

Cambios por seguridad en App.Configs | Web.Configs

Diseño técnico

Contenido

Introducción.....	3
Objetivo.....	3
Audiencia.....	3
Alcance.....	3
Diagrama de Modelo de Dominio.....	4
Diagrama de Componentes.....	6
Diagrama de Secuencia.....	7
Diagrama de Despliegue.....	8
Stack tecnológico.....	9

Introducción

Objetivo

El objetivo de este documento es brindar una especificación de la solución para el proyecto de Envío de ticket por correo.

Audiencia

Este documento va dirigido principalmente a los integrantes del equipo de desarrollo pero también al equipo de soporte y arquitectura que estén interesados en entender la manera en la que se plantea la solución de requerimiento.

Alcance

Este documento contempla la descripción y modelado de:

- Diagrama de modelo de dominio
- Diagrama de componentes
- Diagrama de secuencia
- Diagrama de despliegue

Diagrama de Modelo de Dominio

El siguiente diagrama muestra las clases involucradas que intervienen en la generación y envío de ticket por correo electrónico por medio de ORIGIS.

Las clases principales son:

- SaleView y ReprintView son clases de la aplicación de POS que representan la funcionalidad e interfaces de usuario que se verán afectadas por la nueva funcionalidad de envío de ticket por correo electrónico, habrá cambios en el flujo y captura de información.
- ORIGIS representa el servicio ofrecido por el proveedor ORIGIS.
- Ticket representa la información del ticket necesaria para generar el pdf, además de datos de la transacción como el folio, itu y monedero del cliente más datos de auditoría como fecha de creación, usuario, etc.
- TicketManager servicio que gestiona la información de los tickets y la generación de pdf de los mismos, para la generación se apoya en un TemplateEngine, búsqueda por folio e itu.
- TemplateEngine representa un componente que genera un pdf dado los datos de un ticket y una plantilla, para la solución se plantea reutilizar el motor o mecanismo que utiliza el POS, dejando la posibilidad de que en un futuro se incorporen otros tipos de motores basado por ejemplo en FOP, Html, etc.
- TicketMessageProcessor componente que procesa los mensajes de almacenamiento de la cola de operaciones de Farmax.StoreService.
- TicketMailerMessageProcessor componente que procesa envío de ticket por correo en cola de operaciones de Farmax.StoreService.

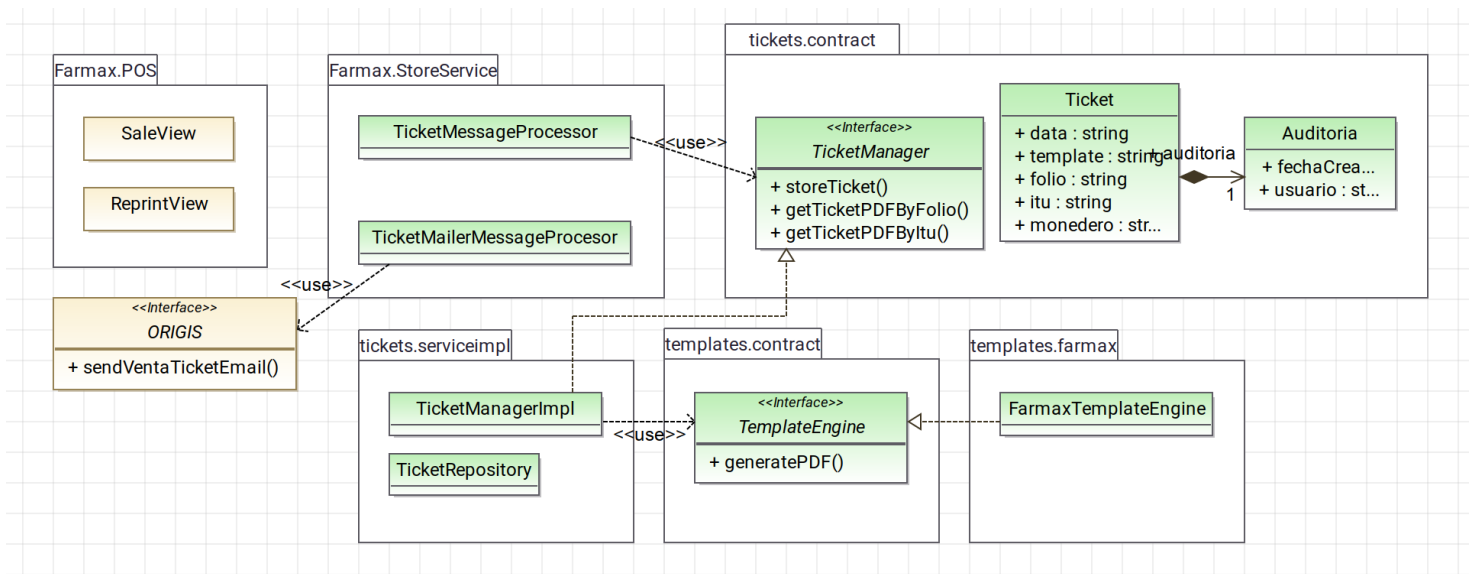




Diagrama de Componentes

En el siguiente diagrama muestra cómo se relacionan entre sí los diferentes componentes en términos de qué interfaces o contratos estos proveen o requieren.

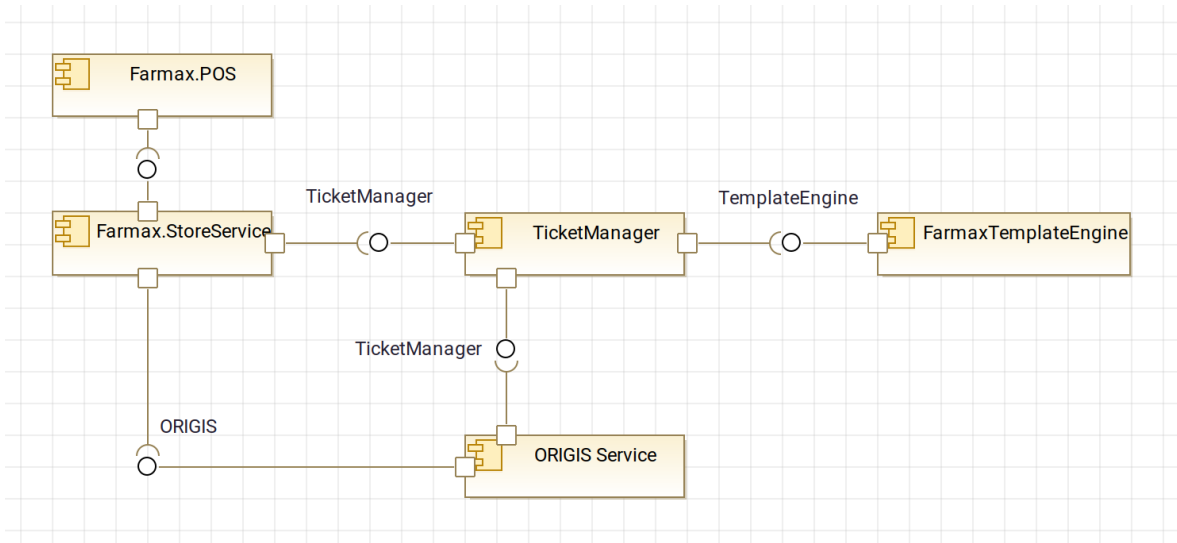


Diagrama de Secuencia

El siguiente diagrama muestra la interacción entre los componentes u objetos para llevar a cabo una funcionalidad, para el caso específico el envío de ticket por correo electrónico.

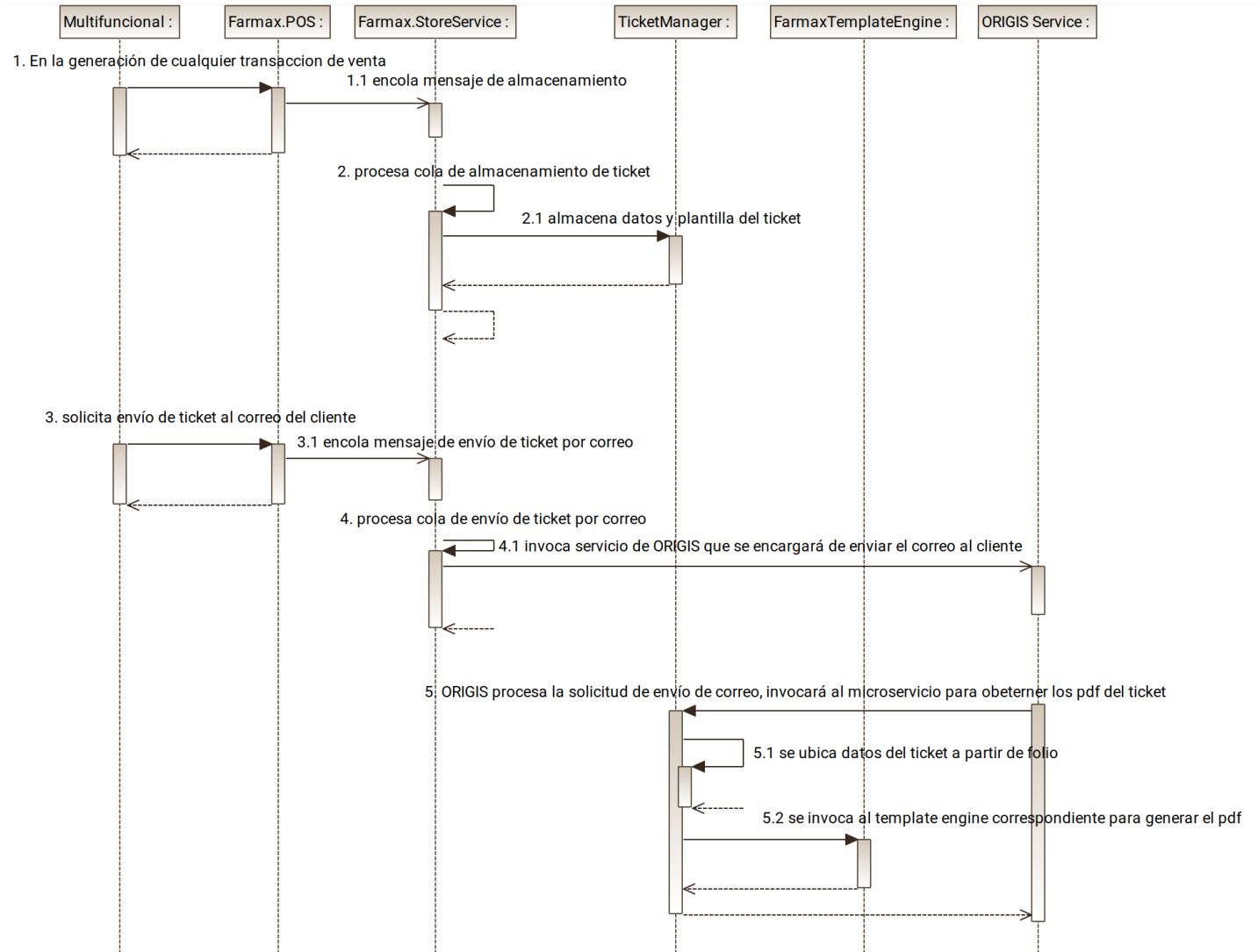
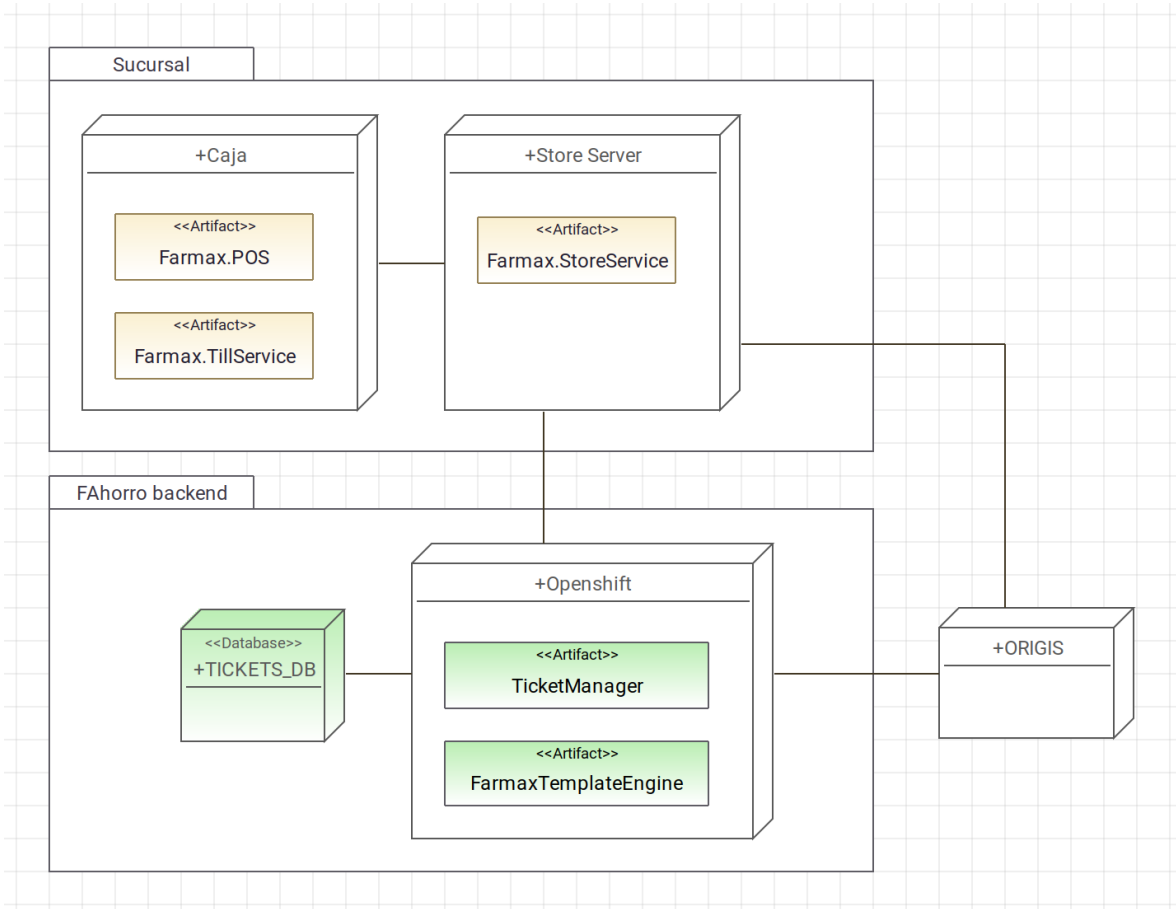


Diagrama de Despliegue

El siguiente diagrama muestra los diferentes artefactos alojados en sus correspondientes nodos. Los microservicios de almacenamiento y generación de ticket se distribuyen como contenedores y estarán desplegados en Openshift.



Stack tecnológico

1. Quarkus para microservicio TicketManager.
2. .NET Core para microservicio de FarmaxTemplateEngine.
3. .NET Framework para las modificaciones en los proyectos del Farmax POS y StoreService.