

Documentación Arquitectura Software

Descripción

Documento que tiene como motivación plasmar el diseño arquitectónico del sistema de farmacias que vamos a implementar para la práctica 2 de Desarrollo de Software.

Fecha	Revision	Autores
18/04/2016	v1	Román Arranz Guerrero Hugo Maldonado Cózar

Tabla de Contenidos

1. Introducción
 - 1.1. Proposito del documento
 - 1.2. Contexto
 - 1.3. Definiciones, Acrónimos y Abreviaciones
 - 1.4. Referencias
2. Representación Arquitectónica
3. Objetivos y restricciones arquitectónicas
4. Casos de Uso
 - 4.1. Casos de uso importantes para la arquitectura
 - 4.2. Breve descripción de los casos de uso
5. Lógica del sistema
 - 5.1. Capa de aplicación
 - 5.2. Capa de Dominio
 - 5.3. Capa de fuente de datos
6. Procesos
7. Despliegue
 - 7.1. Cliente de la aplicación web
 - 7.2. Servidor web
 - 7.3. Base de datos
8. Implementación
 - 8.1. Estructura organizativa de los ficheros
 - 8.2. Estructura de directorios del código
 - 8.3. Organización de los archivos de prueba
 - 8.4. Archivos SQL
 - 8.5. Resumen detallado
9. Incidencias de concurrencia
 - 9.1. Gestión optimista de la concurrencia
10. Patrones
 - 10.1. Factoría
 - 10.2. Visitante
 - 10.3. Bridge
 - 10.4. Delegación
 - 10.5. Inmutable
 - 10.6. Fachada
 - 10.7. Singleton

1. Introducción

1.1. Propósito del documento

Este documento señala el esquema de la arquitectura del sistema de la aplicación web PharmacyS.

Se han utilizado diferentes vistas de arquitectura para ilustrar los distintos aspectos del sistema.

Este documento también presenta las decisiones arquitectónicas importantes que se realizan en el sistema.

1.2. Contexto

El alcance del documento es describir los objetivos y limitaciones arquitectónicas, la vista de casos de uso, la vista lógica, la vista de proceso, la vista de implementación y la vista de implantación. Estas son las vistas basadas en el modelo 4+1 de vistas de Krutchen.

En cuanto al contexto de la aplicación el sistema que se ha solicitado es una aplicación para reservar medicamentos que figuren en el inventario de una farmacia. Se pide incluir geolocalización de las farmacias para seleccionarlas y poder visualizar sus medicamentos, además se pide realizar operaciones de gestión a través del navegador.

1.3. Definiciones, Acrónimos y Abreviaciones

- UML - Unified Modeling Language
- WEAA - Web Enterprise Software Architecture Patterns
- AJAX - Asynchronous Javascript and XML
- JSON - Javascript Object Notation
- REST - Representational State Transfer
- GUI - Graphical User Interface
- UX - User Experience
- JS - Javascript
- HTML - HyperText Markup Language
- Visa Arquitectónica - "Vista de la Arquitectura Software de un Sistema dada una perspectiva. Se centra principalmente en la estructura, modularidad, componentes esenciales, y el principal control de los flujos" (Larman).

1.4. Referencias

- Larman, Craig, Applying UML and Patterns: an Introduction to Object-Oriented Analysis and Design and Iterative Development. 3rd ed, Upper Saddle River: Prentice Hall PTR, 2005.
- Eden, A. and Kazman, R: Architecture Design Implementation: On the Distinction Between Architectural Design and Detailed Design.

2. Representación Arquitectónica

La arquitectura para PharmacyS es representada usando las vistas definidas en el modelo “4+1” de Kazman, pero usando la convención de nombres RUP. Las vistas usadas en el documento de la aplicación PharmacyS son:

Vista de Casos de Uso	
Audiencia	Todos los stakeholders
Area	Describe el conjunto de escenarios y/o casos de uso que son críticos para la arquitectura
Artefactos Relacionados	Documento de casos de uso

Vista Lógica	
Audiencia	Los diseñadores de PharmacyS
Area	Requisitos funcionales que proporcionan la operatividad del sistema.
Artefactos Relacionados	Diagrama de Paquetes

Vista de Procesos	
Audiencia	Integradores del Sistema
Area	Requisitos no funcionales que describen el diseño concurrente y aspectos de sincronización. Como se asocian los componentes de la vista lógica a los procesos.
Artefactos Relacionados	Diagrama de Procesos

Vista de Implementación	
Audiencia	Programadores
Area	Componentes software. Describen la estructura organizativa de los archivos, librerías, test unitarios etc...
Artefactos Relacionados	Diseño de la jerarquía de directorios.

Vista de Implantación	
Audiencia	Ingenieros de sistemas

Area	Topología. Describe el mapeo del software en el hardware y muestra aspectos de sistemas distribuidos.
Artefactos Relacionados	Diagrama de Despliegue UML

3. Objetivos y restricciones arquitectónicas

Las elecciones arquitectónicas se han hecho considerando los siguientes objetivos y restricciones:

- El objetivo de SOEN 343 y SOEN 344 de la WEAA (por ejemplo, Controlador Frontal, Mapeo de Datos, TDG, Mapa de Identidades, Comandos Frontales) y las vistas arquitectónicas (por ejemplo el modelo 4+1). El sistema PharmacyS se ha basado en el diseño propuesto en las transparencias de la asignatura.
- La Autenticación de Usuarios tomará siempre lugar antes que cualquier otro acceso a los datos en PharmacyS. Completando los aspectos de seguridad de los datos a los accesos no autorizados.
- Los requisitos funcionales y no funcionales se mantendrán desde el inicio del proceso de desarrollo con los Casos de Uso especificados.

4. Casos de Uso

“La vista de casos de uso describe una vista de la arquitectura del sistema que cubre el comportamiento del sistema en general desde el punto de vista de los actores involucrados”.

4.1. Casos de uso importantes para la arquitectura

El siguiente conjunto de casos de uso tienen una especial importancia en cuanto a la arquitectura del sistema PharmacyS:

- Creación de Usuarios
- Editar Contenido
- Vista de Inventario
- Vista de Compra (Pasarela de pago)
- Geolocalización

DIAGRAMAS DE CASOS DE USO

4.2. Breve descripción de los casos de uso

- **Creación de Usuarios: UC 4.1**

Un usuario podrá registrarse en el sistema en el caso de que no disponga de una cuenta asociada, para ello tendrá que hacer clic en el botón de **Sign Up** y tendrá que completar un formulario de inscripción. Una vez haya hecho esto recibirá un email en el correo electrónico indicado en el formulario, una vez confirme el enlace que se le ha enviado en ese email podrá tener acceso a la aplicación identificándose con el email y contraseña especificada en el formulario.

- **Editar Contenido**

- **Vista de Inventario**

Un usuario identificado podrá visualizar el inventario de medicamentos de la farmacia que haya seleccionado a través del geolocalizador. Por cada medicamento que ofrece en esta vista el usuario podrá ver su contenido y añadirlo a su carrito de compra.

- **Vista de Compra (Pasarela de pago)**

Un usuario identificado podrá realizar la compra de los productos que tenga en su carrito de la compra accediendo a una pasarela de pago en la que completará un formulario para validar el coste total de su compra.

- **Geolocalización**

Un usuario identificado cuando accede a la aplicación se le mostrará un mapa en el que se vendrán marcadas todas las farmacias cercanas a ese usuario en un radio de n metros, el usuario podrá modificar dicho radio hasta un total de 20km.

5. Vista Lógica

“La organización conceptual del sistema se presenta en la vista lógica en términos de capas, paquetes, subsistemas y clases”.

La vista lógica del sistema PharmacyS se separa en 3 capas principales: la capa de aplicación, la capa de dominio y la capa de fuente de datos.

Estas 3 capas principales se estructuran siguiendo la convención arquitectónica de 3 capas:

- La capa de presentación
- La capa de dominio
- La capa de origen de datos.

DIAGRAMA DE PAQUETES

5.1. Capa de Aplicación

La capa de aplicación en esta aplicación modifica el controlador frontal tradicional con la plantilla de la vista de la organización con componentes AJAX. El controlador frontal ahora se encarga de las solicitudes haciendo referencia a los objetos de dominio desde el cliente hacia el servidor y viceversa. Los beneficios de este enfoque permiten un más robusto y más rico GUI para la aplicación, podrá comportarse con más características que una aplicación nativa de escritorio, además, como la mayoría de los datos se almacenan en caché de forma local, el posterior resultado se renderiza con más agilidad y le libera de carga al servidor de forma que hacemos proporcionamos más UX.

DIAGRAMA DE SECUENCIA DEL CLIENTE

5.2. Capa de Dominio

El paquete del dominio encapsula sub-paquetes utilizados para describir las reglas de la lógica del negocio del sistema PharmacyS que controlan y regulan la información del modo de procesamiento. Los sub-paquetes son el modelo del dominio, el asignados de datos, la unidad de trabajo, el mapa de identidad, la capa de proxy virtual.

El modelo de dominio sub-paquete contiene una clase para cada una de las principales entidades: usuarios, funciones, eventos, medicamentos, farmacias, etc... También contiene una capa para hacer un uso de la unidad genérica de trabajo y un mapa de identidad.

Estos objetos de clase son utilizados por las transacciones de dirigentes y objetos asignador de datos como contenedores de datos. Información sobre una entidad se almacena como atributos de la clase correspondiente. La información acerca de una entidad se hace pasar a través de métodos por los que pasa el objeto de dominio correspondiente como un parámetro. El sub-paquete asignador de datos contiene una clase asignador de datos para cada clase de objeto de dominio se encuentra en el paquete de objetos de dominio. Esta capa también contiene una capa que es Supertype AbstractMapper para hacer uso de la ventaja de patrón de reflexión utilizado por la unidad de trabajo. Estos creadores de mapas de datos crean una capa de direccionamiento indirecto entre la lógica de dominio y la fuente de datos. Las transacciones no saben acerca de la estructura de la fuente de datos, así como las diferentes consultas para acceder a la información.

La unidad de trabajo se utiliza en la aplicación de la unidad de ritmos de trabajo y como se describe en la sección 10.1. The aplicación identidad se utiliza en la aplicación del modelo de aplicación identidad tal como se describe en la sección 10.2. The capa de proxy virtual contiene un conjunto de clases de proxy y las clases de cargador que se utilizan en la aplicación del modelo de carga perezosa como se describe en la sección 10.3.

5.3. Capa de Fuente de Datos

El paquete fuente de datos contiene las clases que definen la infraestructura de servicios técnicos utilizados para almacenar datos persistentes. Este paquete se utiliza por la aplicación I3EM para interactuar con la base de datos ubicada en el servidor Stu03. El paquete incluye un archivo de origen por cada clase que presente en el asignador de datos sub-paquete. Estos objetos de origen de datos (tabla de datos Gateways) son utilizados por los objetos del asignador de datos para acceder a la base de datos directamente para almacenar o recuperar información.

6. Vista de Proceso

"El punto de vista del proceso muestra las responsabilidades y colaboraciones entre los procesos y los hilos del sistema", así como la asignación de elementos lógicos a ellos. En esta sección, se describen las tareas, que incluyen los procesos y subprocesos que están involucrados en la ejecución y las interacciones del sistema. El siguiente modelo describe el proceso de la solicitud expresa **Ver Usuario** como procesos ejecutables junto a sus negociaciones.

DIAGRAMA DE PROCESO

El diagrama ilustra una petición que he hecho por el usuario identificado para obtener detalles de cuenta de usuario específica en el sistema PharmacyS.

El usuario lanza un navegador web se encuentra en un equipo remoto que crea el proceso de cliente web. Una vez que el usuario se ha autenticado su o ella misma, aparece la página principal del usuario.

Al seleccionar la opción de usuario de vista, el proceso de cliente web crea y envía una solicitud al proceso del servidor Tomcat que ha sido y sigue siendo en línea y disponible.

Este hilo petición permanece en la memoria hasta que un resultado se recibe desde el servidor Tomcat. El hilo de solicitud se pasa luego al módulo de servicio adecuado para procesamiento adicional. El método de usuario en Administrador de usuarios recuperar el ID de usuario toma como entrada e interactúa con el UserMapper que a su vez interactúa con la capa de origen de datos. Posteriormente, devuelve los objetos de usuario en el sistema cliente en turnos produce la vista actualizada.

7. Vista de Implantación

“La vista de implantación muestra la distribución de los elementos de vista lógica para los nodos de procesamiento físico, y la configuración de red física entre nodos.” El sistema PharmacyS está diseñado como una aplicación web y está destinado a ser utilizado como un cliente / servidor. La aplicación PharmacyS se va a desplegar en el servidor web de una RaspberryPi alojada en la casa de uno de los integrantes.

En el siguiente diagrama de despliegue de UML, se presentan los elementos de red físicos involucrados en la implementación del sistema. En la siguiente sección describiremos una descripción de cada elemento de red.

7.1. Cliente de la aplicación Web

Los usuarios típicos utilizan un navegador web del cliente, como Internet Explorer se ejecuta en para acceder al Portal Web IEEE Montreal. La comunicación entre el cliente y el servidor está garantizada por el protocolo de red HTTP. El usuario utilizará una interfaz gráfica presentada como una página web a través del cual él / ella puede seleccionar los comandos deseados.

7.2. Servidor Web

La aplicación web I3EM está actualmente instalado y opera desde el servidor Stu03. Este servidor permanece bajo el control del departamento de Ingeniería de Software de la Universidad de Concordia y se integra a la red de laboratorio de computación MITI. El servidor web es mantenido por el departamento de Ingeniería de Software y su disponibilidad puede estar sujeto a la red ENCS trabajos de mantenimiento programado (o no programado).

El despliegue de la aplicación Web debe ser realizado por un túnel SSH con el fin de respetar las restricciones de seguridad de la Universidad de Concordia. Una aplicación cliente, tales como la arcilla, se puede utilizar para conectarse al servidor de Stu03 y realizar las acciones necesarias despliegue de documentación (I & C).

Otras restricciones de seguridad, tales como los siguientes, pueden ocurrir durante la fase de despliegue:

- Conexiones de base de datos se restringen a nivel local (es decir, debe conectarse al servidor local)
- Archivos Servlets lecturas / escrituras se limitan a la creación de políticas de permisos de archivo Stu03
- Tablas de base de datos se limitan a los archivos de comandos SQL Déployé

7.3. Base de datos

Base de datos La base de datos utilizada por la aplicación web I3EM es MySQL 5. El servidor de base de datos se encuentra en el servidor Stu03 sobre un sistema operativo basado en Linux. la creación de cualquier base de datos está restringido por los archivos de comandos SQL disponibles en el SQL.

Ningún usuario puede crear nuevas tablas a través del sistema I3EM. Sin embargo, cualquier persona que tenga derechos de acceso al servidor de Stu03 puede editar los archivos de comandos SQL actuales para agregar nuevas tablas, crear nuevos archivos de comandos SQL o simplemente acceder a la base de datos MySQL a través de una línea de comandos.

DIAGRAMAS DE DESPLIEGUE

8. Vista de Implementación

8.1. Estructura organizativa de los ficheros

La organización estructural de los archivos del sistema PharmacyS cuelga de una carpeta raíz llamada Pharmacy. Todos los archivos fuente de Java se encuentran dentro de la carpeta src. Todos los archivos SQL se encuentran dentro de la carpeta SQL.

8.2. Estructura organizativa de los directorios del código

Los archivos de origen de la aplicación PharmacyS se dividen en un esquema de carpeta que en la raíz separa los archivos de servidor y cliente. Después de esto, los componentes están separados adicionalmente en carpetas correspondientes. Un listado detallado está disponible en 8.5

8.3. Organización de los archivos de prueba

Se han programado tres tipos principales de las pruebas sean realizadas: prueba de nivel de sistema, pruebas funcionales y pruebas unitarias. Los archivos de pruebas se agrupan por componentes independientes entre sí. Las pruebas se pueden ejecutar directamente desde un archivo de prueba correspondiente, por ejemplo EventsManagerJUNIT.java para todas las pruebas relacionadas con la gestión de eventos.

8.4. Archivos SQL

Las carpetas contienen cada uno de los archivos SQL con los correspondientes comandos SQL. "init.sql" se utiliza para crear el conjunto de tablas de base de datos utilizados por el sistema PharmacyS. Durante la creación de las tablas estas quedan rellenas con una serie de datos de prueba.

8.5. Resumen detallado

Recurso	Comentarios
PharmacyS	Raíz
PharmacyS/SQL	Ficheros SQL para la BD
PharmacyS/src	Archivos fuente
PharmacyS/src/model	El dominio de la logica, las clases
PharmacyS/src/dao	Capa de abstracción de interfaz entre los objetos y los datos
PharmacyS/src/service	Hooks de la API REST
PharmacyS/src/servlets	Servlets que procesan los formularios
PharmacyS/src/util	Utilidades usadas en el sistema
PharmacyS/target	Classes ejecutables de java y dependencias resueltas por Maven
PharmacyS/WebContent	Contenido del cliente web
PharmacyS/WebContent/META-INF	Librerías asociadas
PharmacyS/WebContent/WEB-INF	Configuración de los Servlets
PharmacyS/WebContent/css	Hojas de estilo
PharmacyS/WebContent/img	Imagenes
PharmacyS/WebContent/js	Eventos Javascript

9. Incidencias de Concurrency

En esta sección se describe cómo se gestiona el uso simultáneo en el sistema portal web PharmacyS.

9.1. Gestión Optimista

Vamos en primer lugar a ilustrar con un ejemplo como se podría gestionar un conflicto:

Se tienen dos usuarios conectados con privilegios de Administrador “Admin_A” y “Admin_B”, dos administradores tratan de actualizar un dato relacionado con la misma cuenta de usuario. En primer lugar los Administradores se tuvieron que identificar para tener acceso al sistema.

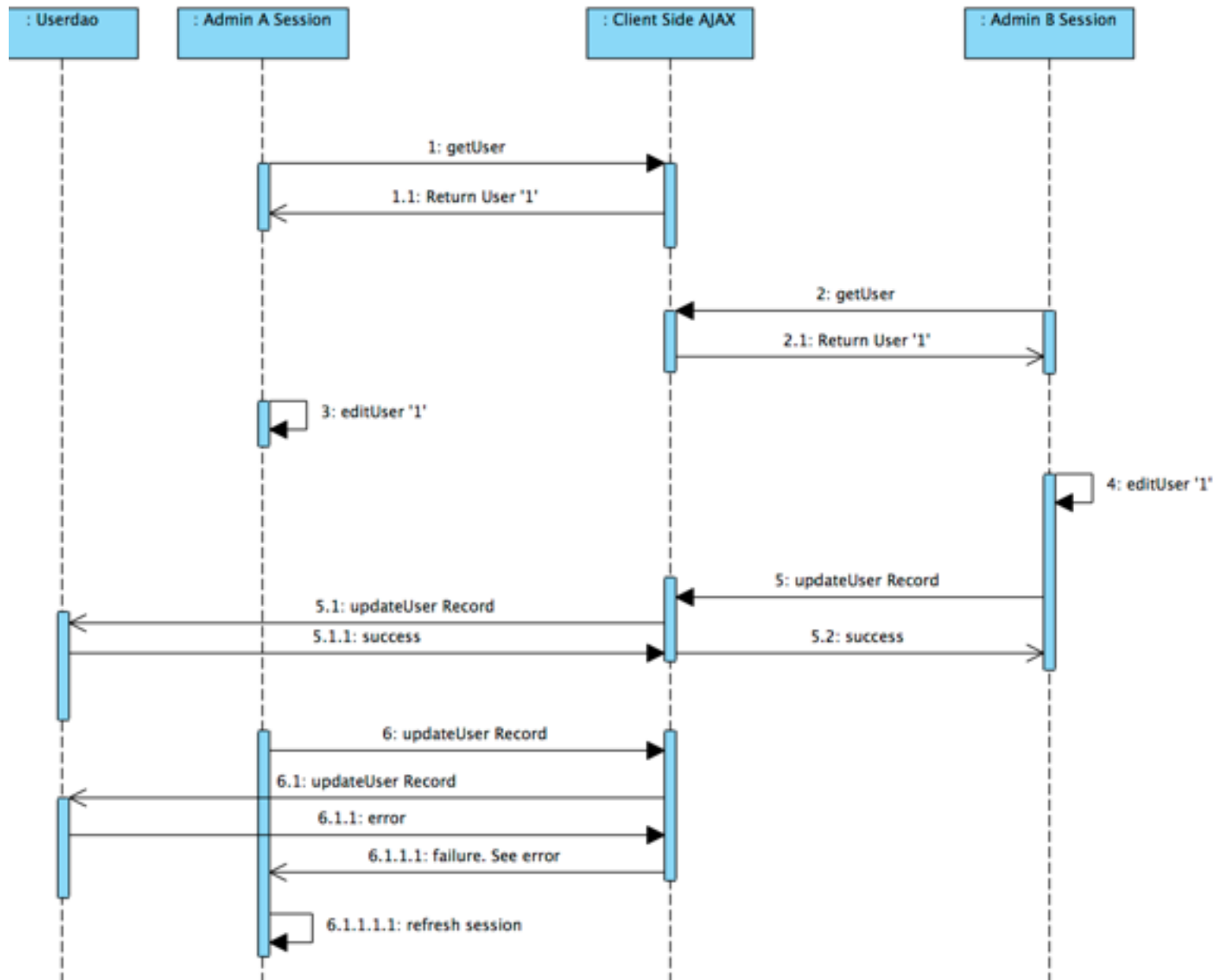
Admin_A se encuentra administrando las cuentas de usuario y selecciona una de ellas para actualizarla. Por lo tanto Admin_A tiene una sesión abierta e hizo un cambio sobre un usuario, esto provoca una actualización del contador de versiones en el registro del usuario.

Otro administrador Admin_B procede a seleccionar el mismo usuario para actualizar casualmente el mismo dato.

Como Admin_A completó la actualización, la aplicación web PharmacyS incrementa el número de versión y almacena el registro de la cuenta de usuario modificado de nuevo en la BD. Si Admin_B completa su actualización, el sistema detectará un fallo ya que no coincide el número de la versión de referencia del registro que quiso actualizar.

En consecuencia el sistema devolverá un mensaje de error que se indica un fallo en la versión del registro. Después de esto se procederá a realizar un reajuste en la sesión del administrador.

Gestión de la Concurrency



10. Patrones