

# FREESTATION

PROYECTO DE FIN DE CARRERA

PLATAFORMA PARA EL  
DESARROLLO DE  
SISTEMAS DE DISTRIBUCIÓN DE  
SOFTWARE LIBRE EN  
PUNTOS DE INFORMACIÓN



14 de Septiembre de 2012

Autor: Ángel Guzmán Maeso  
Director: Carlos González Morcillo





# TABLA DE CONTENIDOS

## PROYECTO DE FIN DE CARRERA

### PRIMERA PARTE

- *Introducción*
- *Objetivos*



# TABLA DE CONTENIDOS

## PROYECTO DE FIN DE CARRERA

### PRIMERA PARTE

- *Introducción*
- *Objetivos*

### SEGUNDA PARTE

- *Estado del Arte*
- *Arquitectura del sistema*
- *Demo*



# TABLA DE CONTENIDOS

## PROYECTO DE FIN DE CARRERA

### PRIMERA PARTE

- *Introducción*
- *Objetivos*

### SEGUNDA PARTE

- *Estado del Arte*
- *Arquitectura del sistema*
- *Demo*

### TERCERA PARTE

- *Metodología y Costes*
- *Conclusiones y trabajo futuro*

# DESCRIPCIÓN

¿Qué es **FreeStation**?



- *Sistema de distribución software*

# DESCRIPCIÓN

¿Qué es **FreeStation**?



- *Sistema de distribución software*
- *Plataforma de desarrollo*

# DESCRIPCIÓN

¿Qué es **FreeStation**?



- *Sistema de distribución software*
- *Plataforma de desarrollo*
- *Creación Puntos de interés*



# DEFINICIÓN POI

POI: **P**oint **O**f **I**nterest

- *Sistema SW y HW de acceso público*





# DEFINICIÓN POI

POI: **P**oint **O**f **I**nterest

- *Sistema SW y HW de acceso público*
- *Orientado a servicios*



# DEFINICIÓN POI

POI: **P**oint **O**f **I**nterest

- *Sistema SW y HW de acceso público*
- *Orientado a servicios*
- *Proporciona información*



# DEFINICIÓN POI

POI: **P**oint **O**f **I**nterest

- *Sistema SW y HW de acceso público*
- *Orientado a servicios*
- *Proporciona información*
- *Interactivo*

# GRADO DE IMPLANTACIÓN

## Importancia

- Difusión contenidos

# GRADO DE IMPLANTACIÓN

## Importancia

- Difusión contenidos
- Interacción del entorno

# GRADO DE IMPLANTACIÓN

## Importancia

- Difusión contenidos
- Interacción del entorno
- Ahorro costes

# GRADO DE IMPLANTACIÓN

## Importancia

- Difusión contenidos
- Interacción del entorno
- Ahorro costes
- Personalización

# GRADO DE IMPLANTACIÓN

## Importancia

## Mercado y Sociedad

- Difusión contenidos
  - Interacción del entorno
  - Ahorro costes
  - Personalización
- Gran facturación (~400M \$ en 2011)



# GRADO DE IMPLANTACIÓN

## Importancia

- Difusión contenidos
- Interacción del entorno
- Ahorro costes
- Personalización

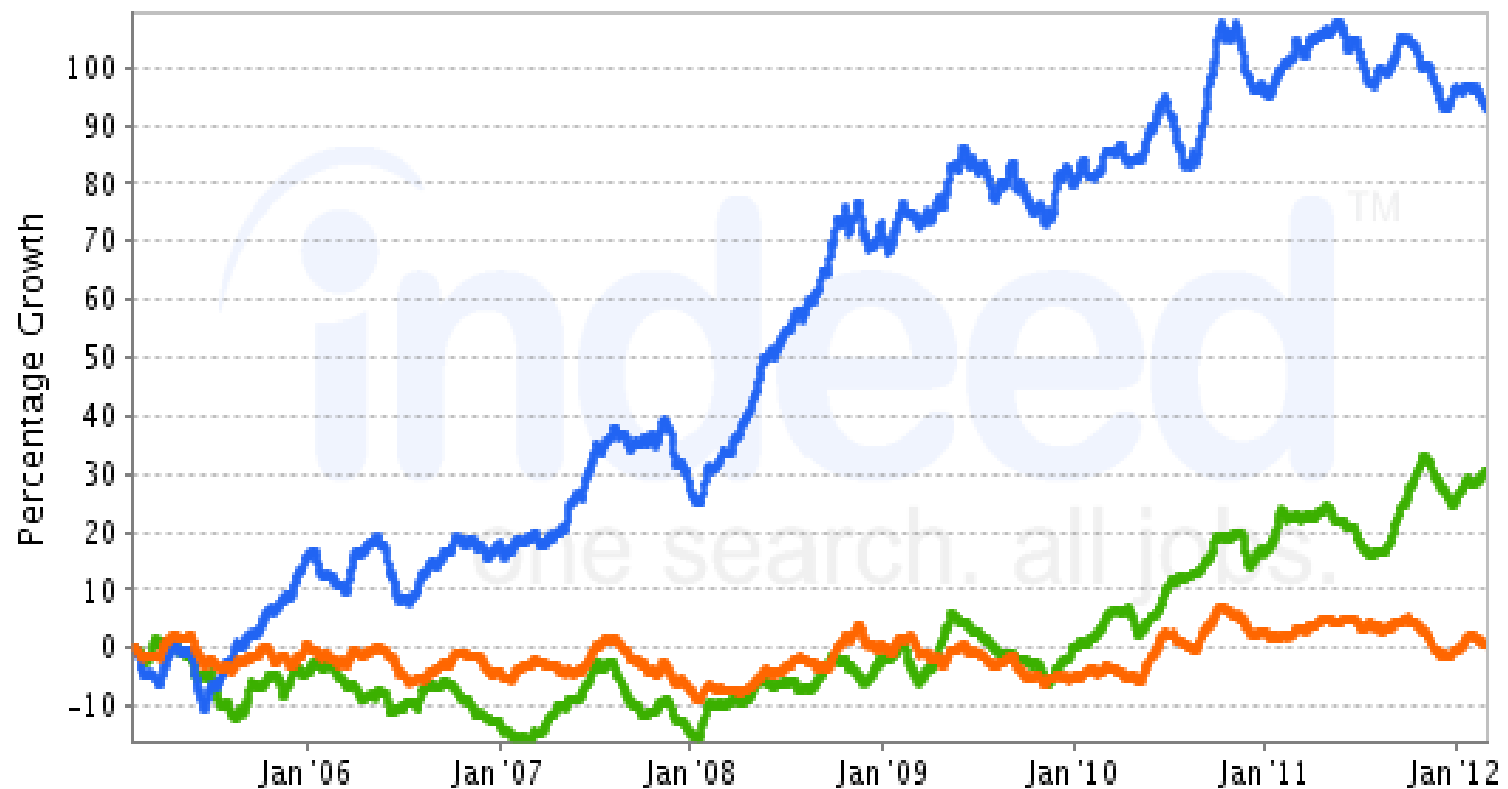
## Mercado y Sociedad

- Gran facturación (~400M \$ en 2011)
- Tecnología en auge

# AUMENTO SW LIBRE

Job Trends from Indeed.com

— windows — linux — mac



# EJEMPLO DE IMPLANTACIÓN

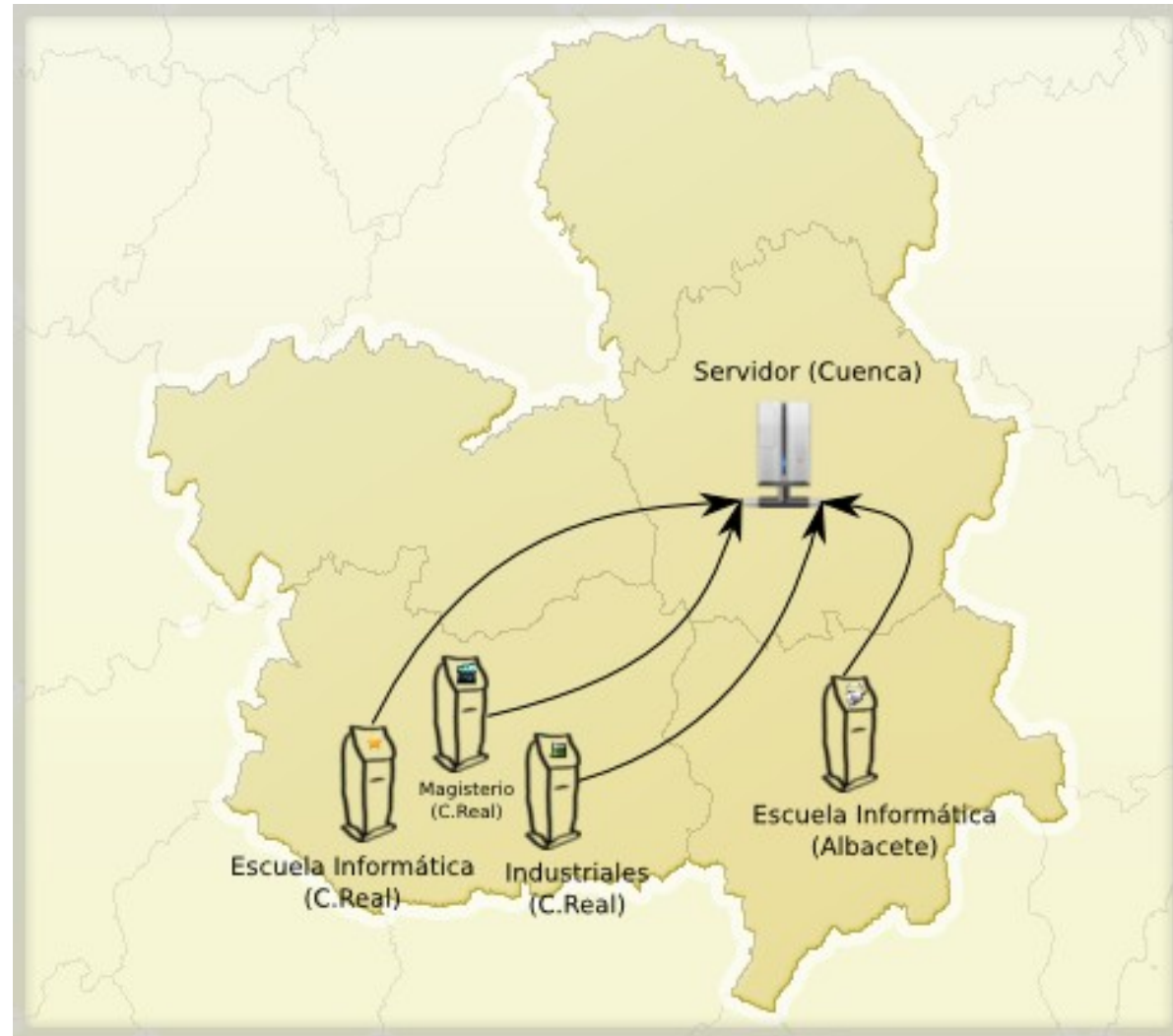
## RED UCLM

- Clientes distribuidos

# EJEMPLO DE IMPLANTACIÓN

## RED UCLM

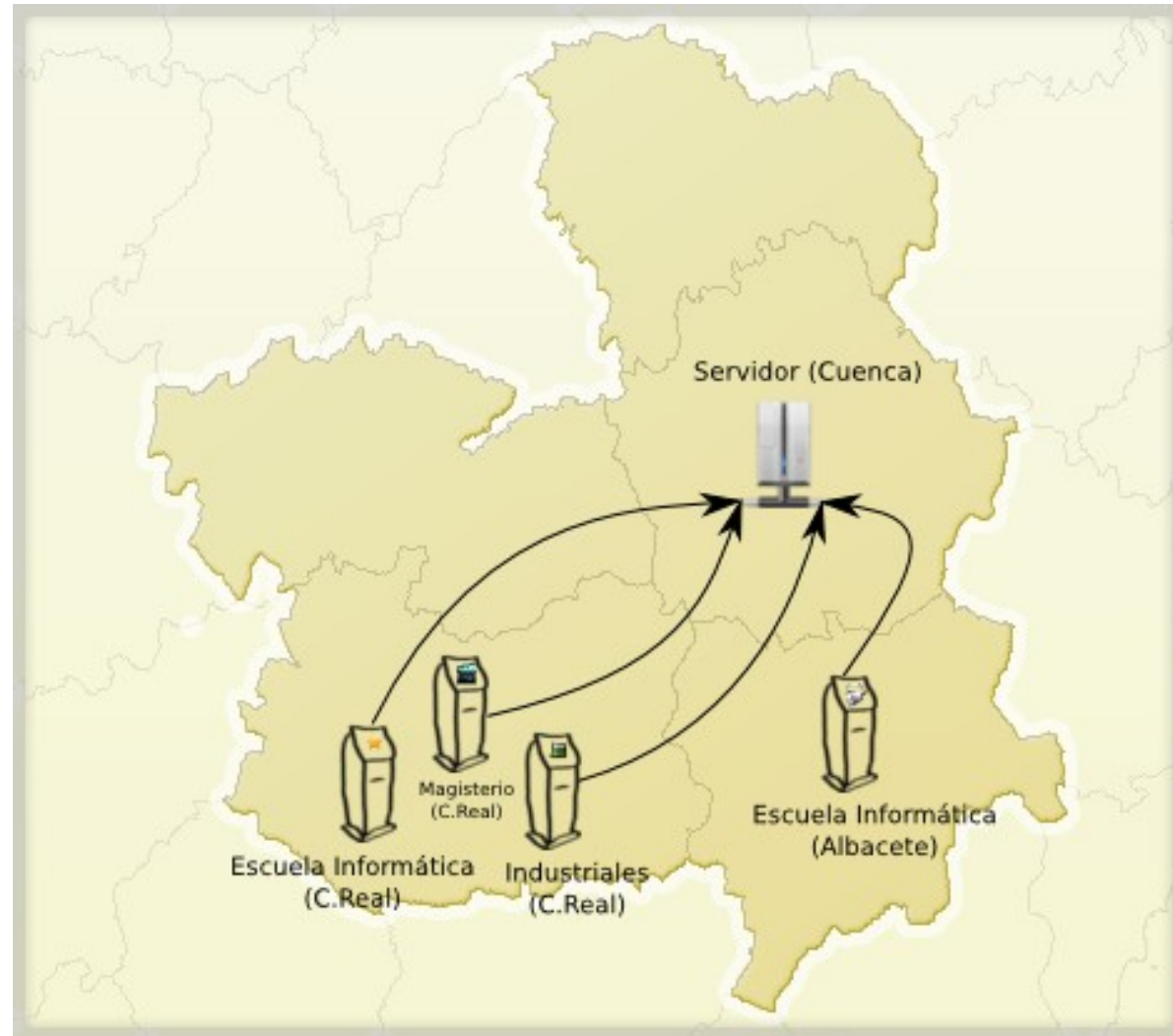
- Clientes distribuidos



# EJEMPLO DE IMPLANTACIÓN

## RED UCLM

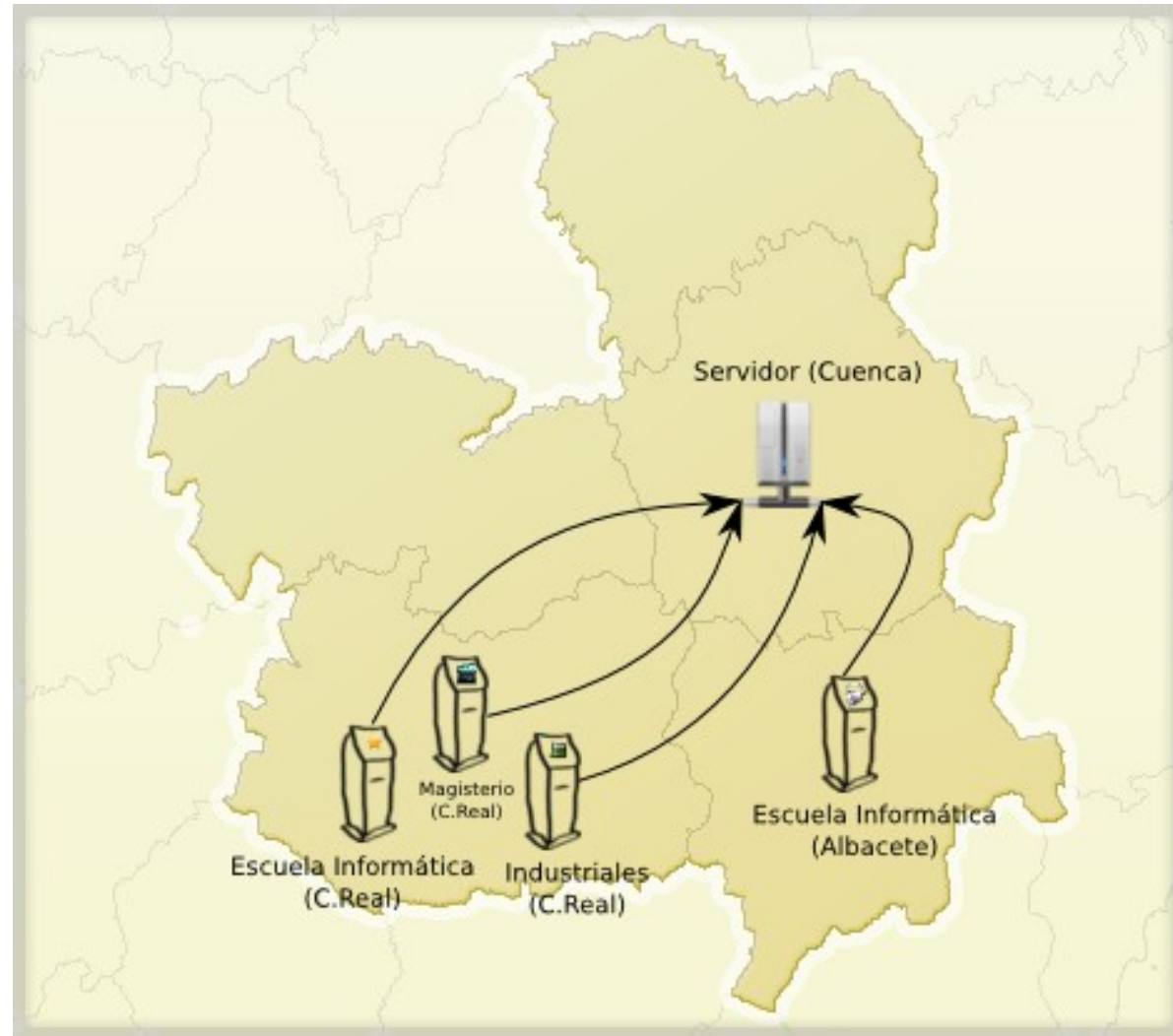
- Clientes distribuidos
- Nodo centralizado



## EJEMPLO DE IMPLANTACIÓN

### RED UCLM

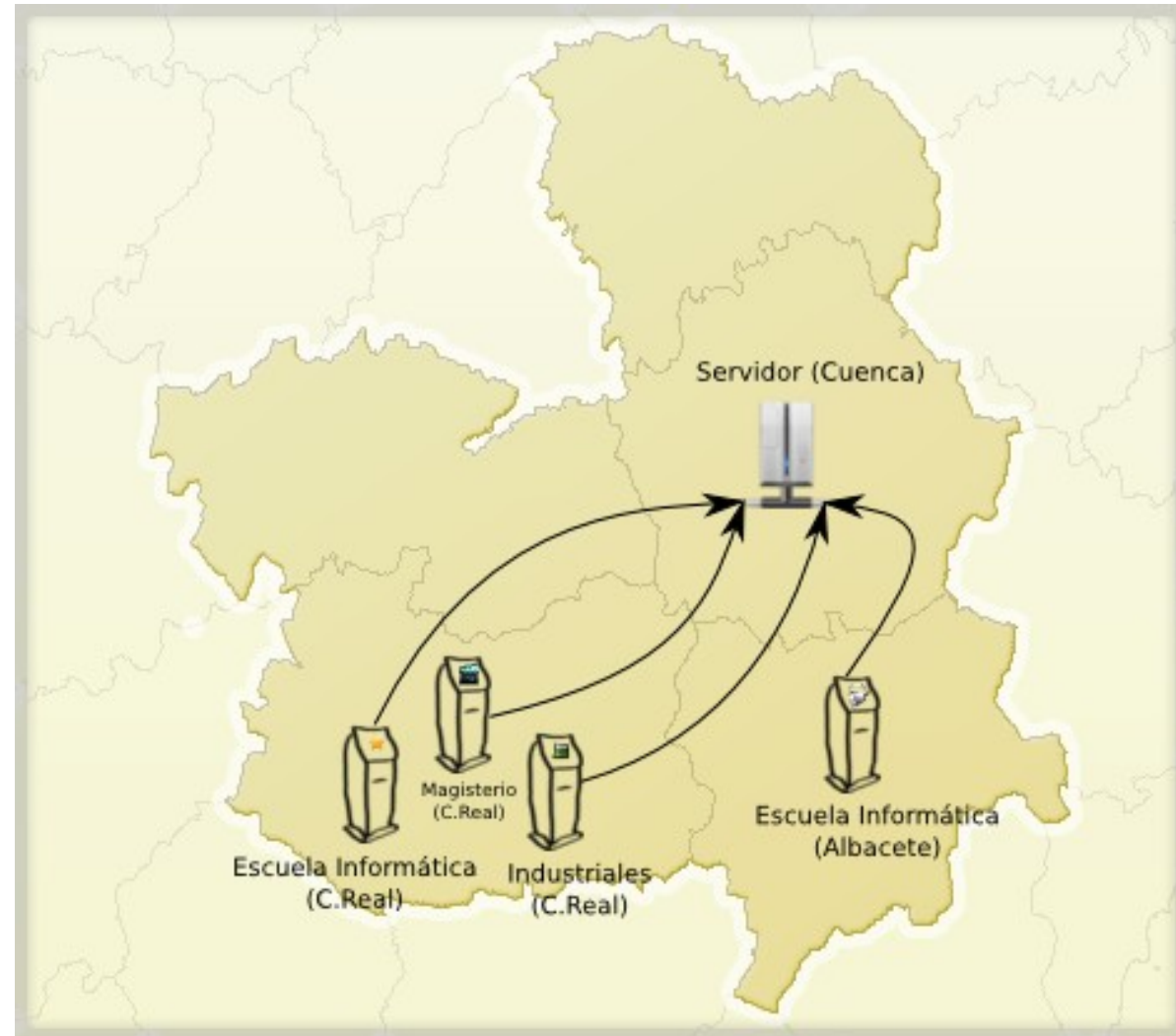
- Clientes distribuidos
- Nodo centralizado
- Diferentes facultades



## EJEMPLO DE IMPLANTACIÓN

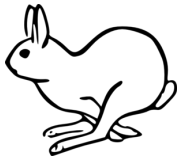
### RED UCLM

- Clientes distribuidos
- Nodo centralizado
- Diferentes facultades
- Personalización POI



# DISTRIBUCIÓN POI

## PRINCIPIOS



Rápido



Escalable



Robusto



# DISTRIBUCIÓN POI

## PRINCIPIOS



Rápido



Escalable



Robusto

## REQUISITOS



Ancho de banda cliente/servidor



Uso intensivo de recursos



Costes equipos (servidores)

# DISTRIBUCIÓN POI

## PRINCIPIOS



Rápido



Escalable



Robusto

## REQUISITOS



Ancho de banda cliente/servidor



Uso intensivo de recursos



Costes equipos (servidores)

## PROBLEMÁTICA



1. *Homogeneización*



2. *Actualizaciones*



3. *Infraestructura*



4. *Parametrización*



5. *Multiplataforma*



# OBJETIVOS

## OBJETIVO PRINCIPAL

- Crear *plataforma* para construir, *desarrollar* y desplegar sistemas de *distribución* de software libre *modificable* a necesidades de organizaciones



# OBJETIVOS

## OBJETIVO PRINCIPAL

- Crear *plataforma* para construir, *desarrollar* y desplegar sistemas de *distribución* de software libre *modificable* a necesidades de organizaciones

## SUBOBJETIVOS

- *Adaptabilidad*



# OBJETIVOS

## OBJETIVO PRINCIPAL

- Crear *plataforma* para construir, *desarrollar* y desplegar sistemas de *distribución* de software libre *modificable* a necesidades de organizaciones

## SUBOBJETIVOS

- *Adaptabilidad*
- *Robustez*



# OBJETIVOS

## OBJETIVO PRINCIPAL

- Crear *plataforma* para construir, *desarrollar* y desplegar sistemas de *distribución* de software libre *modificable* a necesidades de organizaciones

## SUBOBJETIVOS

- *Adaptabilidad*
- *Robustez*
- *Heterogeneidad*



# OBJETIVOS

## OBJETIVO PRINCIPAL

- Crear *plataforma* para construir, *desarrollar* y desplegar sistemas de *distribución* de software libre *modificable* a necesidades de organizaciones

## SUBOBJETIVOS

- *Adaptabilidad*
- *Robustez*
- *Heterogeneidad*
- *Distribuido*



# OBJETIVOS

## OBJETIVO PRINCIPAL

- Crear *plataforma* para construir, *desarrollar* y desplegar sistemas de *distribución* de software libre *modificable* a necesidades de organizaciones

## SUBOBJETIVOS

- *Adaptabilidad*
- *Robustez*
- *Heterogeneidad*
- *Distribuido*
- *Diseño NUI*





# OBJETIVOS

## OBJETIVO PRINCIPAL

- Crear *plataforma* para construir, *desarrollar* y desplegar sistemas de *distribución* de software libre *modificable* a necesidades de organizaciones

## SUBOBJETIVOS

- *Adaptabilidad*
- *Robustez*
- *Heterogeneidad*
- *Distribuido*
- *Diseño NUI*
- *Actualizable*



# OBJETIVOS

## OBJETIVO PRINCIPAL

- Crear *plataforma* para construir, *desarrollar* y desplegar sistemas de *distribución* de software libre *modificable* a necesidades de organizaciones

## SUBOBJETIVOS

- *Adaptabilidad*
- *Robustez*
- *Heterogeneidad*
- *Distribuido*
- *Diseño NUI*
- *Actualizable*
- *Libre*

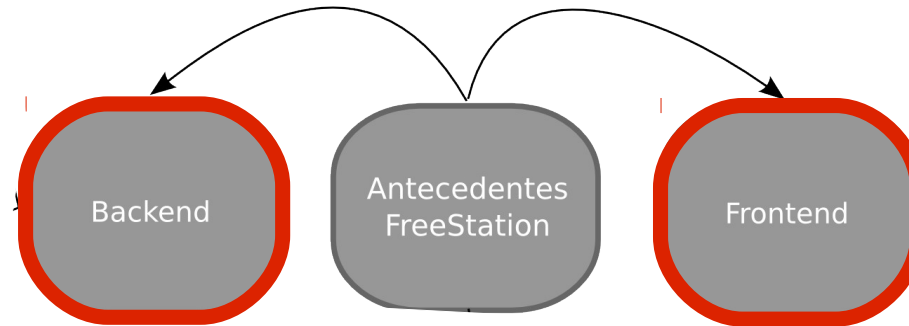


# ESTADO DEL ARTE

Antecedentes  
FreeStation

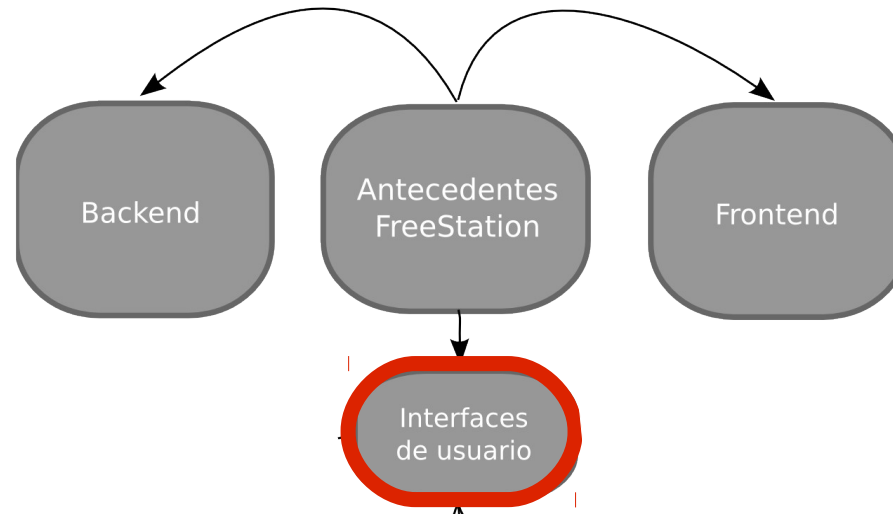


# ESTADO DEL ARTE



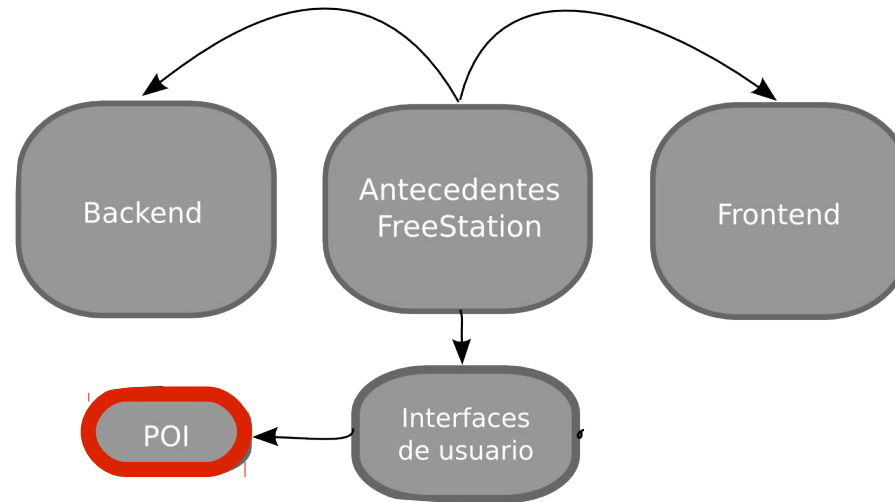


# ESTADO DEL ARTE



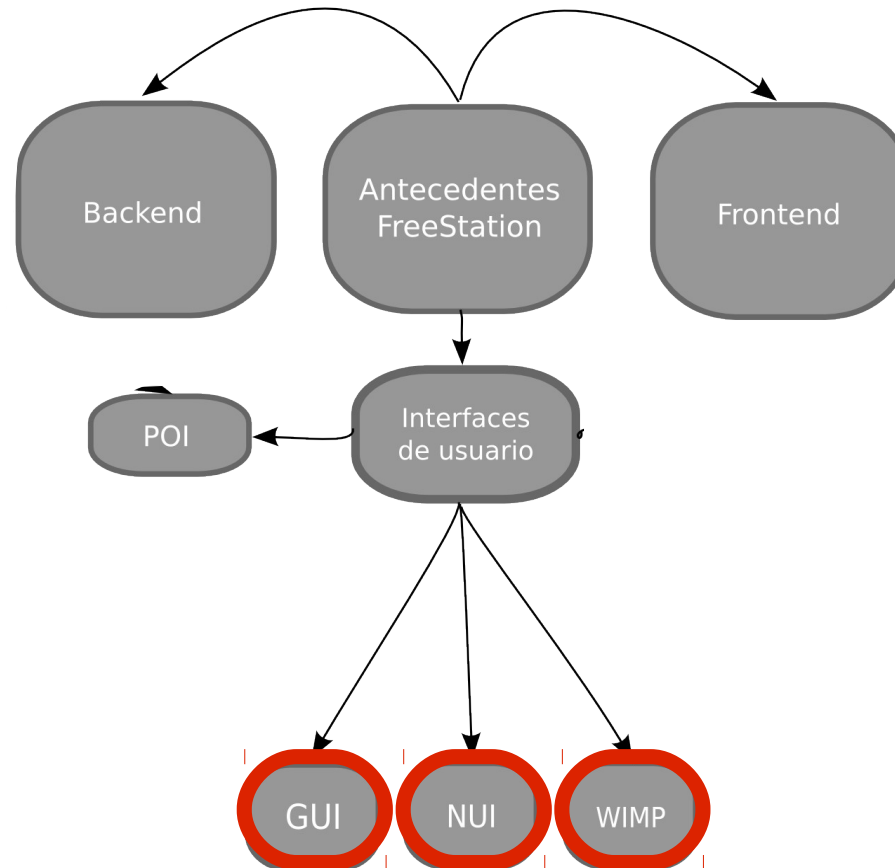


# ESTADO DEL ARTE



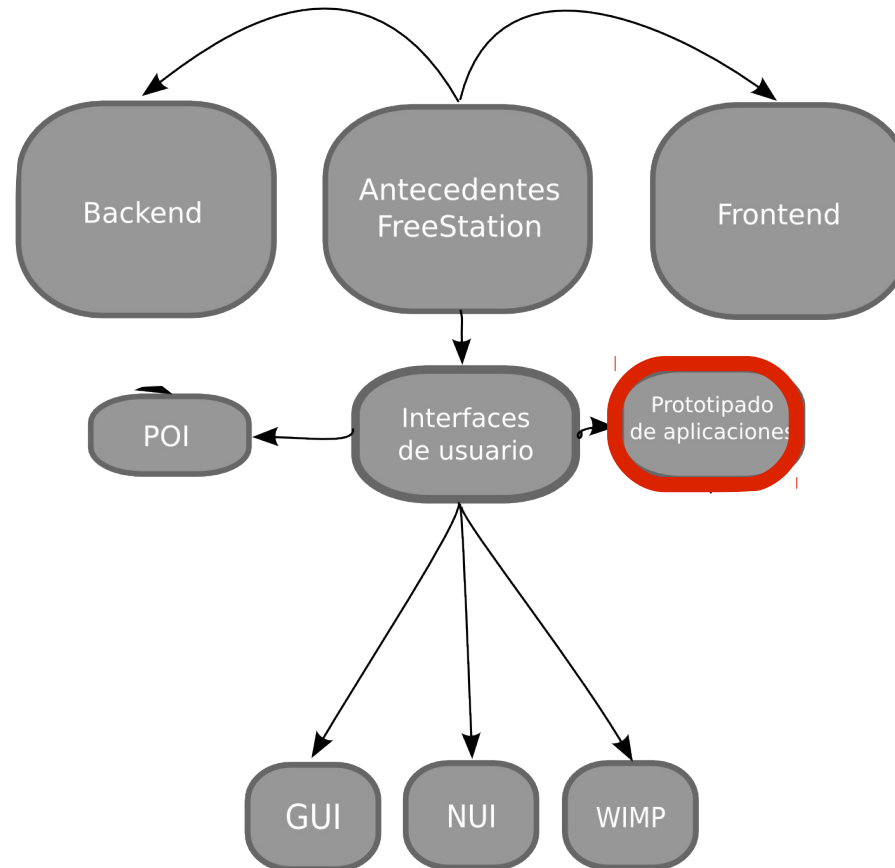


# ESTADO DEL ARTE





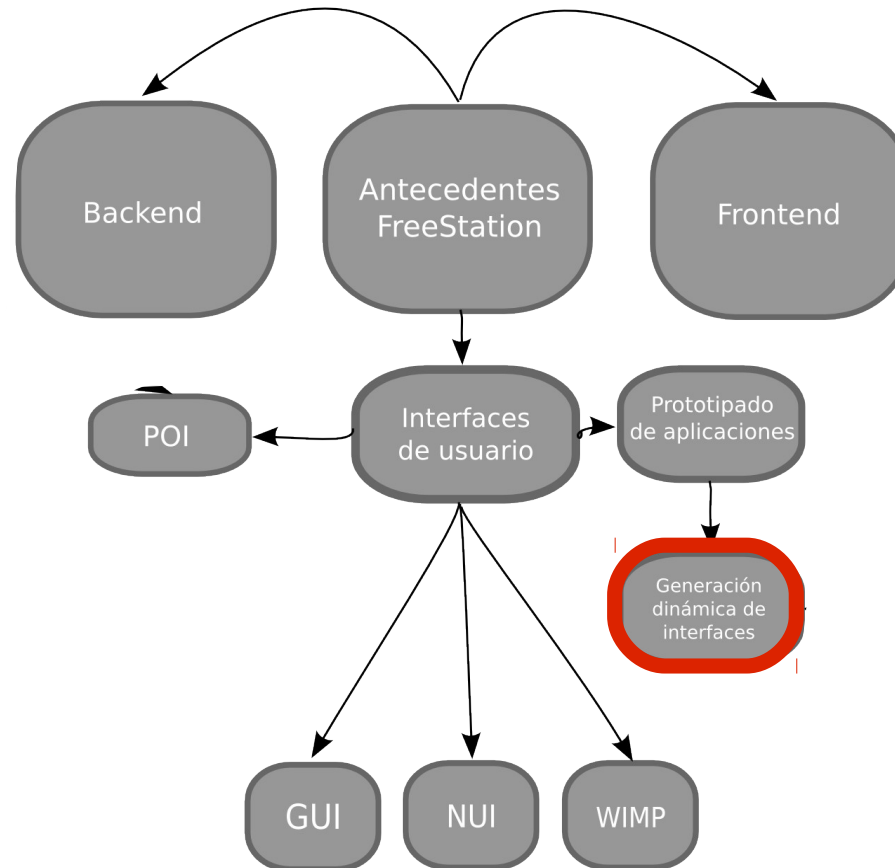
# ESTADO DEL ARTE





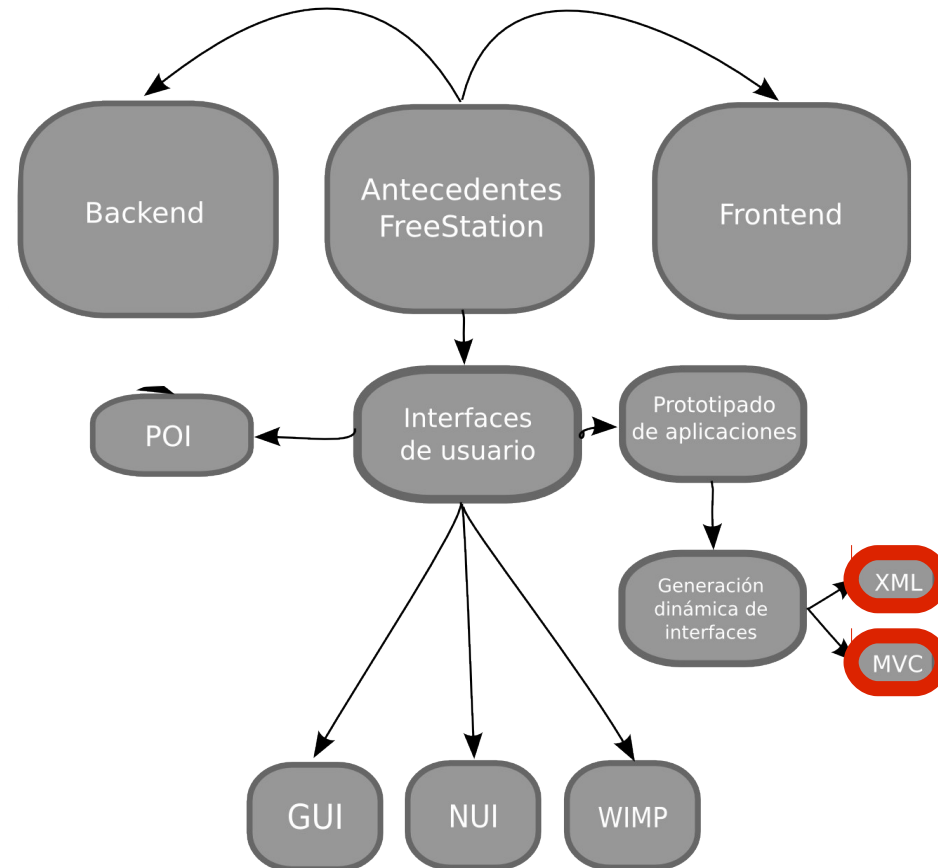


# ESTADO DEL ARTE



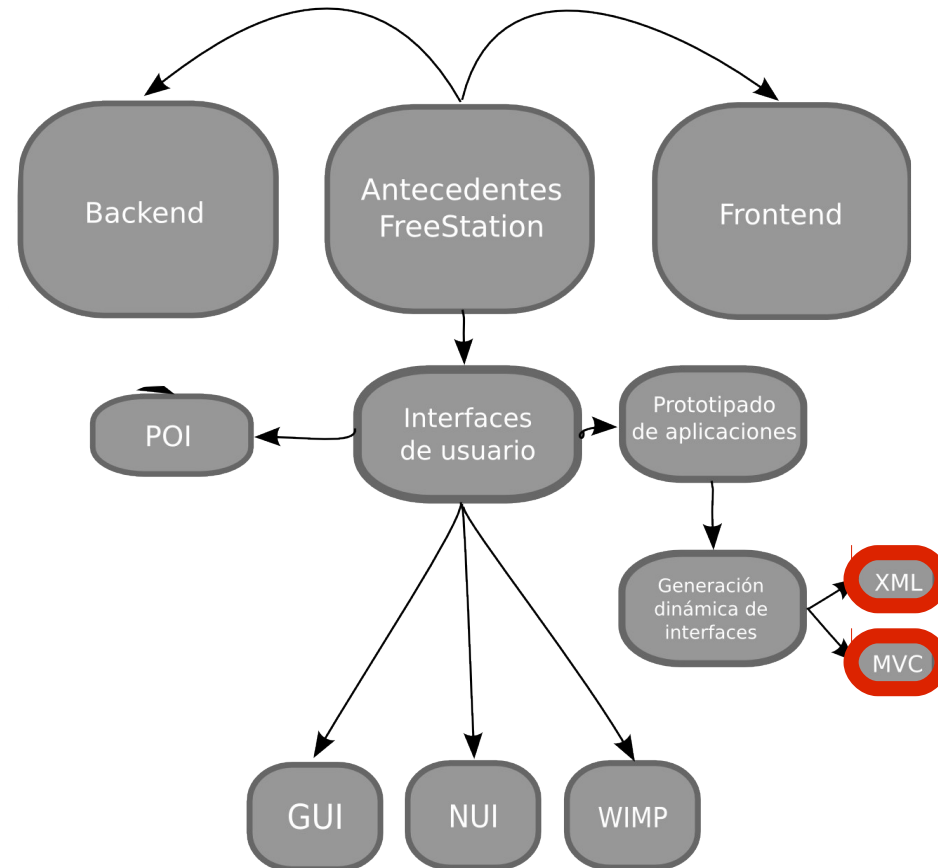


# ESTADO DEL ARTE



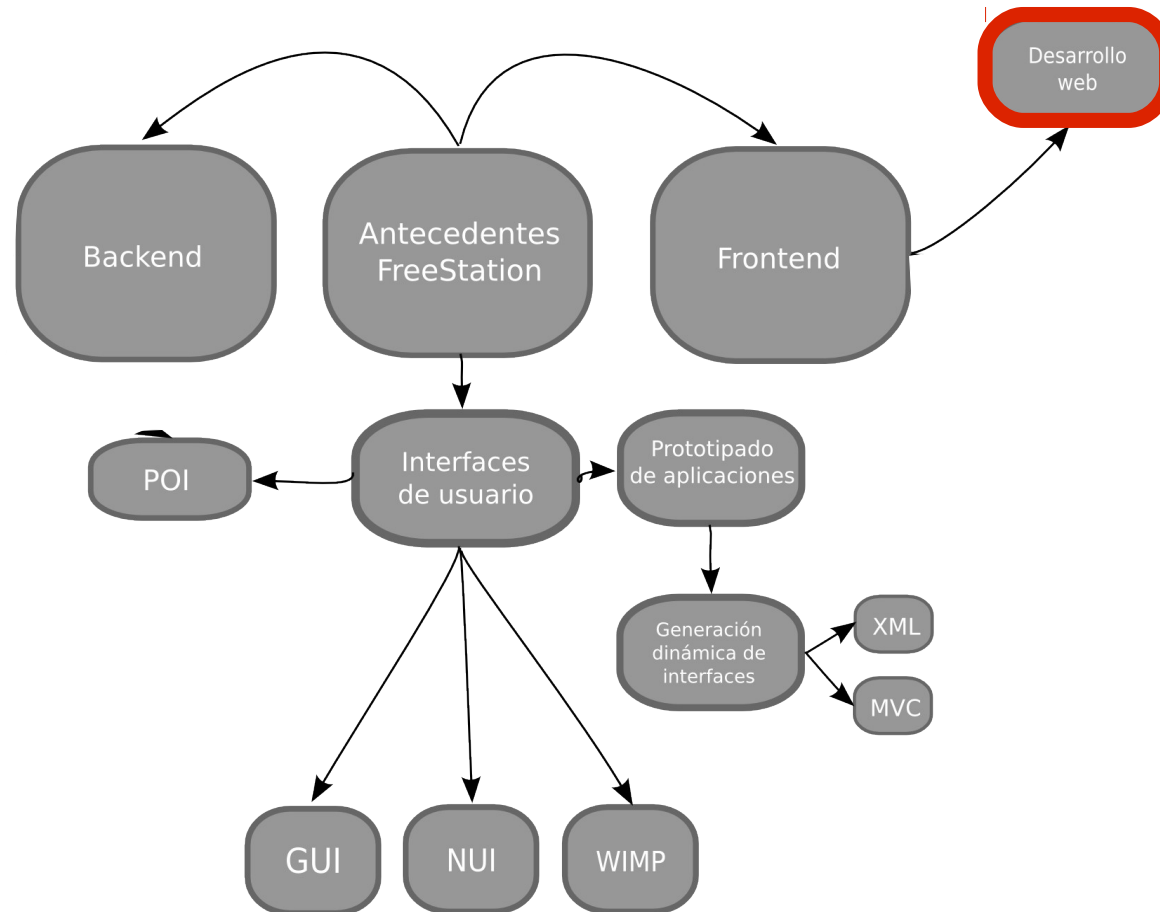


# ESTADO DEL ARTE



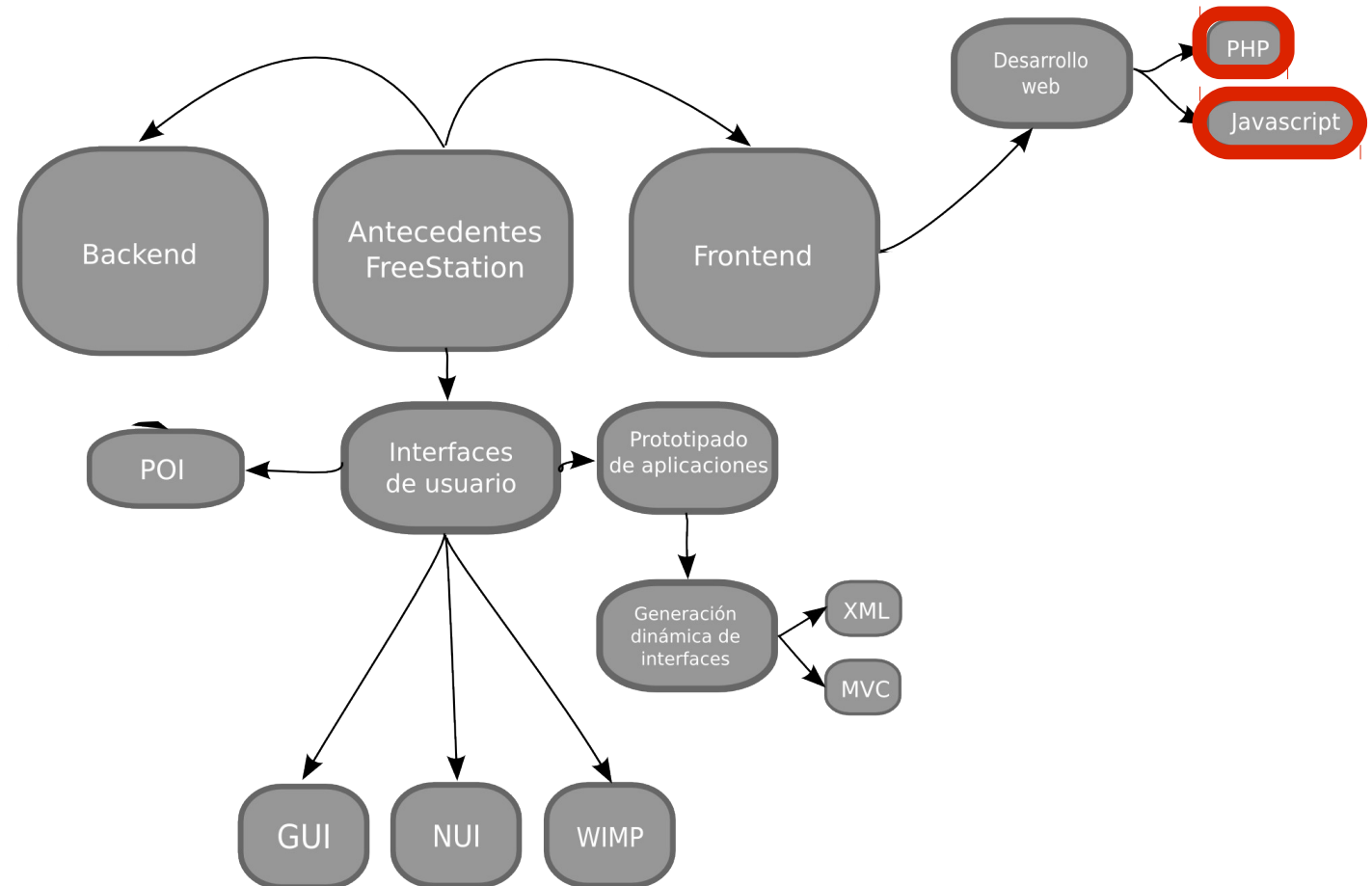


# ESTADO DEL ARTE



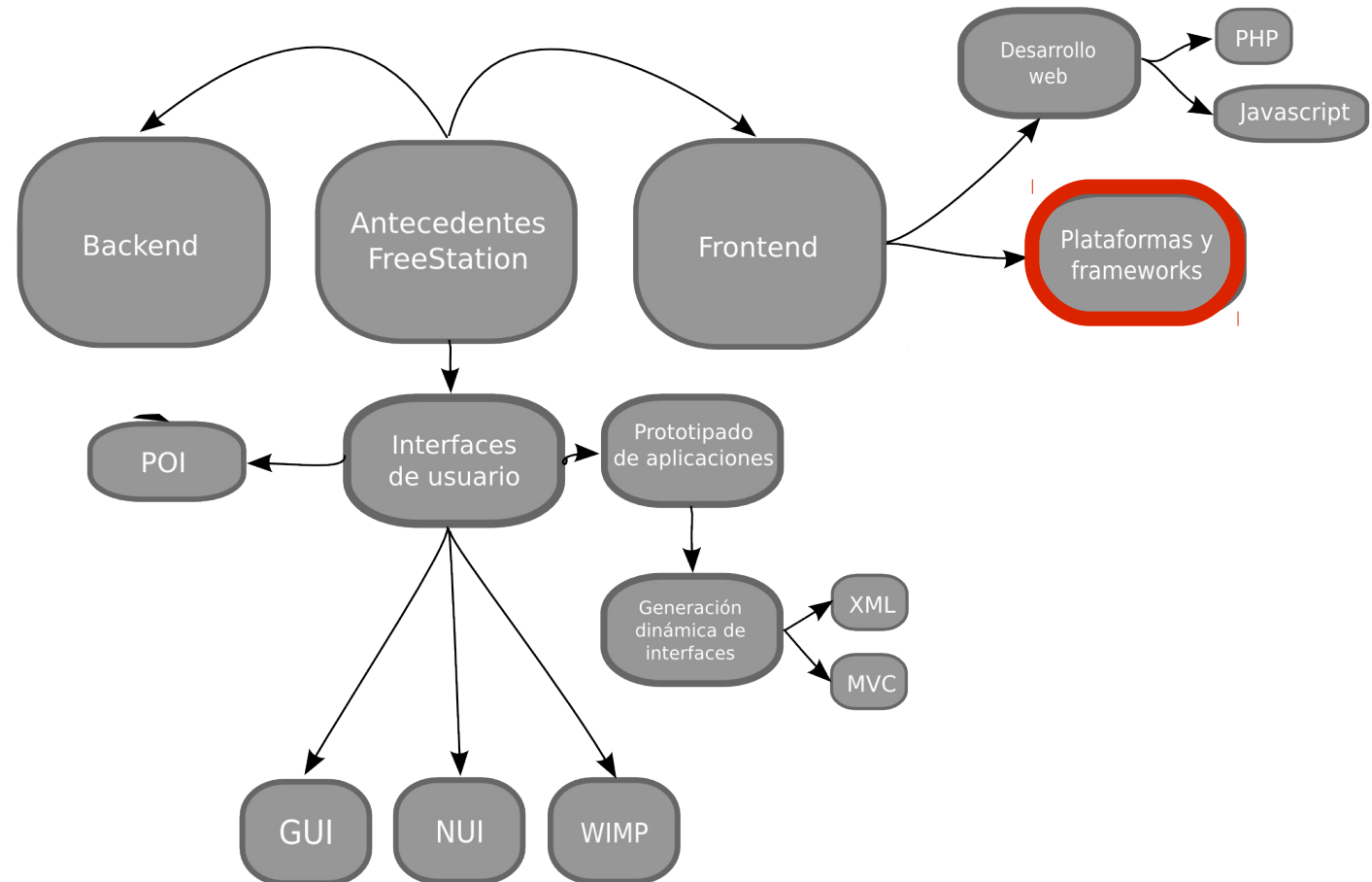


# ESTADO DEL ARTE



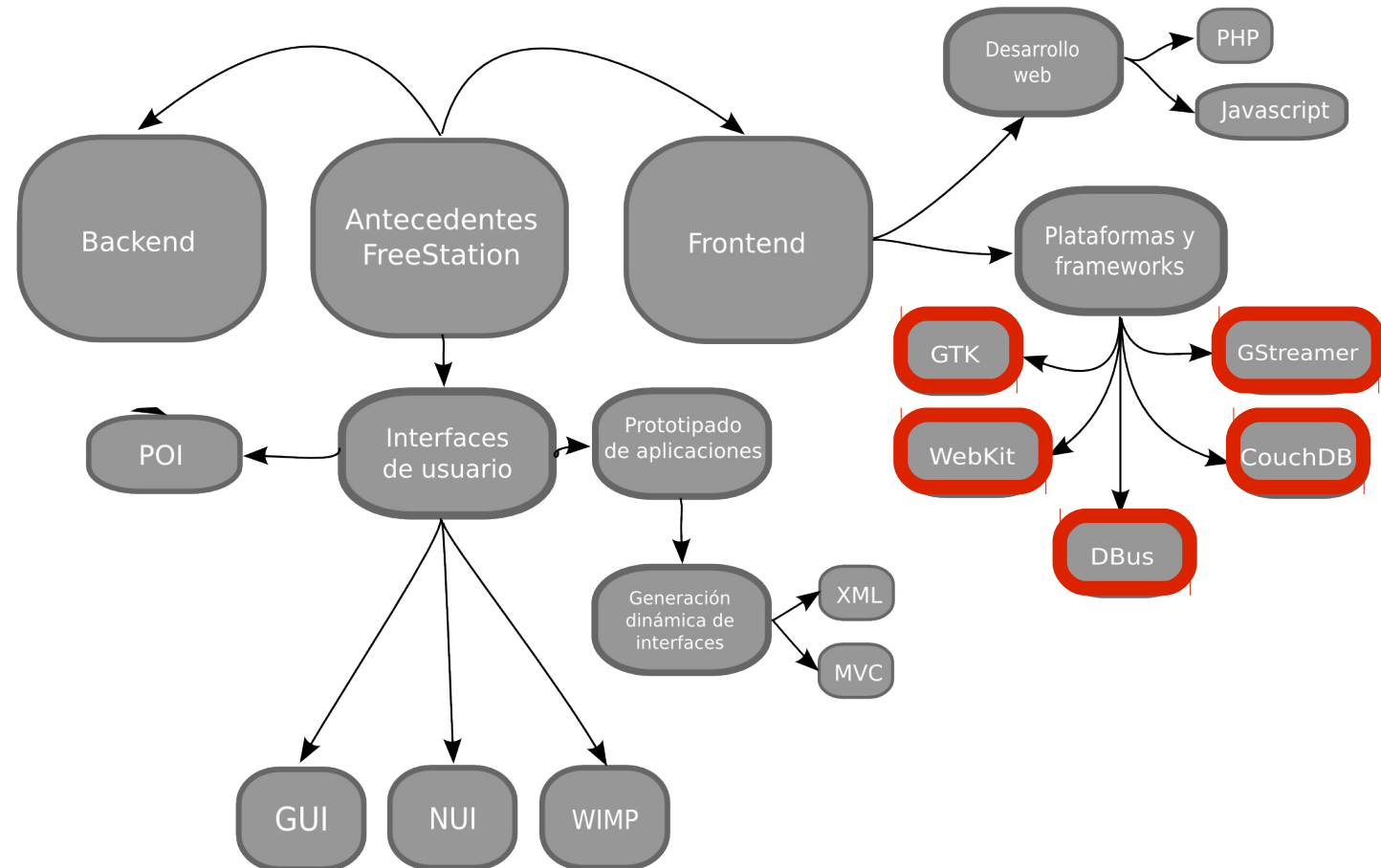


# ESTADO DEL ARTE



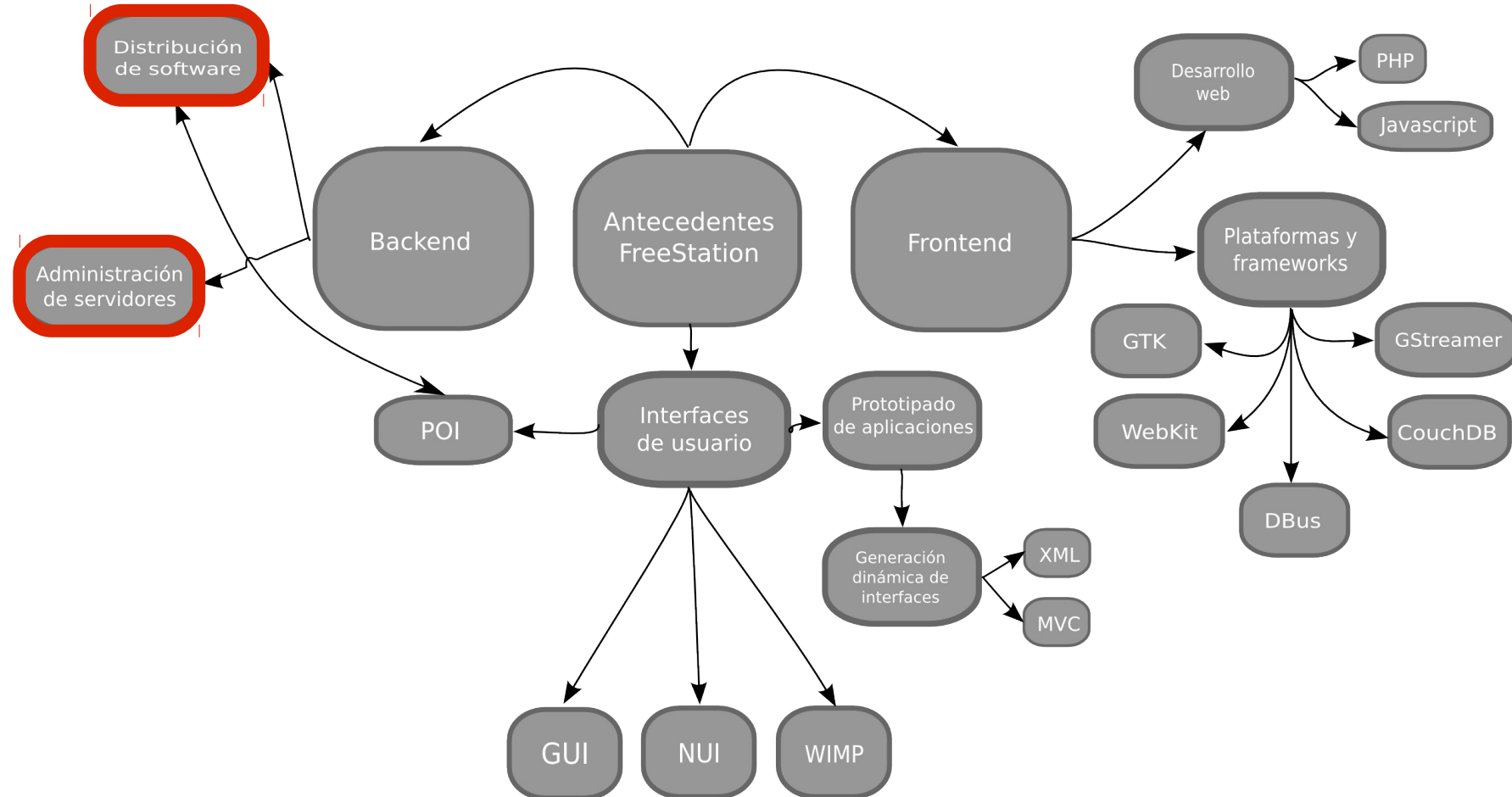


# ESTADO DEL ARTE





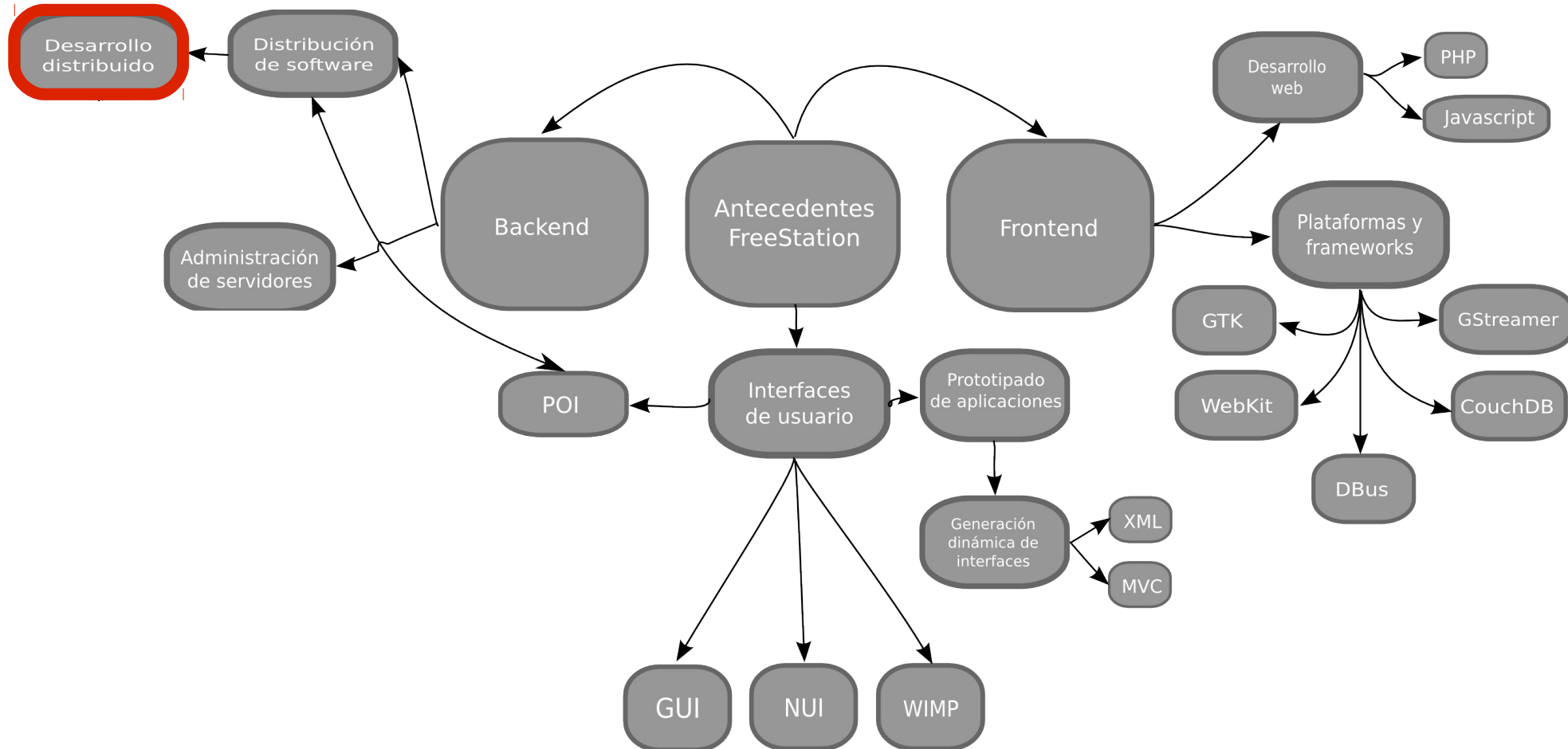
# ESTADO DEL ARTE





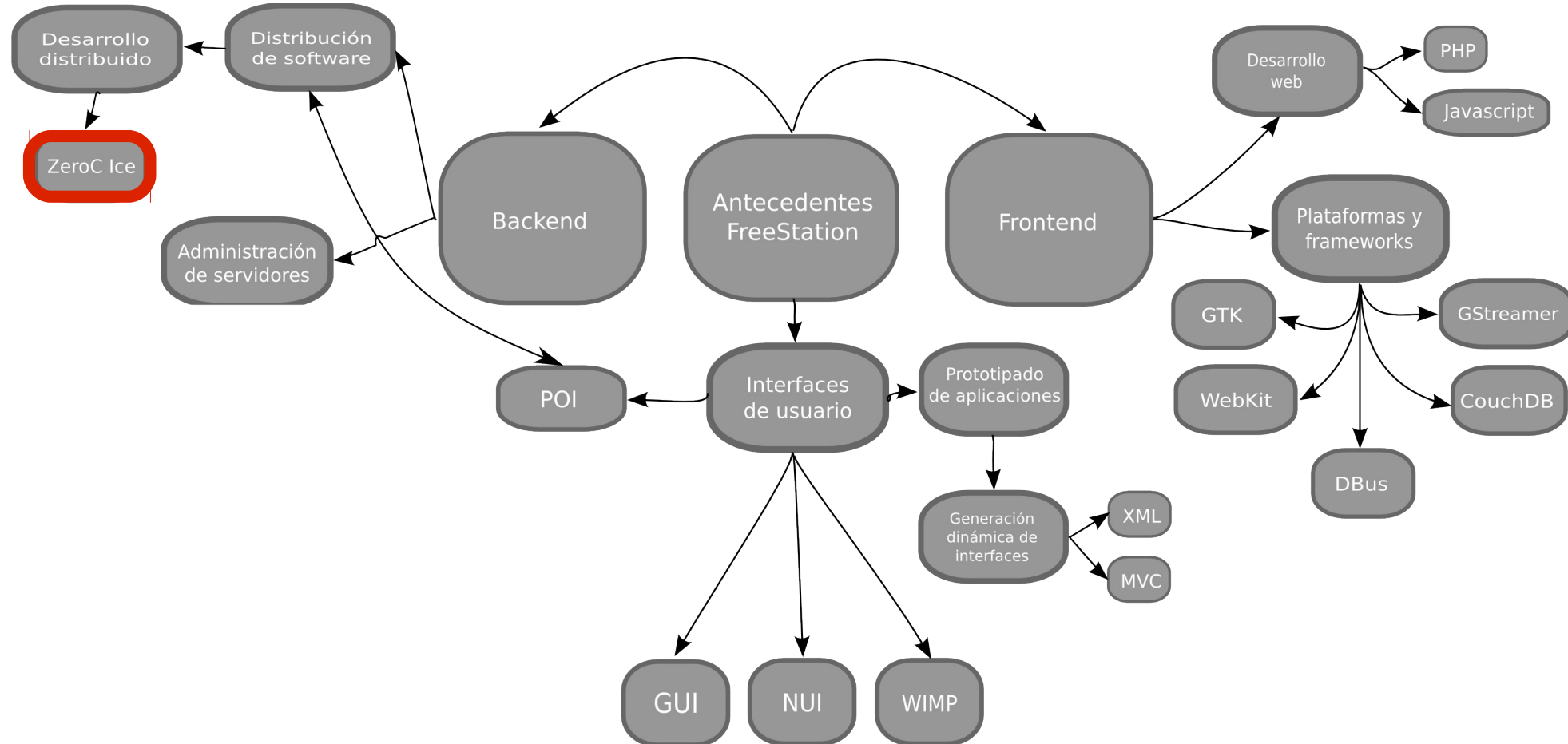


# ESTADO DEL ARTE



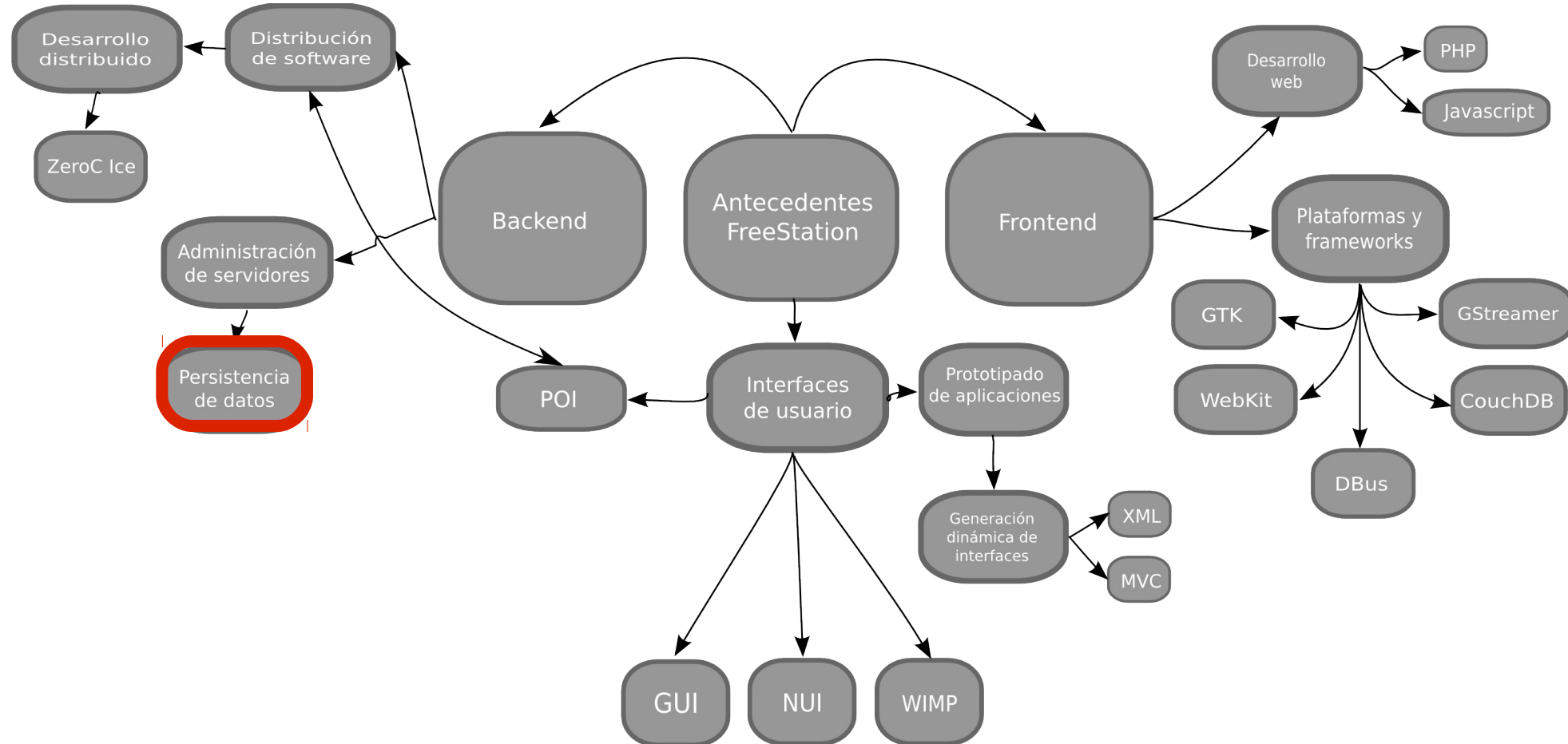


# ESTADO DEL ARTE



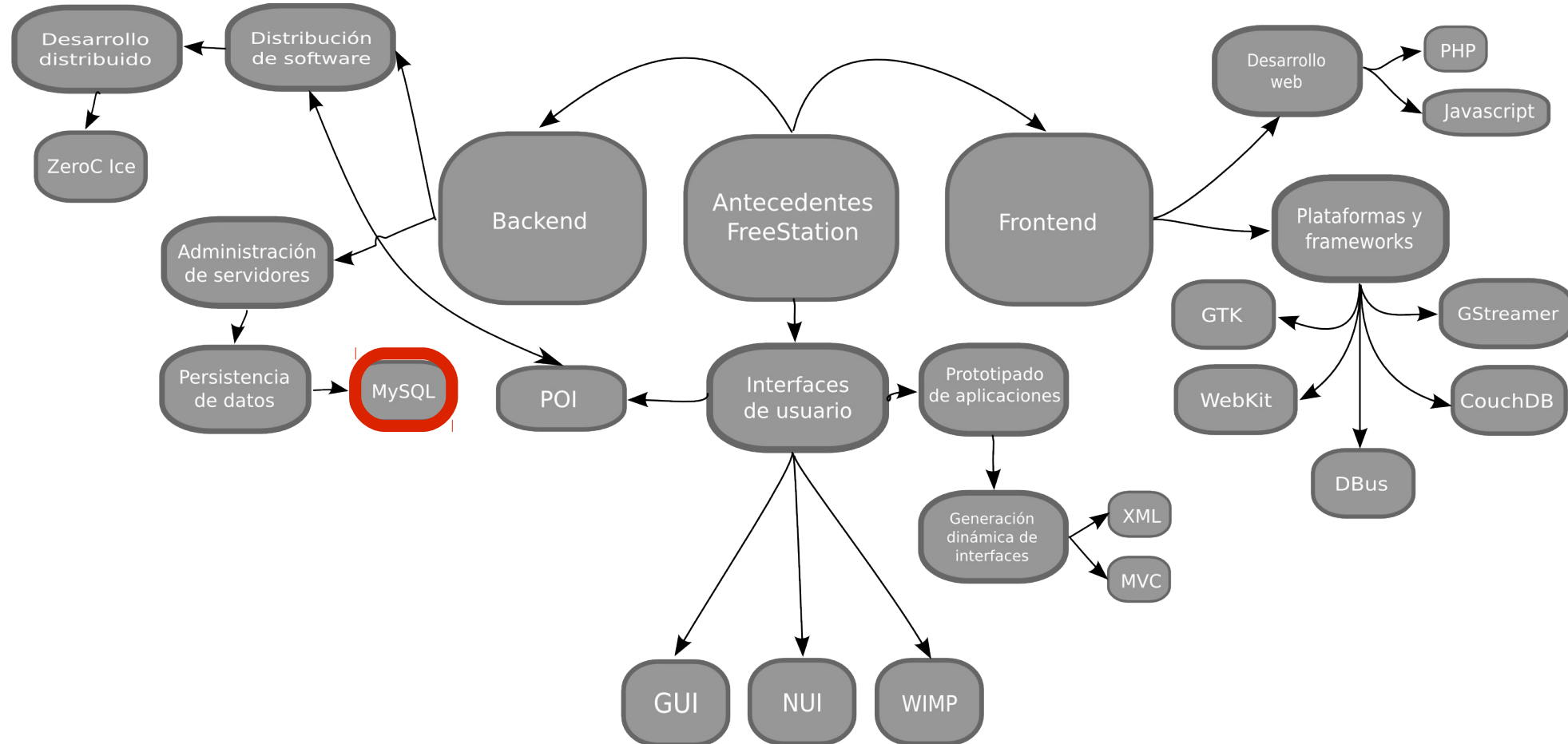


# ESTADO DEL ARTE





# ESTADO DEL ARTE





# ARQUITECTURA DEL SISTEMA

- *Descripción general*
  - *Arquitectura del cliente*
  - *Arquitectura del servidor*

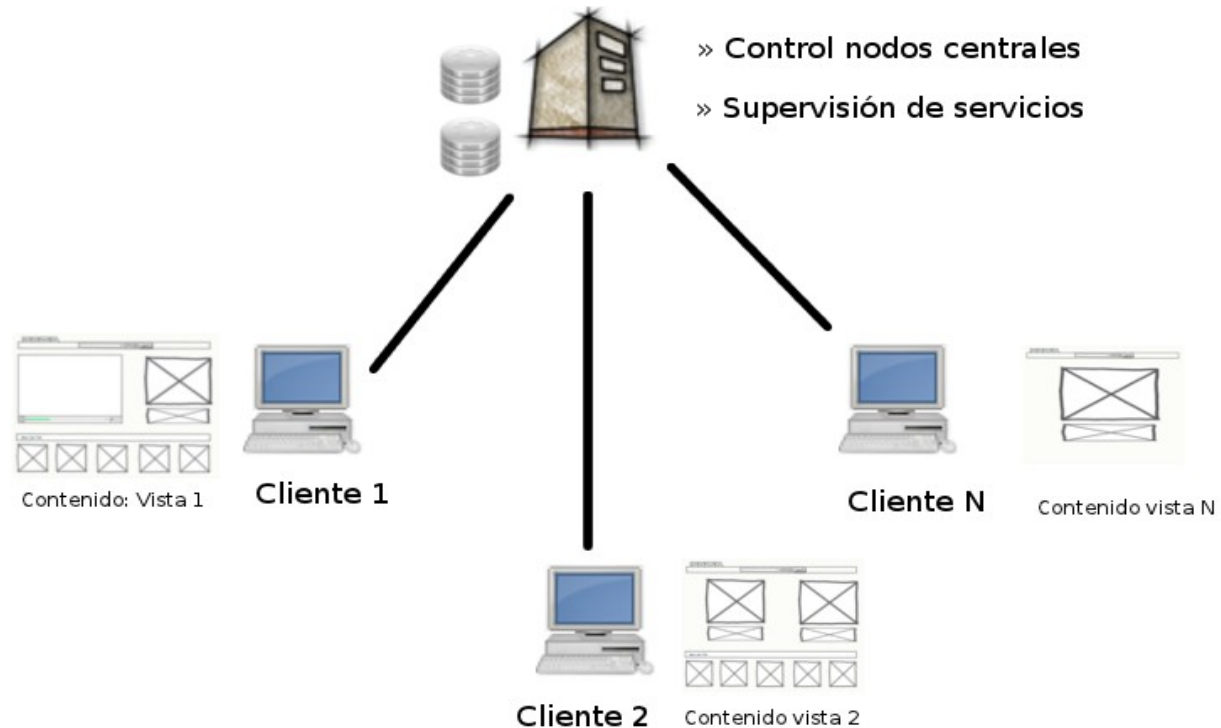


# DESCRIPCIÓN GENERAL

? Control de nodos

☺ Arq Distribuida

✓ Configuración





# ARQUITECTURA DEL SISTEMA

- *Descripción general*
  - *Arquitectura del cliente*
  - *Arquitectura del servidor*



# ARQUITECTURA DEL CLIENTE

- *Descripción general*
- *FreeStationApp*
  - *Arquitectura WatchDog*
  - *Arquitectura Lanzador*
  - *Arquitectura Widgets*
  - *Arquitectura Interfaz Gráfica*





# DESCRIPCIÓN GENERAL

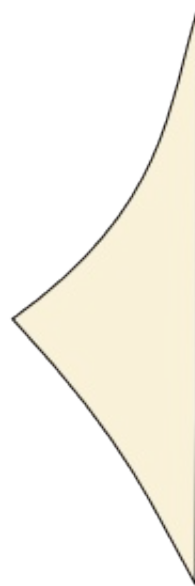
? Conexión servidor

☺ Cliente distribuido

✓ Modularización



POI



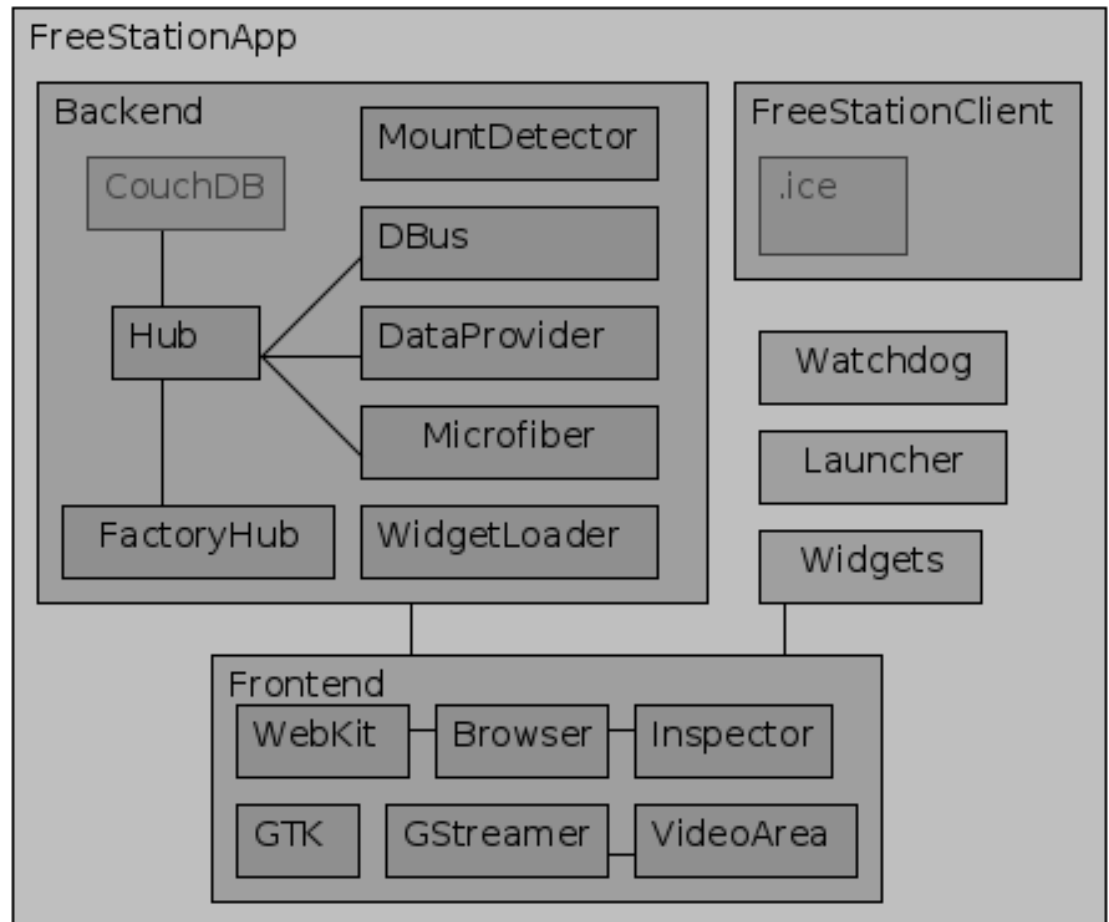


# DIAGRAMA COMPONENTES

? Estructuración cliente

☺ Backend / Frontend

✓ Reutilización / Widgets



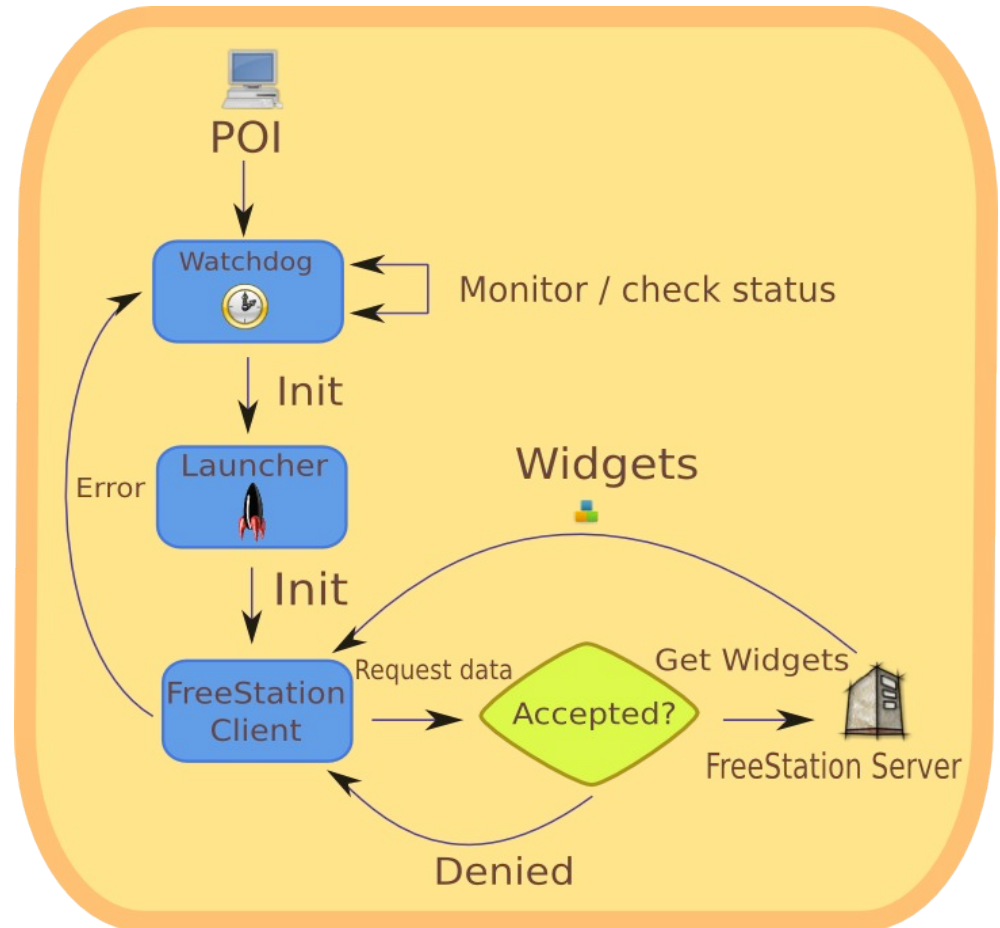


## DIAGRAMA DE FLUJO – CLIENTE

? Flujo definido

☺ Watchdog / Launcher / FreeStationClient

✓ Robusto / Adaptable / Seguro





## ARQUITECTURA WATCHDOG

*Monitorizador de cliente (FreeStationClient)*

*? Tolerancia a errores*

- ☺ *Hilos monitorizables*
- ☺ *Jerarquía de errores*
- ☺ *Reinicio automático*

*✓ Recuperación en caso de fallo*

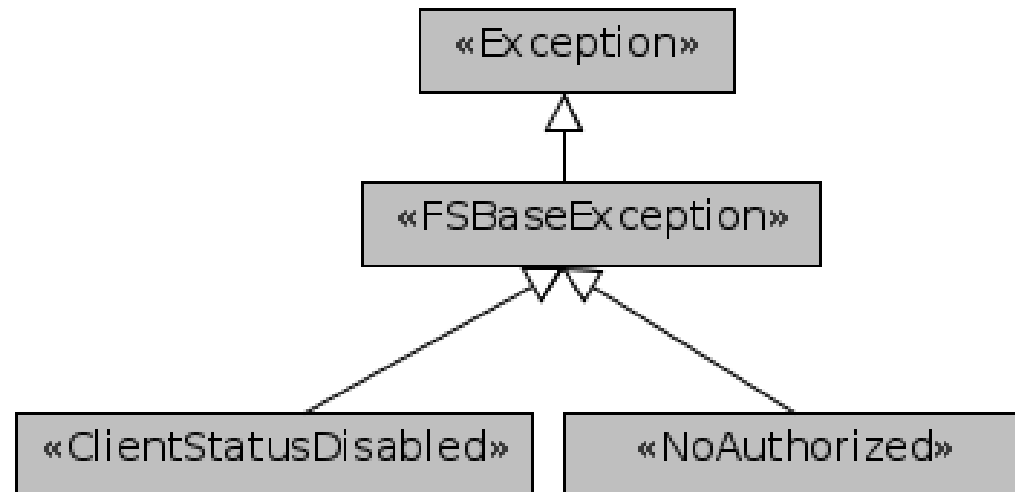


# JERARQUIA DE EXCEPCIONES

? Estructura excepciones

☺ Especialización excepciones

✓ Personalización errores





## ARQUITECTURA LANZADOR

*Elemento intermediario e inicializador de*

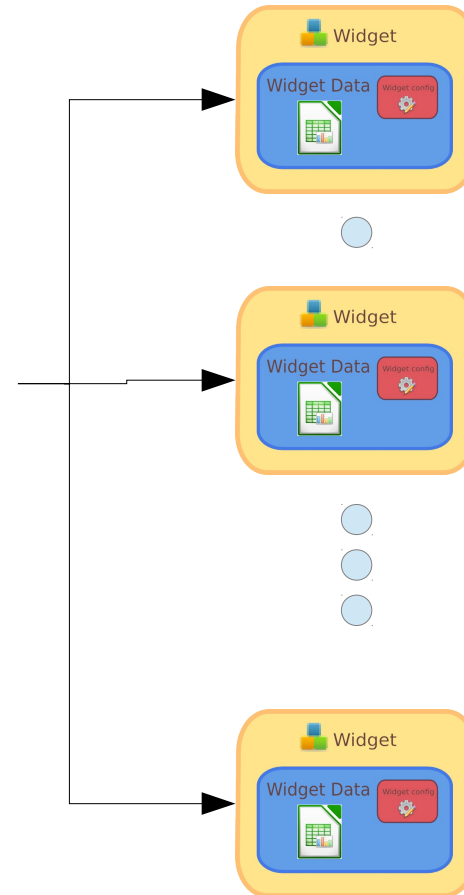
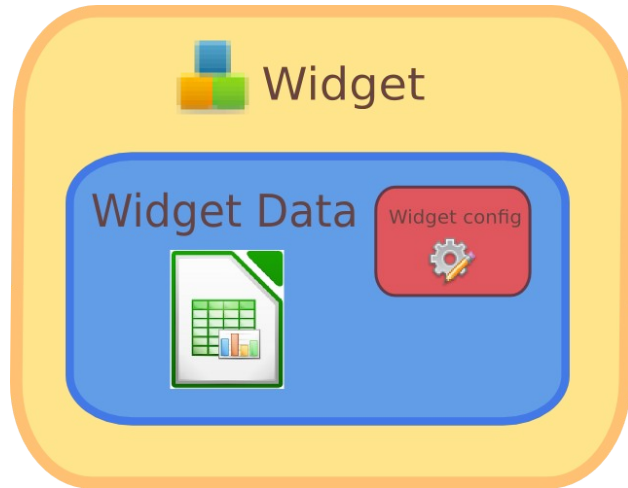
? Carga dinámica interfaz gráfica

☺ Lanzador parametrizable / Hilos

✓ Gestión de errores / Inicialización



# ARQUITECTURA WIDGETS



? Estructura común widgets

☺ Separación Datos/Configuración

✓ Adaptable / Interfaz común

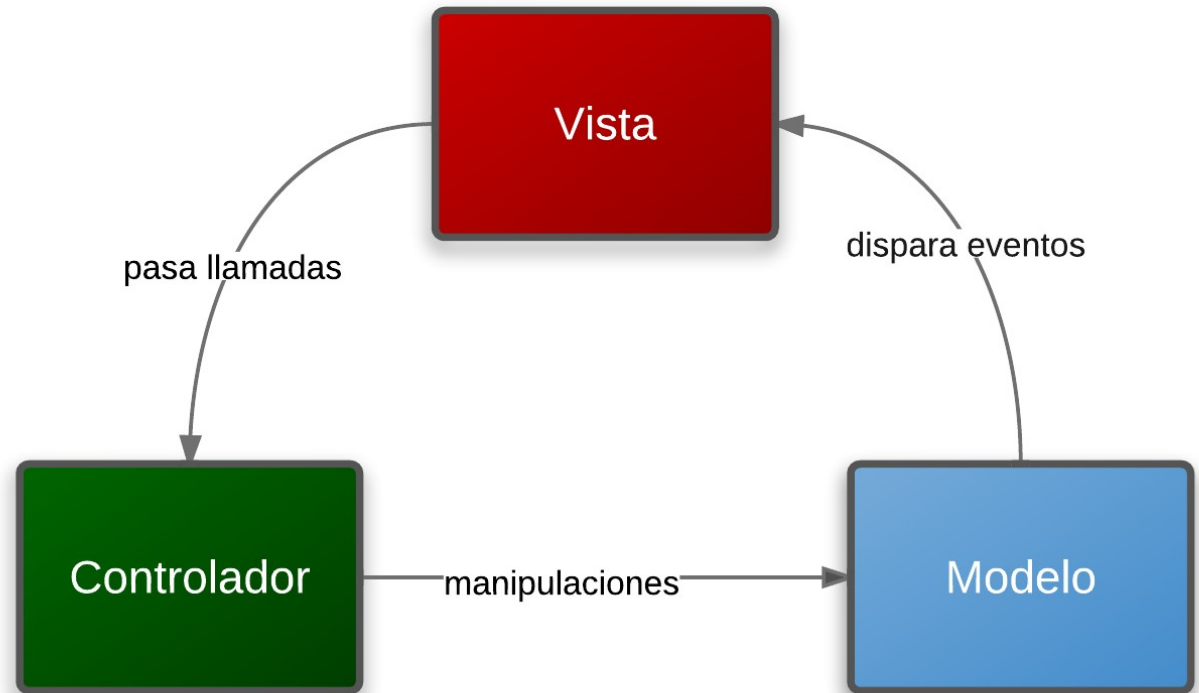


## *MVC: Modelo-Vista-Controlador*

? Patrón de diseño

☺ MVC

✓ Independencia componentes





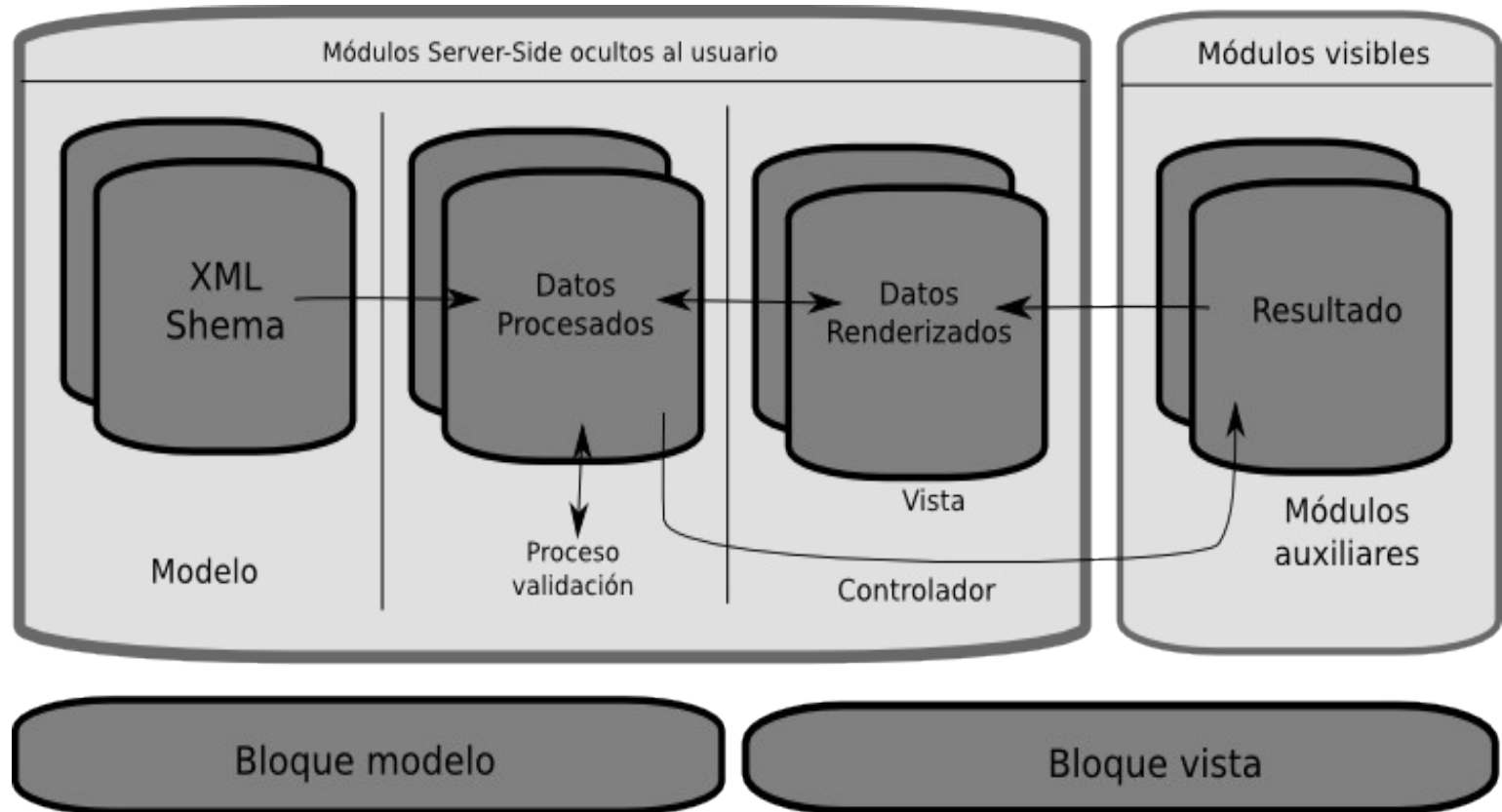


## PROCESAMIENTO DE RESULTADOS CON MVC

? Procesar resultados

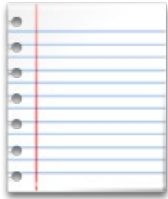
☺ Estándar XML + MVC

✓ Validación





## Listado de widgets iniciales



*MenuActionsArea*



*TitleDisplay*



*LogoArea*



*CouchDB*



*FeedReader*



*NewsReader*



*WhiteLabel*



*VideoArea*



*MountInfo*



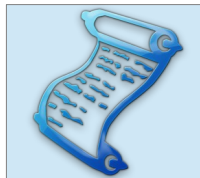
*MountDetector*



*Browser*



*NewsInfo*



*ScrolledBox*



*ScrollableText*



*StarVote*



*ButtonSection*



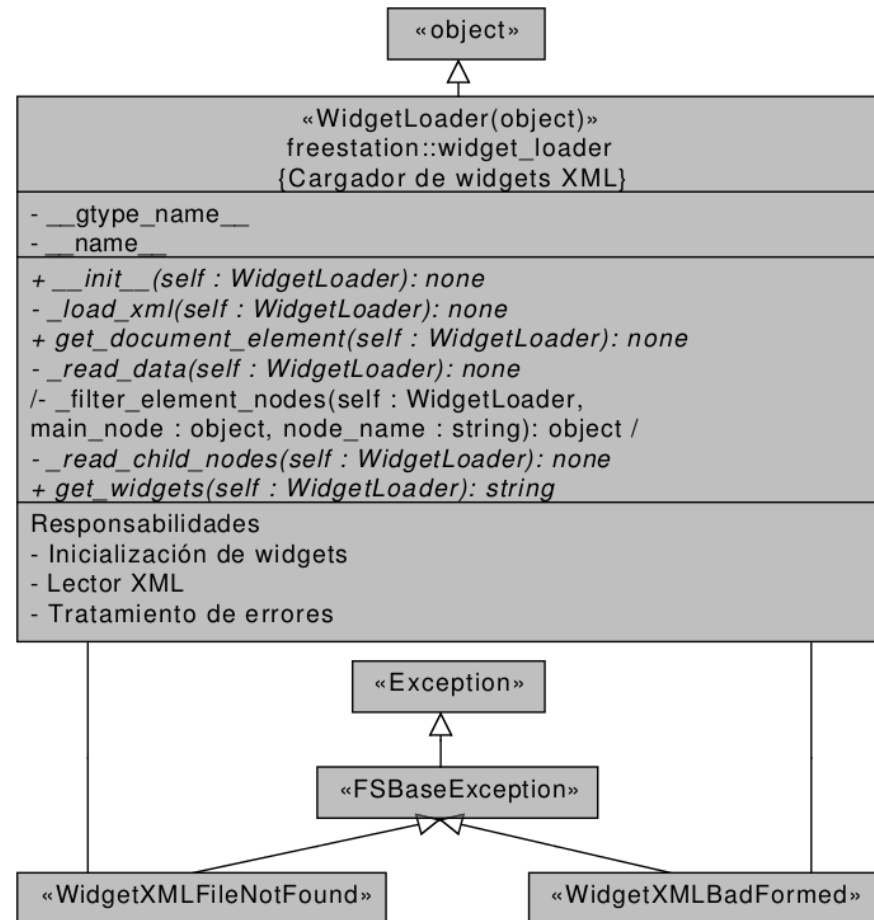
# DIAGRAMA DE CLASES - WIDGETLOADER

## CARGADOR DINÁMICO DE WIDGETS

? Carga rápida y eficiente

- ☺ Detección y validación XML
- ☺ Inicialización widgets
- ☺ Lector configuración

- ✓ Tolerancia a fallos
- ✓ Automatización





# ESTRUCTURA DE ARCHIVO WIDGETS.XML

? Estándar interno

☺ Automatización XML

✓ Extensible

✓ Estructurado

```
<interface>
  <widget>
    <name>TitleDisplay</name>
    <properties>
      <position type="relative" child="0" pack="None|Start|End">MainWindow</position>
      <homogeneous>1</homogeneous>
      <spacing>5</spacing>
      <width>200</width>
      <height>300</height>
      <data>UCLM - FreeStation</data>
    </properties>
  </widget>

  ...

  <widget>
    <name>BrowserView</name>
    <properties>
      <position type="absolute"></position>
      <width>200</width>
      <height>300</height>
    </properties>
  </widget>

  ....

  <widget>
    <name>MountInfo</name>
    <properties>
      <position type="absolute"></position>
      <width>200</width>
      <height>300</height>
    </properties>
  </widget>

</interface>
```

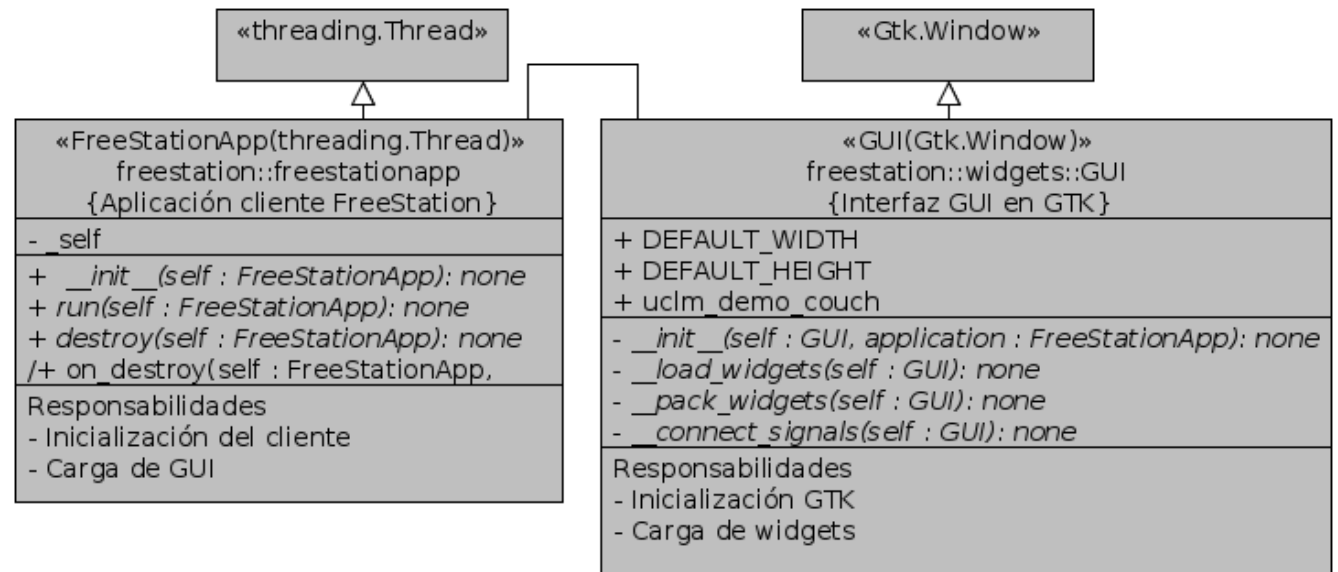


# GUI INTERFAZ GRÁFICA

? Visualización widgets

☺ Interpretar objetos  
analizados (widgets)

✓ Interfaz dinámica  
✓ Generación automática





# ARQUITECTURA APACHE COUCHDB



- *Descripción*
- *Propagación de eventos y señales*
- *Diagrama de clases*
- *Diagrama secuencia entre componentes (API REST)*



# DESCRIPCIÓN

- *Base de datos documental (JSON)*
- *Consultas MapReduce (Javascript)*
- *API REST normalizada (HTTP)*



✓ *Sincronización y replicación*



# PROPAGACIÓN DE EVENTOS

? Actualización datos interfaz

☺ Eventos / Señales CouchDB

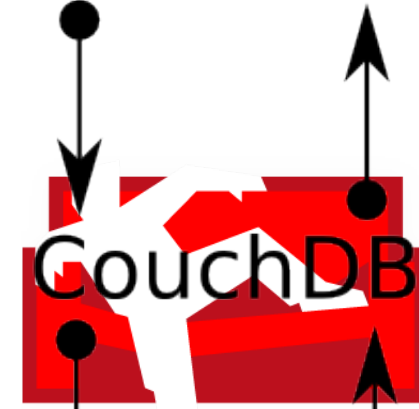
☺ Hub (concentrador)

✓ Rápido

✓ Eficiente

✓ Personalizable

HTML GUI  
con WebKit

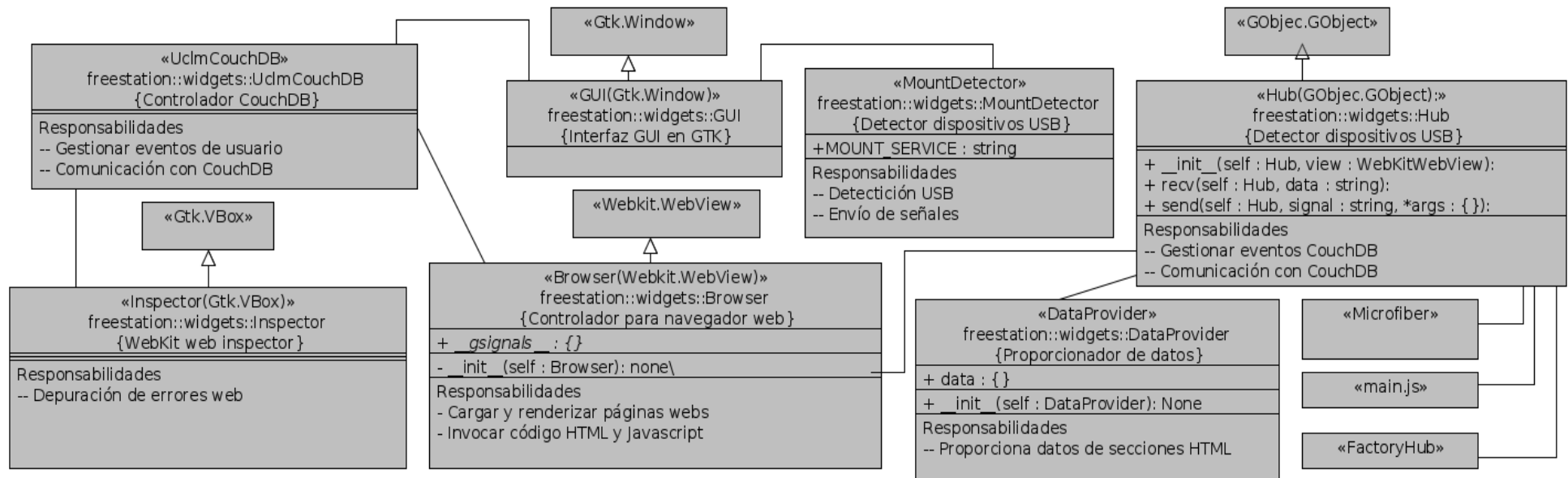


FreeStation  
Client





# ARQUITECTURA COUCHDB - DIAGRAMA DE CLASES

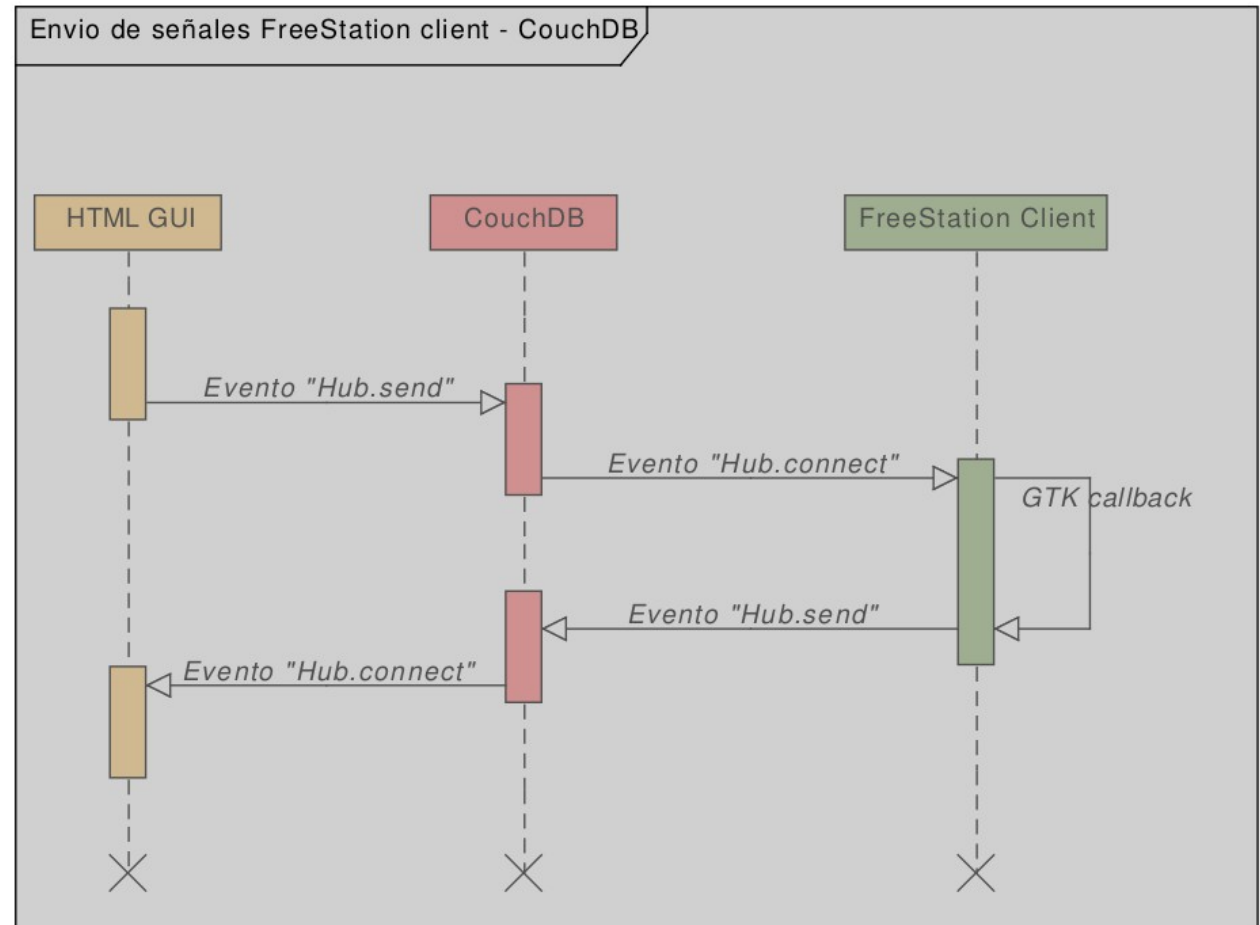




## API REST

1. Enviar evento: `send()`
2. Conectar: `connect()`
3. Interpretar datos (callback)

# DIAGRAMA DE SECUENCIA





# ARQUITECTURA DEL SISTEMA

- *Descripción general*
  - *Arquitectura del cliente*
  - *Arquitectura del servidor*



# ARQUITECTURA DEL SERVIDOR

- *Descripción general*
- *FreeStationServer*
  - *Backend*
  - *Frontend*

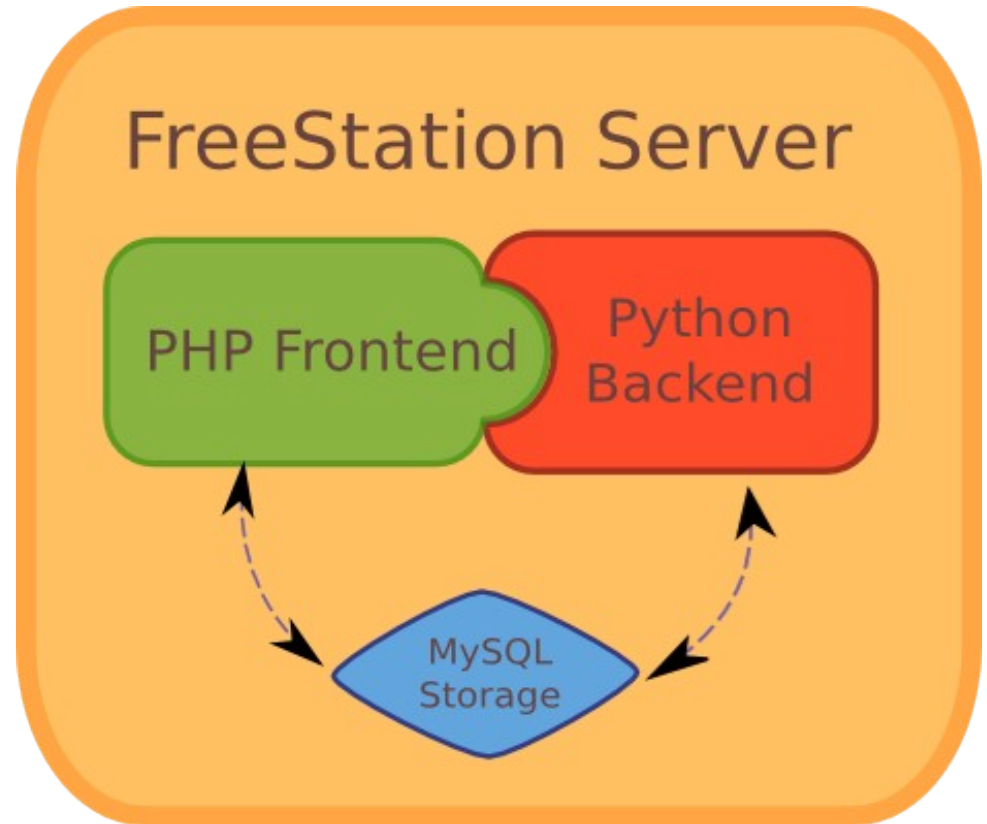


# DESCRIPCIÓN GENERAL

## ? Estructura FreeStationServer

- ☺ Frontend (PHP)
- ☺ Backend (Python)
- ☺ Persistencia (MySQL)

- ✓ Modularizable
- ✓ Independencia





## FRONTEND

- *Subsistema de widgets*
- *Subsistema de clientes*
- *Subsistema de gestión del servidor*
- *Subsistema de administración de contenidos (CMS)*



# DIAGRAMA DE COMPONENTES

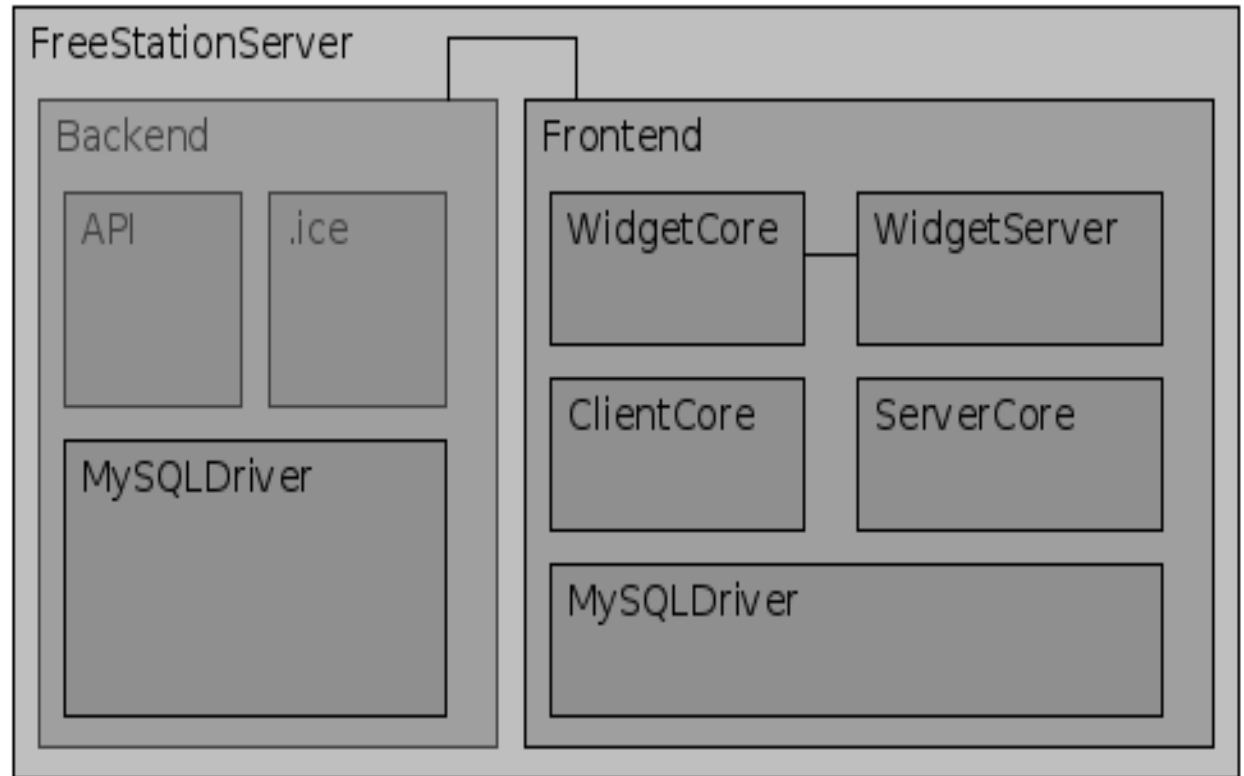
? Desarrollo núcleos

☺ Modelo ClassCore

☺ Modelo Drivers

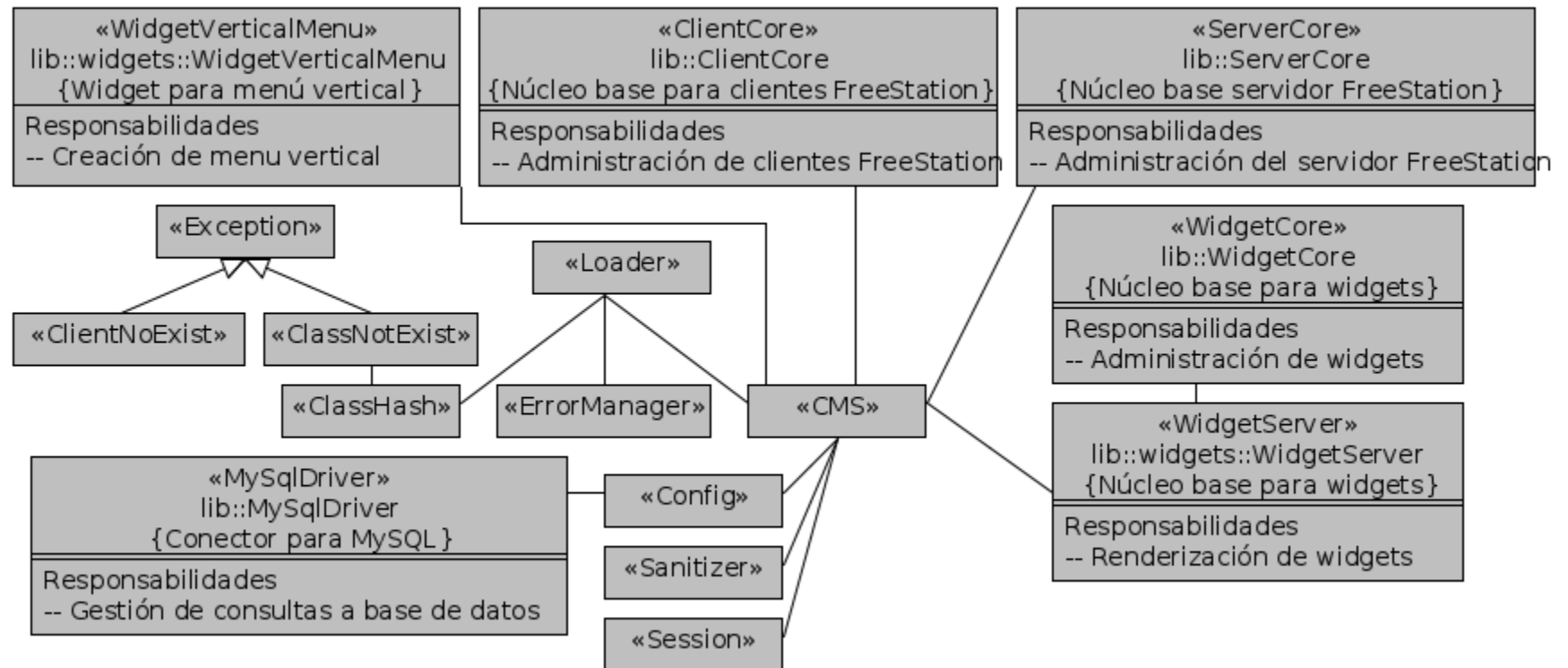
✓ Eficiente

✓ Extensible





# ARQUITECTURA SERVIDOR - FRONTEND







# ARQUITECTURA SERVIDOR - BACKEND

## *Zero C ICE: middleware sistemas distribuidos*

- *Backend Zero C ICE*
  - *Especificación SLICE*
  - *Generación Api basada en módulo FS.Api*



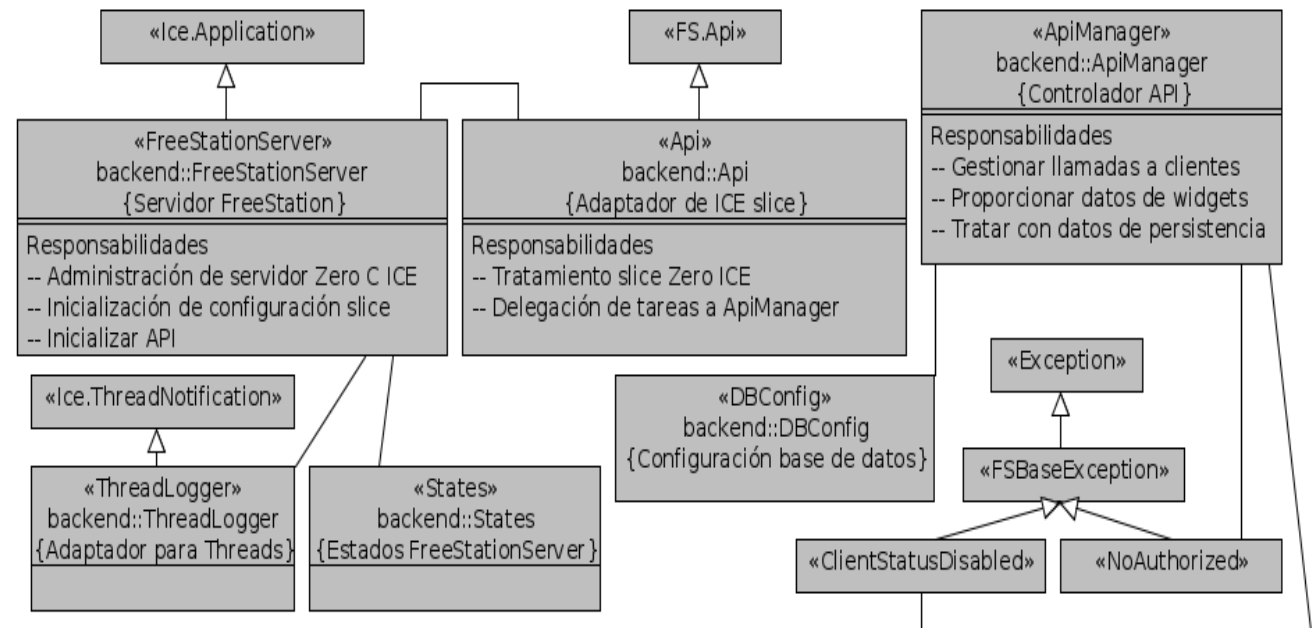
# ARQUITECTURA SERVIDOR - BACKEND

## Generación Api basada en módulo FS.Api

1. Slice genera Fs.Api

2. Utilizado Api

3. Adaptador ApiManager





# CLIENTE ICE

*Conexión al servidor (modelo distribuido)*

- *Descarga de archivos*
  - *Configuración widgets.xml*
  - *Descarga de datos*
- *Estadísticas de descargas*



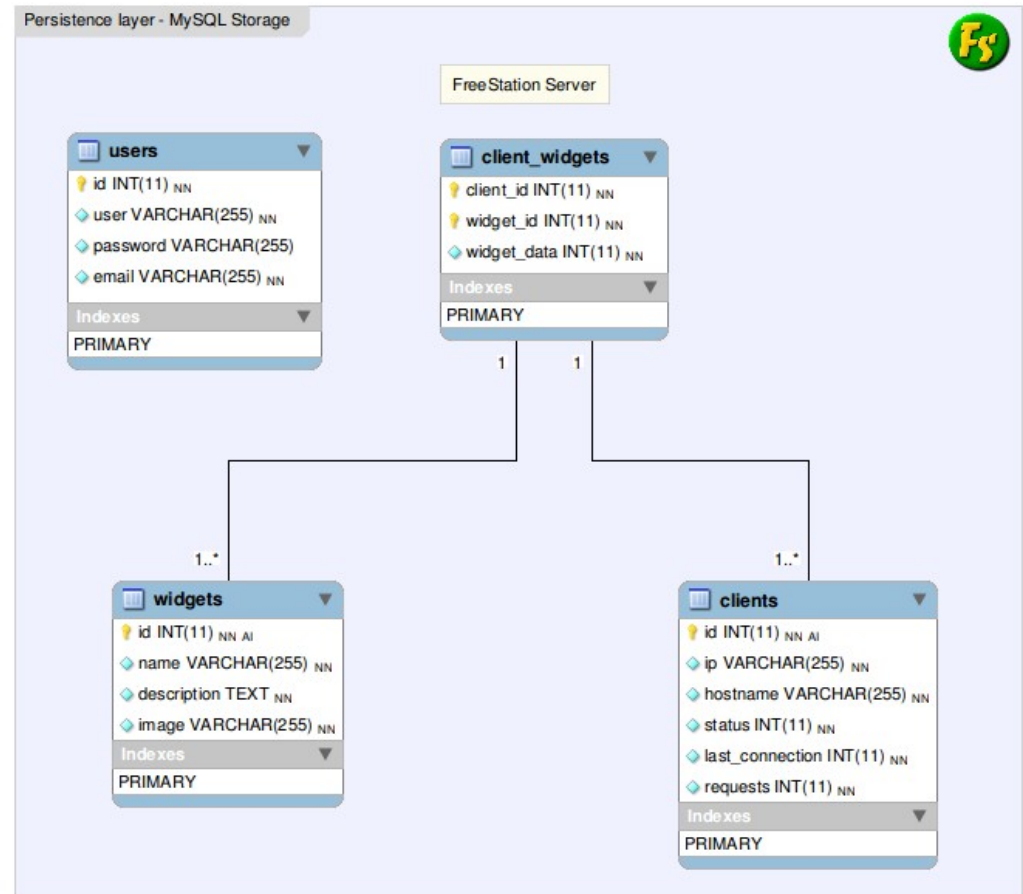
# CAPA DE PERSISTENCIA

## Diagrama ERR

? Estructuración de datos

- 😊 Usuarios
- 😊 Widgets
- 😊 Clientes

- ✓ Modelo Relacional
- ✓ Escalable



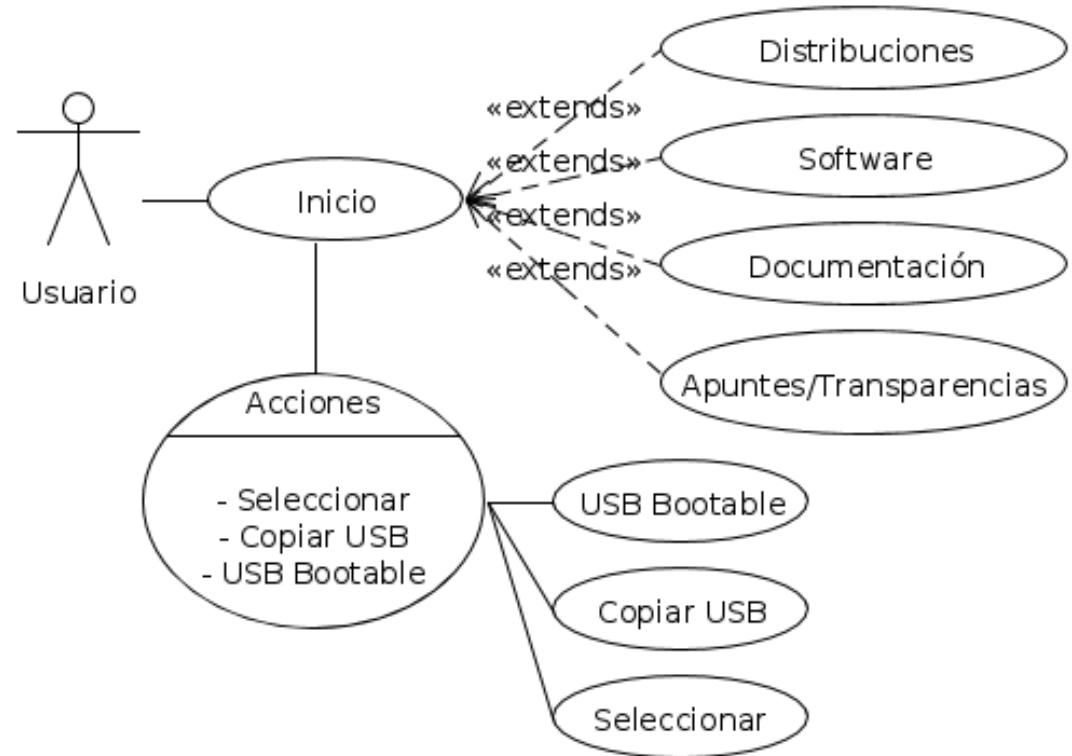


## CASO DE EXPLOTACIÓN POI UCLM

? Ilustrar ejemplo funcional de la plataforma

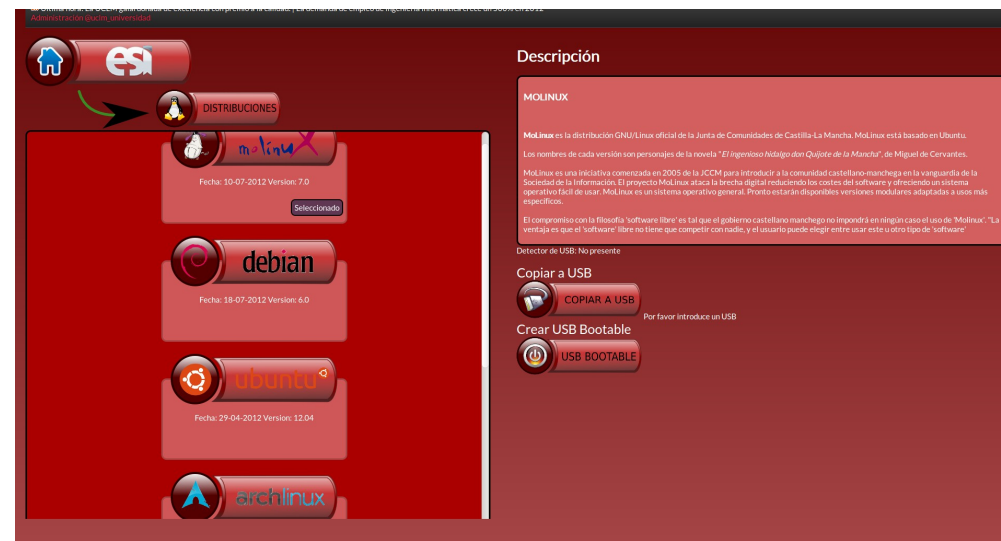
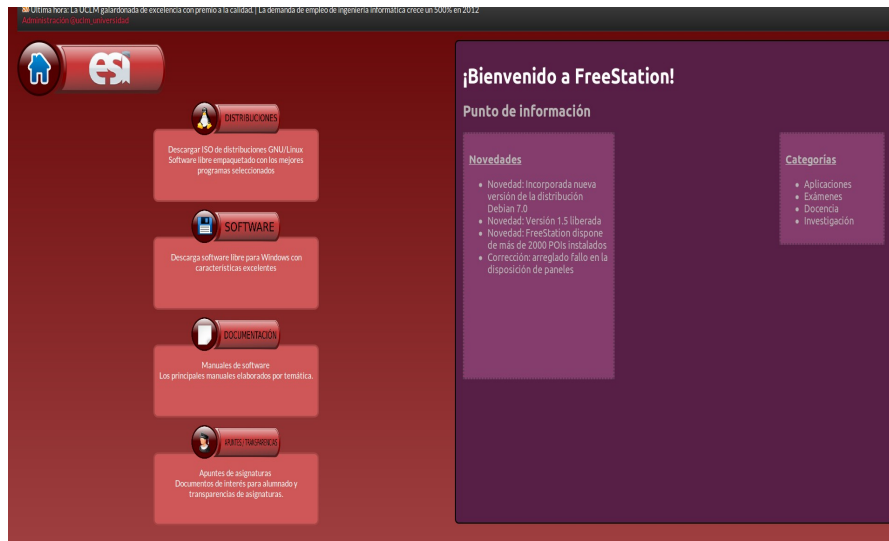
- ☺ Ejemplo widgets (CouchDB)
- ☺ Ejemplo widgets (GTK)

- ✓ Rápido
- ✓ Eficiente
- ✓ Personalizable



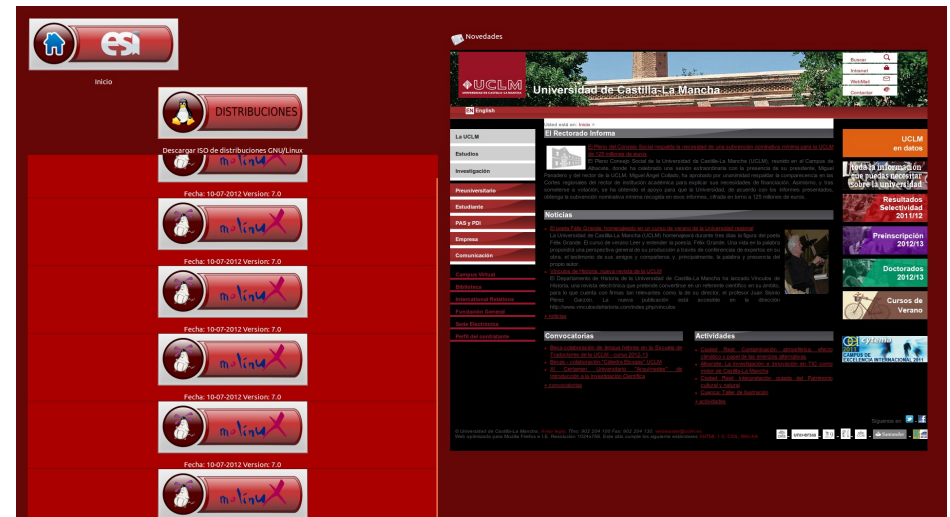
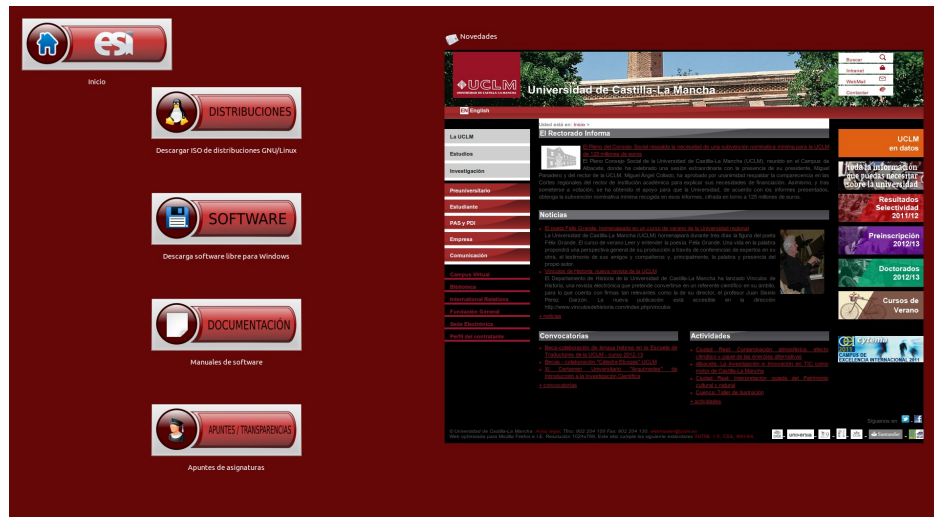


# DEMO (COUCHDB)

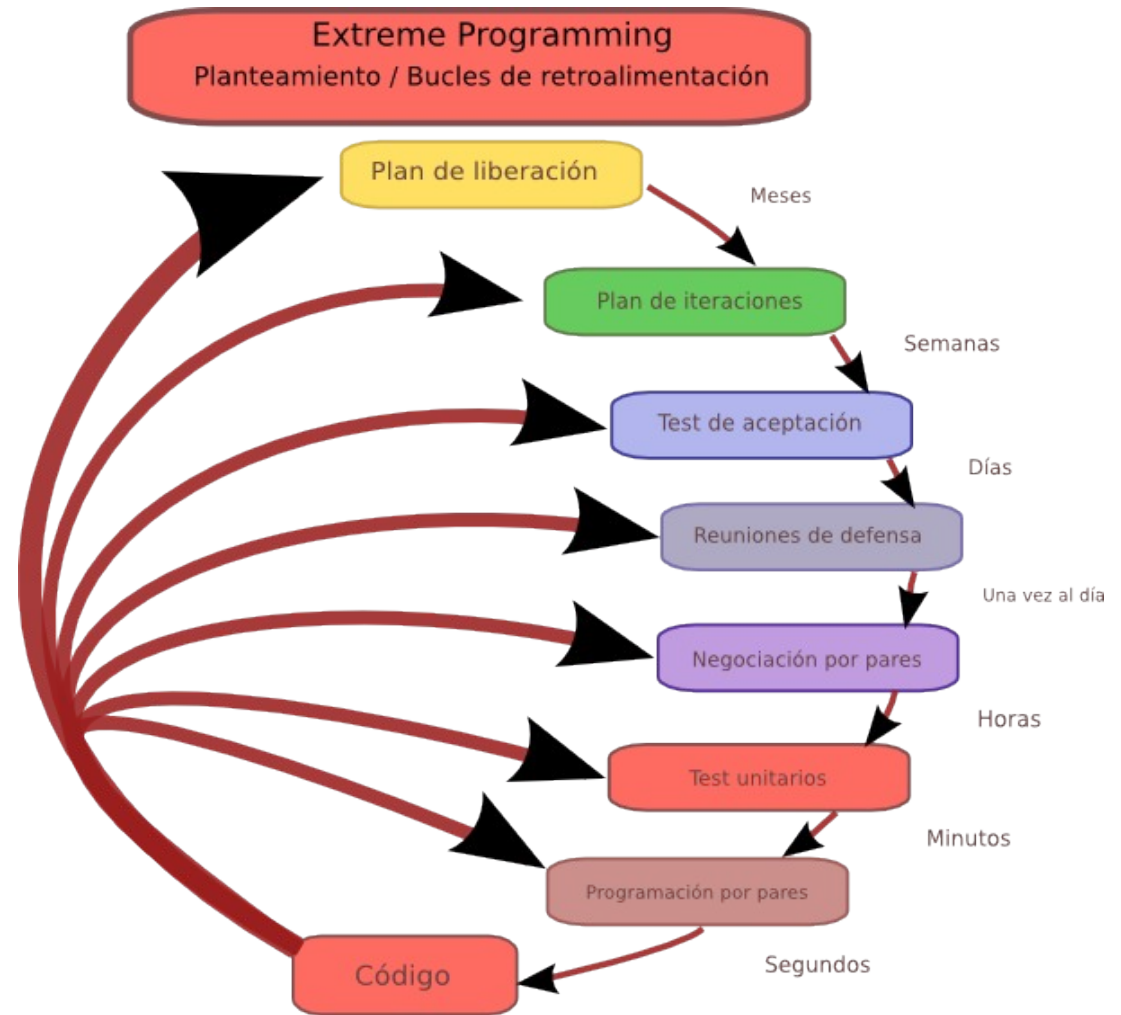




# DEMO (GTK)



- ☺ Desarrollo rápido
- ☺ Plan iteraciones
- ✓ Prototipos
- ✓ Resultados fiables

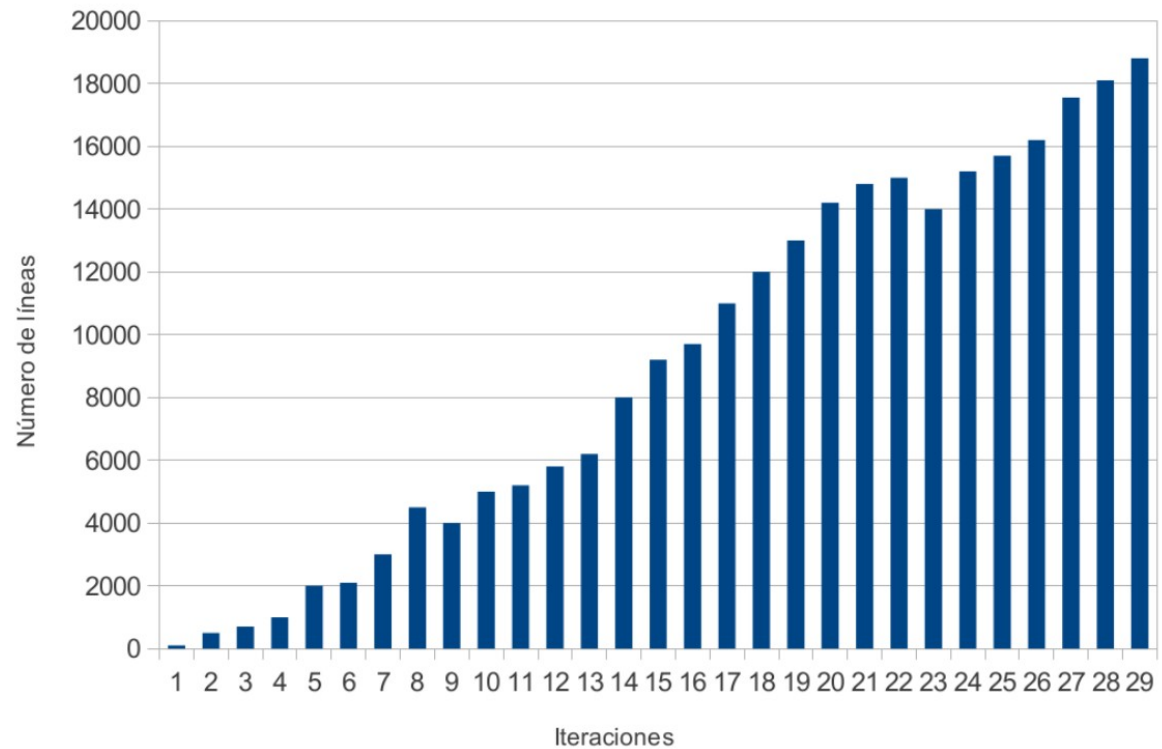




## ESTADÍSTICAS DE ACTIVIDAD EN EL REPOSITORIO

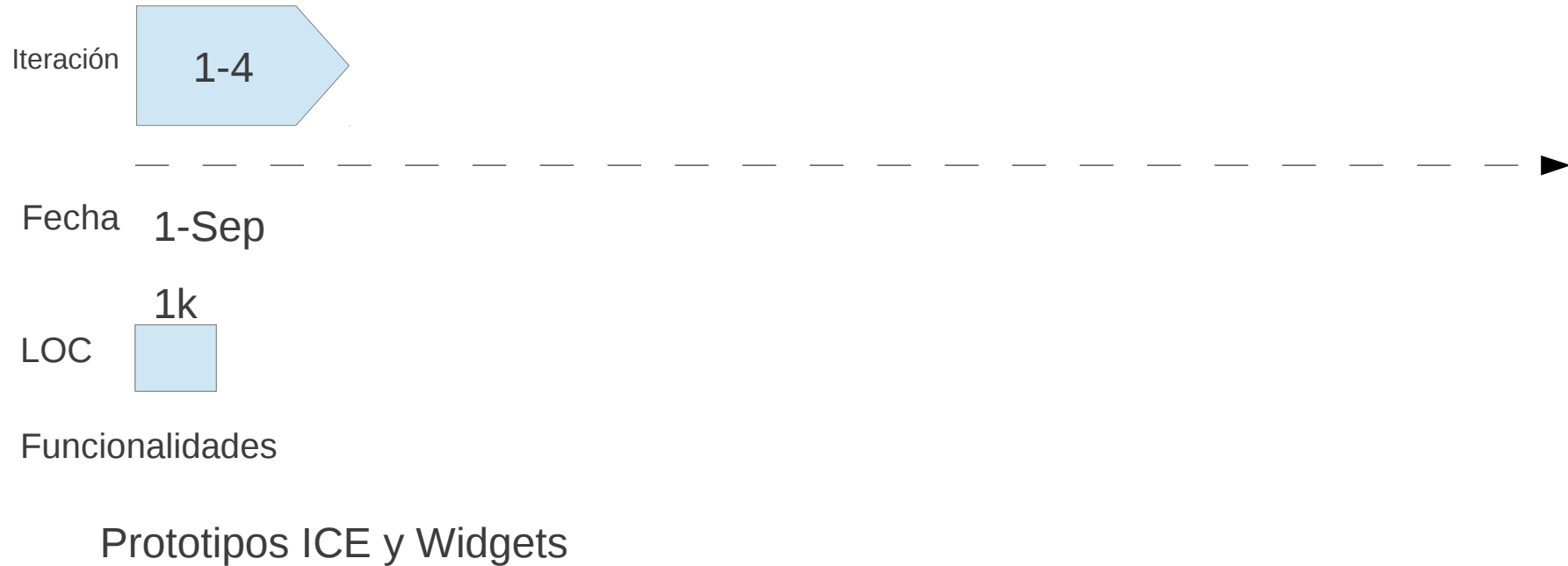
Líneas de código por iteración

Proyecto FreeStation

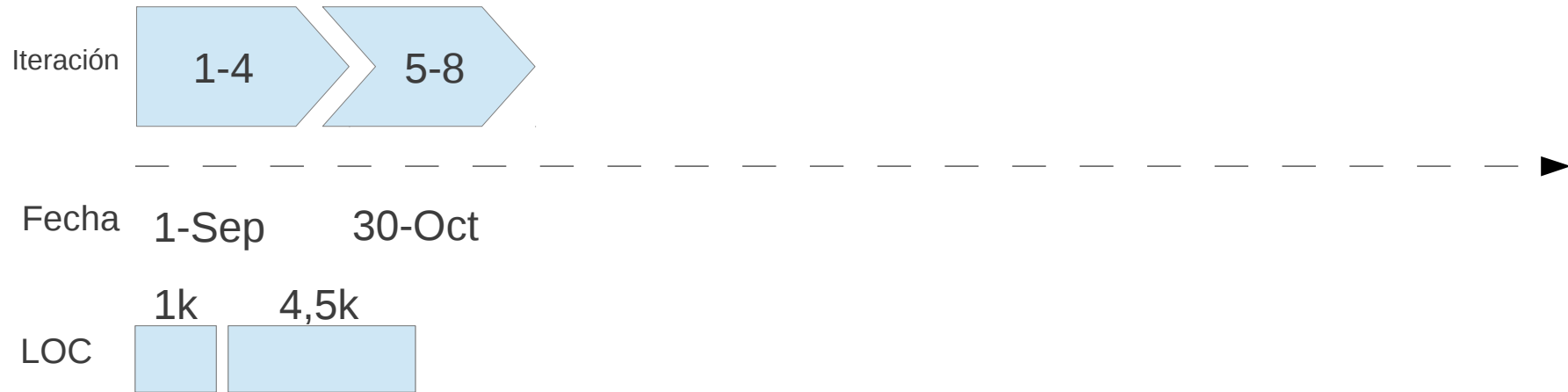


- *29 iteraciones*
- *20.000 mil líneas*
- *11 meses desarrollo*

# *ITERACIONES - EVOLUCIÓN*



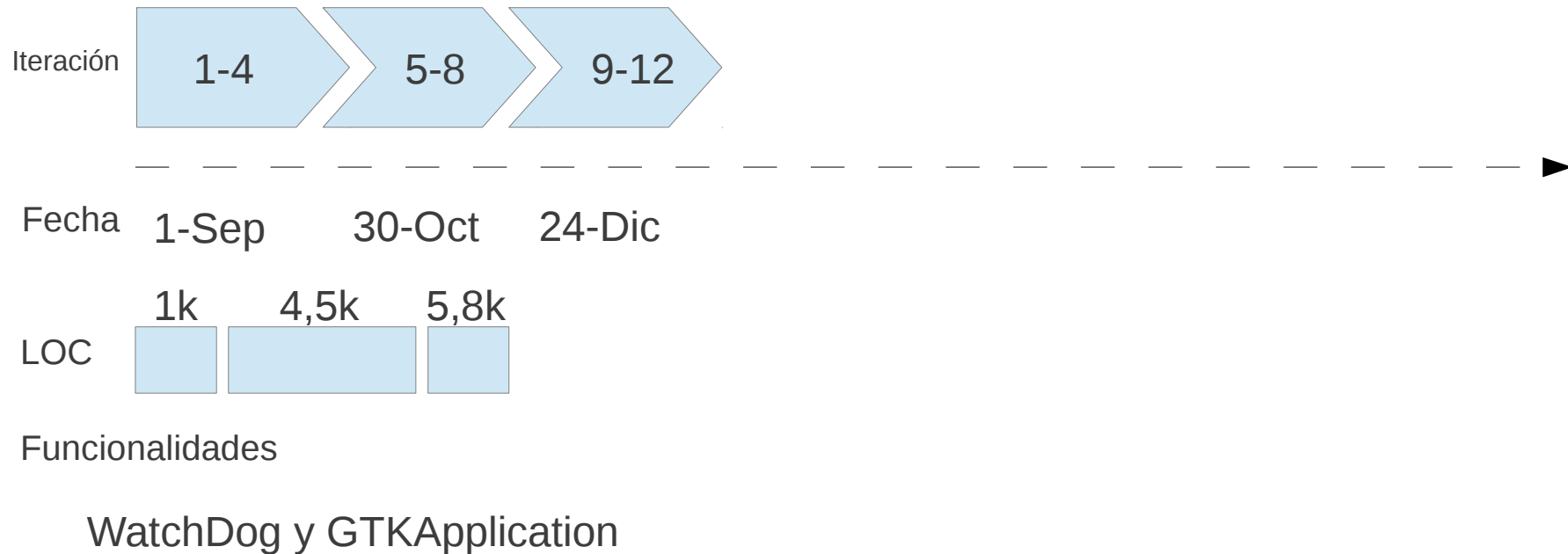
# ITERACIONES - EVOLUCIÓN



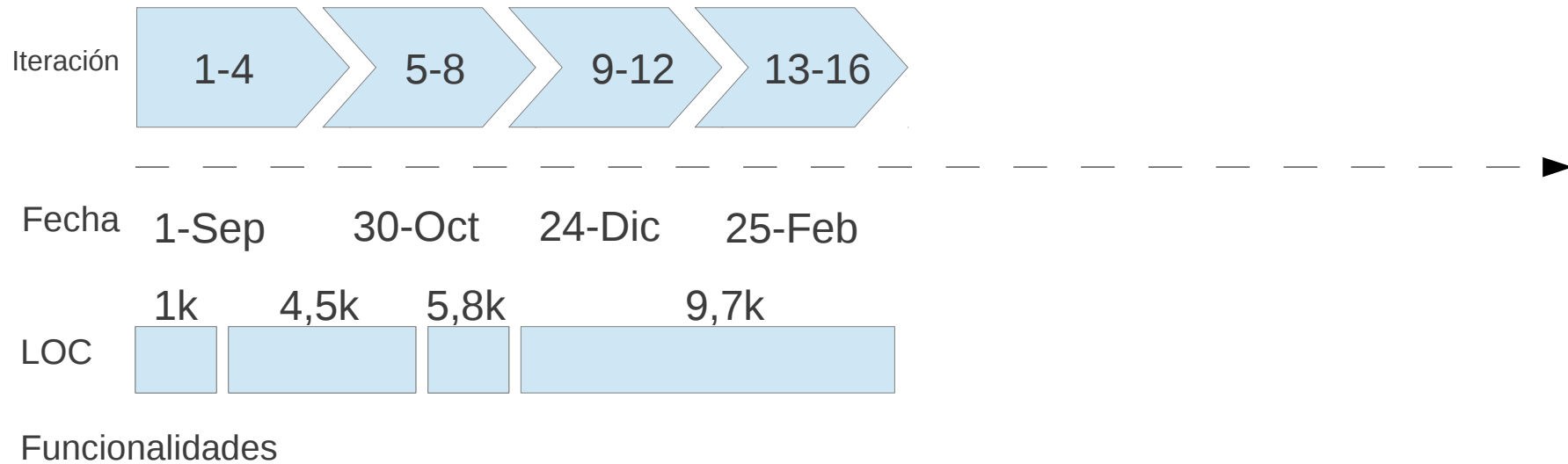
Funcionalidades

Desarrollo Gstreamer y WebKit

# ITERACIONES - EVOLUCIÓN

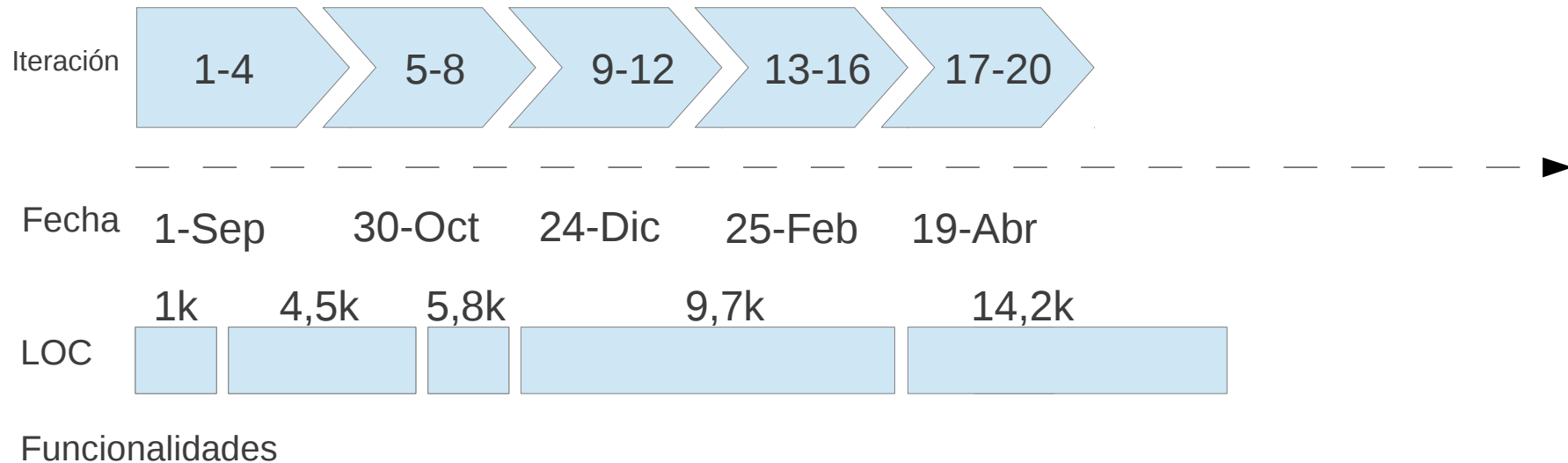


# ITERACIONES - EVOLUCIÓN



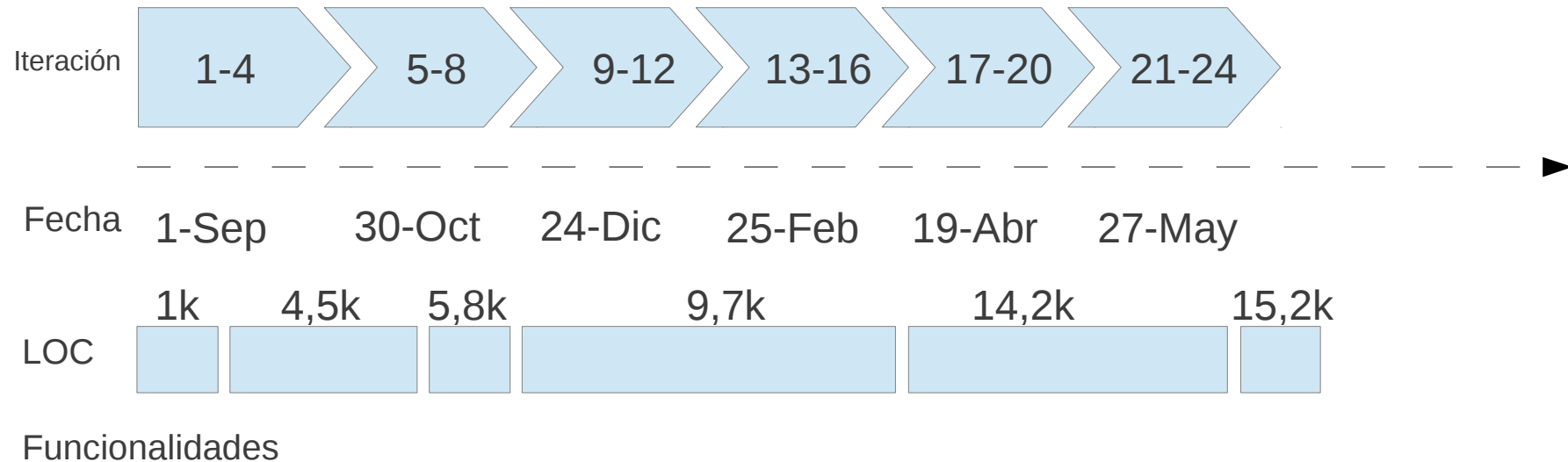
Carga dinámica de widgets y Backend

## ITERACIONES - EVOLUCIÓN



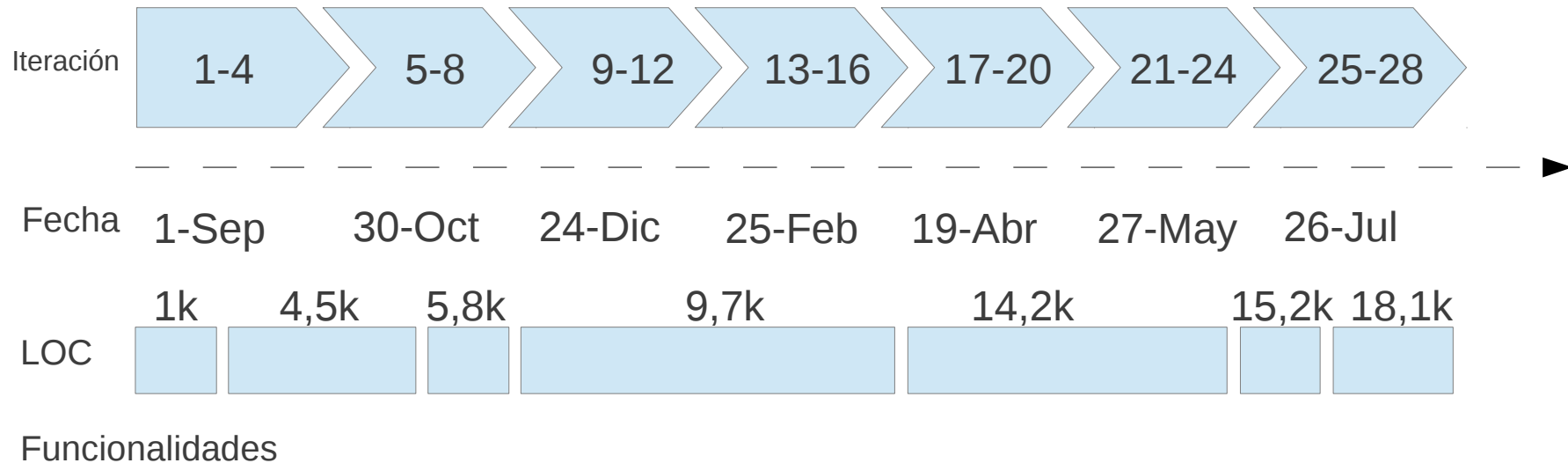
VideoArea y refactorizado WebKit

# ITERACIONES - EVOLUCIÓN



Theming y CouchDB

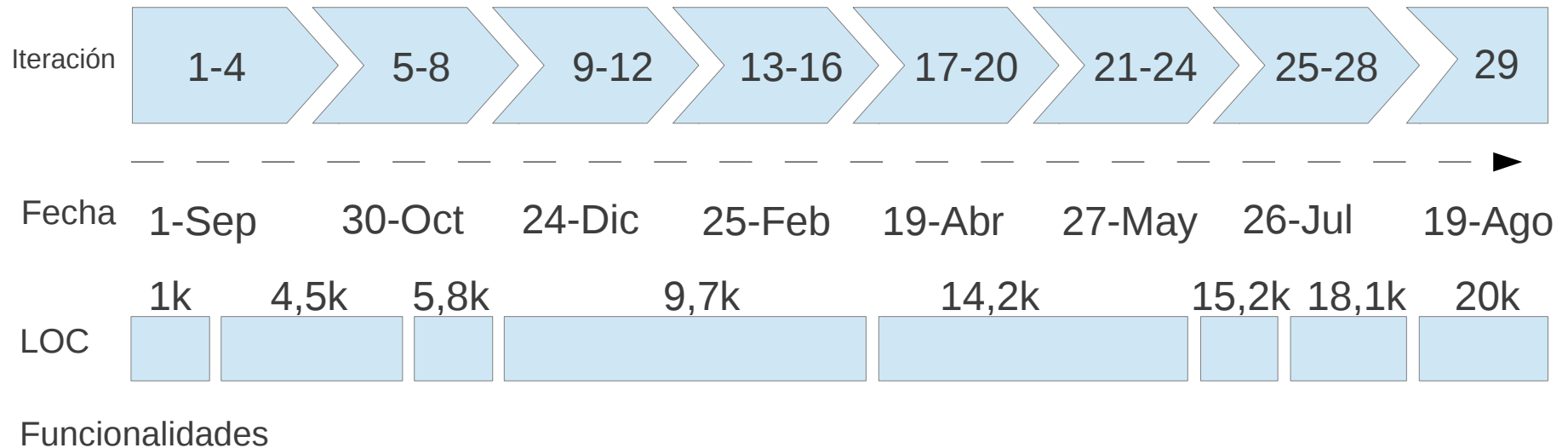
# ITERACIONES - EVOLUCIÓN



Reescritura Python 3 y contribuciones



## ITERACIONES - EVOLUCIÓN



Caso de explotación GTK y CouchDB

## *DESGLOSE CÓDIGO FUENTE*

Idiomas agrupados por lenguaje dominante primero				
Lenguaje	Archivos	Espacios en blanco	Comentarios	Líneas de código
Javascript	19	2901	2177	10495
HTML	18	505	5	2797
CSS	14	507	141	2405
PHP	27	402	930	1999
Python	8	367	348	876
make	1	24	5	124
SQL	2	37	57	82
Bourne Shell	2	22	6	48
XML	1	0	0	10
Total	92	4765	3709	18836

## *RESUMEN COSTE FINAL DEL PROYECTO*

- *Equipos informáticos = 3765 €*
- *Licencias = 225 €*
- *Material fungible = 205 €*
- *Recursos humanos: 56.220 €*

*Total: 60.720 €*

# RESULTADOS

## *Test Profiling*

- *CouchDB (aprox 200 eventos/seg)*
- *GTK ( $\leq 1$  seg iniciar app)*

## *Contribuciones proyectos Software Libre*

- *Gnome Bug #631901 - Proof of concept fundamental support (2011-11-01)*  
*Ejemplo test para Gstreamer (vídeo) en PyGObject*
- *Merge proposal #119261 (12-08-2012)*  
*Mejora WebKitNetworkRequest (Novacut)*
- *Launchpad Bug #1023770 - Store as persitent the log actions (12-06-2012)*  
*Captura de mensajes de consola WebKit para Novacut*
- *Mención Proyecto Novacut (08-08-2012) (Blog y Google+)*

## CONCLUSIONES

- *Alcance total objetivos previstos.*
- *Funcionalidad completa.*
- *Uso extensivo de patrones y principios.*
- *Reutilización de componentes.*
- *Adaptabilidad, homogeneización*

## TRABAJO FUTURO

- *Ampliación de widgets*  
*Estimación de tiempo: 2-3 meses*
- *Sustitución del almacenamiento de datos para widgets*  
*Estimación de tiempo: 1-2 meses*
- *Incorporar un caso de explotación con localización de idiomas*  
*Estimación: 1-2 meses.*

# GRACIAS POR SU ATENCIÓN

