva intervenció.



Desenvolupar un dashboard de KPI widgets amb informació actualitzada en temps real.

Objectiu del projecte

1. Llenguatge comú (Ubiquitous Language)



<< Cal definir un llenguatge comú i rigorós entre els desenvolupadors i els usuaris >>

<http://martinfowler.com/bliki/UbiquitousLanguage.html>

Tècnic	Negoci	Llenguatge comú
Dashboard	Pantalla	Dashboard
Widget	Crear petites aplicacions	Widget
Instància de Widget	Petita aplicació amb dades	Instància de Widget
Dades	Dades	Widget Data
Usuari	Client	User

DASHBOARD

My Dashboard

USER

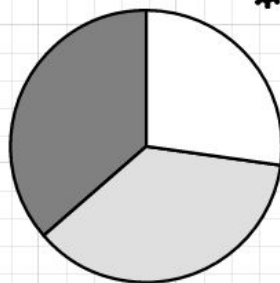
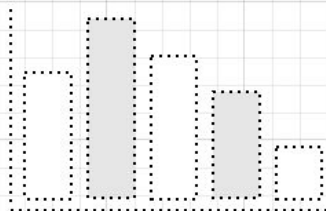
[Logout eduheredia](#)

General

Performance

Alerts

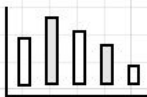
+ Add new tab



DADES

INSTANCIA DE WIDGET

Stats Google



WIDGETS

2. Blocs del domini

A partir d'un conjunt de coneixement, s'ha d'extreure informació. L'objectiu és **modelar** aquest domini amb ajuda dels següents elements :

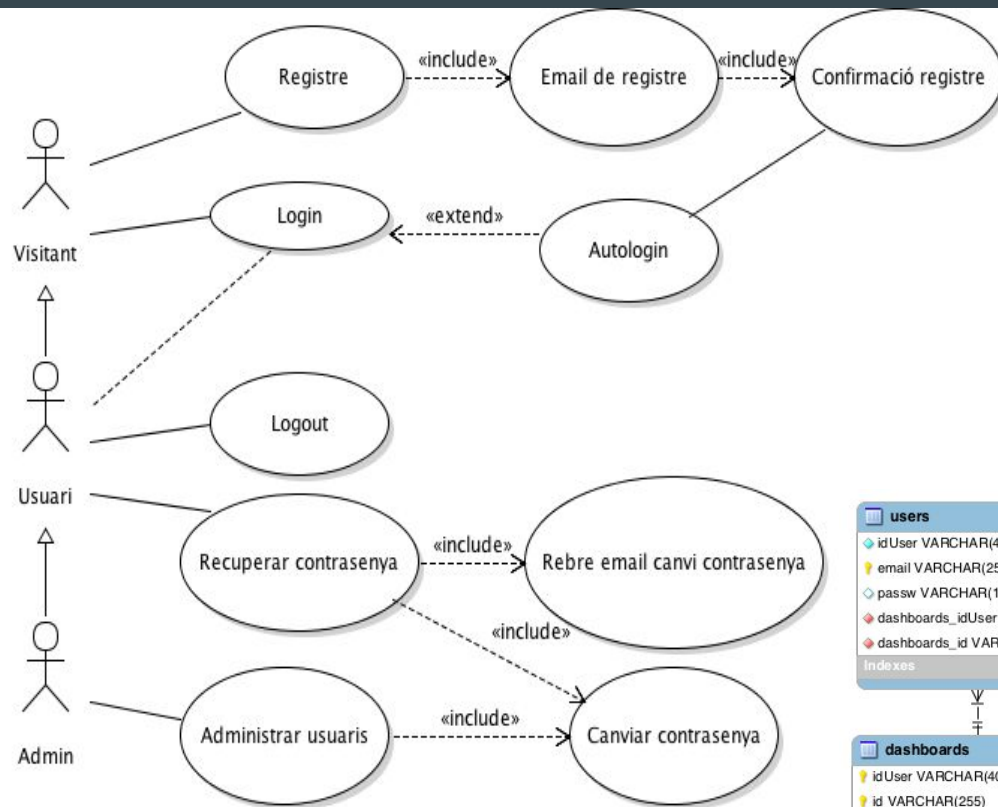
	Definició	Exemples
Entitats	Es un objecte que te una identitat i unes accions	Dashboard, Widget ...
Events de domini	Es un succés que s'ha de produir	Generació de noves dades
Repositori	Per obtenir o persistir els models	Dashboard en Mysql
Value Objects	Son objectes sense identitat pròpies immutable	Widget Data

2. Blocs del domini

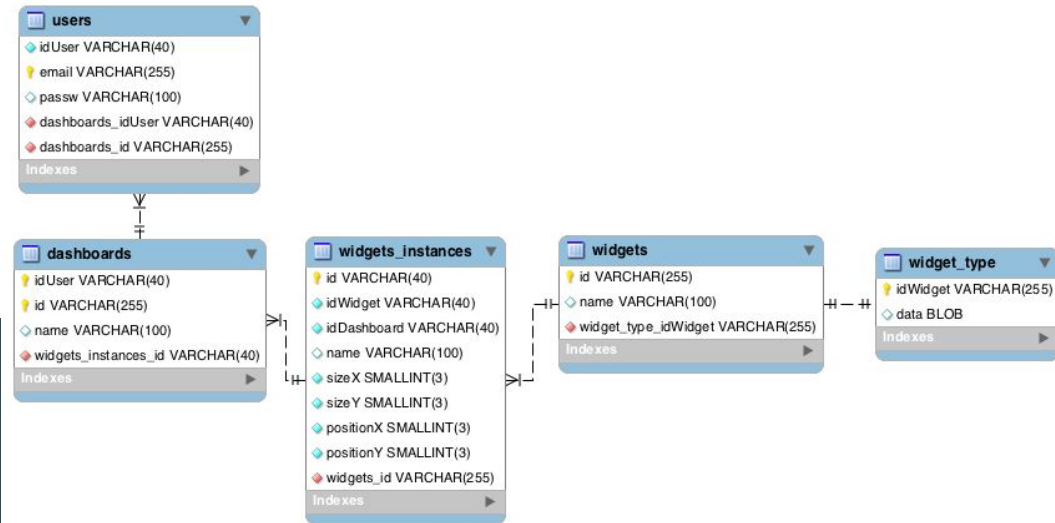


A partir d'un conjunt de coneixement, s'ha d'extreure informació. L'objectiu es **modelar** aquest domini amb ajuda dels següents elements :

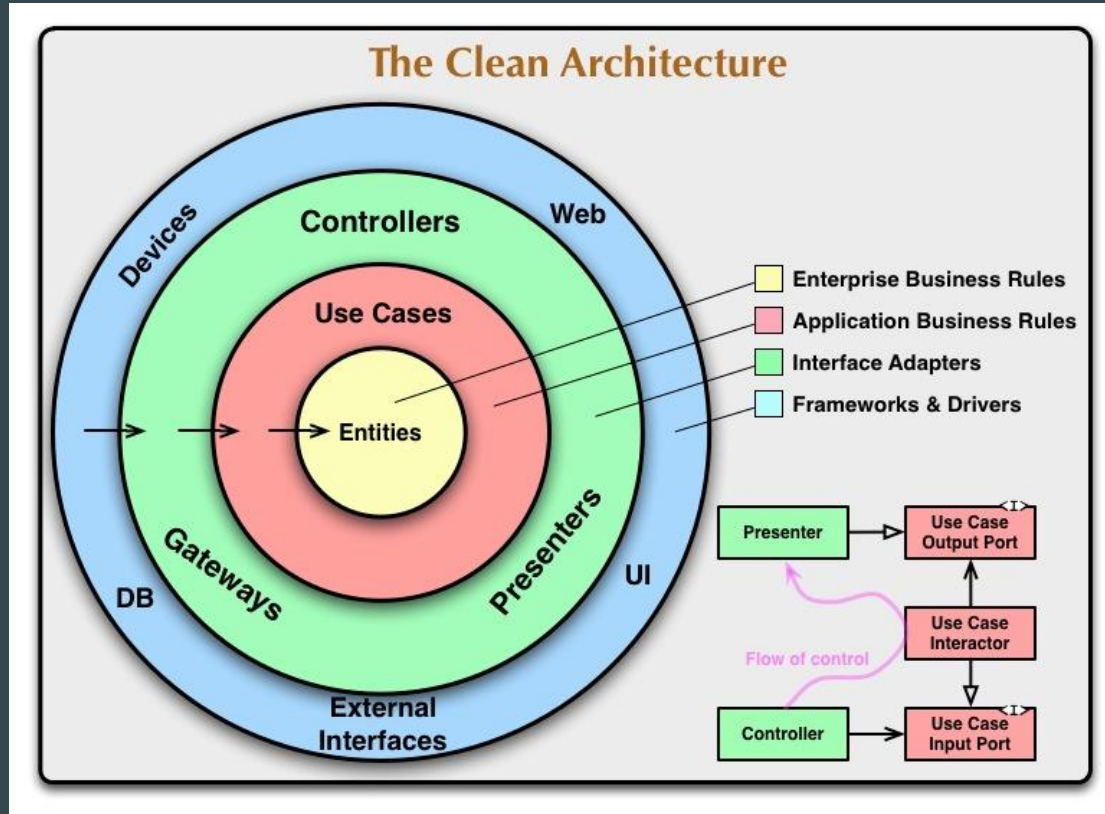
	Definició	Exemples
Casos d'ús / Serveis	Una acció en concret	Crear un dashboard, posicionar un widget
Agregats	Tenen una entitat comú	Instància de Widget amb les dades
Factories	Especialitzar entitats	Instància de Widget de memòria o CPU



Obtenir els casos d'ús, les entitats i els repositoris es fa de la forma habitual. No requereix esforç extra



3. Arquitectura neta (Clean architecture)



3. Arquitectura neta (Clean architecture)



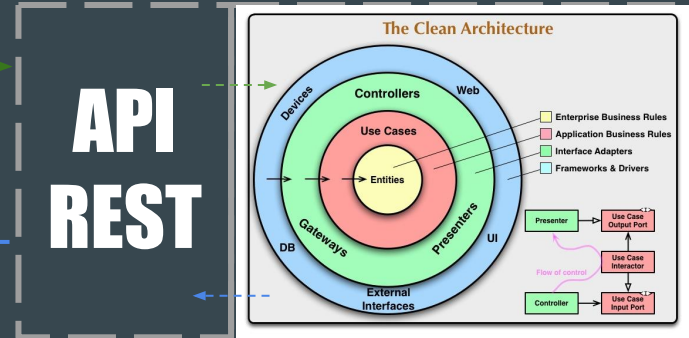
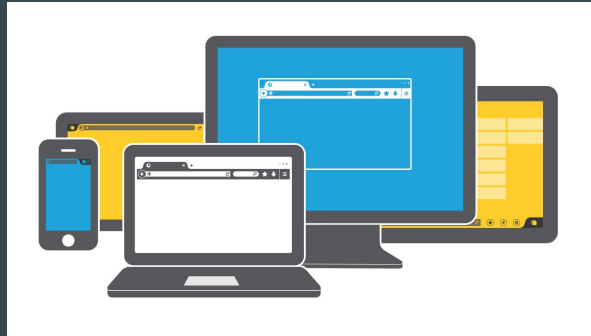
Organitzant l'arquitectura en capes, obtenim els següents beneficis inicials :

- Independència entre frameworks, UI ...
- Resultats reproduïbles més d'una vegada (si es donen les mateixes condicions)
- Separació de responsabilitats
- Adaptació entre les dades d'entrada i de sortida
- Intercanvi de components (les capes es comuniquen per contractes-interfaces)

4. Separació de contextes (Bounded context)



Els casos d'ús es poden convertir en recursos d'una API REST. Aquests poden ser consumits fora del bounded context



5. Codificació amb bones pràctiques

Amb els 4 punts anteriors es posen les bases per arribar a programar un codi d'alta qualitat. Aplicant SOLID a l'arquitectura neta anterior, s'obtenen beneficis immediats, alguns poden ser:

- Disminució de la complexitat ciclomàtica
- Facilitat en el testeig
- Reemplaçabilitat de components
- ...

5. Codificació amb bones pràctiques



A dia d'avui, amb les eines disponibles no es pot entendre en desenvolupament sense aquestes bones pràctiques:

- Testejar el codi (imprescindible el domini)
- Estàndard de codificació unificat
- Duplicació de codi el mínim imprescindible
- Refactoritzar per millorar el codi
- Automatitzar tasques (exemple passar els tests en el pre-commit)
- Històric de canvis (sistema de versions)
- Gestor de dependències
- Ús de patrons

Exemples d'eines de qualitat utilitzats

```
eduhheredia@MacBook-Air-de-Edu:~/PhpstormProjects/UOC_projecte/dashboard$ make run-all-tests  
mocha src/js/Tests/
```

```
api  
  Verify if api is running  
  ✓ Api is up (1459ms)  
  Invalid route  
  ✓ /api/aaaaaaa is a invalid route (1187ms)  
  
api routes  
  GET /api/widgets  
  ✓ return all widgets (966ms)  
  
3 passing (4s)
```

Inspection Results for Inspection Profile 'Project Default'

- dashboard (51 items)
 - Code Smell (5 items)
 - HTML (1 item)
 - Php Inspections (EA Extended)
 - PHPDoc (1 item)
 - Probable bugs (2 items)
 - Route**
 - Undefined (37 items)
 - Unused (5 items)

Maintainability Index **106**

Cyclomatic complexity 1

Bugs by file 0.08

Lines of code 105

Logical lines of code 20

Comment lines of code 32

Difficulty 4.19

Intelligent content 46.89

Vocabulary 21.33

Analyzed files 3

```
eduhheredia@MacBook-Air-de-Edu:~/PhpstormProjects/UOC_projecte/dashboard$ make run-php-tests  
./vendor/phpunit/phpunit/phpunit --bootstrap=vendor/autoload.php -c phpunit.xml.dist  
PHPUnit 4.8.20 by Sebastian Bergmann and contributors.
```

```
.....  
Time: 946 ms, Memory: 9.25Mb
```

```
OK (39 tests, 77 assertions)
```

Search: (No Grouping)

Incl.	Self	Called	Function	Location	
	137 824	762	(0)	{main}	app.php
	112 671	396	1	Symfony\Component\HttpK...	HttpCach
	108 716	88	1	Symfony\Component\HttpK...	HttpCach
	107 358	204	1	Symfony\Component\HttpK...	HttpCach
	106 975	341	1	Symfony\Bundle\Framewor...	HttpCach
	97 259	292	1	Symfony\Component\HttpK...	HttpCach
	95 403	53	1	Symfony\Component\HttpK...	bootstrap
	90 883	126	1	Symfony\Component\HttpK...	bootstrap
	90 538	108	1	Symfony\Component\HttpK...	bootstrap
	90 375	281	1	Symfony\Component\HttpK...	bootstrap
	78 111	3 008	5	<cycle 6>	(unknown)
	57 416	13 722	37	<cycle 2>	(unknown)
	52 774	556	6	Symfony\Component\Event...	classes.p
	52 209	324	6	Symfony\Component\Event...	classes.p
	51 755	1 245	8	Symfony\Component\Event...	classes.p

6. Desacoplar amb events

Per desacoplar encara mes el codi, s'introdueix el concepte d'Event de domini. Aporta els següents beneficis:

- Permet paral·lelitzar accions de llarga durada
- Reaprofitament de codi
- Codi més endreçat
- Seguiment d'accions

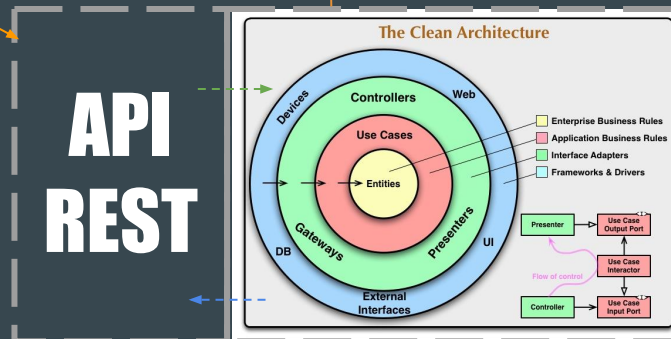
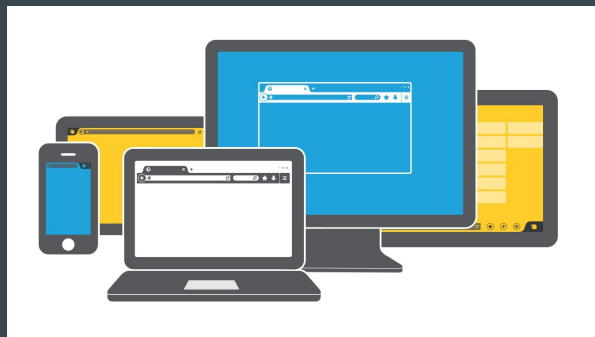
Exemple : Un usuari que es registra a la web, el procés de registre i enviament de mail estan relacionades però son accions diferents. Per tant el registre envia un event d'enviament de mail i es despreocupa dels detalls.

6. Desacoplar amb events



Event Manager

Event Bus



Tecnologies utilitzades?

Son DETALLS d'implementació



BACKBONE.JS

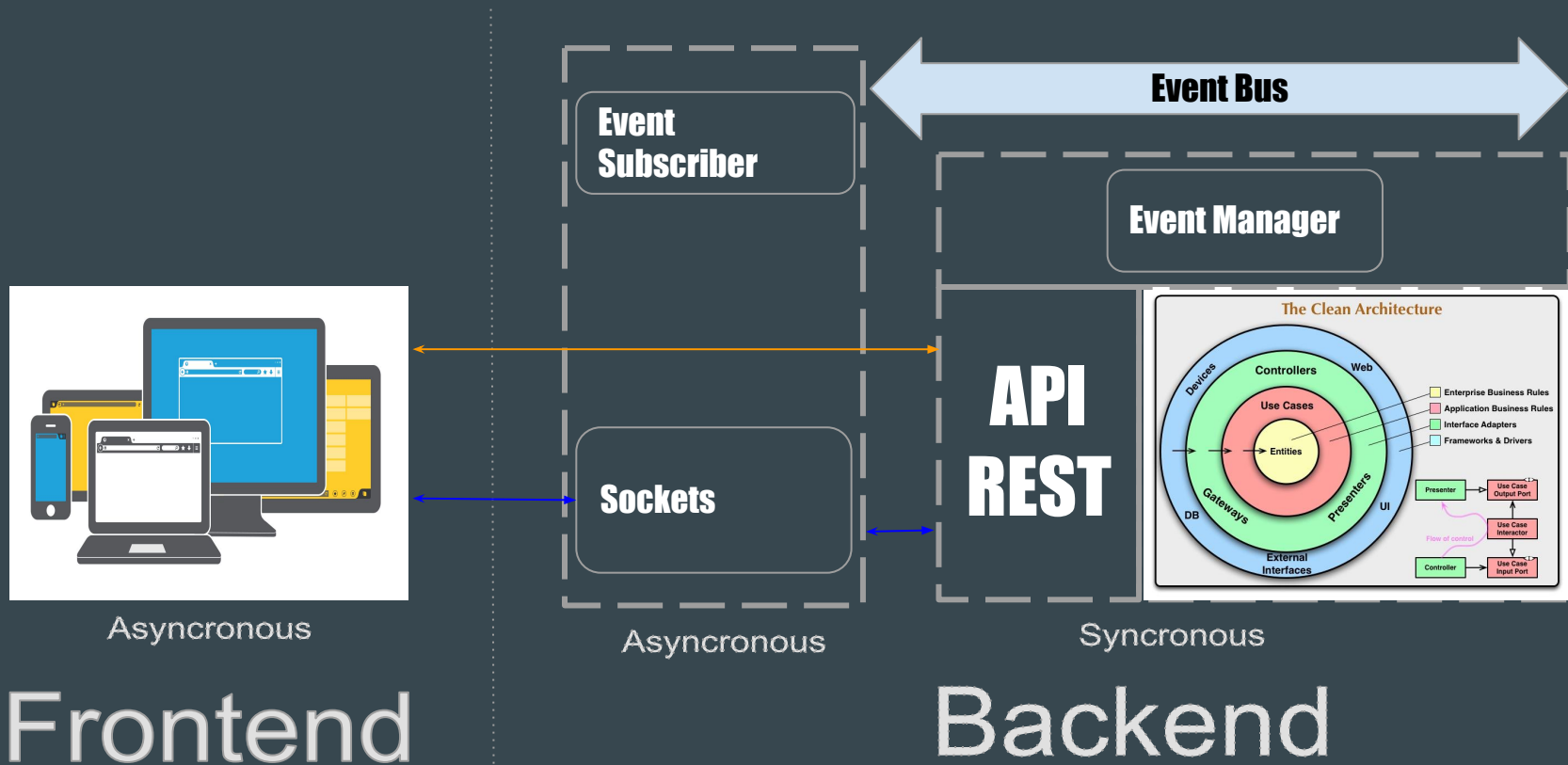


Symfony



redis

Organització de l'aplicació Dashboard KPI en temps real

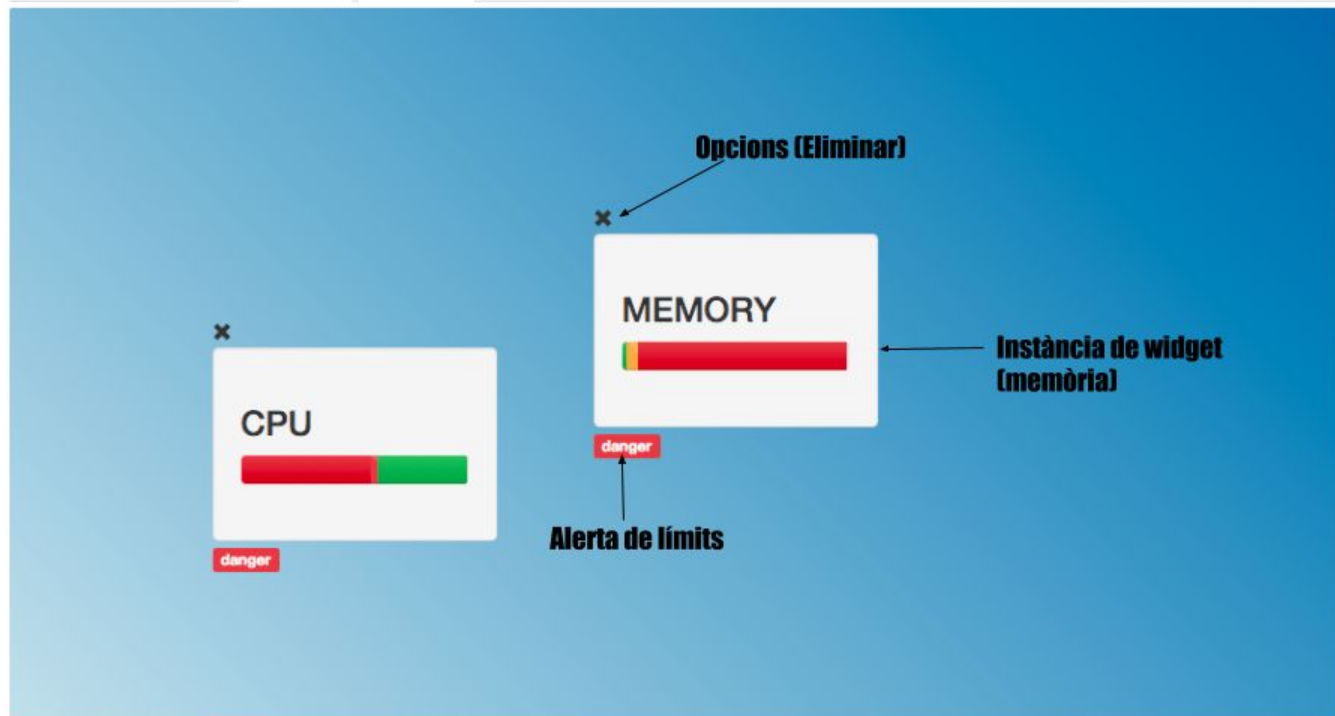


Add new dashboard

realtime

prueba2

Navegació entre dashboards



Sincronització sockets



Widgets base

```
▼ [{"id": "544f8b78-a1be-4ef2-b605-1cf3cfe03616", idWidget: "1",...},...]  
▶ 0: {id: "544f8b78-a1be-4ef2-b605-1cf3cfe03616", idWidget: "1",...}  
▶ 1: {id: "74af428b-7574-4465-8d8f-be9698ba0df1", idWidget: "3",...}  
▶ 2: {id: "96ac6211-c83b-4a4e-a0f6-00936786628b", idWidget: "2",...}  
▼ 3: {id: "b1714ad8-546d-4959-b074-e1bd7b90d9db", idWidget: "4",...}  
    id: "b1714ad8-546d-4959-b074-e1bd7b90d9db"  
    idDashboard: "3515ea14-0fac-4e24-9aec-bc7df4c38e25"  
    idWidget: "4"  
    tpl: "<h3>MEMORY</h3><div class='progress'><div class='progress-bar progress-bar-success' style="
```

```
▶ data: {free: 4, other: 10, used: 85}  
  positionX: 427  
  positionY: 94  
  refresh: 5  
  sizeX: 200  
  sizeY: 200  
▶ status: ["danger"]
```

Real time data

Widget Instance &
template

Events data

```
[vagrant@precise64:~$ redis-cli monitor  
OK  
1452640460.105622 [0 127.0.0.1:57874] "GET" "widgetInstance:datad08f617e-5c1e-47b5-8c10-3dc711c8e66a"  
1452640460.270986 [0 127.0.0.1:57874] "GET" "event:44887c0c93474101280f769da57b2b7d"  
1452640460.276613 [0 127.0.0.1:57874] "SET" "event:44887c0c93474101280f769da57b2b7d" "3db06ba0-2662-4eaa-8d1c-73ad5f2eed15:d08f617e-5c1e-47b5-8c10-3dc711c8e66a:300"  
1452640460.291911 [0 127.0.0.1:57874] "EXPIRE" "event:44887c0c93474101280f769da57b2b7d" "299"  
1452640460.296840 [0 127.0.0.1:57874] "PUBLISH" "event:generatedata" "3db06ba0-2662-4eaa-8d1c-73ad5f2eed15:d08f617e-5c1e-47b5-8c10-3dc711c8e66a:300"
```

Exemple en viu

<http://www.dashboarduoc.net>

(*)

(*) Nota: Cal actualitzar el fitxer /etc/hosts
188.166.45.150 www.dashboarduoc.net

Avantatges del projecte

Actualitzar les dades en temps real, permet reaccionar abans a possibles problemes i reduir costos.

Té un sistema d'alertes quan una mètrica no és correcte

Fàcilment ampliable i customitzable (Projecte obert)

Les parts mes crítiques es poden escalar al estar desacoplat

Conclusions

Desenvolupar el projecte es l'objectiu, però no a qualsevol preu. **L'objectiu real seria maximitzar la mantenibilitat i disminuir el cost de modificació sense comprometre ni el cost i ni l'abast.**

La qualitat no es crítica amb la consecució del èxit d'un projecte.

El dashboard en temps real por ser útil per estalviar pèrdues en les empreses

El projecte te molt marge de millora i ampliació, les peces que el componen permeten alta mantenibilitat i el cost de modificació o ampliació es baix.

Preguntes ?

Moltes gràcies