



Sistema Integrado de Apoyo y Asistencia para Polimedicados

E5.1b – Diseño Arquitectónico

Estado: FINAL

CEIEC-UFV

Este documento contiene el diseño arquitectónico de la solución propuesta para el Sistema PAUTA

Referencia: PAUTA_DOC_E5.1b DiseñoArquitectonico v1.4



plan
avanza2»»



<http://pauta.ceiec.es>

Control Documental

Control documental

Proyecto:	PAUTA
Tarea:	E5.1
Destinatario:	Consortio

Título:	Diseño Arquitectónico
Referencia:	PAUTA_DOC_E5.1b DiseñoArquitectonico
Versión:	1.4
Fecha de creación:	09-10-2011
Distribución:	Restringida
Autor/es:	Álvaro José García Tejedor (CEIEC-UFV)

Fichero:	PAUTA_DOC_E5.1 DiseñoArquitectonico v1.4.DOCX
Editado con:	WORD 2010

Control de versiones

Versión	Parte que cambia	Descripción del cambio	Fecha
1.0	Versión original	N/A	14-02-2011

Índice

1. Introducción	1
1.1. Propósito	1
1.2. Alcance	1
1.3. Definiciones, acrónimos y abreviaciones	1
1.3.1. Abreviaciones	1
1.3.2. Definiciones	2
1.4. Referencias	2
1.5. Vista general	3
2. Descripción del Sistema	4
3. Contexto del Sistema	5
4. Diseño del Sistema	6
4.1. Método de diseño	6
4.2. Descripción de la descomposición	7
5. Descripción de los componentes	9
5.1. C1_Interfaz HTML/PHP	9
5.1.1. Tipo	9
5.1.2. Propósito	9
5.1.3. Función	9
5.1.4. Subordinados	10
5.1.5. Dependencias	10
5.1.6. Interfaces	10
5.1.7. Recursos	10
5.1.8. Referencias	10
5.1.9. Procesamiento	10
5.1.10. Datos	11
5.2. C2_Javascript	11
5.2.1. Tipo	11
5.2.2. Propósito	11
5.2.3. Función	11
5.2.4. Subordinados	12
5.2.5. Dependencias	12
5.2.6. Interfaces	12
5.2.7. Recursos	12
5.2.8. Referencias	12
5.2.9. Procesamiento	12

5.2.10. Datos	13
5.3. C21_Funciones	13
5.3.1. Tipo	13
5.3.2. Propósito	13
5.3.3. Función	14
5.3.4. Subordinados	14
5.3.5. Dependencias	14
5.3.6. Interfaces	14
5.3.7. Recursos	14
5.3.8. Referencias	14
5.3.9. Procesamiento	14
5.3.10. Datos	15
5.4. C22_Librerías	15
5.4.1. Tipo	15
5.4.2. Propósito	15
5.4.3. Función	15
5.4.4. Subordinados	16
5.4.5. Dependencias	16
5.4.6. Interfaces	16
5.4.7. Recursos	16
5.4.8. Referencias	16
5.4.9. Procesamiento	16
5.4.10. Datos	16
5.5. C3_Controlador	17
5.5.1. Tipo	17
5.5.2. Propósito	17
5.5.3. Función	17
5.5.4. Subordinados	17
5.5.5. Dependencias	17
5.5.6. Interfaces	18
5.5.7. Recursos	18
5.5.8. Referencias	18
5.5.9. Procesamiento	18
5.5.10. Datos	18
5.6. C31_Clases	18
5.6.1. Tipo	18
5.6.2. Propósito	18
5.6.3. Función	19
5.6.4. Subordinados	19
5.6.5. Dependencias	19
5.6.6. Interfaces	19
5.6.7. Recursos	19
5.6.8. Referencias	20

5.6.9. Procesamiento	20
5.6.10. Datos	20
5.7. C4_Acceso a Datos	20
5.7.1. Tipo	20
5.7.2. Propósito	20
5.7.3. Función	21
5.7.4. Subordinados	21
5.7.5. Dependencias	21
5.7.6. Interfaces	21
5.7.7. Recursos	21
5.7.8. Referencias	21
5.7.9. Procesamiento	21
5.7.10. Datos	22
5.8. C41_Conexiones	22
5.8.1. Tipo	22
5.8.2. Propósito	22
5.8.3. Función	22
5.8.4. Subordinados	22
5.8.5. Dependencias	23
5.8.6. Interfaces	23
5.8.7. Recursos	23
5.8.8. Referencias	23
5.8.9. Procesamiento	23
5.8.10. Datos	23
6. Viabilidad y estimación de recursos	24
6.1. Desarrollo	24

1. Introducción

En este documento se reflejan los distintos módulos que compondrán el sistema fijo de pautado para los médicos y cuáles son sus conexiones e interfaces con los demás módulos que lo componen.

1.1. Propósito

Este documento describe el diseño arquitectónico de la aplicación médica del sistema Pauta. Se describe cada uno de los componentes que forman el software y sus contenidos.

Este documento está destinado a los desarrolladores y analistas del sistema, de manera que puedan realizar la correcta implementación del software.

1.2. Alcance

Los diferentes productos software que se desarrollarán en este proyecto, en el área de la plataforma fija o aplicación de pautado.

Estos productos software son:

- Módulo de acceso a la aplicación
- Módulo de pautado
- Módulo de modificación
- Módulo de consultas
 - Pautas
 - Buscador de pautas
- Módulo de administración
- Módulo de impresión

1.3. Definiciones, acrónimos y abreviaciones

A continuación se definen las definiciones, acrónimos y abreviaciones presentes en este documento.

1.3.1. Abreviaciones

ABREVIACIÓN	DESCRIPCIÓN
-------------	-------------

DEM	Dispensador electrónico de medicamentos
PDA	Personal Digital Assistant
LOPD	Ley Orgánica de Protección de Datos
PC	Requisito Capacidad del sistema PAUTA
PR	Requisito Restricción del sistema PAUTA
DC	Requisito Capacidad del dispositivo DEM
DR	Requisito Restricción del dispositivo DEM
TC	Requisito Capacidad del sistema de Tele-asistencia
TR	Requisito Restricción del sistema de Tele-asistencia
MC	Requisito Capacidad del dispositivo móvil
MR	Requisito Restricción del dispositivo móvil
ESA	European Spacial Agency (Agencia Espacial Europea)

1.3.2. Definiciones

TÉRMINO	DEFINICIÓN
Smartphone	Teléfono móvil que ofrece más funciones que un teléfono móvil común, como posicionamiento GPS, acelerómetro o conexión a Internet.

1.4. Referencias

A continuación se muestran los documentos a los que hace referencia este documento:

DOCUMENTO	REFERENCIA/APLICABLE
Software Engineering Standards ESA PSS-05-02 Issue 1 Revision 1 (March 1995) GUIDE TO USER REQUIREMENTS DEFINITION PHASE	Documento Referencia/Aplicable
PAUTA_Memoria_AVANZA_2010_v7	Documento Referencia
PAUTA_INF_T2.2 ReuniónDoctores v1.2	Documento Referencia
PAUTA_INF_T2.2 ReuniónDoctores v1.4	Documento Referencia

1.5. Vista general

El documento contiene una amplia descripción de todos los factores que puedan afectar al proyecto PAUTA y sus requerimientos analizándolos de forma general, permitiendo al lector una aproximación a la especificación del sistema.

La segunda parte nos presenta un listado con todos los requerimientos funcionales y no funcionales de la aplicación PAUTA, es decir, todo lo que la aplicación debe hacer y sus restricciones.

2. Descripción del Sistema

El sistema de pautado es el encargado de realizar la recopilación de la información correspondiente a una pauta terapéutica compleja y almacenarla en el sistema, de manera que quede constancia de la misma y ésta pueda ser enviada al dispositivo móvil y al servicio de teleasistencia en formato XML.

De esta manera, el dispositivo móvil es capaz de realizar la gestión diaria de la pauta, emitiendo los avisos necesarios para que el paciente no pierda ninguna toma de su medicación y enviar información de control al servicio de teleasistencia, para dar un servicio completo al paciente.

Se trata de un sistema web, que alberga la información necesaria sobre los pacientes, doctores y medicamentos para poder generar y almacenar una pauta terapéutica compleja. El sistema está estructurado siguiendo el patrón de diseño MVC (Modelo-Vista-Controlador), arquitectura muy extendida en el diseño de aplicaciones web por su facilidad de implementación y su posibilidad de ampliación futura sin necesidad de hacer una reforma del sistema existente.

Una de las ventajas de usar este patrón, aparte de las ya mencionadas, es que permite el uso de objetos en PHP, lo que permite mejorar mucho el desarrollo de la aplicación, permitiendo al mismo tiempo diferenciar correctamente los diferentes componentes de la aplicación y encapsular las distintas funcionalidades de la misma.

Inicialmente, la aplicación se nutre de información precargada, aunque dispone de un administrador que permite gestionar toda la información almacenada y añadir nueva información a la base de datos o modificar la información ya almacenada. Toda esta información pre-cargada es la que posteriormente se mostrará en el gestor de pautas, permitiendo elegirla convenientemente para la creación de las pautas.

3. Contexto del Sistema

El contexto en el que se engloba la aplicación de pautado está representado en la Figura 1.

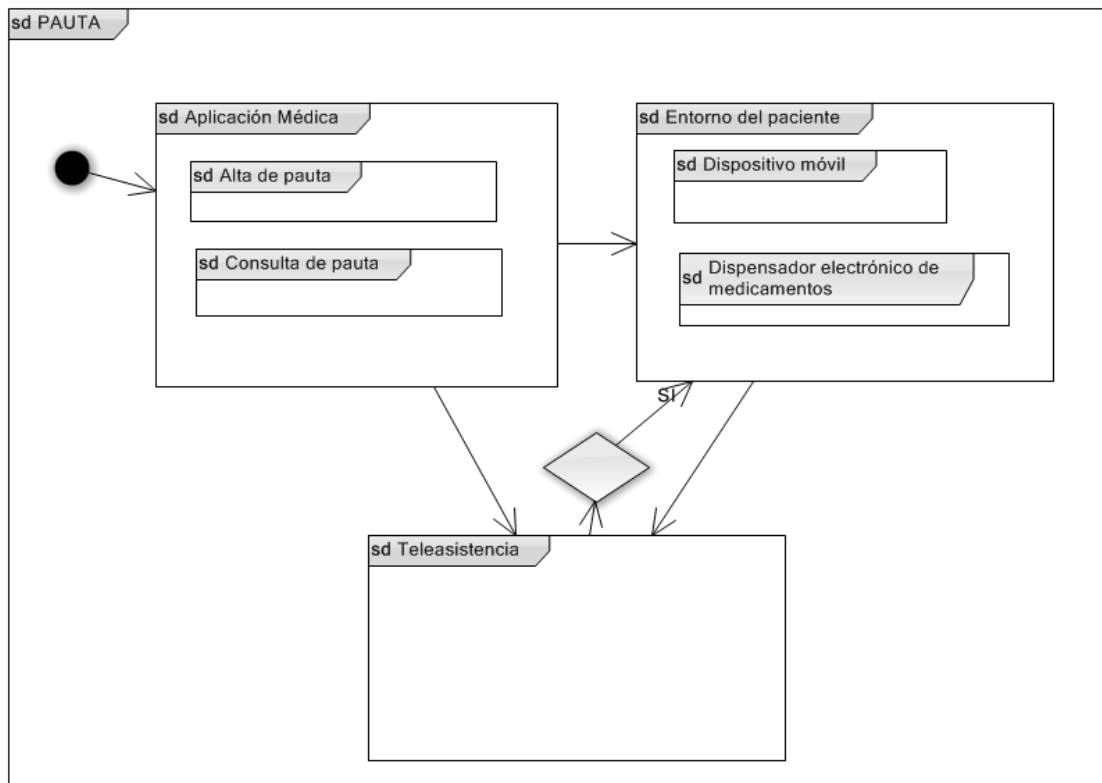


Figura 1 Diagrama de contexto de la aplicación de pautado

Como queda patente, la aplicación de pautado queda englobada dentro del sistema PAUTA, y es la responsable de la generación y la consulta de las pautas terapéuticas complejas prescritas al paciente y la comunicación con los otros sistemas implicados en PAUTA. La comunicación con estos sistemas se realiza mediante un fichero XML que contiene toda la información de la pauta prescrita.

La aplicación de pautado del sistema PAUTA dispone de dos interfaces directas:

1. Servicio de teleasistencia: Se remite al servicio de teleasistencia la pauta prescrita, de manera que desde el servicio puedan monitorizar el seguimiento, por parte del paciente, de la pauta.
2. Dispositivo móvil: Se remite al dispositivo móvil, mediante banda ancha móvil, el fichero XML con la pauta, de manera que éste la pueda interpretar y generar los avisos pertinentes a las horas marcadas para que el paciente se tome la medicación, indicándole además información adicional sobre el medicamento que debe tomar.

4. Diseño del Sistema

4.1. Método de diseño

El método de diseño elegido es el MVC (modelo-vista-controlador). MVC es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. El patrón de llamada y retorno MVC, se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista y remitirlos al modelo.

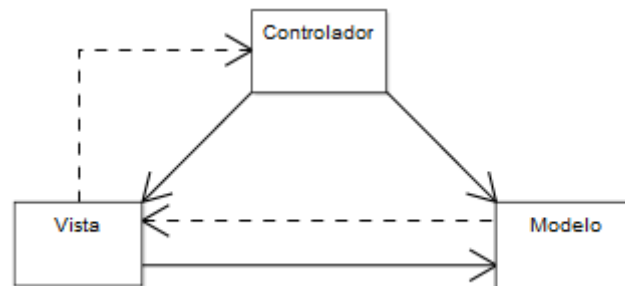


Figura 2 Diagrama MVC

- **Modelo:** Esta es la representación específica de la información con la cual el sistema opera. En resumen, el modelo se limita a lo relativo de la *vista* y su *controlador* facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado.
- **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- **Controlador:** Este responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista.

Muchos de los sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos: en líneas generales del **MVC** corresponde al modelo. La unión entre capa de presentación y capa de negocio conocido en el paradigma de la Programación por capas representaría la integración entre **Vista** y su correspondiente **Controlador** de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio y capa de presentación pero si pretende separar la capa visual gráfica de su correspondiente programación y acceso a datos, algo que mejora el desarrollo y mantenimiento de la Vista y el Controlador en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí.

Aunque se pueden encontrar diferentes implementaciones de **MVC**, el flujo que sigue el control generalmente es el siguiente:

- El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)
- El controlador recibe (por parte de los objetos de la interfaz) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler o manejador) o callback.
- El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
- El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, se podría utilizar el patrón Observador para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice. Nota: En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.
- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

4.2. Descripción de la descomposición

La arquitectura definida (MVC) se ha subdividido en diferentes componentes, aparte de los ya establecidos de manera que sea más sencillo desarrollar cada uno de ellos intentando minimizar el acoplamiento entre todos los componentes.

La descomposición de componentes propuesta es la siguiente:

- Vista/Presentación
 - C1_Interfaz
 - C2_Javascript
 - C21_Funciones
 - C22_Librerías-JS
- Controlador
 - C3_Controlador
 - C31_Clases
- Modelo
 - C4_Acceso-a-datos
 - C41_Gestión-de-conexiones

En la Figura 3 puede verse el diagrama de componentes del sistema:

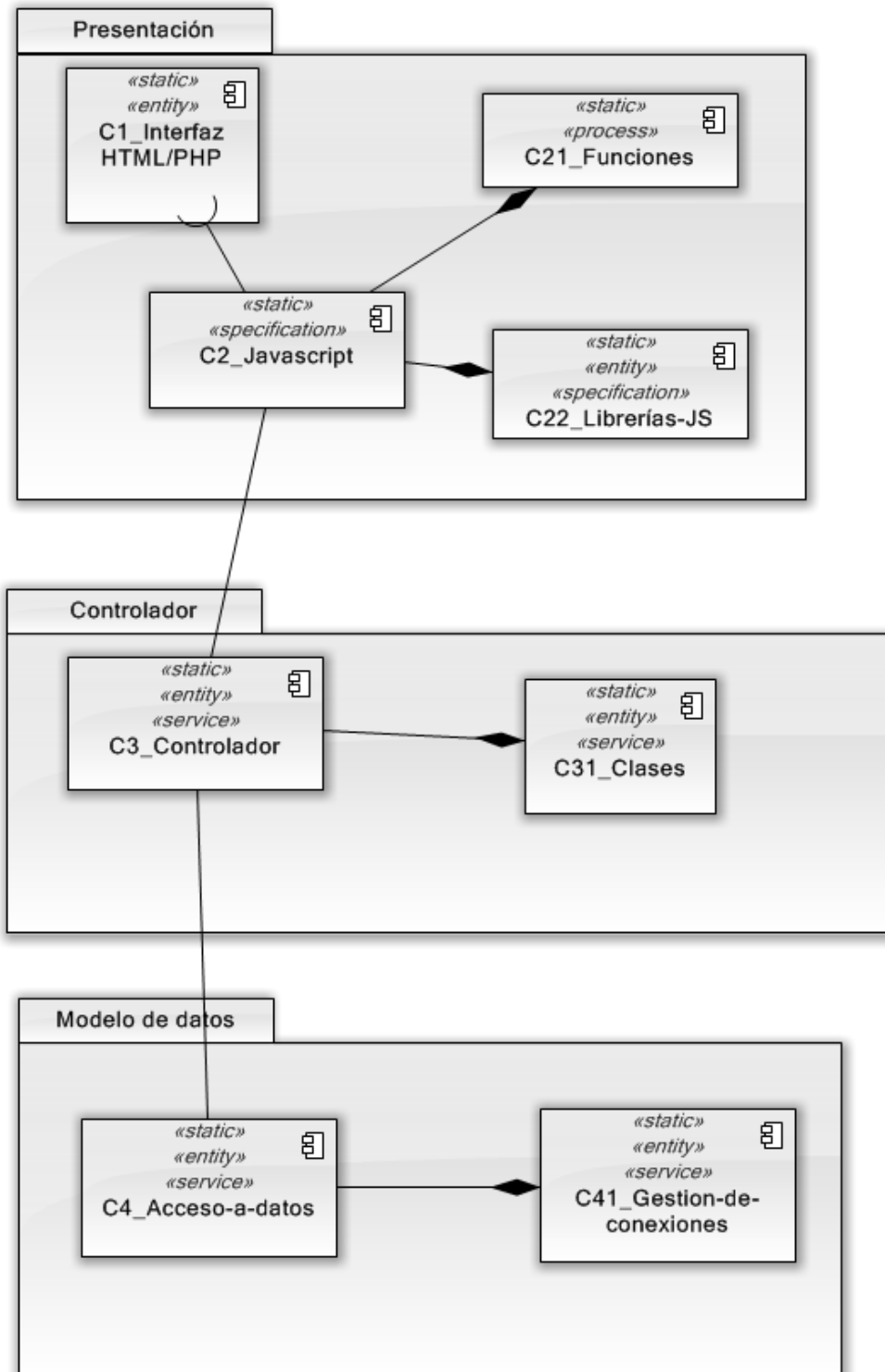


Figura 3 Diagrama de componentes de la aplicación

5. Descripción de los componentes

A continuación se describe cada uno de los diferentes componentes que forman el subsistema fijo del sistema Pauta.

5.1. C1_Interfaz HTML/PHP

5.1.1. Tipo

Este componente es una entidad, de tipo estático, que pertenece a la categoría Vista del patrón de diseño MVC.

5.1.2. Propósito

Este componente cumple los siguientes requisitos software, presentes en el documento de requerimientos software:

- SRF-007
- SRF-009
- SRF-010
- SRF-011
- SRF-012
- SRF-013
- SRF-014
- SRF-015
- SRF-016
- SRF-017
- SRF-018
- SRF-019
- SRF-020
- SRF-021
- SRF-022
- SRO-001

5.1.3. Función

Este componente es el encargado de generar la interfaz con el usuario. Genera y presenta los formularios necesarios para introducir la información relativa a la gestión de pautas terapéuticas, administración de la aplicación, administración de la información.

Se trata de código HTML estático, que podrá ser modificado mediante la interfaz presente con el componente C2_Javascript.

La información obtenida en el componente C2_Javascript será mostrada en este componente.

5.1.4. Subordinados

C2_Javascript

5.1.5. Dependencias

La ejecución de este componente depende principalmente del estado del entorno de producción, ya que deben estar funcionando y totalmente disponibles los servidores de páginas web y de base de datos elegidos.

5.1.6. Interfaces

La interfaz principal de este componente es con el componente C2_Javascript. El flujo de datos hacia el componente C2_Javascript consiste en los parámetros recopilados por este componente, relativos a:

- Gestión del sistema (
- Gestión de la información almacenada (pautas, usuarios)
- Parámetros de control

La información que recibe este componente corresponde a las respuestas que devuelve el componente C2_Javascript, con la información solicitada en tiempo de ejecución o un mensaje de error que permita identificar qué ha ocurrido durante la operación.

5.1.7. Recursos

Se requiere de un servidor web y un servidor de bases de datos para la correcta ejecución del sistema.

Para acceder al mismo es necesario un PC con sistema operativo que disponga de conexión a la red y un navegador web compatible.

5.1.8. Referencias

No aplica.

5.1.9. Procesamiento

Este componente está compuesto por ficheros, mayoritariamente, con lo que no se realiza procesamiento en el mismo. El procesamiento se realizará en el componente C2_Javascript.

5.1.10. Datos

Los formularios de recogida de datos se encontrarán inicialmente vacíos, es decir, no contendrán datos, lo que será representado mediante la cadena vacía "". Este componente no dispondrá de estructuras de datos complejas, ya que los datos se mostrarán cada uno en su contenedor.

Los tipos de datos que serán habituales serán los datos formados por tipos String (datos alfanuméricos considerados como cadenas de texto) y números enteros.

5.2. C2_Javascript

5.2.1. Tipo

Este componente es una entidad, de tipo estático, que pertenece a la categoría Vista del patrón de diseño MVC, encargado de realizar operaciones sobre el componente Interfaz para dotarle de contenidos dinámicos.

5.2.2. Propósito

Este componente cumple los siguientes requisitos software, presentes en el documento de requisitos software:

- SRF-007
- SRF-009
- SRF-010
- SRF-011
- SRF-012
- SRF-013
- SRF-014
- SRF-015
- SRF-016
- SRF-017
- SRF-018
- SRF-019
- SRF-020
- SRF-021
- SRF-022
- SRO-001

5.2.3. Función

La función de este componente es la de servir de conexión entre el componente C1_Interfaz y el componente C3_Controlador.

Es el encargado de recopilar los datos introducidos a través de la interfaz y pasarlos al controlador, para que puedan ser tratados, ya sea para almacenar información o como

parámetros para los distintos métodos. También es el encargado de recuperar la información generada por el controlador y presentarla en la interfaz, mediante funciones AJAX, que permiten modificar la interfaz en tiempo de ejecución sin necesidad de realizar actualizaciones de la interfaz completa.

5.2.4. Subordinados

- C21_Funciones
- C22_Librerías

5.2.5. Dependencias

Este componente depende de las librerías de Javascript incorporadas y del componente C1_Interfaz, del que toma los datos o parámetros y al que devuelve los resultados generados por el componente C3_Controlador.

También depende directamente del componente C3_Controlador, encargado de tratar la información facilitada por este componente y de recibir los resultados generados por él.

5.2.6. Interfaces

Este componente dispone de interfaces con los siguientes componentes:

C1_Interfaz

C21_Funciones

C22_Librerías Javascript

C3_Controlador

5.2.7. Recursos

Se requiere de un servidor web y un servidor de bases de datos para la correcta ejecución del sistema.

Para acceder al mismo es necesario un PC con sistema operativo que disponga de conexión a la red y un navegador web compatible.

5.2.8. Referencias

No aplica.

5.2.9. Procesamiento

Este componente realiza dos tipos de procesamiento:

1. Gestión de los datos tomados desde la interfaz y creación de las estructuras de datos necesarias para remitirlas al componente C3_Controlador.

2. Recuperar los datos generados en el componente C3_Controlador y mostrarlos en el componente C1_Interfaz.

El acceso a los datos de la interfaz se realiza mediante el acceso al DOM (Document Object Model) de la página web, a través de los identificadores de cada elemento presente.

Una vez se dispone del acceso al elemento, se puede recuperar la información contenida en él o modificarla, así como el resto de atributos del mismo.

5.2.10. Datos

Los formularios de recogida de datos se encontrarán inicialmente vacíos, es decir, no contendrán datos, lo que será representado mediante la cadena vacía "". Este componente no dispondrá de estructuras de datos complejas, ya que los datos se mostrarán cada uno en su contenedor.

Los tipos de datos que serán habituales serán los datos formados por tipos String (datos alfanuméricos considerados como cadenas de texto) y números enteros.

5.3. C21_Funciones

5.3.1. Tipo

Este componente es una entidad, de tipo estático, que pertenece a la categoría Vista del patrón de diseño MVC, contiene los diferentes métodos y funciones que actúan entre la interfaz y el controlador.

5.3.2. Propósito

Este componente cumple los siguientes requisitos software, presentes en el documento de requisitos software:

- SRF-007
- SRF-009
- SRF-010
- SRF-011
- SRF-012
- SRF-013
- SRF-014
- SRF-015
- SRF-016
- SRF-017
- SRF-018
- SRF-019
- SRF-020
- SRF-021
- SRF-022

- SRO-001

5.3.3. Función

Este componente estará formado por uno o varios ficheros, los cuales contendrán los distintos métodos o funciones encargados de acceder al componente C1_Interfaz para recopilar la información contenida en la interfaz web que se presenta al usuario y que éste ha introducido o modificado. La función de este componente consiste también en enviar información al componente C3_Controlador, en forma de parámetros o flags de control, para que se realicen las operaciones necesarias sobre los datos enviados. Asimismo, recuperar á información del componente C3_Controlador para mostrarla en la interfaz

5.3.4. Subordinados

- No aplica

5.3.5. Dependencias

Este componente depende de las librerías de Javascript incorporadas y del componente C1_Interfaz, del que toma los datos o parámetros y al que devuelve los resultados generados por el componente C3_Controlador.

También depende directamente del componente C3_Controlador, encargado de tratar la información facilitada por este componente y de recibir los resultados generados por él.

5.3.6. Interfaces

Este componente dispone de interfaces con los siguientes componentes:

C1_Interfaz

5.3.7. Recursos

Es necesario que el navegador en el que se ejecute la aplicación tenga activado y disponga de soporte para Javascript. También son necesarias las librerías Javascript pertinentes enlazadas con la página principal de la aplicación.

5.3.8. Referencias

No aplica.

5.3.9. Procesamiento

Este componente, estructurado en distintos métodos, recupera información de la interfaz accediendo al DOM mediante comandos Javascript. Posteriormente, esta información se remite al controlador de la aplicación, estructurando la información recibida en parámetros, junto con un identificador de acción, usando funciones AJAX.

Una vez la información es procesada por el controlador, el método recupera la información y actualiza la interfaz con los resultados recuperados.

5.3.10. Datos

Los formularios de recogida de datos se encontrarán inicialmente vacíos, es decir, no contendrán datos, lo que será representado mediante la cadena vacía `""`. Este componente no dispondrá de estructuras de datos complejas, ya que los datos se mostrarán cada uno en su contenedor.

Los tipos de datos que serán habituales serán los datos formados por tipos String (datos alfanuméricos considerados como cadenas de texto) y números enteros.

5.4. C22_Librerías

5.4.1. Tipo

Este componente es una entidad, de tipo estático, que pertenece a la categoría Vista del patrón de diseño MVC, contiene las librerías básicas de Javascript para el código AJAX y la interfaz de la aplicación.

5.4.2. Propósito

Este componente cumple los siguientes requisitos software, presentes en el documento de requerimientos software:

- SRF-007
- SRF-009
- SRF-010
- SRF-011
- SRF-012
- SRF-013
- SRF-014
- SRF-015
- SRF-016
- SRF-017
- SRF-018
- SRF-019
- SRF-020
- SRF-021
- SRF-022
- SRO-001

5.4.3. Función

Este componente estará formado por uno o varios ficheros, los cuales contendrán los distintos métodos o funciones en los que se basará el módulo C21_Funciones, para la realización de

todas las funciones AJAX disponibles en la librería, con el fin de facilitar el acceso a los diferentes elementos de la Interfaz y el acceso a los servicios que provee el controlador.

5.4.4. Subordinados

No aplica

5.4.5. Dependencias

No aplica

5.4.6. Interfaces

Este componente dispone de interfaces con los siguientes componentes:

- C21_Interfaz

5.4.7. Recursos

Es necesario que el navegador en el que se ejecute la aplicación tenga activado y disponga de soporte para Javascript.

5.4.8. Referencias

No aplica.

5.4.9. Procesamiento

Este componente provee todos los métodos necesarios para el uso de funciones avanzadas AJAX, basadas en el uso de la librería JQUERY, que da al usuario una serie de métodos estructurados que permite explotar al máximo las ventajas de esta librería.

5.4.10. Datos

Los métodos contenidos en esta librería trabajarán con los datos que se le pasen como parámetros desde el componente C21_Funciones. Estos datos serán cadenas de datos String y números enteros.

5.5. C3_Controlador

5.5.1. Tipo

Este componente es una entidad, de tipo estático, que pertenece a la categoría Controlador del patrón de diseño MVC, contiene los métodos necesarios para procesar la información recibida desde la interfaz web.

5.5.2. Propósito

Este componente cumple los siguientes requisitos software, presentes en el documento de requerimientos software:

- SRF-001
- SRF-010
- SRF-011
- SRF-012
- SRF-013
- SRF-015
- SRF-016
- SRF-017
- SRF-019
- SRF-023
- SRF-024
- SRI-001
- SRI-002
- SRI-003
- SRY-001
- SRY-003
- SRM-001
- SRM-002
- SRM-003

5.5.3. Función

Este componente estará formado por uno o varios ficheros, los cuales contendrán los distintos métodos que gestionarán la información recibida desde la interfaz, en forma de parámetros y que se encargará de hacer los accesos necesarios al componente C31_Clases para gestionar los datos de la base de datos, ya sean inserciones o recuperaciones de los datos.

5.5.4. Subordinados

C31_Clases.

5.5.5. Dependencias

C21

5.5.6. Interfaces

Este componente dispone de interfaces con los siguientes componentes:

- C21_Interfaz

5.5.7. Recursos

Es necesario que el navegador en el que se ejecute la aplicación tenga activado y disponga de un intérprete de PHP en la versión 5.3.

5.5.8. Referencias

No aplica.

5.5.9. Procesamiento

El componente C2_Javascript remitirá la información, en forma de parámetros, al componente C3_Controlador, usando métodos POST. El componente C3_Controlador se encargará de procesar la información recibida, de acuerdo a las reglas de pautado. Entre las operaciones que realiza, se encuentran las siguientes:

- Comprobación de los datos que se reciben.
- Proceso de los datos recibidos
 - Almacenamiento de los datos
 - Recuperación de datos acorde a los parámetros recibidos
- Estructuración de los datos en estructuras más complejas

5.5.10. Datos

Los datos manejados en este componente estarán formados, inicialmente, por los strings y los números enteros remitidos desde el componente C2_Javascript. Según sea necesario el tratamiento de estos datos, se crearán estructuras más complejas, como arrays o listas, llegando al uso de objetos en PHP para el tratamiento correcto de los datos.

5.6. C31_Clases

5.6.1. Tipo

Este componente es una entidad, de tipo estático, que pertenece a la categoría Controlador del patrón de diseño MVC.

5.6.2. Propósito

Este componente cumple los siguientes requisitos software, presentes en el documento de requerimientos software:

- SRF-001
- SRF-010
- SRF-011
- SRF-012
- SRF-013
- SRF-015
- SRF-016
- SRF-017
- SRF-019
- SRF-023
- SRF-024
- SRI-001
- SRI-002
- SRI-003
- SRY-001
- SRY-003
- SRM-001
- SRM-002
- SRM-003

5.6.3. Función

Este componente, compuesto por varias clases en PHP, es el encargado de gestionar los datos, agrupándolos en estructuras más complejas aprovechando la capacidad de PHP de gestionar objetos. Permite acceso a los datos estructurados en estructuras complejas y su agrupación para crear nuevas estructuras.

Una de estas clases, será la encargada de crear el fichero XML que contendrá la pauta prescrita.

5.6.4. Subordinados

No aplica.

5.6.5. Dependencias

Este componente depende de C3_Controlador.

5.6.6. Interfaces

Este componente dispone de interfaces con los siguientes componentes:

- C3_Controlador

5.6.7. Recursos

Es necesario que el navegador en el que se ejecute la aplicación tenga activado y disponga de un intérprete de PHP en la versión 5.3.

5.6.8. Referencias

No aplica.

5.6.9. Procesamiento

Cada una de las clases contenidas en este componente tendrá una función específica acorde a los datos que contenga. Su misión es proveer una interfaz a cada estructura de datos contenida y realizar operaciones sobre estos datos, volviendo a almacenarlos en la propia clase o remitiéndolos al componente C3_Controlador, para su posterior envío a la base de datos o a la interfaz.

5.6.10. Datos

Los datos manejados en este componente estarán formados, inicialmente, por los strings y los números enteros remitidos desde el componente C3_Controlador. Según sea necesario el tratamiento de estos datos, se crearán estructuras más complejas, como arrays o listas, llegando al uso de objetos en PHP para el tratamiento correcto de los datos.

5.7. C4_Acceso a Datos

5.7.1. Tipo

Este componente es una entidad, de tipo estático, que pertenece a la categoría Modelo del patrón de diseño MVC.

5.7.2. Propósito

Este componente cumple los siguientes requisitos software, presentes en el documento de requerimientos software:

- SRF-023
- SRO-003
- SRS-001
- SRS-004
- SRU-001
- SRT-002
- SRY-001
- SRY-003
- SRY-004
- SRP-001
- SRQ-002

5.7.3. Función

Este componente, compuesto por varias clases en PHP, es el encargado de realizar el acceso a los datos contenidos en la base de datos, de manera que contiene los métodos necesarios para realizar la extracción y la inserción de datos en la base de datos, manteniendo la integridad de los datos.

5.7.4. Subordinados

C41_Datos.

5.7.5. Dependencias

Este componente depende de C42_Conexiones.

5.7.6. Interfaces

Este componente dispone de interfaces con los siguientes componentes:

- C42_Conexiones

5.7.7. Recursos

Es necesario que el navegador en el que se ejecute la aplicación tenga activado y disponga de un intérprete de PHP en la versión 5.3 y de un servidor de bases de datos.

5.7.8. Referencias

No aplica.

5.7.9. Procesamiento

Este componente del sistema será el encargado de realizar las consultas generadas en el componente C3_Controlador a la base de datos. Las consultas a la base de datos podrán ser de dos tipos:

- Recuperar datos: El componente recibirá como parámetros la consulta que se debe realizar y devolverá una serie de objetos con los datos recuperados desde la base de datos. Si la recuperación de datos es correcta, se remitirá también un mensaje de aviso indicando que todo fue correcto. En caso contrario, se indicará un mensaje de error con información relevante para su depuración.
- Inserción de datos. El componente recibirá la consulta con los datos que se deben insertar en la base de datos. Si la inserción se realiza correctamente se devolverá un mensaje indicando que todo fue correcto. En caso contrario, se indicará cuál fue el error ocurrido.

5.7.10. Datos

Los datos que manejará este componente serán, por un lado, las consultas recibidas desde el componente C3_Controlador, y por otro, los datos almacenados directamente en la base de datos, correspondientes a la información usuarios, medicamentos y pautas.

5.8. C41_Conexiones

5.8.1. Tipo

Este componente es una entidad, de tipo estático, que pertenece a la categoría Modelo del patrón de diseño MVC.

5.8.2. Propósito

Este componente cumple los siguientes requisitos software, presentes en el documento de requerimientos software:

- SRF-023
- SRO-003
- SRS-001
- SRS-004
- SRU-001
- SRT-002
- SRY-001
- SRY-003
- SRY-004
- SRP-001
- SRQ-002

5.8.3. Función

Este componente, será el encargado de gestionar los datos de conexión del sistema con la base de datos elegida. Inicialmente, controlará la conexión con una base de datos de tipo MySQL, dejando al sistema preparado para poder realizar consultas de todo tipo sobre la misma. El hecho de separar la conexión del sistema a la base de datos permitirá integrar, en caso necesario, conexiones a otros sistemas de bases de datos o a otras bases de datos alojadas en otras ubicaciones.

5.8.4. Subordinados

No aplica.

5.8.5. Dependencias

No aplica.

5.8.6. Interfaces

Este componente dispone de interfaces con los siguientes componentes:

- C4_Acceso a datos

5.8.7. Recursos

Es necesario que el navegador en el que se ejecute la aplicación tenga activado y disponga de un intérprete de PHP en la versión 5.3 y de un servidor de bases de datos.

5.8.8. Referencias

No aplica.

5.8.9. Procesamiento

Este componente será invocado desde el componente C4_Acceso a datos cada vez se vaya a realizar un acceso a los datos contenidos en la base de datos. Antes de poder realizar la consulta será necesario establecer la conexión a la base de datos, con los privilegios adecuados. Una vez se realiza la consulta, es necesario cerrar la misma usando un método específico para tal fin.

5.8.10. Datos

Los datos que manejará este componente serán, por un lado, las consultas recibidas desde el componente C3_Controlador, y por otro, los datos almacenados directamente en la base de datos, correspondientes a la información usuarios, medicamentos y pautas.

6. Viabilidad y estimación de recursos

Los recursos necesarios para la implementación de la aplicación y su posterior despliegue y mantenimiento se han estimado en base a la disposición de tres entornos, a saber, entorno de desarrollo, entorno de pre-producción y entorno de producción. Para cada uno de los entornos se establecen unos requisitos mínimos para su correcto funcionamiento.

6.1. Desarrollo

Para el desarrollo de la aplicación se estiman necesarios los siguientes recursos:

- Equipo PC o Mac con procesador a 2.0 GHz o superior
- Entorno Wamp o Lamp