

Christian Javier Bonilla Herrera  
Miguel Angel Guerrero Forigua

Dirigido por:  
Alba Consuelo Nieto Lemus  
Co-Director:  
Henry Alberto Diosa

Integración de motores de  
orquestración de procesos y de  
gestión de reglas de negocio a un  
sistema legado basado en  
componentes.  
Caso de estudio SOA-FenixSGA



UNIVERSIDAD DISTRITAL  
FRANCISCO JOSÉ DE CALDAS



# Agenda



1

## Introducción

- Descripción del problema
- Objetivo General
- Objetivos Específicos

2

## Marco Referencial

- Software Legado
- SOA
- Bus Empresarial de Servicios
- Orquestación
- BPMS
- BRMS

3

## Desarrollo del proyecto

- Metodología Híbrida SOA + DSBC
- Demostración
- Pruebas

4

## Conclusiones

- Trabajo Futuro
- Preguntas

# Descripción del problema



Motivado por FenixSGA: Sistema de Gestión Académica para instituciones de educación superior propuesto como solución a la escalabilidad.

- SOA
- Base de Datos Distribuida
- Gestiona procesos de matrícula e inscripción de asignaturas



# Objetivo General



Extender el desarrollo realizado para la primera versión de FenixSGA utilizando un motor de orquestación de procesos de negocio y un motor de reglas de negocio para componer dinámicamente los servicios asociados a un proceso.

# Objetivos Específicos



- Definir una estrategia de reutilización de artefactos de diseño y componentes de software del desarrollo anterior.
- Identificar e implementar reglas de negocio que determinen el flujo de ejecución de actividades de una nueva funcionalidad que dará soporte a un proceso de negocio utilizando un motor de reglas de negocio.
- Evaluar las condiciones de integración del motor de procesos de negocio y del motor de reglas de negocio con el bus de servicios de FenixSGA.
- Integrar y configurar un motor de orquestación de procesos de negocio para controlar dinámicamente el flujo de ejecución de los servicios.
- Diseñar y aplicar pruebas a los nuevos procesos de negocio automatizados para evaluar su desempeño, robustez, mantenibilidad y extensibilidad.

# Marco Referencial

Created in

Master

PDF

Editor

Created in

Master

PDF

Editor

# Software Legado



COBOL  
fortran



Representación de software legado

Fuente: Commodore 64C system with 1541-II floppy drive and 1084S RGB monitor. Bill Bertham

Un software o sistema legado es una solución informática que automatiza procesos clave para un negocio que es difícil de mantener o que ha sido desarrollada utilizando tecnología que resulta obsoleta.

# Modernización de Software Legado



## Re-Desarrollo

Desarrollar nuevamente el sistema

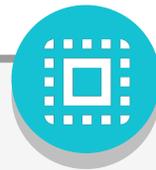
- GUI



## Envoltura

Envolver interfaces existentes exponiendo nuevas interfaces

- Unidades de servicio



## Migración

Mover el sistema a una nueva plataforma



# SOA



Arquitectura Orientada a Servicios por sus siglas en inglés. Es un estilo arquitectural utilizado para construir soluciones empresariales basadas en servicios.

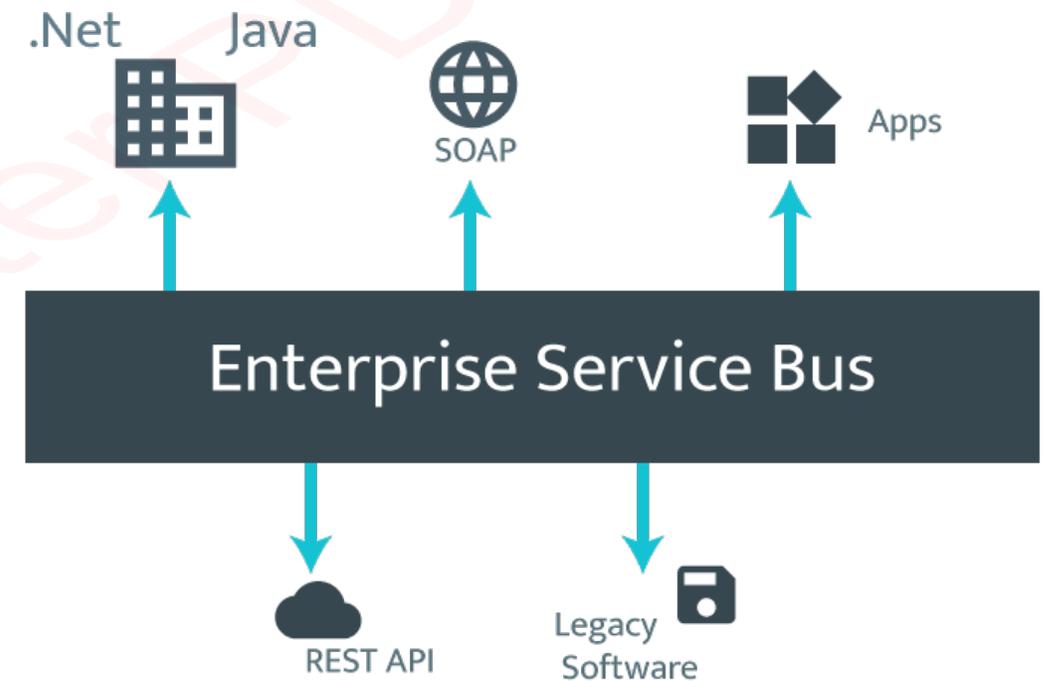
SOA promete:

- Flexibilidad
- Reutilización de servicios tecnológicos
- Integración de sistemas existentes

# Bus empresarial de servicios (ESB)



Un bus empresarial de servicios es una infraestructura de middleware que provee virtualización y gestión de interacciones de servicios, incluyendo soporte para la comunicación, mediación, transformación e integración de tecnologías requeridas por los servicios.



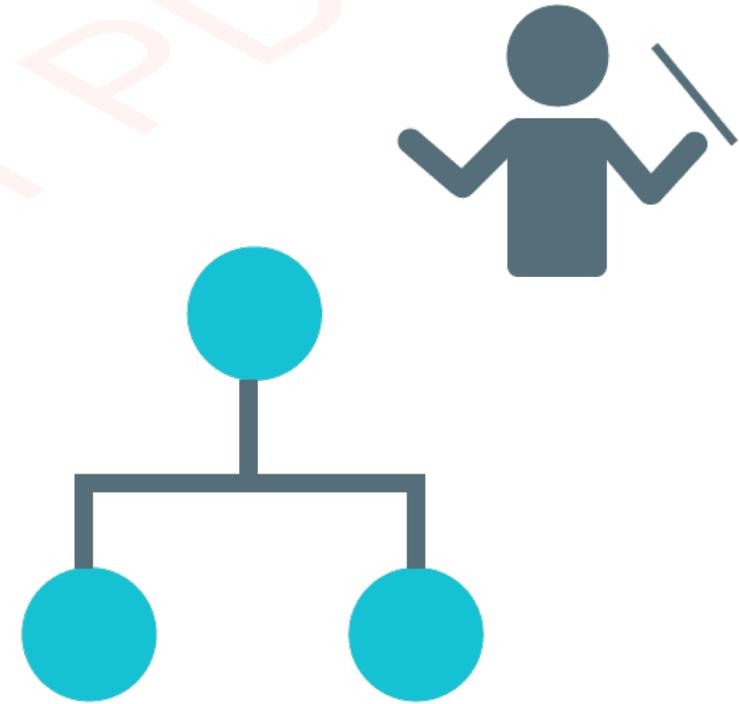
Representación de bus empresarial de servicios

Fuente: Este trabajo

# Orquestación de servicios



Es el arreglo, coordinación y gestión automatizada de servicios. Es un tipo de composición de servicios que describe el flujo de trabajo de cómo interactúan los servicios, incluyendo lógica de negocio y el orden de las interacciones.



Representación de orquestación de servicios

Fuente: Este trabajo

# Motor de procesos de negocio



Es un sistema que permite la ejecución de procesos de negocio expresados en un lenguaje de orquestación de procesos de negocio (BPEL).



# Motor de reglas de negocio



Un motor de reglas de negocio, permite evaluar declaraciones que definen o limitan algún aspecto del negocio, expresadas en un lenguaje ejecutable de reglas de negocio (Drools).



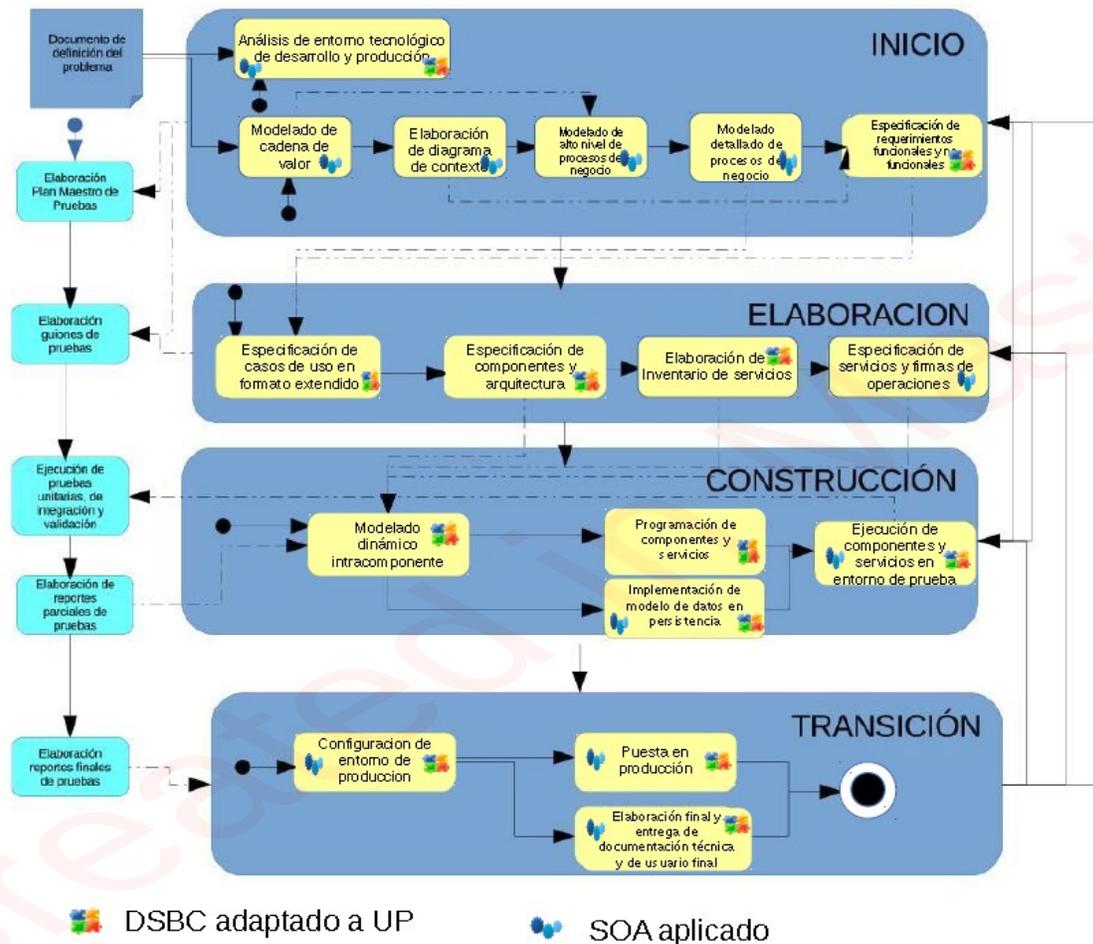
WS<sub>2</sub>

Business Rules Server

Desarrollo del proyecto



# Metodología Híbrida SOA + DSBC



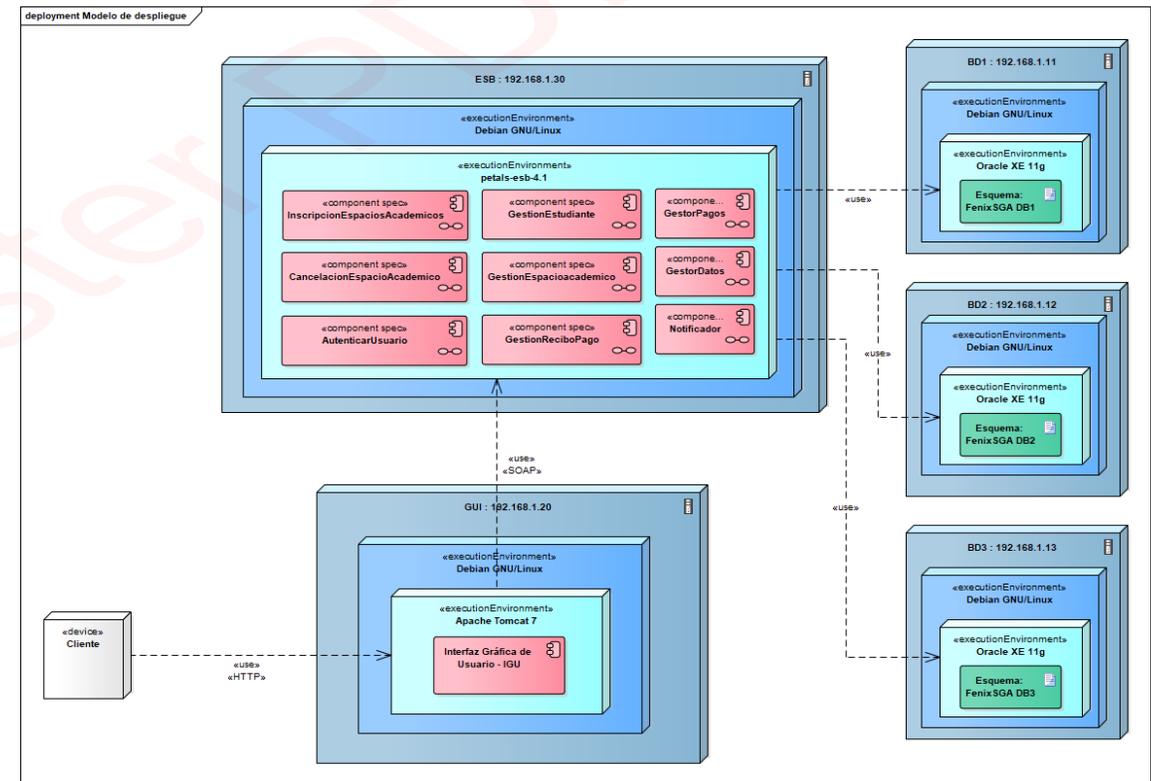
**Ilustración de híbrido metodológico**

Fuente: Alba Consuelo Nieto L. Diego Alejandro Maldonado C. Henry Alberto Diosas Sandra Patricia Cala D. Híbrido metodológico SOA + DSBC para el desarrollo de software orientado a servicios.

# Análisis del entorno tecnológico



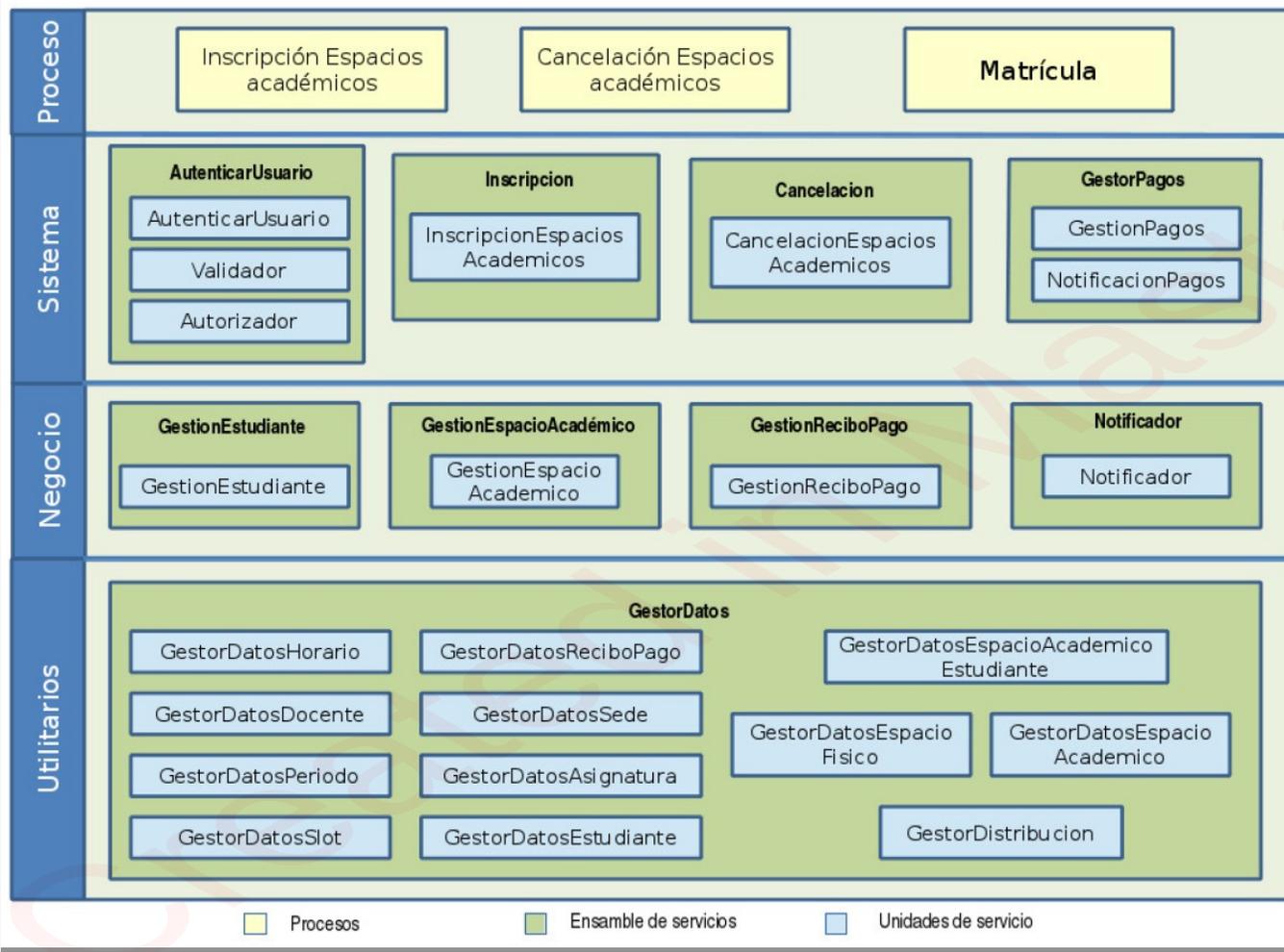
- Bus Empresarial de Servicios: Petals ESB
- Base de datos distribuida: Oracle XE
- Servidor de interfaz gráfica de usuario: Apache Tomcat 7



Representación de Diagrama de despliegue para FenixSGAv1

Fuente: Este trabajo

# Inventario de servicios



## Inventario de servicios para FenixSGAv1

Fuente: Sandra Cala, Diego Maldonado y David Peña. Desarrollo de un prototipo de software para gestionar los procesos universitarios de matrícula e inscripción de espacios académicos usando ISOS(Ingeniería de Software Orientada a Servicios)

# Procesos de una IES

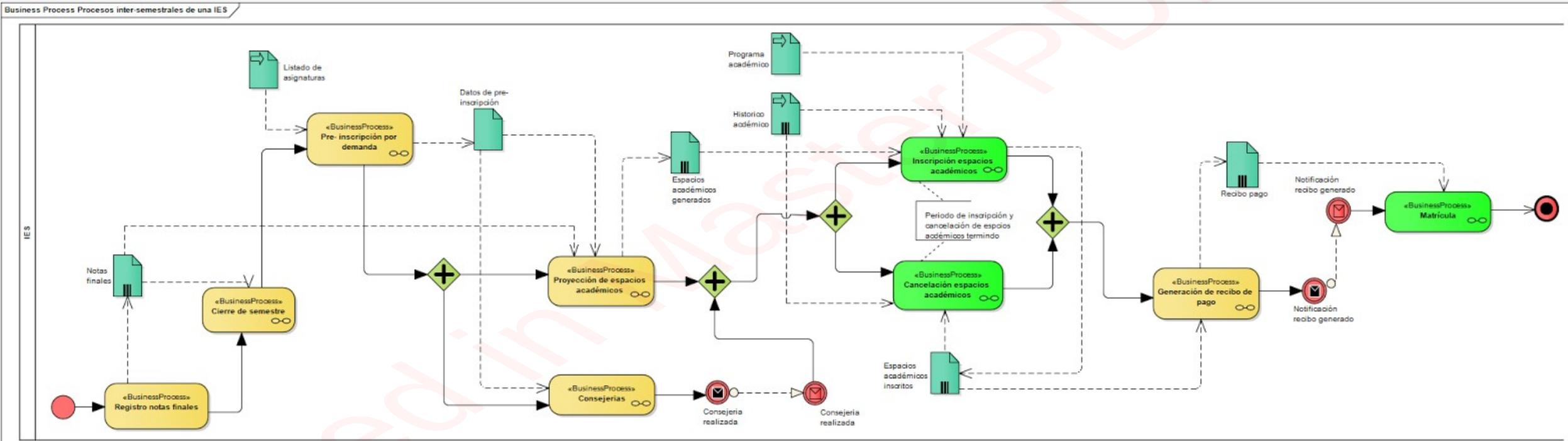


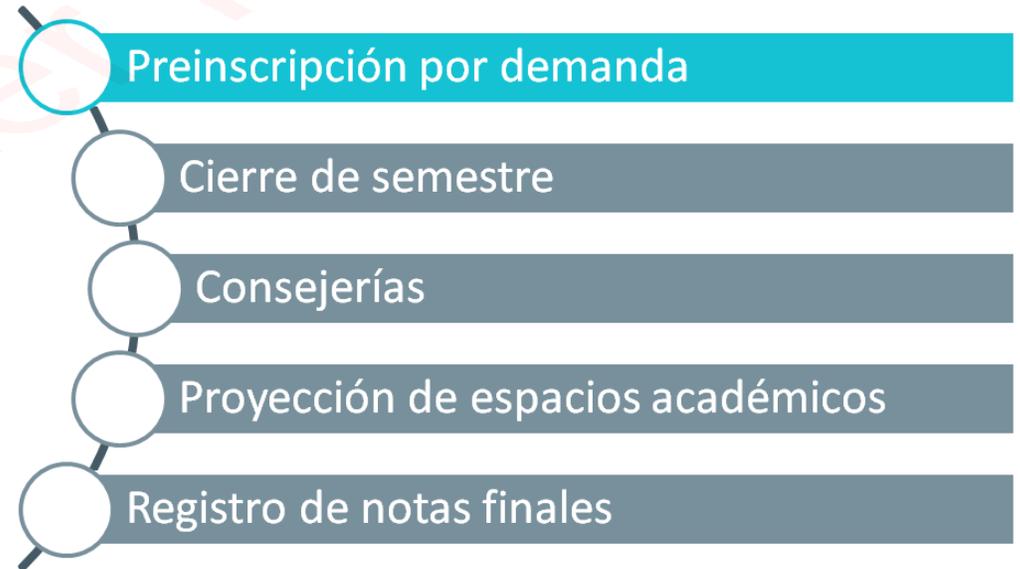
Diagrama de procesos de negocio de alto nivel

Fuente: Sandra Cala, Diego Maldonado y David Peña. Desarrollo de un prototipo de software para gestionar los procesos universitarios de matrícula e inscripción de espacios académicos usando ISOS(Ingeniería de Software Orientada a Servicios)

# Criterios de Selección de Proceso de Negocio



- Debe ser un proceso de negocio que no se haya implementado en una versión anterior del aplicativo.
- Debe ser un proceso de negocio que permita la orquestación de servicios. Es decir, debe incluir varios servicios, posiblemente con distintos flujos de ejecución de acuerdo al contexto.
- Debe ser un proceso de negocio que incluya reglas de decisión cambiantes que puedan implementarse usando el motor de reglas de negocio.
- Debe ser un proceso de negocio que incluya por lo menos un servicio desarrollado en una versión anterior del aplicativo.



# Proceso de negocio detallado

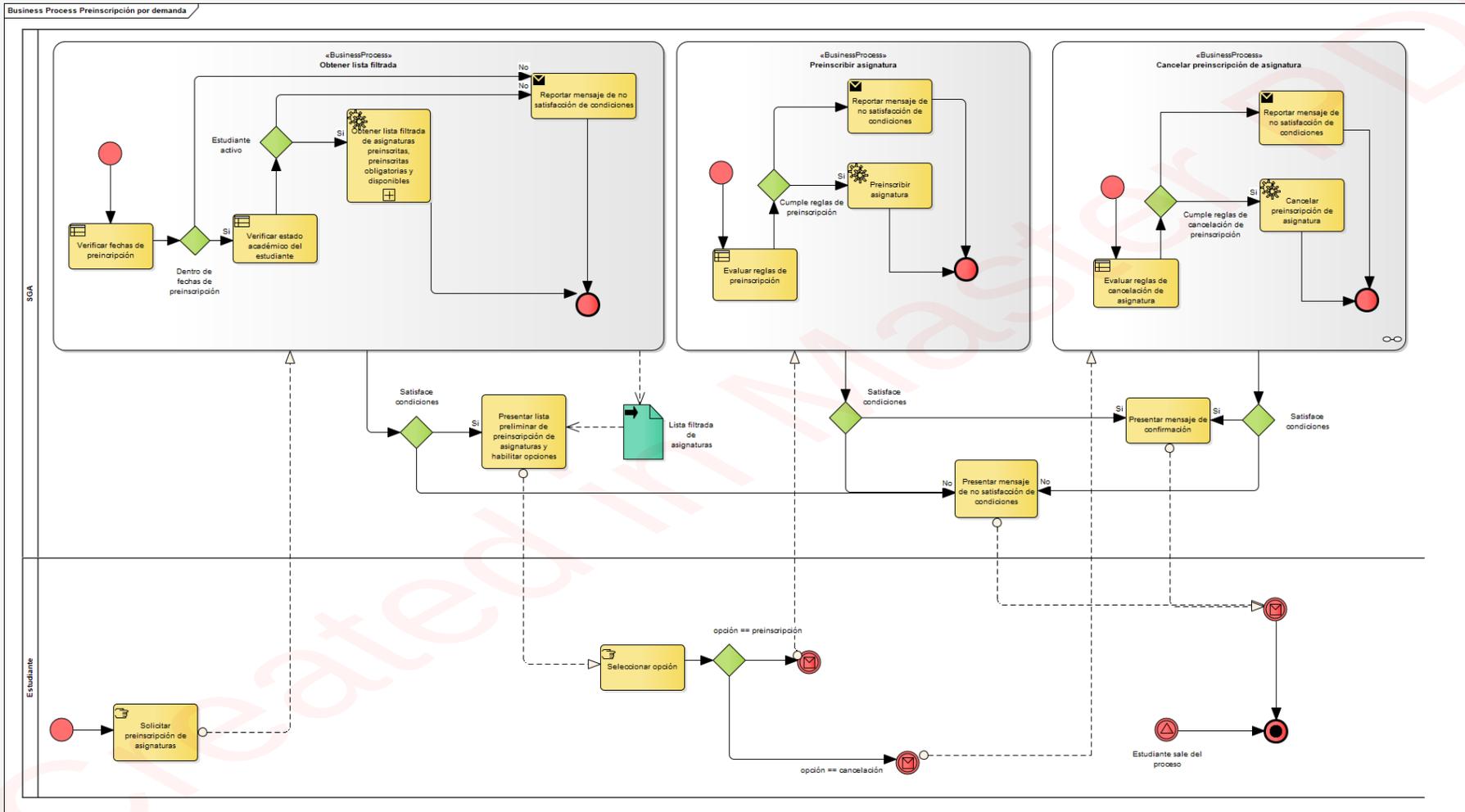


Diagrama de proceso de negocio detallado del proceso preinscripción de asignaturas

Fuente: Este trabajo

# Diagrama de casos de uso

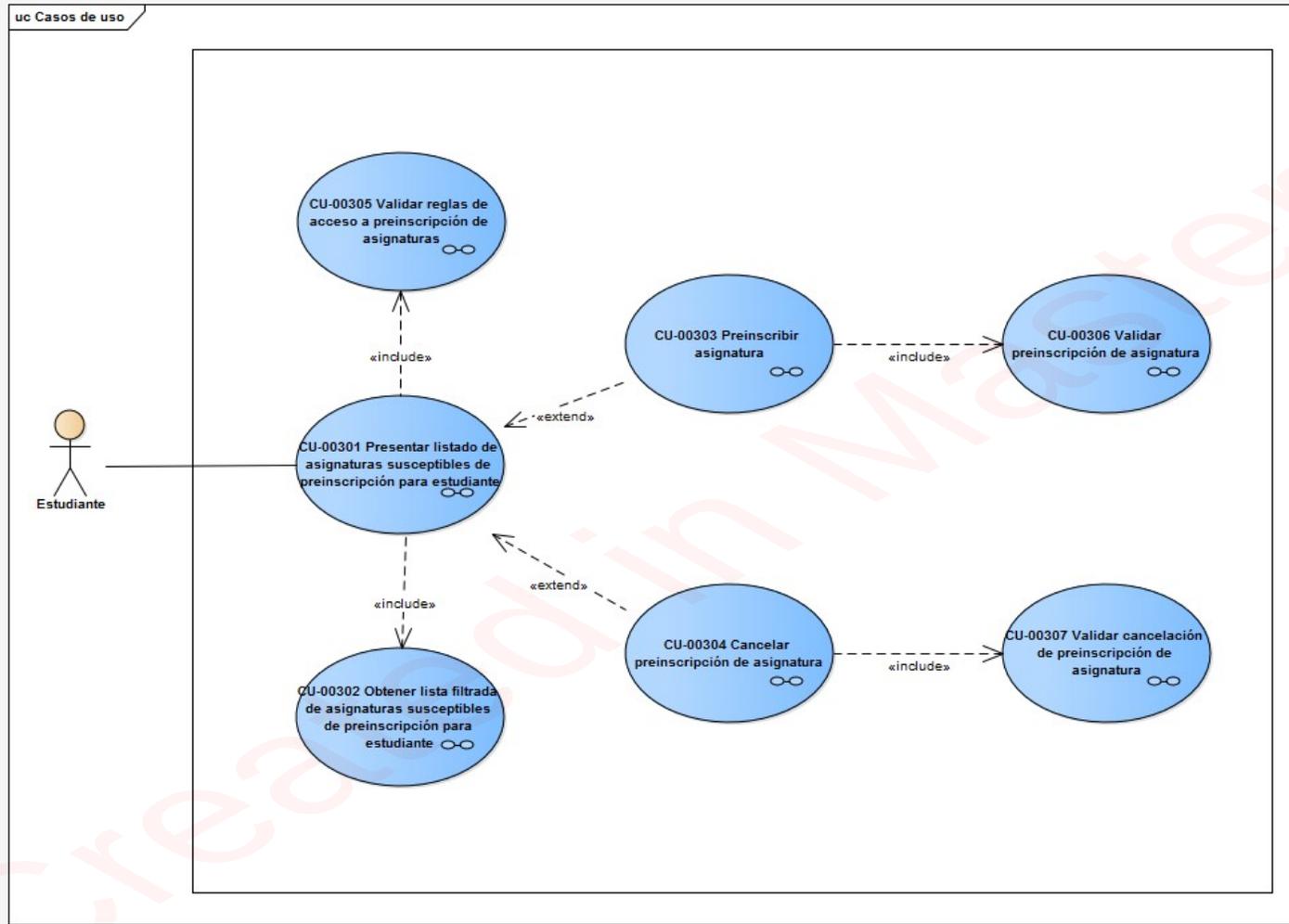
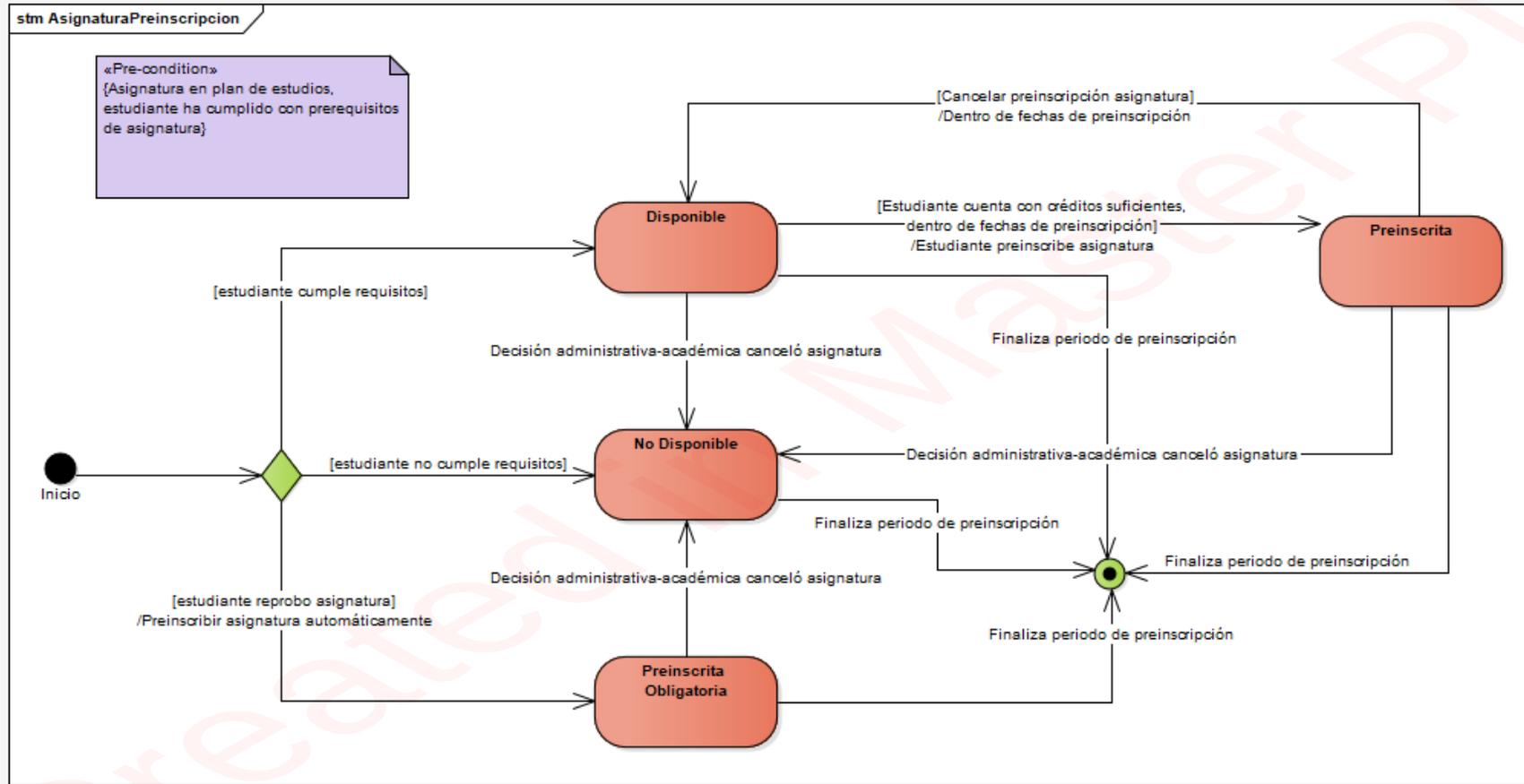


Diagrama de casos de uso  
para preinscripción de  
asignaturas

Fuente: Este trabajo

# Maquina de estados

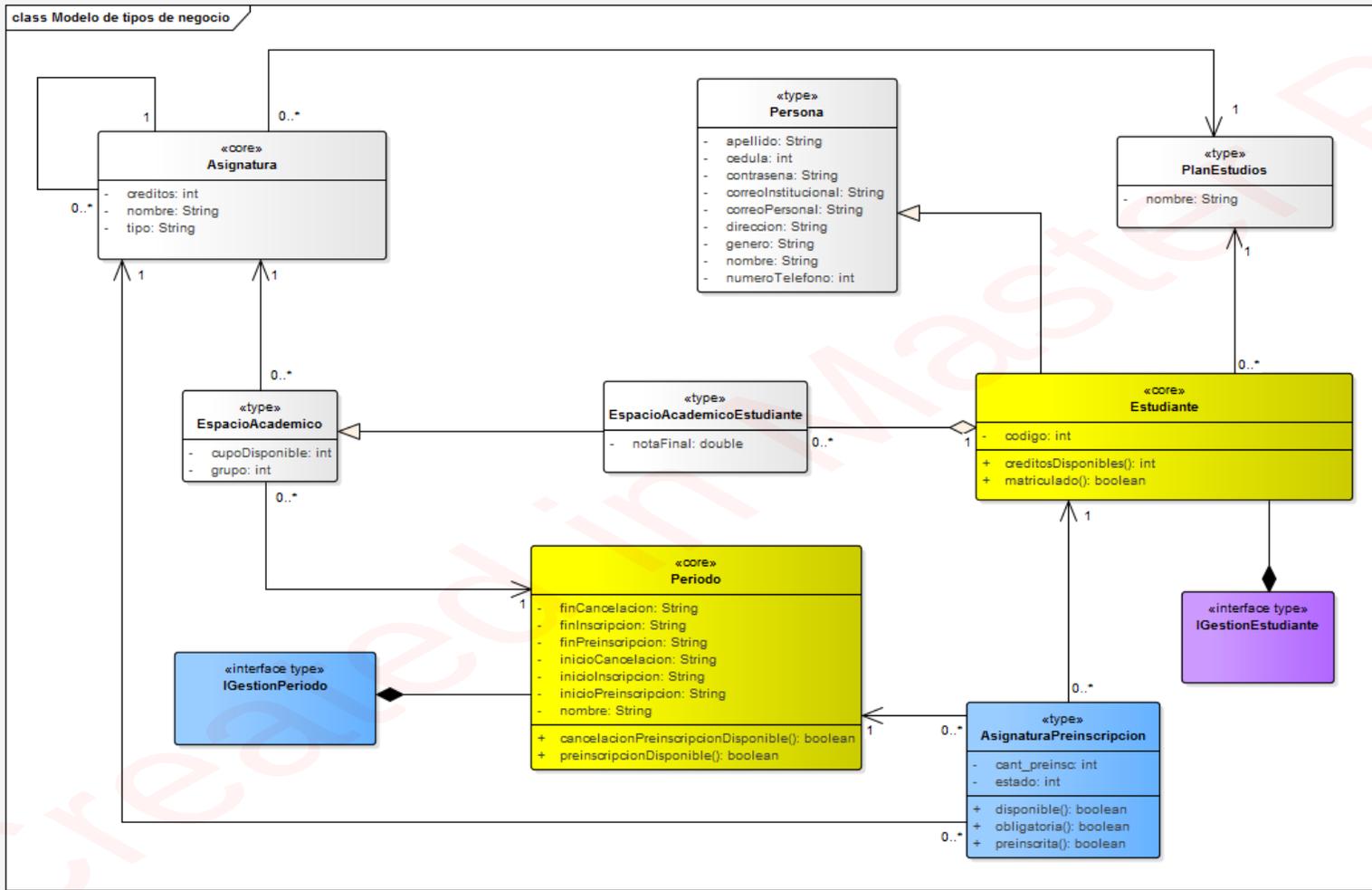
Preinscripción de asignaturas



**Maquina de estados para una asignatura susceptible a preinscripción**

Fuente: Este trabajo

# Modelo de tipos de negocio



Modelo de tipos de negocio  
para soportar preinscripción  
de asignaturas

Fuente: Este trabajo

# Interacción inter-componente

Ejemplo

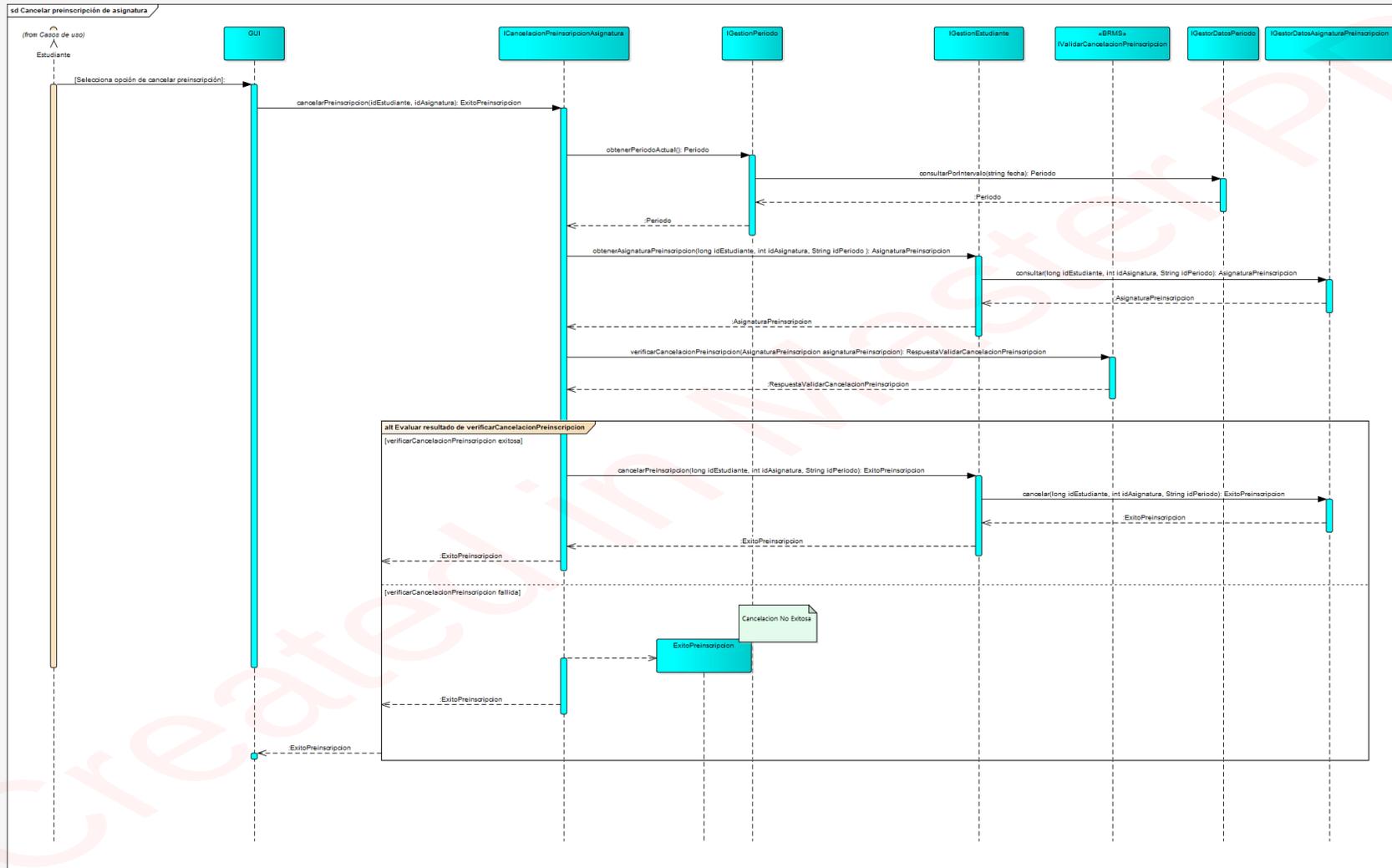
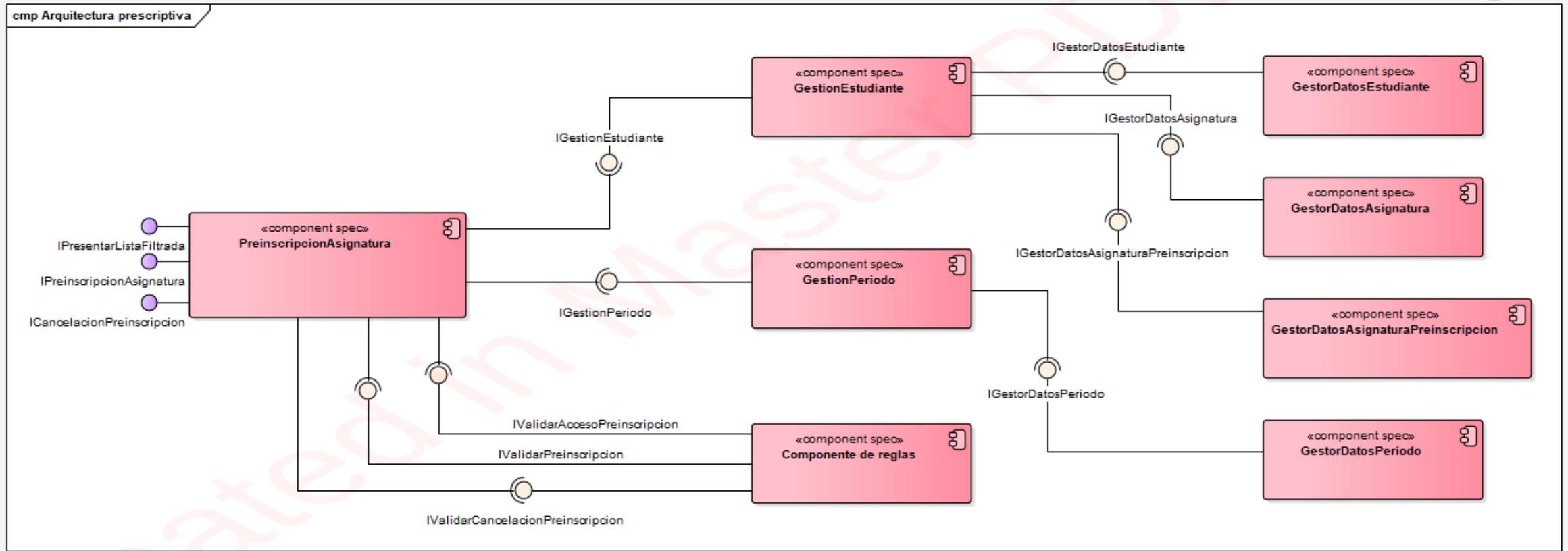


Diagrama de Interacción inter-componente para cancelar preinscripción de asignatura

Fuente: Este trabajo

# Arquitectura prescriptiva

FenixSGAv2

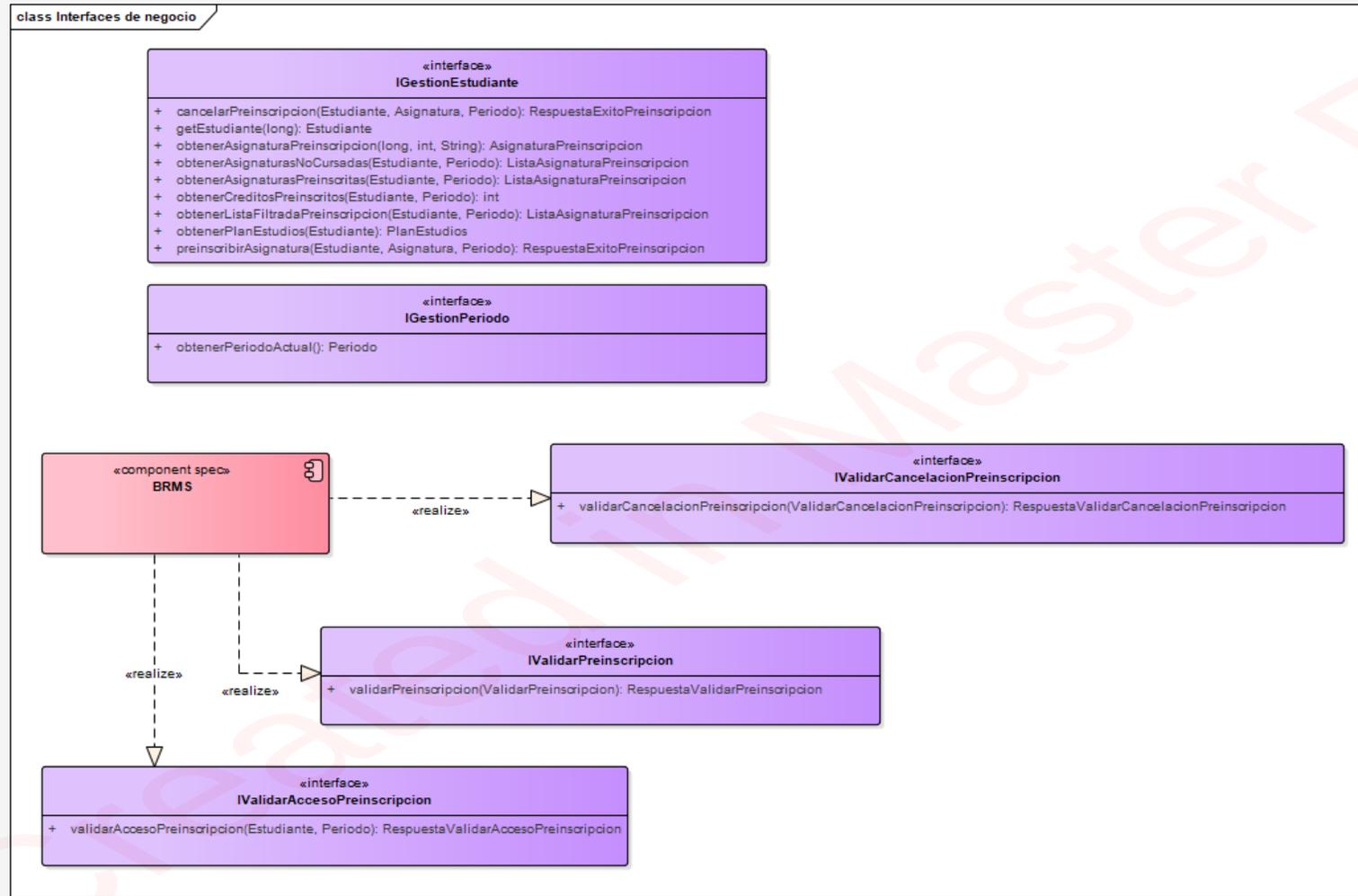


Arquitectura prescriptiva para FenixSGAv2

Fuente: Este trabajo

# Especificación de interfaces de servicios

Ejemplo

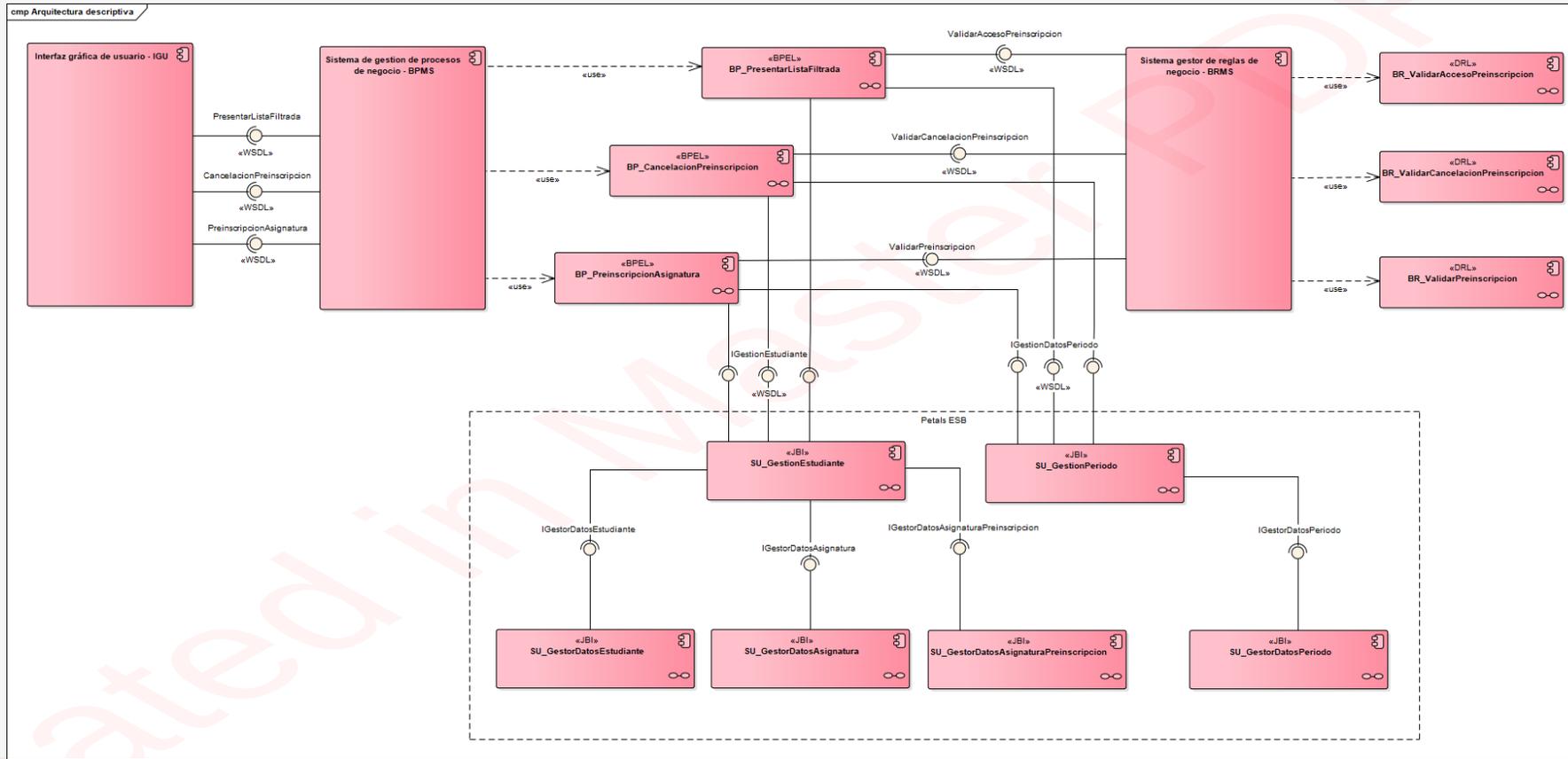


**Diagrama de especificación de interfaces de negocio para FenixSGAv2**

Fuente: Este trabajo

# Arquitectura descriptiva

FenixSGAv2



Arquitectura descriptiva para FenixSGAv2

Fuente: Este trabajo

# Inventario de servicios

FenixSGAv2

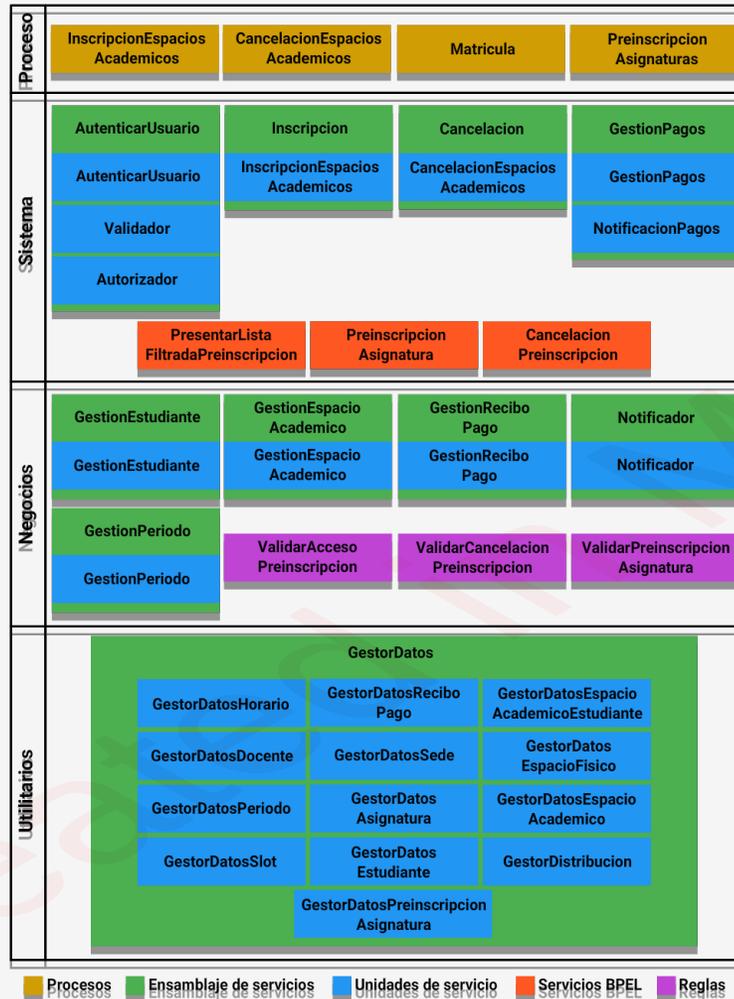


Diagrama de inventario de servicios para FenixSGAv2

Fuente: Este trabajo

# Demostración

Created in [www.dreamstime.com/stock-photo-editor](http://www.dreamstime.com/stock-photo-editor)

# Diagrama de despliegue

FenixSGAv2

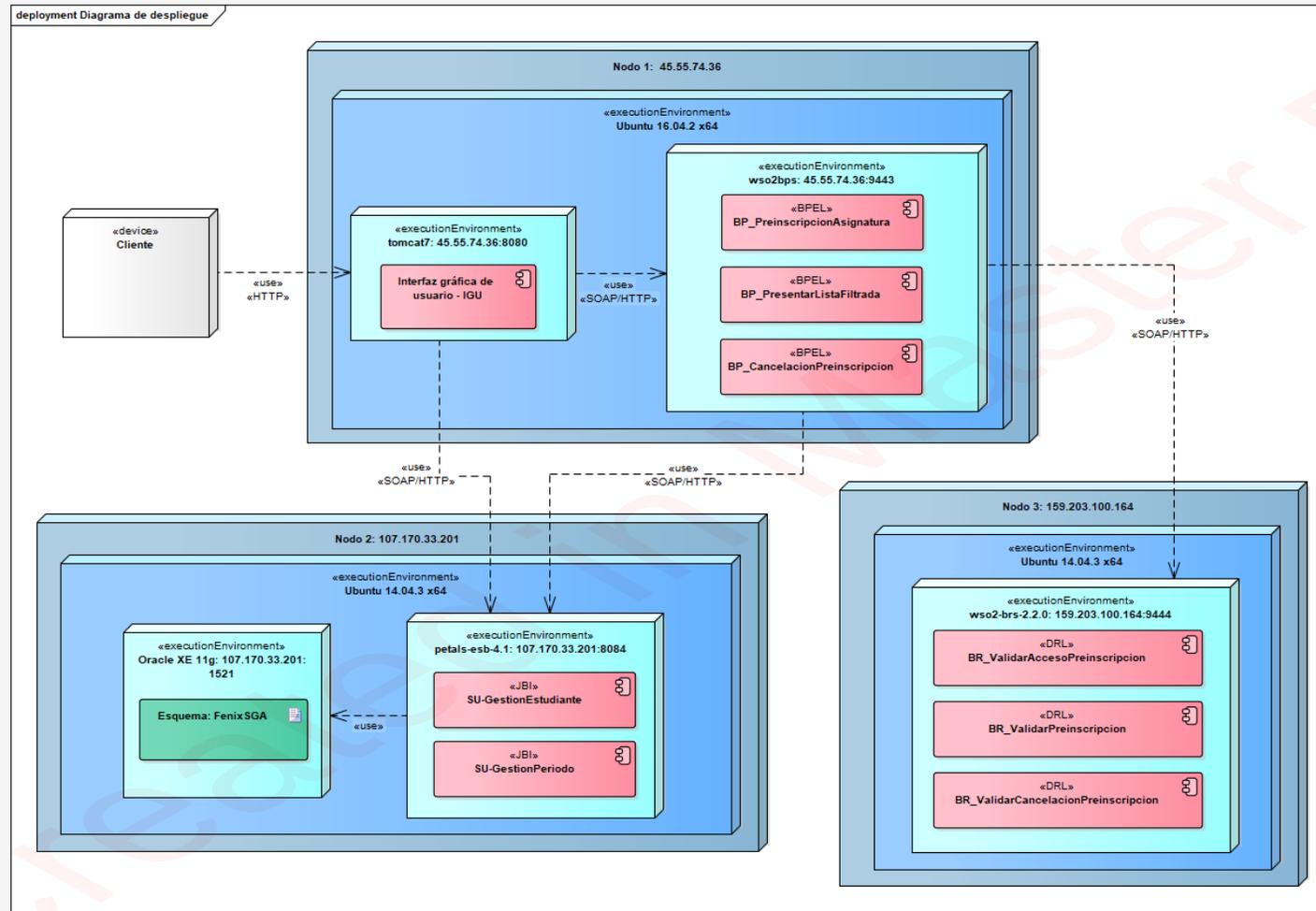


Diagrama de despliegue para FenixSGAv2

Fuente: Este trabajo

# Pruebas

# Pruebas funcionales



- Pruebas de servicios
- Pruebas de integración de servicios

## Resultados

- 500 casos de pruebas
- 38 fallos (18 solucionados)



■ No solucionados ■ Casos de prueba ■ Solucionados

# Pruebas no funcionales



## Pruebas realizadas:

- Pruebas de línea base
- Pruebas de carga
- Pruebas de estrés
- Comparación de rendimiento (servicio orquestado versus coreografiado)
- Análisis de mantenibilidad y extensibilidad

## Resultados:

- Línea base 50 P/s;  $t < 1.5s$ .
- Respuestas aceptables hasta 200 P/s;  $t < 4s$ .
- Punto de quiebre 1500 P/s.
- El servicio coreografiado tiene un tiempo de respuesta un 45% más rápido.
- 89.8% de conformidad

# Conclusiones



- La documentación y el modelo de referencia con los que contaba la primera versión del prototipo apoyaron el desarrollo de la nueva iteración permitiendo conservar principios de calidad de software como separación de responsabilidades, modularidad y flexibilidad.
- La metodología híbrida utilizada dio una guía para el diseño y elaboración de componentes que se construyeron como servicios.
- La orquestación de procesos de negocio permite componer dinámicamente servicios de negocio en un flujo fácil de entender.
- Es necesario evaluar la volatilidad de las reglas de negocio para determinar si es necesario desarrollar una regla de negocio.

# Conclusiones



- Se logró representar el flujo de objetos dentro de los servicios de procesos de negocio utilizando diagramas dinámicos inter-servicio mediante la notación UML.
- Las pruebas funcionales permitieron descubrir fallos relacionados a aspectos no funcionales que no se tuvieron en cuenta en el diseño del prototipo.
- Las pruebas de rendimiento realizadas sobre dos versiones del mismo servicio, evidenciaron un costo en rendimiento en la composición por orquestación.

# Trabajo futuro



- Implementar los procesos de negocio restantes para obtener un prototipo funcional que pueda llevarse a fase de producción.
- Implementar el proceso de proyección de espacios académicos mediante el uso de técnicas de inteligencia artificial.
- Implementar características no funcionales contempladas por el modelo de referencia SOA, como: seguridad, gobernanza y versionamiento de servicios.



¡Gracias!

¿Preguntas?