**DOCUMENTO RESUMIDO - DESCRIPCION DE ARQUITECTURA**

***Obligatorio Arquitectura de Software***

***27-11-2014***

***143904 Richard Caetano***

***156773 Fernando Godoy***

**Índice**

[1. Introducción 3](#_Toc391389969)

[1.1 Propósito 3](#_Toc391389970)

[2. Antecedentes 3](#_Toc391389971)

[2.1 Propósito del sistema 3](#_Toc391389972)

[2.2 Requerimientos significativos de Arquitectura 3](#_Toc391389973)

[2.2.1 Resumen de Requerimientos Funcionales 3](#_Toc391389974)

[2.2.2 Resumen de Requerimientos NoFuncionales 3](#_Toc391389975)

[3. Documentación de la arquitectura 4](#_Toc391389976)

[3.1 Vistas de Módulos 4](#_Toc391389977)

[3.1.1 Vista de Descomposición 4](#_Toc391389978)

[3.1.1.1 Representación primaria 4](#_Toc391389979)

[3.1.1.2 Catálogo de elementos (opcional) 5](#_Toc391389980)

[3.1.1.3 Decisiones de diseño (opcional) 5](#_Toc391389981)

[3.1.2 Vista de Uso 5](#_Toc391389982)

[3.1.2.1 Representación primaria 5](#_Toc391389983)

[] 5](#_Toc391389984)

[3.1.2.2 Catálogo de elementos (opcional) 5](#_Toc391389985)

[3.1.2.3 Decisiones de diseño 5](#_Toc391389986)

[3.1.3 Vista de Layers 6](#_Toc391389987)

[3.1.3.1 Representación primaria 6](#_Toc391389988)

[] 6](#_Toc391389989)

[3.1.3.2 Catálogo de elementos (opcional) 6](#_Toc391389990)

[3.1.3.3 Decisiones de diseño (opcional) 6](#_Toc391389991)

[3.1.3.4 Otras vistas de módulos 6](#_Toc391389992)

[3.1.4 Catálogo de elementos 6](#_Toc391389993)

[3.1.5 Interfaces 7](#_Toc391389994)

[3.1.6 Comportamiento 7](#_Toc391389995)

[3.2 Vistas de Componentes y conectores 8](#_Toc391389996)

[3.2.1 Vista de Tiers 8](#_Toc391389997)

[3.2.1.1 Representación primaria 8](#_Toc391389998)

[[Diagramas ] 8](#_Toc391389999)

[3.2.1.2 Catalogo de elementos 8](#_Toc391390000)

[3.2.1.3 Interfaces 8](#_Toc391390001)

[3.2.1.4 Comportamiento 8](#_Toc391390002)

[3.2.1.5 Relación con elementos lógicos 9](#_Toc391390003)

[3.2.1.6 Decisiones de diseño 9](#_Toc391390004)

[3.2.1.7 Otras vistas de C&C 9](#_Toc391390005)

[3.3 Vistas de Asignación 10](#_Toc391390006)

[3.3.1 Vista de Despliegue 10](#_Toc391390007)

[3.3.1.1 Representación primaria 10](#_Toc391390008)

[3.3.1.2 Catalogo de elementos 10](#_Toc391390009)

[3.3.1.3 Relación con componentes 11](#_Toc391390010)

[3.3.1.4 Decisiones de diseño 11](#_Toc391390011)

# Introducción

Propósito

El propósito del presente documento es proveer una especificación completa de la arquitectura del ***Sistema*** *TuRecetaAlToque*

# Antecedentes

Propósito del sistema

*TuRecetaAlToque se trata de una aplicación que permite a esa persona que llega a su hogar y no sabe que puede cocinar, ingresar los ingredientes principales que tiene a su alcance en la cocina y le devolverá una lista de recetas (ordenadas por valoración) que puede realizar con esos ingredientes.*

*Se quiere que las recetas puedan consultar de múltiples plataformas, como ser teléfonos celulares y navegadores. El registro de recetas debe poder realizarse de manera asíncrona*

*Los interesados en esta gran base de datos de recetas y servicios para consultarlas, son los responsables de elGourmetInc.com que se encargaran de desarrollar las distintas interfaces para cada usuario potencial.*

Requerimientos significativos de Arquitectura

### Resumen de Requerimientos Funcionales

|  |  |
| --- | --- |
| **ID Requerimiento** | **Descripción** |
| 1. Registro de usuario | *Permite registrar un usuario brindando algunos datos basicos* |
| 2. Ingreso al sistema | *Permite a un usuario registrado, activar una sesion por 5 minutos* |
| 3. Buscar por ingredientes | *Permite a un usuario activo, realizar una busqueda de recetas ingresando hasta 4 ingredientes basicos* |
| 4. Subir receta | *Permite a un usuario con una sesion activa, agregar una receta al sistema* |
| 5. Valorar receta | *Permite a un usuario con una sesion activa, agregar una valoracion a una receta existente* |
| 6. Top 10 búsquedas | *Permite obtener la lista de ingredientes mas buscados* |
| 7. Top 10 usuarios mejor valorados | *Permite obtener la lista de los usuarios con mejores valoraciones en sus recetas* |

### Resumen de Requerimientos NoFuncionales

**Interoperabilidad:**

RNF 1 -Multiplataforma, la aplicación podrá ser consumida por distintos clientes para el registro/ingreso de usuarios y búsqueda de recetas

**Performance**:

RNF 2 -La funcionalidad de subir y valorar receta será asíncrona

RNF 3 - El tiempo de respuesta es importante, se espera que los usuarios reciban el resultado de una consulta en menos de 5 segs. Las otras operaciones no tienen requerimientos de velocidad de respuesta, aunque deben ser aceptables teniendo como base cualquier aplicación web del mercado.

RNF 4 - Se espera que la aplicación sea usada desde dispositivos móviles, por lo tanto el formato de intercambio de datos deberá ser liviano

**Modificabilidad**:

RNF 5 - Es importante que la aplicación sea independiente del motor de base de datos

RNF 6 - Se debe desarrollar de manera evolutiva/incremental

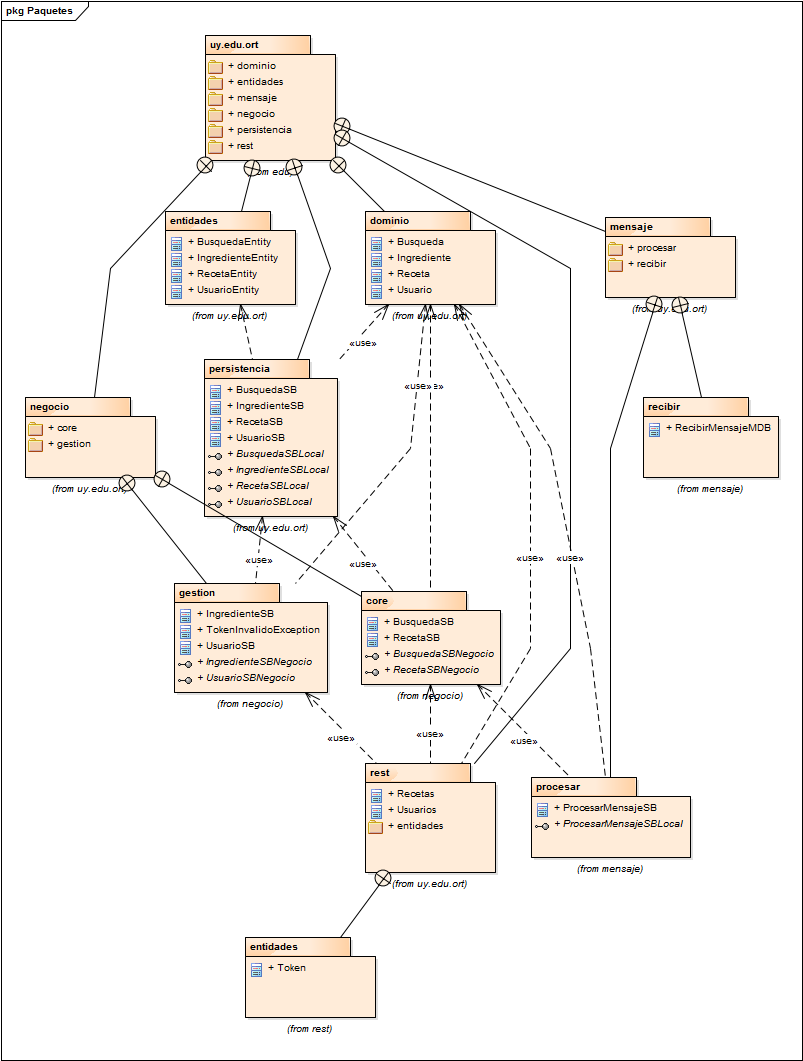
RNF 7 - La comunicación de la aplicación siempre se dará a través del protocolo HTTP pero es importante que la aplicación pueda comunicarse con distintos dispositivos.

# Documentación de la arquitectura

Vistas de Módulos

### Vista de Descomposicion y Uso

#### Representación primaria



#### Catálogo de Elementos

|  |  |
| --- | --- |
| **Elemento** | **Descripción** |
| dominio | Paquete con los DTO del dominio del problema |
| persistencia | Paquete con los sessionBeans para la persistencia. Expone Interfaces para ocultar su implementación |
| entidades | Paquete con las entidades a persistir |
| negocio.core | Paquete con los sessionBeans para las operaciones core del negocio, Recetas y Búsquedas |
| negocio.gestion | Paquete con los sessionBeans para la gestión del negocio, Ingredientes y Usuarios |
| rest | Paquete de Servicios REST/json para las operaciones |
| rest.entidades | Paquete que encapsula el formato de los datos necesarios en los servicios REST |
| mensaje.procesar | Paquete encargado del procesamiento de los mensajes recibidos en la Queue |
| mensaje.recibir | Paquete encargado de la instanciación y recepción de mensajería, invocando al procesador al recibir un mensaje |

**Interfaces**

|  |  |
| --- | --- |
| **Interfaz** | **Descripción** |
| BusquedaSBLocal | Interfaz para abstraer la implementación de la persistencia y recuperación de búsquedas |
| IngredienteSBLocal | Interfaz para abstraer la implementación de la persistencia y recuperación de ingredientes |
| RecetaSBLocal | Interfaz para abstraer la implementación de la persistencia y recuperación de recetas |
| UsuarioSBLocal | Interfaz para abstraer la implementación de la persistencia y recuperación de usuarios |
| IngredienteSBNegocio | Interfaz para abstraer la implementación de la lógica de ingredientes |
| UsuarioSBNegocio | Interfaz para abstraer la implementación de la lógica de usuarios |
| BusquedaSBNegocio | Interfaz para abstraer la implementación de la lógica de las búsquedas |
| RecetaSBNegocio | Interfaz para abstraer la implementación de la lógica de las recetas |
| ProcesarMensajeSBLocal | Interfaz para abstraer la implementación del procesamiento de mensajes |

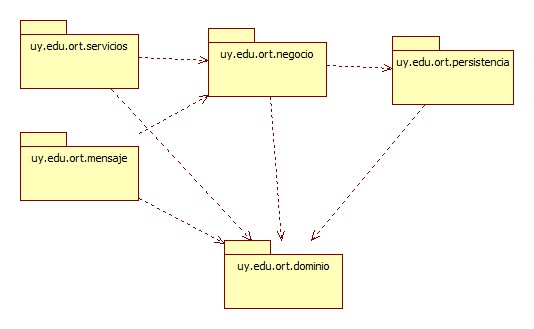
#### Decisiones de Diseño

La separación en módulos se realiza utilizando la táctica Increase Semantic Coherence, también utilizamos las tácticas Use an Intermediary agregando un paquete de negocio, intermediario entre los servicios y la persistencia, y Restrict Dependencies evitando las dependencias cíclicas para favorecer los requerimientos de Modificabilidad.

Para atacar el requerimiento de independencia del motor de base de datos, se abstrae la implementacion exponiendo interfaces. De esta manera, para cambiar el mecanismo de persistencia se puede realizar implementando las interfaces sin necesidad de cambiar otro modulo.

### Vista de Layers

#### Representación primaria

**

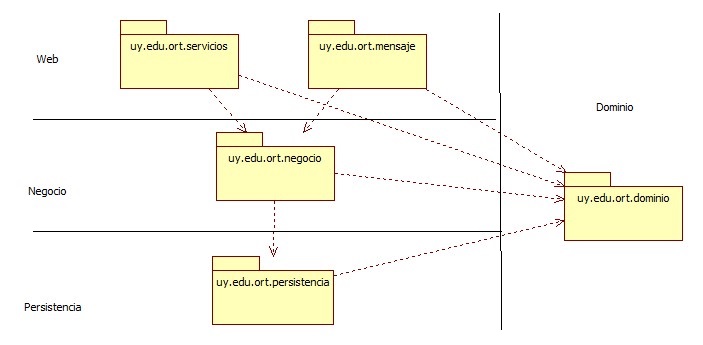
#### Decisiones de diseño

*Se realiza la distribución de las clases de acuerdo a las distintas responsabilidades, buscando alta cohesión y bajo acoplamiento de las mismas.*

### Vista de Layers

Podemos observar en la siguiente vista la disposición de los paquetes en las 4 capas lógicas definidas para esta solución.

#### Representación primaria



#### Decisiones de diseño

*Utilizamos el patrón de capas (Layers) para beneficiar la eficiencia y la disponibilidad, soportando el reuso.*

### Catálogo de elementos

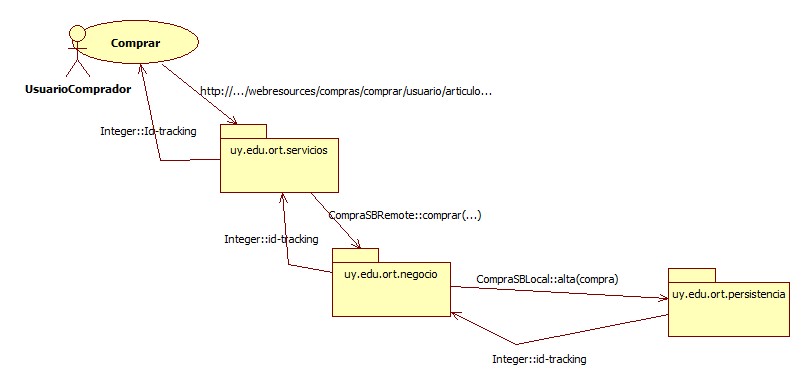
|  |  |
| --- | --- |
| **Elemento** | **Responsabilidades** |
| ***Uy.edu.ort.persistencia*** | ***Reponsable del acceso al servidor de base de datos y persistencia de las entidades del sistema*** |
| ***Uy.edu.ort.negocio*** | ***Responsable de realizar todas las operaciones y validaciones correspondientes a la lógica del negocio*** |
| ***Uy.edu.ort.dominio*** | ***Definición de los DTO del sistema*** |
| ***Uy.edu.ort.servicios*** | ***Responsable de exponer las funcionalidades del sistema mediante Rest*** |
| ***Uy.edu.ort.mensaje*** | ***Responsable de recibir y procesar los Java Messages depositados en la Queue definida para la actualización de estados asincronica*** |

### Interfaces

|  |  |
| --- | --- |
| **Interfaz:** | **Interfaces del negocio** |
| **Paquete que la implementa:** | **Uy.edu.ort.negocio** |
| ***Servicio*** | ***Descripción*** |
| *Expone las operaciones del negocio* | *Se generan interfaces remotas para abstraer la implementación de la lógica de negocio y separar el acceso a datos del negocio de las capas expuestas a la red pública.* |

|  |  |
| --- | --- |
| **Interfaz:** | **Interfaces de Persistencia** |
| **Paquete que la implementa:** | **Uy.edu.ort.persistencia** |
| ***Servicio*** | ***Descripción*** |
| *Expone las operaciones de acceso a datos* | *Se generan interfaces locales para abstraer la implementación de acceso a datos de la lógica de negocio,* |

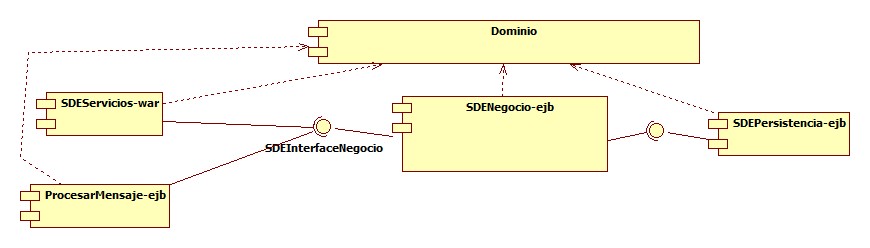
### Comportamiento

**

Vistas de Componentes y conectores

### Diagrama de componentes y conectores

#### Representación primaria



#### Catálogo de elementos

|  |  |
| --- | --- |
| **Componente/conector** | **Descripción** |
| ***SDEServicios-war*** | ***War que agrupa los servicios Rest*** |
| ***ProcesarMensaje-ejb*** | ***EJB que contiene el procesamiento de Messages*** |
| ***SDENegocio-ejb*** | ***EJB de lógica del negocio*** |
| ***SDEPersistencia-ejb*** | ***EJB de acceso a datos*** |
| ***Dominio*** | ***Jar con DTOs del problema*** |

#### Interfaces

|  |  |
| --- | --- |
| **Interfaz:** | **Interfaces del negocio** |
| **Paquete que la implementa:** | **Uy.edu.ort.negocio** |
| ***Servicio*** | ***Descripción*** |
| *Expone las operaciones del negocio* | *Se generan interfaces remotas para abstraer la implementación de la lógica de negocio y separar el acceso a datos del negocio de las capas expuestas a la red pública.* |

|  |  |
| --- | --- |
| **Interfaz:** | **Interfaces de Persistencia** |
| **Paquete que la implementa:** | **Uy.edu.ort.persistencia** |
| ***Servicio*** | ***Descripción*** |
| *Expone las operaciones de acceso a datos* | *Se generan interfaces locales para abstraer la implementación de acceso a datos de la lógica de negocio,* |

#### Comportamiento

#### Relación con elementos lógicos

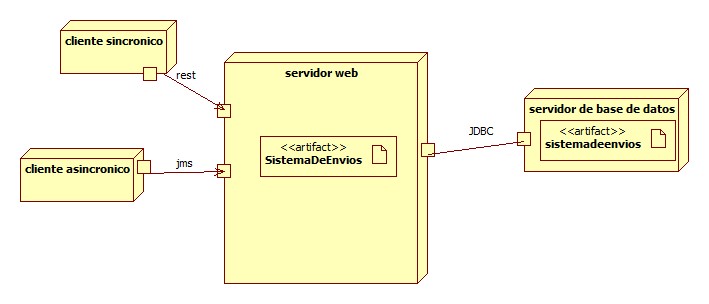
|  |  |
| --- | --- |
| **Componente** | **Paquetes** |
| *SDEServicios-war* | *Uy.edu.ort.servicios.\** |
| *SDEPersistencia-ejb* | *Uy.edu.ort.persistencia.\** |
| *SDENegocio-ejb* | *Uy.edu.ort.negocio.\** |
| *ProcesarMensaje* | *Uy.edu.ort.mensaje.\** |
| *Dominio* | *Uy.edu.ort.dominio* |

#### Decisiones de diseño

*Para responder a los RNF se implementan los componentes SDEServicios y ProcesarMensaje, donde SDEServicios-war contiene la implementación de los REST services y ProcesarMensaje el procesamiento de los mensajes para los clientes asincrónicos.*

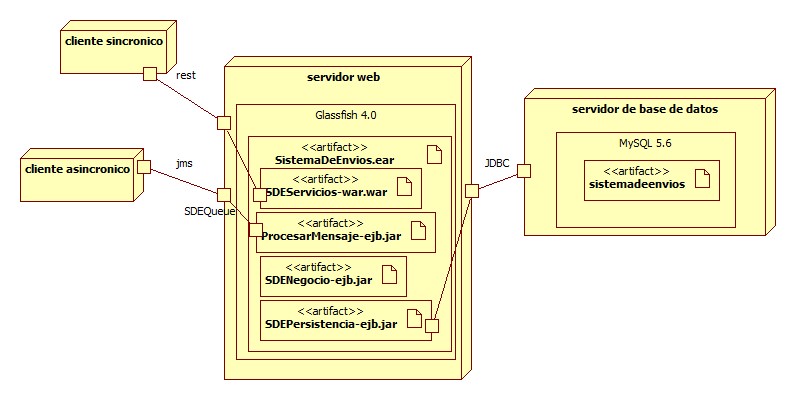
Vistas de Asignación

*Se decide implementar esta solución con 3 capas físicas, cliente, servidor y base de datos. Donde lo que refiere a la implantación se trata de un ear conteniendo cada componente necesario para ejecutar la aplicación.*

**

### Vista de Despliegue

#### Representación primaria

**

# Anexos

Condiciones de Entregable

*En esta entrega lamentamos no tener la funcionalidad requerida. La falta de conocimiento del framework y de la tecnología junto con el comportamiento del ide y el servidor web nos jugaron una muy mala pasada. No fue posible hacer funcionar los servicios consumiendo DTOs por lo que se debieron cambiar para consumir tipos básicos y crear los DTO en la comunicación con el negocio. Esto no es todo, ya que el retorno de objetos complejos tampoco fue posible, asi el único listado que probamos funciona es el de Articulos y confirmado que el de Usuarios falla y no encontramos la solución al momento*