ARQUITECTURA DEL CLIENTE

- *Descripción general*

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

ARQUITECTURA WATCHDOG

Monitorizador de cliente (FreeStationClient)

? Tolerancia a errores

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

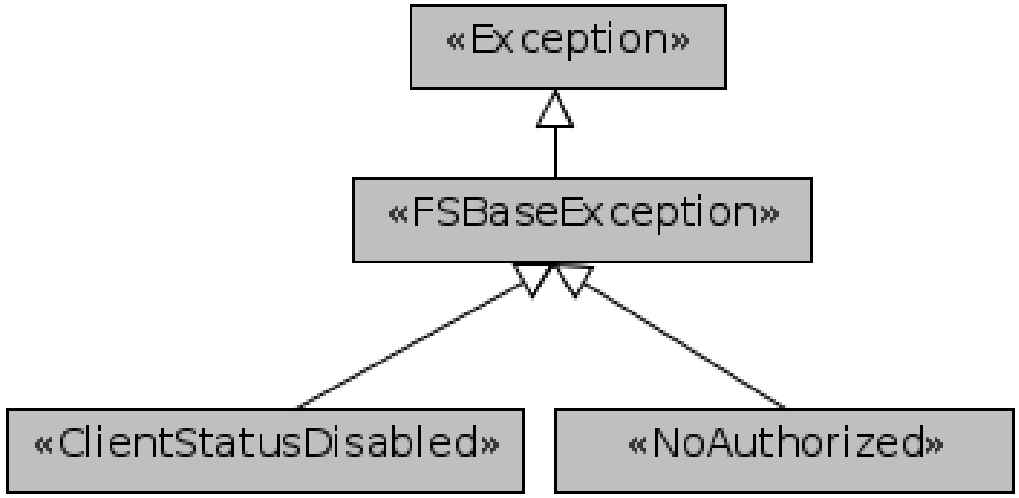
✓ *Recuperación en caso de fallo*

JERARQUIA DE EXCEPCIONES

? Estructura excepciones

☺ Especialización excepciones

✓ Personalizacion errores



ARQUITECTURA LANZADOR

Elemento intermediario e inicializador de

? Carga dinámica interfaz gráfica

☺ Lanzador parametrizable / Hilos

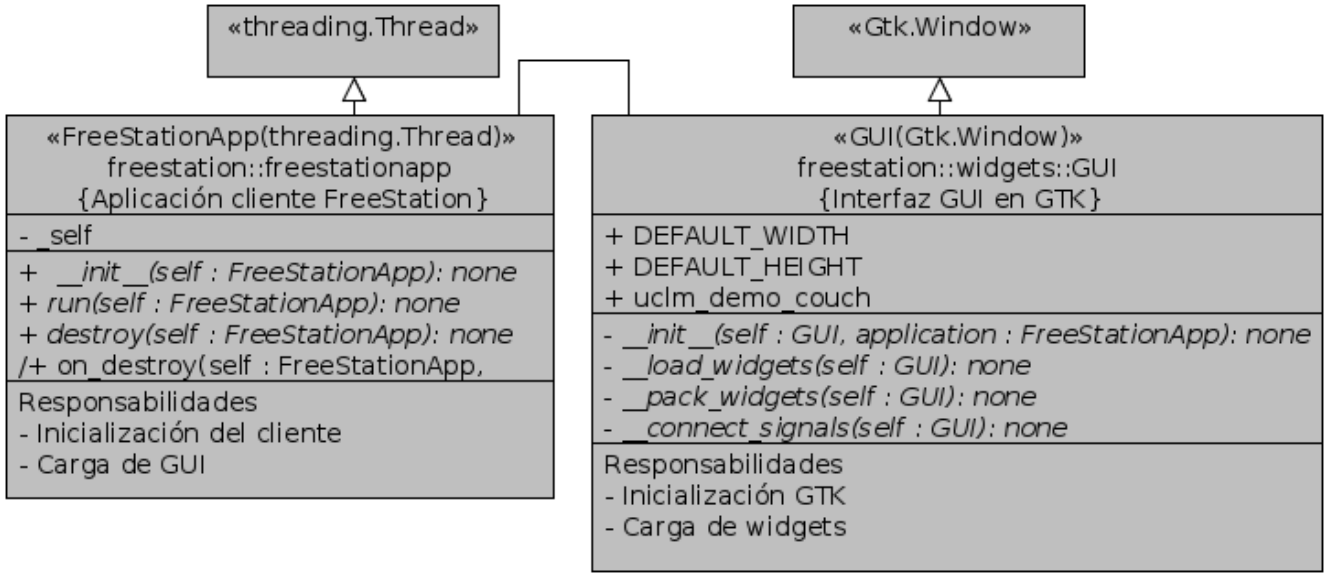
✓ Gestión de errores / Inicialización

GUI INTERFAZ GRÁFICA

? Visualización widgets

☺Interpretar objetos analizados (widgets)

✓Interfaz dinámica
✓Generación automática



ARQUITECTURA DEL SISTEMA

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

ARQUITECTURA DEL SERVIDOR

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

Subsistema de widgets

? Acceso a datos y configuración widgets

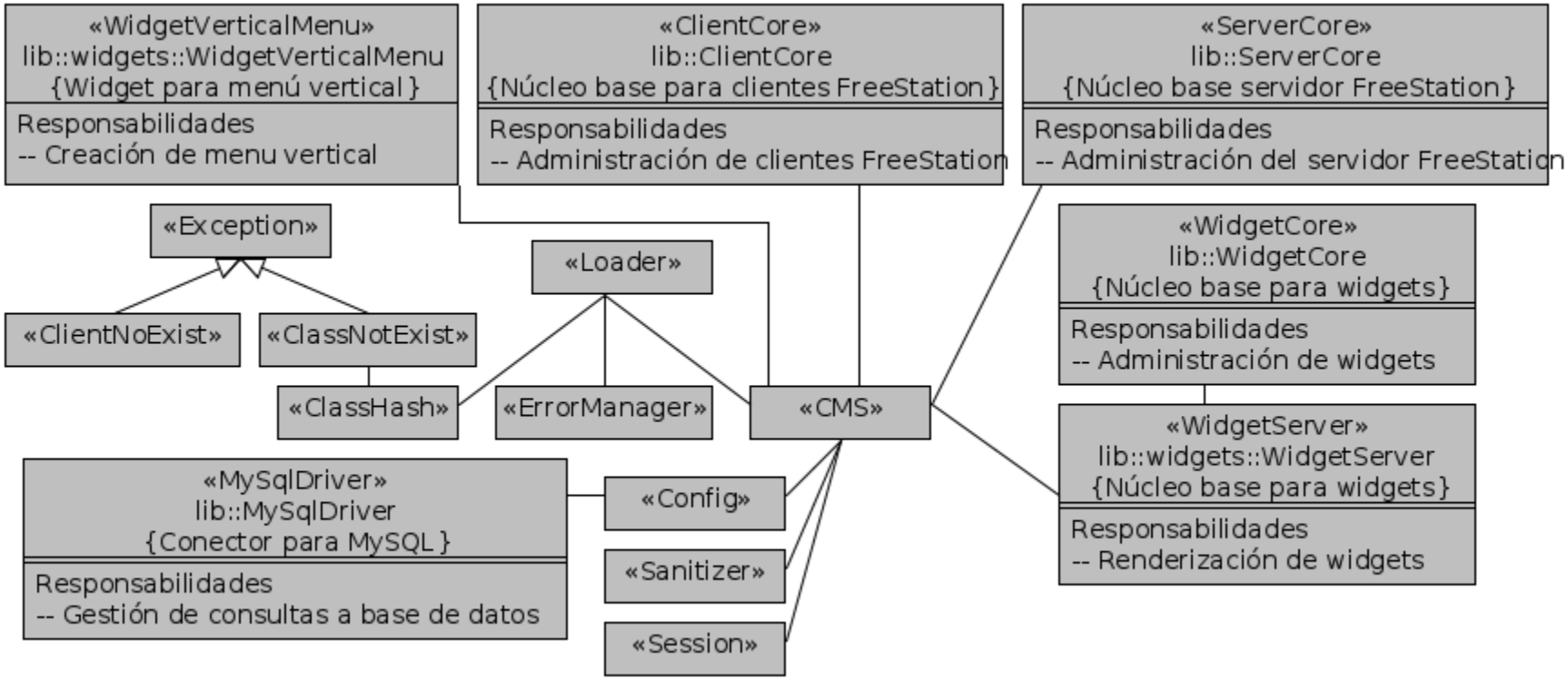
☺ WidgetCore (administración de widgets)
☺ WidgetServer (administración widget servidor)

✓ Modularizable
✓ Extensible

FRONTEND

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

ARQUITECTURA SERVIDOR - FRONTEND



Subsistema de clientes

? Acceso a datos y configuración clientes

☺ ClienteCore (administración de cliente)

✓ Persistencia
✓ Control de acceso

Subsistema de administración de contenidos (CMS)

? Visualización e interfaz web

☺ Carga por demanda
☺ Gestión de errores
☺ *Niveles asociados logs*

✓ Eficiencia
✓ Desarrollo rápido

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

Zero C ICE: middleware sistemas distribuidos

- Backend Zero C ICE
  - Especificación SLICE
  - Creación:
    - Comunicador
    - Adaptador
    - Sirviente
  - Generación Api basada en módulo FS.Api

ARQUITECTURA SERVIDOR - BACKEND

Especificación SLICE

- *Lenguaje de definición de interfaces*
- *Contrato servidor/cliente*
- *Autogenera especificación (lenguaje independiente)*

ARQUITECTURA SERVIDOR - BACKEND

Creación de componentes ICE

- *Comunicador:*
  - *Contexto implícito*
  - *Gestión eventos*
- *Adaptador*
  - *Mapear objetos ICE*
  - *Asociar puntos conexión*
- *Sirviente*
  - *Mantener objetos ICE (memoria)*

CLIENTE ICE

Conexión al servidor (modelo distribuido)

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

CLIENTE ICE

Estadísticas de descargas de archivos

Pruebas de rendimiento por tamaño de chunk			
Tamaño archivo	Tiempo	Tamaño chunk	Tasa transferencia
1.5 MB	137 sec	1 KB	10,94 KB/s
1.5 MB	16 sec	10 KB	93,75 KB/s
1.5 MB	4 sec	50 KB	375,00 KB/s
1.5 MB	2.84 sec	100 KB	528,16 KB/s
1.5 MB	2.67 sec	150 KB	561,79 KB/s
1.5 MB	2.26 sec	500 KB	663,71 KB/s
700 MB	961 sec	150 kb	728,40 KB/s

PATRONES

Singleton: creacional - clases de Frontend del servidor como CMS, Session, Error y ClassHash.

Factory o Fábrica: creacional - clases relacionadas con CouchDB como HubFactory y la creación de widgets.

Mosca o peso ligero(Flyweight): estructural - creación de widgets para una carga más ligera.

Facade (Fachada): estructural - clases de Frontend (server) – API y ApiManager

Proxy: estructural - conexión de Zero C ICE (cliente y servidor)

Adapter (Adaptador): estructural - conexión de Zero C ICE (cliente y servidor)

Pruebas unitarias

Python – unittest

Clases criticas

Watchdog
FreeStationApp
Browser

PHPUnitTest (frontend server)

MapReduce

Función Map()

Mapea pares datos (paralelo) a un dominio de datos (lista de pares)

Map(k1,v1) -> list(k2,v2)

Función Reduce()

Reduce las listas con mismas claves(paralelo) y obtiene colección de valores

Reduce(k2, list (v2)) -> list(v2)