

# Consultas Medicas

## Documento de arquitectura de software

### Analistas y Desarrolladores

Nombre	Apellido	Colaboración	Profesión
David	Guzmán	PM/Lead Backend Developer	Estudiante de Ingeniería en Sistemas
Rossangela	López	Analyst and Designer/Architecture	Estudiante de Ingeniería en Sistemas
Abel	Méndez	DBA/Frontend Developer	Estudiante de Ingeniería en Sistemas

### Tabla de contenido

- [1. Introduccion](#)
  - [1.1 Proposito](#)
  - [1.2 Alcance](#)
  - [1.3 Definiciones, acrónimos y abreviaturas](#)
  - [1.4 Referencias](#)
  - [1.5 Visión general](#)
- [2. Representación arquitectónica](#)
- [3. Metas y restricciones arquitectónicas](#)
- [4. Vista de casos de uso](#)
- [5. Vista lógica](#)
  - [5.1 Visión de conjunto](#)
  - [5.2 Paquetes de diseño arquitectónicamente significativos](#)
  - [5.3 Realizaciones de casos de uso](#)
- [6. Vista de proceso](#)
- [7. Vista de implementación](#)
- [8. Vista de implementación](#)
  - [8.1 Visión general](#)
  - [8.2 Capas](#)
- [9. Vista de datos \(optional\)](#)
- [10. Tamaño y rendimiento](#)
- [11. Calidad](#)

## Documento de arquitectura de software

### 1. Introducción

El procesamiento de los datos ha sido una de las mayores preocupaciones de las organizaciones a través de la historia, de tal manera que se han venido creando y evolucionando herramientas que cumplen con la tarea de ayudar a la obtención de información por medio de un proceso adecuado a los datos. En este aspecto, las tecnologías informáticas han puesto a disposición de la humanidad una serie de herramientas computarizadas que permiten el manejo de la información en forma veraz rápida y oportuna, además de contar con un alto grado de confiabilidad. Es por esto que es tan importante el control sobre las consultas clínicas, tener facilidad para accederlos y al mismo tiempo ejercer el control sobre las personas que tienen acceso a estas. Con esta idea se crea una herramienta capaz de brindar una solución a esta situación de orden, dándole al usuario un esquema o procedimiento a seguir con el fin de organizar, archivar y acceder con facilidad las consultas medicas de un paciente. El objetivo principal del software "Consultas Medicas" es proporcionar un sistema eficiente y seguro. A través de esto, la atención al paciente se integra en diferentes departamentos de atención médica.

#### 1.1 Propósito

Sistema de Control de Consultas Medicas con el fin recopilar los datos del paciente, incluida la información personal del paciente, historias clínicas y medicamentos, en formato digital, almacenados y accedidos por usuarios autorizados dentro de una plataforma digital segura.

#### 1.2 Alcance

Proporcionar consultas medicas sobre la informacion de la medicación, los procedimientos y el curso general de recuperación del paciente, esto podriamos decirlo que serviría como un diario digital para los Doctores que trabajan en el hospital ya el software esta diseñado para la información clínica del paciente de un hospital específico, incluyen el registro computarizado del paciente, las consultas de atención médica, el registro personal de salud del paciente y los de datos clínicos del paciente.

### 1.3 Definiciones, acrónimos y abreviaciones

Requisito: Base de Datos: MYSQL Fronted en un servidor:apache HTTP Backend en un servidor remoto: tomcat o glassfish  
Tecnologias: react, Spring Boot, JPA y JS

Funcionalidad: Mostrar los registros de las consultas de los pacientes atraves de una base de datos para llevar el control de procesos en el software

Interfaces Externas: harware:computadoras o laptops software: aplicacion del proyecto ya ejecutado en un servidor

Cliente/Usuarios:podra verificar agregar y consultar información de los pacientes del hospital a si mismo ver su diagnostico específico.

### 1.4 Referencias

Pressman. Ingeniería del Software: Un Enfoque Práctico. 7 Ed. McGraw Hill, 2010.

Sommerville Ian. Ingeniería del Software. Pearson 7a. Edición. 2005.

S. Sanchez, M. Sicilia, D. Rodriguez. Ingenieria del software. Un enfoque desde la guía SWEBOK. Alfaomega. 2012.

Rumbaugh J., Jacobson I., Booch G. El Lenguaje Unificado de Modelado. Manual de Referencia. Editorial Addison-Wesley – 2000

Braude Eric. Ingeniería de Software. Una perspectiva orientada a objetos. Alfaomega Grupo Editor S.A. 2003.

Daniel Bolaños, Almudena Sierra, Idoia Alarcón. Pruebas de software y JUnit. Un análisis en profundidad y ejemplos prácticos. Pearson Education. 2007.

### 1.5 Visión general

Análisis y diseño de una aplicacion de un software que permita tener consultas medicas de los pacientes hecha con un backend y fronted utilizando un servidor glassfish y una base de datos remota MYSQL para la aplicacion y asi darle el usuario la facilidad para la creación de consultas medicas.Así para facilitarle la gestión de información al personal administrativo y médico.

## 2. Representación arquitectónica

Vista de Caso de Uso: La interacción del médico con el sistema de control de consultas médicas asi como insertar, leer, modificar y eliminar una consulta del sistema.

Vista Lógica: El sistema esta compuesto por 4 tapas

*la capa del modelo:donde las entidades de la base de datos se relacionan entre sí atraves de interfaces.*

*la capa del Dao: donde se encuentran todas las entidades específicas atraves de interfaces como medio de interacción.*

*la capa de servicio: donde se encontraran todos los servicios funcionales y principales de la aplicación del sistema.*

*la capa del controlador:donde separa los datos de la aplicacion y se encuentra la interfaz de usuario.*

Vista de Procesos: Se encuentran todas las interacciones que tiene las interfaces del sistema para conectarlas en un mismo fin que sería la capa del controlador y la capa de servicio del sistema.

Vista de Despligue: Se desplegara la aplicación en un servidor en la nube donde el usuario tiene el poder de modificar, eliminar o buscar una consulta de la base de dato, en este caso sería MSQl.

Vista de Implementacion:

*entidades: se implemento 9*

*entidades:Consultation,Disease,DoctorEntity,DoctorSpecialization,Medicine,MedicineForDisease,PatientEntity,PrescriptionEntity,SpecializationEntity*

*Interfaces:interfaces de las entidades ya dichas tanto en la capa del modelo,Dao, servicios y del controlador del sistema.*

*capas:capa del modelo, controlador, servicio y Dao.*

*Frontend en un servidor :(Apache HTTP)*

*Backend en otro servidor:(Glassfish, Tomcat)*

*Base de datos remota (MySQL, Firebase)*

## 3. Metas y restricciones arquitectónicas

## METAS:

Garantizar la conservación de las consultas clínicas.

Reducir el volumen de archivado, el tiempo y los recursos de búsqueda de historias clínicas de esas dichas consultas.

Reducir los gastos con la implementación del software.

Cumplir el periodo de 1 mes, establecido para la realización de este proyecto.

Aplicar los conocimientos de programación básica y avanzada, base de datos y análisis de sistemas para el desarrollo y avance del proyecto.

Cumplir con los objetivos de manera clara y precisa en el tiempo establecido.

## RESTRICCIONES

Falta de compromiso por parte de miembros del equipo.

Conflicto en el equipo del proyecto, con respecto a las inasistencias en las reuniones previamente pactadas.

No entregar a tiempo los entregables.

Conflicto con el ordenador al momento de poner a prueba el prototipo. (el ordenador tenga virus, o que existan fallas eléctricas que afecten al ordenador)

## 4. Vista de casos de uso

Caso de Uso: el usuario es el único que tiene contacto con el sistema en este caso serían los médicos y trabajadores del hospital.

### FLUJO PRINCIPAL:

paso1: El usuario selecciona la opción de insertar consulta, llenando el formulario con los datos deseados.

paso2: El usuario guarda los datos a través de una base de datos MySQL.

paso3: El usuario puede modificar la consulta que ha sido propuesta anteriormente.

paso4: El usuario puede eliminar la consulta deseada y no se registrará en la base de datos.

### EXCEPCIONES:

Error al guardar la información en la base de datos

falta de datos al momento de llenar el formulario

siempre llenar todos los datos obligatorios del formulario

## 5. vista lógica

### 1. PAQUETES DE NIVEL SUPERIOR

a. as2final\_hospital

### 2. PAQUETES DEL SUBSISTEMA DE DISEÑO

a. gt

b. edu

c. umg

e. as2final\_hospital(from umg)

### 3. PAQUETES DEL SISTEMA

a. controller(from as2final\_hospital)/import service-model

b. service(from as2final\_hospital)/import dao-model

c. dao(from as2final\_hospital)/import model/model

e. model(from as2final\_hospital)import model

## 5.1 visión de conjunto

### 1.CAPAS DEL SISTEMA

- a.capa del modelo
- b.capa de servicio
- c.capa del controlador
- d.capa del DAO

### 2.ENTIDADES Y ATRIBUTOS

- a.Consultation:id\_consultation,id\_paciente,id\_doctor,complaints,diagnosis,treatment,date\_recorded
- b.Disease:id\_disease,disease\_name,id\_medicineForDisease
- c.DoctorEntity:id,first\_name,middle\_name,last\_name,maiden\_name,address1,address2,gender,birthdate,collegiate\_number
- d.DoctorSpecialization:id\_doctor,id\_specialization
- e.Medicine:medicine\_name,id\_medicineForDisease,id\_medicine
- f.MedicineForDisease:id\_medicine,id\_medicineForDisease
- g.PatientEntity:first\_name,middle\_name, maiden\_name,last\_name,address1,address2,phone1,phone2,gender,birthdate
- h.PrescriptionEntity:id\_consultation,id\_medicine
- i.SpecializationEntity:id\_specialization.id\_doctor

### 3.INTERFACES

- a.interfaces del controlador
- b.interfaces del dao
- c.interfaces del modelo
- d.interfaces del servicio

## 5.2 Paquetes de diseño arquitectónicamente significativos

### 1.PAQUETES DE NIVEL SUPERIOR

- a. as2final\_hospital:paquete donde se encuentra las capas del sistema que se esta realizando

### 2.PAQUETES DEL SUBSISTEMA DE DISEÑO

- a.gt:subpaquete del sistema
- b.edu:subpaquete del sistema
- c.umg:subpaquete del sistema
- e. as2final\_hospital(from umg):subpaquete del sistema

### 3.PAQUETES DEL SISTEMA

- a.controller(from as2final\_hospital)/import service-modeles :es la capa que separa los datos de la aplicación, la interfaz de usuario y la lógica de los componentes del sistema
- b.service(from as2final\_hospital)/import dao-model: es donde se encuentran desplegados los datos del sistema como una capa de presentacion
- c.dao(from as2final\_hospital)/import model/model : es la capa que le da acceso a los datos del sistema
- e.model(from as2final\_hospital)import model : es la capa donde se encuentra la estructura aquitectònica del sistema y donde se evaluan las entidades y atributos.

## 5.3 Realizaciones de casos de uso

## 1.ACTOR:

a.usuario: personas de la área médica del hospital

## 2.ACCIONES:

a. insertar:El usuario selecciona la opción de insertar un nueva consulta.

b.leer consulta :el usuario lee la consulta agregada

c. modificar:El usuario puede modificar la consulta si esta errónea actualizandola en la base de datos

d. eliminar :El usuario puede eliminar una consulta agregada y no se guardara en la base de datos

## 6. Vista de proceso

### 1.PROCESOS:

El diseño de este subsistema quedó fuera del alcance del proyecto. Se prescinde del mismo recurriendo a una herramienta externa para la implementación del framework. La salida del trabajo de este subsistema es una representación intermedia del proceso de negocio dicha representación intermedia es entrada del subsistema "Ejecución de Procesos" que se describe a continuación.

### 2.EJECUCION DE PROCESOS

```
*model(from as2final_hospital)import model
```

```
*dao(from as2final_hospital)/import model/model
```

Estos dos componentes o capas del sistema tienen como una representación intermedia del proceso de negocio con el ejecutable del sistema

\*Estos dos componentes o capas del sistema tienen como una representación intermedia del proceso de negocio con el ejecutable del sistema

```
*service(from as2final_hospital)/import dao-model
```

Estas capas tienen como representación del ejecutable obtenida de la aplicación que estamos trabajando en el proyecto

El resultado de la resolución de un "servicio" puede ser una ontología; dentro de esta ontología pueden estar contenidos elementos a ser utilizados como resultados intermedios del proceso con los que se precise hacer algún cálculo o ser entrada de una nueva invocación a un servicio

### 3.COMPONENTES DEL PROCESO:

a.Transformación: recibe la representación intermedia de un proceso de negocio y genera un archivo con la representación ejecutable del mismo.

b.Instalación: se encarga de tomar la representación ejecutable de un proceso de negocio, resultado del trabajo del módulo Transformación, y dejarla disponible en un motor de ejecución de procesos de negocio de forma que pueda ser ejecutada.

c.Motor de ejecución: este módulo representa el motor de ejecución de procesos de negocio; los procesos son deployados dentro de este componente y se hacen accesibles para ser ejecutados generando una interfaz para cada uno de ellos; en general son expuestos en forma de web service(glassfish)

## 7. Vista de Deployment

A continuación se describen los nodos presentes del proyecto "consultas Médicas":

- Servidor Glassfish: representa el equipo donde correrá el subsistema Ejecución de Procesos. Por tanto, se ejecutan aquí las funcionalidades de transformación e instalación de procesos de as2final\_hospital así como el motor de ejecución de procesos. Tales servicios son expuestos mediante respectivas interfaces web service.

- PC Analista: En este nodo correrá el subsistema Definición de Procesos. Se comunica con el Servidor glassfish para la transformación e instalación del proceso definido; la comunicación es vía servidor a través de la interfaz web service expuesta por dicho nodo.

- Cliente de Proceso: representa el equipo donde corre la aplicación cliente as2final\_hospital que desea ejecutar un proceso de negocio disponible en el Servidor glassfish. No hay componentes del framework corriendo en este nodo. Los procesos son ejecutados invocando el web service publicado para cada uno de ellos.

- Servidor Resolución de Servicios: representa el equipo donde correrá el subsistema Resolución de Servicios. Recibe pedidos del motor de ejecución de servicios corriendo en el Servidor glassfish. La comunicación es nuevamente a través de una interfaz web service.

• **Repositorio de Servicios:** representa un equipo donde se mantendrá un registro de servicios disponibles a los cuales podrá acceder el subsistema Resolución de Servicios para buscar los servicios adecuados para una determinada solicitud. En este caso se usará los repositorios de github con los nombres: `as2final_backend` y `as2final_hospital_froned`.

## 8. Vista de Implementación

### COMPONENTES:

- **resolve:** es la operación que dispara la resolución de un servicio. Es invocada por el subsistema Ejecución de Procesos, en particular por el componente de ejecución "controlador". Recibe como parámetro de entrada un objeto "servicio". La forma de este objeto podrá variar de acuerdo al lenguaje empleado para el manejo semántico de los servicios pero sustancialmente debe contener la descripción semántica del servicio a resolver y los parámetros de invocación. El resultado de esta operación se encapsula en un objeto de tipo de "entidad" el cual contiene los parámetros de retorno del servicio invocado y el objeto "controlador" con el que se realizó la invocación.
- **find:** se encarga de hallar el conjunto de servicios que matcheen el perfil de servicio deseado. A partir de un parámetro de tipo único que indica dónde se halla la ontología que describe el tipo de servicio requerido, consulta el repositorio de servicios (github) en busca de los que matcheen el perfil y retorna en un objeto de tipo "lista" la lista de servicios obtenida y el grado de matcheo de cada servicio.
- **select:** se encarga de seleccionar el servicio a invocar de los incluidos en el objeto "lista" devuelto por la función `find`. El criterio para seleccionar el servicio se representa mediante un objeto "inserción", el cual permite la parametrización de la selección en caso de que el componente que implemente "controlador" así lo permita. El resultado de esta operación es el servicio elegido representado por un objeto "lista"; es decir, un elemento de la lista "resultado" recibida como entrada.
- **invoke:** mediante esta operación se ejecuta el servicio en el controlador del sistema.

### 8.1 Visión General

#### 1. Diseño de Proceso de `as2final_hospital`:

- Representa una de las funcionalidades más importantes en la gestión de procesos de negocio, el diseño de los mismos.
- Introduce la necesidad de contar con un lenguaje gráfico de representación de procesos de "consultas médicas" así como un editor con el cual poder construir dichos procesos.
- Este componente agrega la complejidad de manejar conceptos semánticos gráficamente de forma amigable e intuitiva para el usuario. Este usuario será en general un analista de negocio que posiblemente no tenga conocimientos profundos a nivel técnico informático.

#### 2. Transformación e Instalación de un Proceso

- Esta funcionalidad es sumamente importante pues permite pasar de un nivel de abstracción a otro dentro del modelo propuesto.
- Tanto este caso como el anterior genera la necesidad de buscar un formato de representación intermedia para la especificación del proceso.
- Una vez fijado dicho formato se debe especificar un mecanismo de transformación de uno al otro. Más precisamente poder generar una representación ejecutable del proceso a partir de su representación intermedia.

#### 3. Ejecución de Proceso

- Funcionalidad básica que debe proveer el framework para posibilitar la ejecución del proceso.
- La ejecución de un proceso de dicho sistema "`as2final_hospital`" es iniciada por una aplicación cliente a través de una interfaz de servicio provista por el framework.
- Como parte esencial de la ejecución de un proceso se debe resolver la invocación de servicios a partir de una descripción semántica de los mismos.
- La invocación de un servicio a partir de una descripción semántica implica tres procesos fundamentales:
  - \* Búsqueda de servicios que se correspondan semánticamente con la descripción del perfil de servicio requerido.
  - \* Contar con un mecanismo para la selección del mejor servicio, dentro de los hallados en el proceso anterior, en base a algún criterio preestablecido. Un ejemplo de esto es seleccionar el más adecuado en base a indicadores de la calidad de servicio de los mismos.
  - \* En la invocación del servicio propiamente dicha se deben manejar los parámetros teniendo en cuenta sus diferentes tipos de datos, incluyendo ontologías.
  - \* Introduce además la necesidad de especificar un formato de mensaje para la invocación de servicios y para la devolución de resultados.

### 8.2 Capas

## CAPAS DEL SISTEMA

a.controller(from as2final\_hospital)/import service-modeles :es la capa que separa los datos de la aplicación, la interfaz de usuario y la lógica de los componentes del sistema.

b.service(from as2final\_hospital)/import dao-model: es donde se encuentran desplegados los datos del sistema como una capa de presentacion.

c.dao(from as2final\_hospital)/import model/model : es la capa que le da acceso a los datos del sistema.

e.model(from as2final\_hospital)import model : es la capa donde se encuentra la estructura aquitectónica del sistema y donde se evaluan las entidades y atributos.

## 9. Vista de datos(opcional)

Una aplicación cliente invoca la ejecución de un proceso de "consultas médicas" a través de la interfaz de servicio expuesta para tal fin. El sistema comienza la ejecución del proceso resolviendo cada una de las invocaciones a servicios especificadas. Dependiendo del tipo de invocación a resolver, y en caso de ser necesario, el sistema busca un servicio concreto que cumpla con un perfil de servicio especificado.

## 10. Tamaño y rendimiento

### OBJETIVOS

La historia clínica es el documento fundamental de nuestra consulta, tanto desde un punto de vista asistencial como investigador y docente. Además tiene gran importancia desde un punto de vista legal. Por todo ello resulta fácil comprender la importancia que tiene el disponer de una buena historia clínica, entendiendo por "buena" una historia clínica en la que la información sea lo más clara posible y lo más accesible posible.

Un sistema informatizado pretende conseguir las siguientes metas:

- \*Identificación correcta de los pacientes atendidos.
- \*Disminución de los errores en el manejo de la información asistencial.
- \*Registro de signos y síntomas del paciente.
- \*Registro de los actos y decisiones asistenciales tomadas por los profesionales.
- \*Recuperación, interrelación, análisis y presentación adecuada de los datos registrados en cualquier punto del proceso asistencial.
- \* Presentación y prescripción adecuada de las instrucciones e informes asistenciales a los pacientes.
- \*Mayor interrelación entre los profesionales.
- \*Creación progresiva de bases de conocimiento médico para ayudar en el proceso de toma de decisiones clínicas.

### RENDIMIENTO:

Las características fundamentales del servidor son:

- \* Multiplataforma.
- \* Es un servidor de web conforme al protocolo HTTP/1.1.
- \*Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- \* Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- \* Se desarrolla de forma abierta.
- \*Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca react, un lenguaje de programación del lado del servidor.

## 11. Calidad

- \*Un sistema de consultas médicas facilita el proceso de asignación, cancelación y cumplimiento de citas médicas tanto para médicos como para pacientes, evitando largas colas en consultorios y esperas en la línea telefónica.
- \* La utilización de sistemas informáticos dentro de centros de salud y hospitales es indispensable, hoy en día, debido a las gigantescas

cantidades de información que se manejan. Así mismo, estos sistemas deben contar con altos estándares de seguridad ya que dicha información puede ser delicada y confidencial.

\*Debido a las herramientas utilizadas para el desarrollo del sistema y las tecnologías de alto nivel, el sistema desarrollado para esta disertación cuenta con los estándares de seguridad necesarios para un sistema de citas médicas, así como también la facilidad y rapidez de uso.

\*El sistema brinda un servicio al paciente de alta calidad, donde él se podrá sentir seguro y le facilitará mucho el proceso de petición de consultas médicas. Le ahorrará el tiempo que le tomaría esperando en la línea telefónica hasta que le asignen una cita o acercándose hasta el consultorio médico directamente.

\*El sistema debe ser utilizado como apoyo del médico para la asignación de citas médicas y sincronizarlo con las citas que se realicen por teléfono o presencialmente. Esto para evitar solapamiento de horarios.

\*Los médicos deben aprovechar la herramienta para organizar horarios y tiempos para la atención de pacientes, esto les facilitaría el trabajo a ellos y les ahorraría tiempo a sus pacientes.