

# Transformación de Sistemas Heredados

... o cómo cambiar el motor de un 747 en pleno vuelo

Noviembre, 2019





01.

# Planteamiento general

Open Canarias y la  
Modernización



# Acerca de Open Canarias

Visión de la compañía y sus principales clientes.

Empresa fundada en 1996, con origen de la actividad en Canarias a partir de un grupo de trabajadores de la delegación de IBM en las islas. Desde ese punto de partida, se ha convertido en un referente en Desarrollo y Modernización Tecnológica TIC, con un importante crecimiento a nivel nacional e internacional, que incluye más de 300 clientes públicos y privados, fundamentado en las capacidades de sus 80 empleados (+70% titulación superior).

En su trayectoria, Open Canarias destaca por su compromiso por la I+D+i, entiendo como el eje sobre el que articula la estrategia empresarial. Con múltiples aportaciones científicas, las aportaciones a dos estándares del OMG, y varias contribuciones en proyectos de la plataforma Eclipse, Open Canarias ha alcanzado una alta reputación nacional e internacional.





# El problema

## El lastre de los monolitos.

**Los procesos de transformación digital demandan una mayor agilidad en la evolución de los sistemas, siendo ésta en muchos casos asimétrica. No existe una única arquitectura que soporte adecuadamente todo tipo de productos y servicios, con sus correspondientes necesidades de escalado, y por eso los sistemas monolíticos suponen un lastre en la mencionada transformación**

Las aplicaciones monolíticas son aquellas en las que el software se estructura en grupos funcionales muy acoplados,, involucrando los aspectos referidos a la presentación, procesamiento y almacenamiento de la información. Suelen estar contruidos sobre lenguajes y plataformas antiguas, y en muchos casos su desarrollo se ha producido a lo largo de varias décadas.

En la transformación de aplicaciones monolíticas nos encontramos con tres escenarios: reescritura, reemplazamiento y migración. Cualquiera de estos escenarios comporta enormes riesgos y posibilidades de fracaso si se aborda como un Big Bang. La palabras del CIO de un gran Banco son muy clarificadoras: “es como cambiar el motor de un 747... en pleno vuelo”. Otra frase del blog de Martin Fowler: “la única cosa que la reescritura Big Bang garantiza es un Big Bang”.

En el caso concreto de la migración, nos encontramos con que la mayoría de herramientas automatizadas de migración son puramente traslacionistas (cambian el lenguaje pero mantienen la arquitectura). La brecha entre la arquitectura de los supuestos con respecto a las nuevas arquitecturas es enorme. Se migran también los problemas (código muerto, código duplicado, complejidad ciclomática, ...).

# Modernización

Definición.

**Proceso** mediante el cual se aborda la transición de **sistemas heredados** hacia otros sistemas que respondan a **nuevas necesidades**, con un **enfoque integral** y orientado a que dicha transición sea **ordenada**, **controlada** y **adaptada** a las especificidades del cliente.



# Métodos

Las principales aproximaciones a la modernización.



## **ADM (Architecture Driven Modernization)**

Proceso para la comprensión y evolución de los activos de software existentes para mejorarlo, modificarlo, interoperar, refactorizar, reestructurar, reutilizar, migrar, ...

## **SABA Framework**

Es un framework de alto nivel para la planificación de la evolución y migración de sistemas heredados, y que vincula tanto los aspectos tecnológicos como los organizacionales

## **Visaggio's Decision Model (VDM)**

Es un método y modelo de decisión para determinar los procesos de renovación de software adecuados a nivel de componente, en función de las cualidades técnicas y económicas de dichos componentes.

## **Renaissance**

Es un método para evaluar iterativamente sistemas heredados, desde perspectivas técnicas, comerciales y organizacionales, con prácticas de reingeniería y aproximación incremental



# 04. ADA

Análisis y  
Descubrimiento de la  
Arquitectura



# Arquitectura ADA

## Principales componentes.

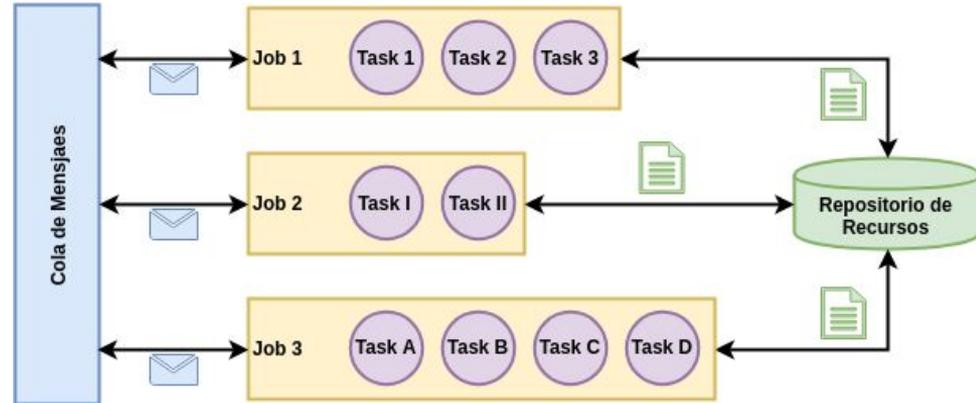
**Kafka** como cola de mensajes.

**Akka Actors** como librería para implementar la coordinación y ejecución de las tareas siguiendo el modelo de actores.

**HDFS (Hadoop Distributed File System)** como repositorio para almacenar recursos. El sistema permite la contribución de diferentes repositorios como sistema de ficheros local, etc.

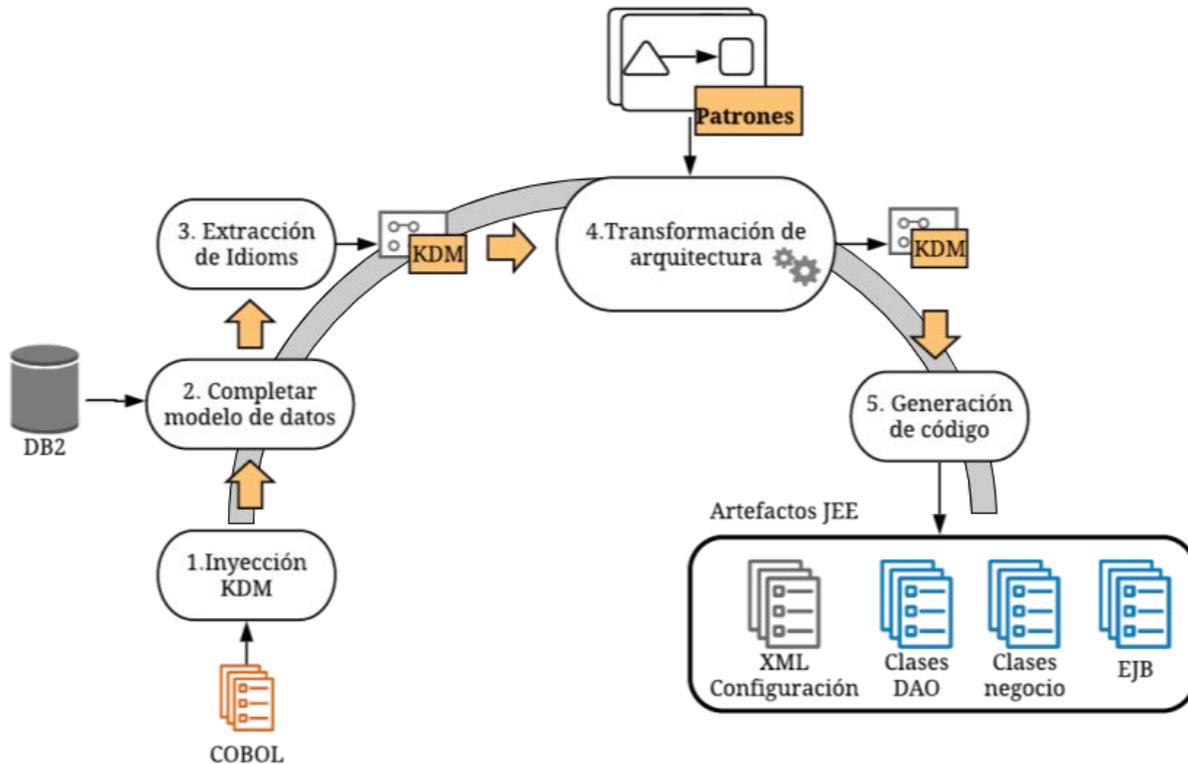
**Cassandra** como base de datos para almacenar información adicional derivada de la ejecución de las tareas así como el registro de la ejecución de las mismas.

**OrientDB** como base de datos para representar relaciones entre artefactos y realizar análisis basadas en grafos.



# Proceso de migración

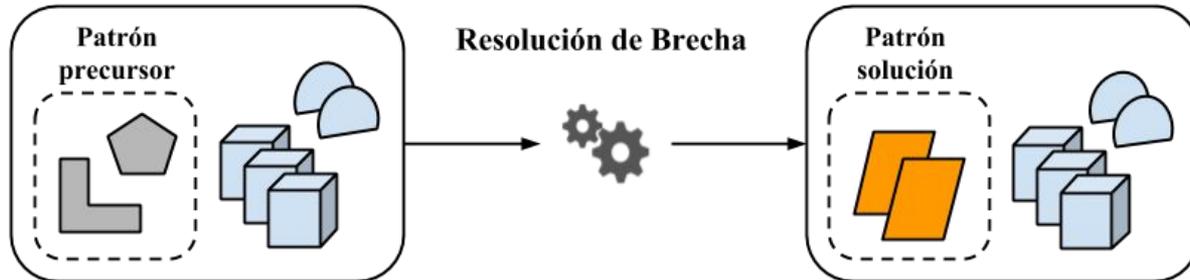
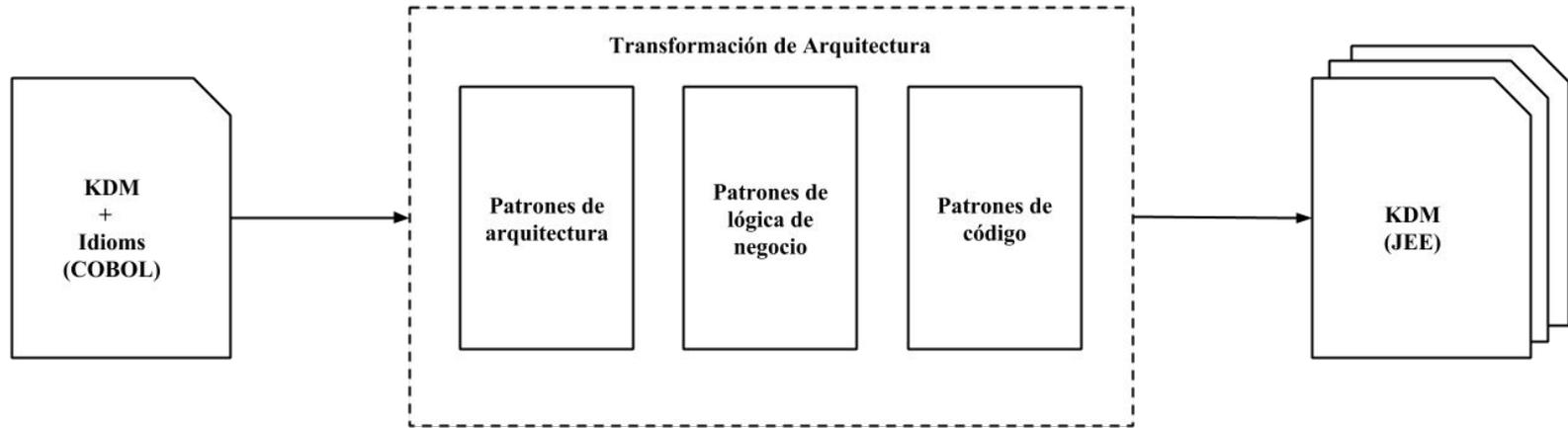
Descripción.





# Patrones

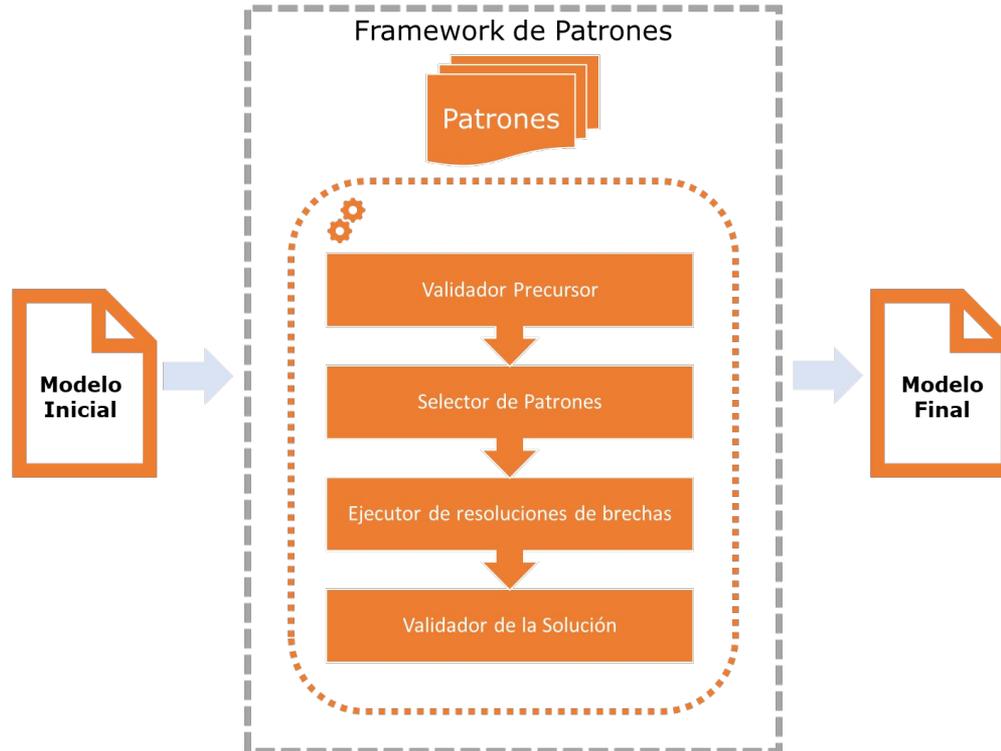
Herramienta formal para la transformación.





# Aplicación de patrones

## Descripción del framework





02.

# Marco Estratégico

Definición de un plan  
de transición



# Marco Estrategico

La aplicación estranguladora.

**La estrategia que ha demostrado ser un éxito es la que Martin Fowler denomina *StranglerApplication*.**

**El nombre viene de la Higuera Estranguladora australiana, un tipo de liana que crece alrededor de los árboles, consumiendolos hasta matarlo.**

**La aplicación nueva se va construyendo alrededor del monolito, dando soporte a nuevos servicios y sacando otros.**

Esta estrategia requiere de la definición de un marco de trabajo que facilite la transición incremental. Debemos tener en cuenta que dicha transición implica un amplio periodo de tiempo, en el que probablemente las necesidades del negocio y los cambios de la tecnología determinen nuevos escenarios para la transición.



# Definición

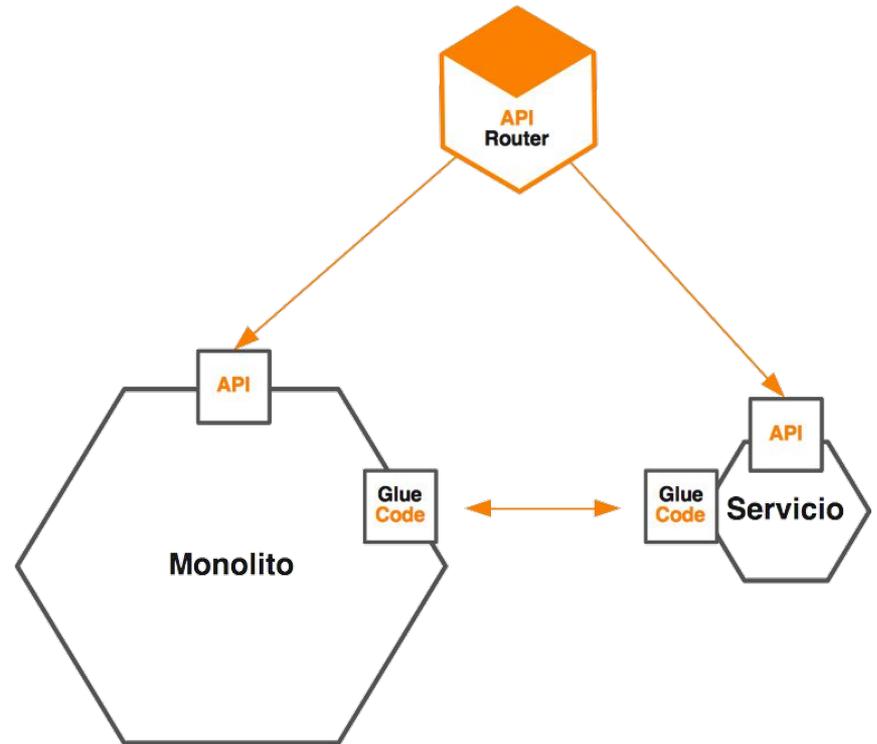
Definir la estrategia de estrangulación.

Establecer la arquitectura que va a permitir desarrollar nuevos servicios y extraer los antiguos del monolito

Validar la solución

Establecer criterios de priorización

Definir plan de transición





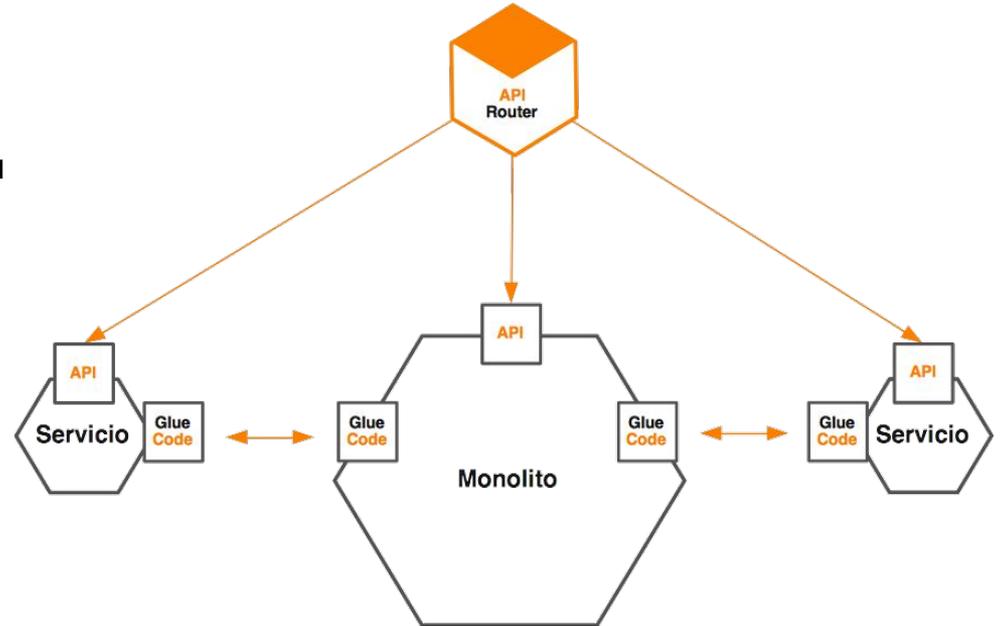
# Formalización

Dejar de cavar.

Todos los nuevos desarrollos siguen la nueva arquitectura

Añadir nuevas estrategias de estrangulamiento

Refinamientos del plan de transición

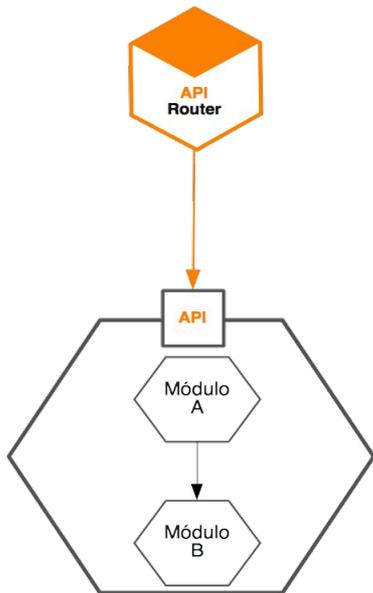




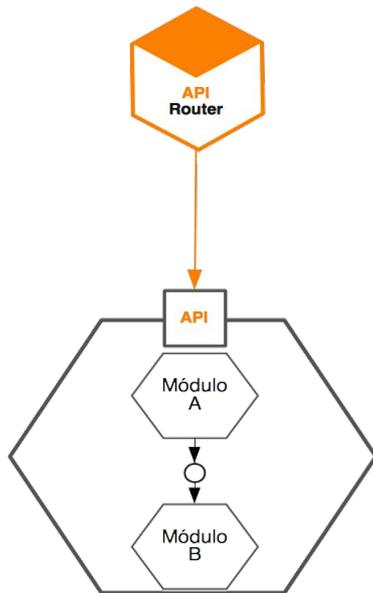
# Transformación

Meter el cuchillo y cortar.

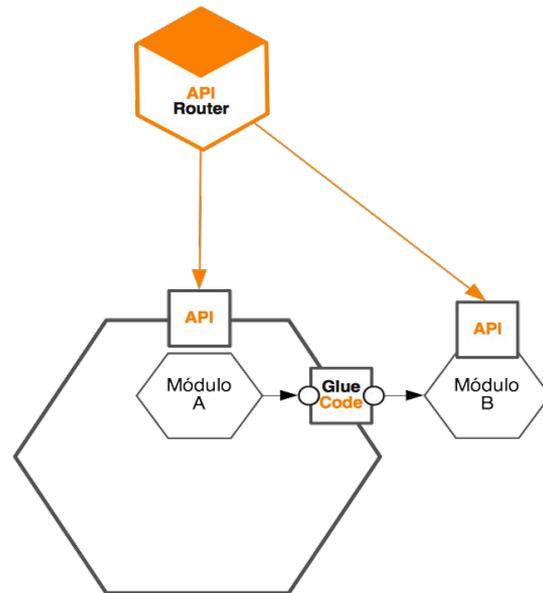
Identificar el módulo a extraer del monolito



Desacoplar el módulo de otros a través de una fachada



Reescribir el módulo fuera del monolito e integrarlo





# Resultado de la hoja de ruta

Una visión realista





**¡Necesitamos  
un Big Bang!**



04.

# Migración de código

Nuevos  
planteamientos



# Objetivos

Reducción drástica del proceso de transición

Arquitectura  
de destino

Adaptación  
de la  
herramienta

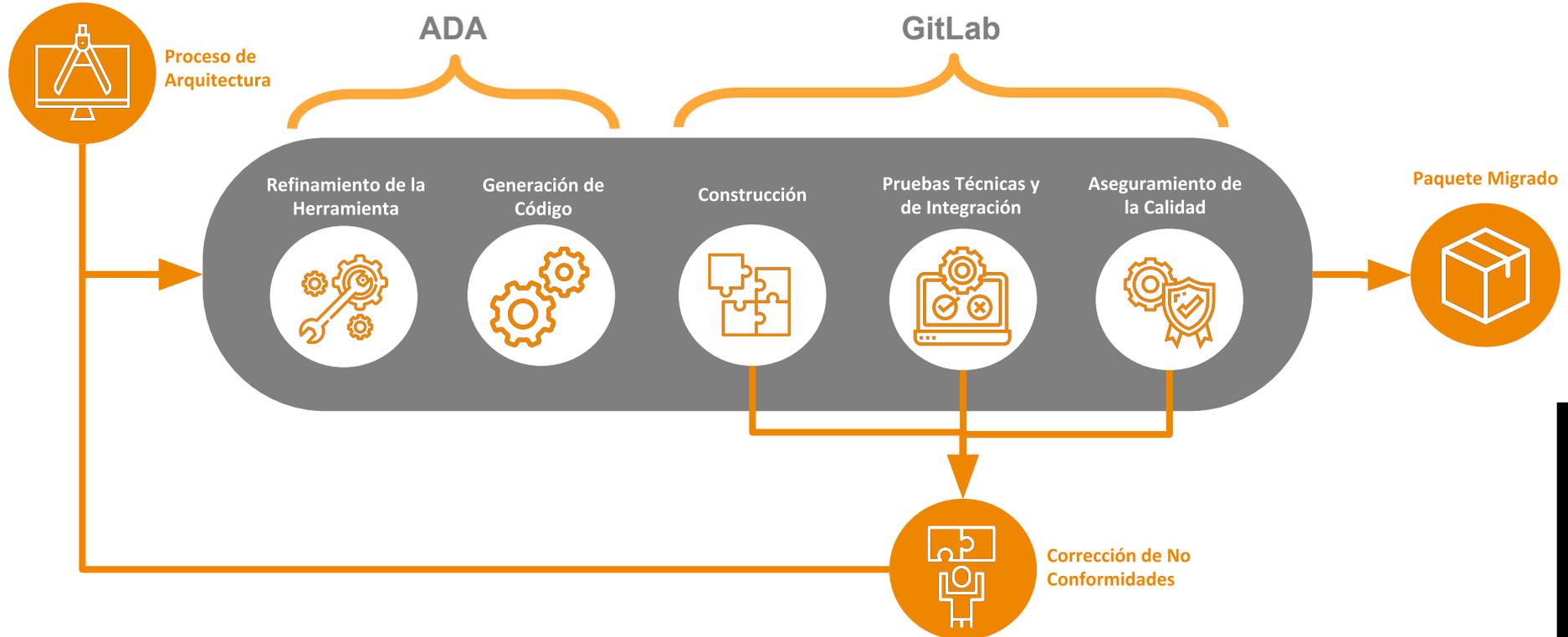
Validación  
de los  
resultados

Migración  
del código



# Proceso de migración

Descripción detallada del proceso

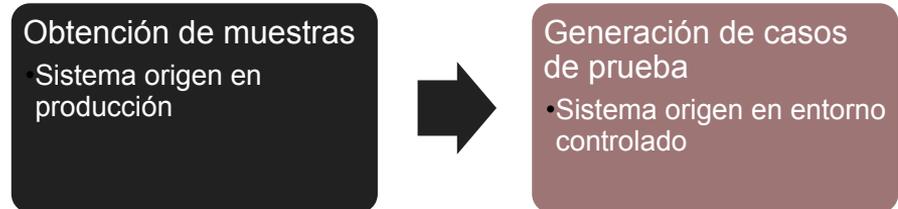




# Casos de prueba

## Proceso de generación de casos de prueba

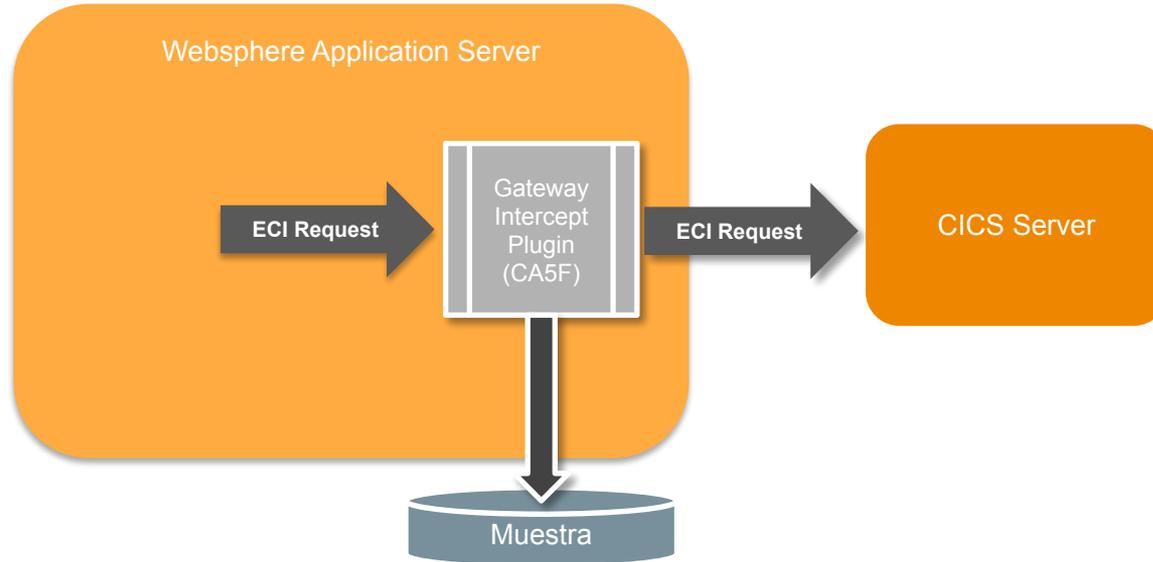
- Con el análisis estático no podemos determinar la intencionalidad del código por lo que las pruebas serán de **caja negra**.
- Generación de los **casos de prueba** en un escenario determinista, que garantice el valor de los resultados de las pruebas.
- Automatización de la grabación de **casos de prueba**, considerando la necesidad de disponer de múltiples muestras en diferentes ventanas temporales, además de las motivadas por los cambios en el código.
- Evitar los falsos positivos en las pruebas, validando por cada **caso de prueba** no sólo la E/S, sino también el estado de la base de datos.
- Sustentar la ejecución de las **pruebas automatizadas** en una plataforma de **CI/CD** con capacidad de escalado horizontal de forma elástica.





# Obtención de muestras

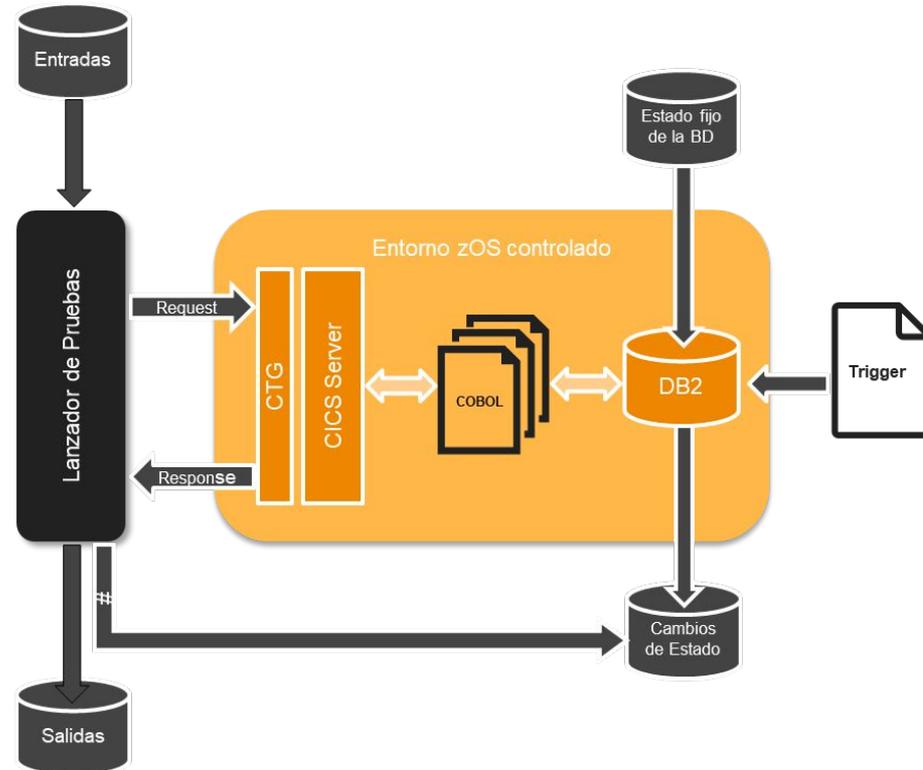
Sistema en producción





# Generación de casos

## Entorno controlado



Gracias por su atención....



Elías Ramos González, 4, ofc. 304  
38001 Santa Cruz de Tenerife

© 2017 Open Canarias