



Sistema de Gestión de historias clínicas de nutrición

Pontificia Universidad Javeriana Cali

Procesos de Ingeniería De Software

Sistema de Gestión de historias clínicas de nutrición

DOCUMENTO DE DISEÑO

Katherine Camacho Calderón

Juan José Hoyos Urcué

Jeison Fernando Garcés

Mayo 29 del 2020

Versión 2.0



Sistema de Gestión de historias clínicas de nutrición

Pontificia Universidad Javeriana Cali

Procesos de Ingeniería De Software

Tabla de Contenido

1. INTRODUCCIÓN
 - 1.1. Propósito.
 - 1.2. Alcance.
 - 1.3. Definiciones,acrónimos y abreviaciones
 - 1.4. Referencias
2. REPRESENTACIÓN ARQUITECTÓNICA
3. OBJETIVOS Y RESTRICCIONES DE LA ARQUITECTURA
4. CASOS DE USO
5. VISTA LÓGICA
6. VISTA DE PROCESOS
7. VISTA FÍSICA
8. VISTA DE DESPLIEGUE
9. BASE DE DATOS
10. ANEXOS



Sistema de Gestión de historias clínicas de nutrición

Pontificia Universidad Javeriana Cali

Procesos de Ingeniería De Software

1. Introducción

1.1 Propósito

El propósito del documento de diseño del Sistema de gestión de historias clínicas de nutrición es proporcionar diagramas UML que permiten detallar la manera en la cual se debe hacer la implementación del sistema desde el punto de vista arquitectónico, lógico , físico , de procesos y de despliegue.

1.2 Alcance

Este documento tiene como objetivo proveer las herramientas necesarias para que los desarrolladores implementen el aplicativo web de gestión de historias clínicas en un lenguaje de programación orientado a objetos, con un servidor remoto que aloja la página y otro en el cual estará la base de datos.

1.3 Definiciones, acrónimos y abreviaciones

Aplicativo web: En la ingeniería de software se denomina aplicación web a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de internet o de una intranet mediante un navegador

Arquitectura de software: El concepto de arquitectura de software se refiere a la estructuración del sistema que, idealmente, se crea en etapas tempranas del desarrollo.



Sistema de Gestión de historias clínicas de nutrición

Pontificia Universidad Javeriana Cali

Procesos de Ingeniería De Software

Capa: Grupo de componentes que interactúan mediante ciertas interfaces para realizar una funcionalidad en específico

Disponibilidad: Se denomina disponibilidad a la posibilidad de una cosa o persona de estar presente cuando se la necesita.

Escalabilidad: La escalabilidad, término tomado en préstamo del idioma inglés, es la propiedad deseable de un sistema, una red o un proceso, que indica su habilidad para reaccionar y adaptarse sin perder calidad, o bien manejar el crecimiento continuo de trabajo de manera fluida, o bien para estar preparado para hacerse más grande

Integralidad: El término integridad de datos se refiere la correctitud y completitud de la información en una base de datos o en un sistema particular.

MER: Modelo entidad relación

Servidor: Un servidor es una aplicación en ejecución capaz de atender las peticiones de un cliente y devolverle una respuesta en concordancia. Los servidores se pueden ejecutar en cualquier tipo de computadora, incluso en computadoras con bombillo dedicadas a las cuales se les conoce individualmente como «el servidor».

1.4 Referencias

[1] A. Schiaffarino. (2019, marzo 12). Modelo cliente servidor. [Online]. Available: <https://blog.infranetworking.com/modelo-cliente-servidor/>

[2] V. Gómez. (2015, mayo 23). Arquitectura en Tres Capas. [Online]. Available: <https://instintobinario.com/arquitectura-en-tres-capas/>

[3] A. Lafuente. (2018, sep 27). Bases de datos relacionales vs. no relacionales: ¿qué es mejor?. [Online]. Available: <https://aukera.es/blog/bases-de-datos-relacionales-vs-no-relacionales/>



Sistema de Gestión de historias clínicas de nutrición

Pontificia Universidad Javeriana Cali

Procesos de Ingeniería De Software

[4]Evaluando Software. (2018, nov 16).Qué es la arquitectura de microservicios. [Online]. Available:<https://www.evaluandosoftware.com/que-es-la-arquitectura-de-microservicios/>

[5]Raúl Herranz Gómez (2014).BASES DE DATOS NOSQL: ARQUITECTURA Y EJEMPLOS DE APLICACIÓN[Online]. Available:<https://core.ac.uk/download/pdf/44310803.pdf>

2. Representación Arquitectónica

Arquitecturas Candidatas

CLIENTE SERVIDOR [1]

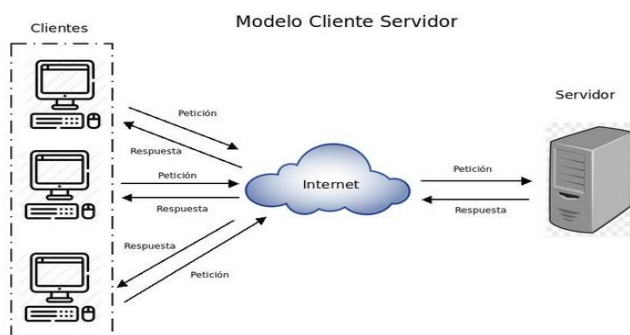


Figura 01. Esquema del Modelo Cliente Servidor

Esta arquitectura tiene dos partes claramente diferenciadas, el servidor es una de ellas y la otra parte está conformada por un cliente o grupo de clientes.

Donde el cliente suele ser las estaciones de trabajo que solicitan varios servicios al servidor, mientras que el servidor es una máquina que actúa como depósito de datos y a su vez como gestor de base de datos, esta se encarga de dar la respuesta demandada por el cliente.

Los componentes que se deben tener en cuenta a la hora de usar esta arquitectura son:



Sistema de Gestión de historias clínicas de nutrición

Pontificia Universidad Javeriana Cali

Procesos de Ingeniería De Software

Red: Conjunto de clientes, servidores y bases de datos unidos de forma física o no física.

Cliente: Es un demandante de servicios, puede ser un ordenador como también una aplicación de informática.

Servidor: Es un proveedor de servicios, puede ser un ordenador o una aplicación informática que envía información a los demás agentes de la red.

Protocolo: Conjunto de normas o reglas sobre el flujo de información en una red estructurada.

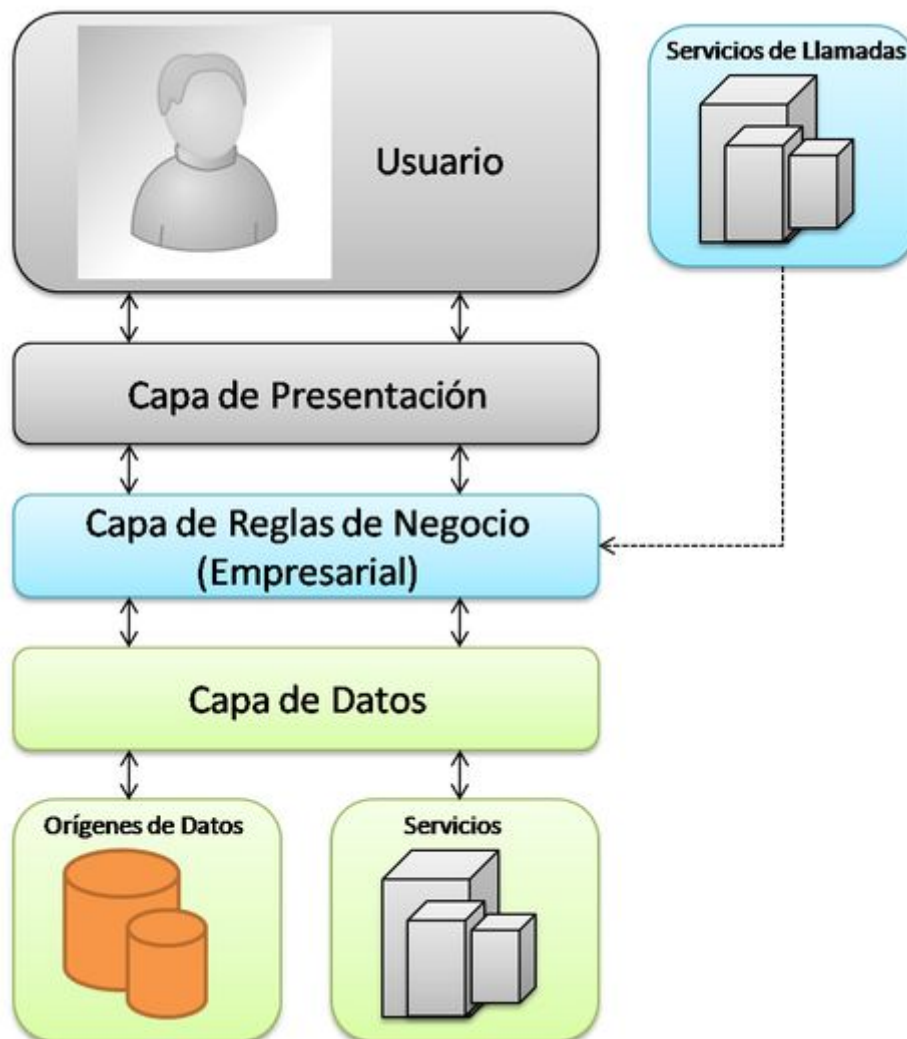
Servicios: Conjunto de información que busca responder las necesidades de un cliente.

Base de datos: Bancos de información ordenada, categorizada y clasificada que forman parte de la red.

ARQUITECTURA POR CAPAS [2]

La arquitectura de capas consiste en separar un sistema grande y complejo en grupos de componentes más sencillos que realicen funciones específicas y que tengan alto nivel de independencia. Estas capas están organizadas de manera vertical en forma de pila, por eso para realizar una operación desde la capa de más alto nivel es necesario pasar por todas las capas intermedias hasta llegar a la capa que provee el servicio solicitado, cada capa le agrega más funcionalidades al sistema, de tal manera que en la capa de más alto nivel el usuario tiene acceso a todas las capas inferiores pero implícitamente, es decir, desde la capa más alta no se puede conocer exactamente cómo está implementado el servicio o funcionalidad, sin embargo, si se pueden usar. El fuerte de esta arquitectura es que cada capa está totalmente aislada de las demás, algo que permite elaborar proyectos dinámicos con mayor facilidad, pues se pueden realizar cambios sin

afectar de manera drástica el sistema, solo a la capa local en la cual se efectúa dicho cambio. A continuación podemos observar el esquema de arquitectura de 3 capas:



Capa de presentación : En esta capa se crea la interfaz del usuario. Su única función es pasarle las acciones que realice el usuario a la capa de negocio.

Capa modelo de negocio : En esta capa se gestiona la lógica de la aplicación. Es donde se dice que se hace con los datos. Estará conectada con la capa de datos para poder realizar sus funciones.



Sistema de Gestión de historias clínicas de nutrición

Pontificia Universidad Javeriana Cali

Procesos de Ingeniería De Software

Capa de datos: Esta capa se encarga de guardar los datos. Será donde se gestione todo lo relativo a la base de datos y a la creación, edición y borrado de datos de ésta.

ARQUITECTURA DE MICROSERVICIOS[4]

Los microservicios, o arquitectura de microservicios, es un enfoque para el desarrollo de software en el que una aplicación grande se construye como un conjunto de componentes o servicios modulares.

Cada módulo admite una tarea específica o un objetivo de negocio y utiliza una interfaz simple y bien definida, como una interfaz de programación de aplicaciones (API), para comunicarse con otros conjuntos de servicios.

En la arquitectura de microservicios, una aplicación se divide en servicios. Cada uno ejecuta un proceso único y generalmente administra su propia base de datos. Un servicio puede generar alertas, registrar datos, admitir interfaces de usuario (UI), manejar la identificación o autenticación del usuario y realizar otras tareas.

El paradigma de microservicio proporciona a los equipos de desarrollo un enfoque más descentralizado para construir software. Los microservicios permiten que cada servicio sea aislado, reconstruido, re desplegado y administrado de forma independiente. Por ejemplo, si un programa no genera informes correctamente, puede ser más fácil rastrear el problema a ese servicio específico. Ese servicio específico podría luego probarse, reiniciarse, parchearse y volver a implementarse según sea necesario, independientemente de otros servicios.



Sistema de Gestión de historias clínicas de nutrición

Pontificia Universidad Javeriana Cali

Procesos de Ingeniería De Software

Arquitecturas Seleccionada

Analizando las anteriores arquitecturas decidimos que la más apta para el desarrollo del sistema de gestión de historias clínicas de nutrición es la arquitectura de 3 capas, ya que esta nos presenta tres capas totalmente diferenciables, lo que permite implementar el sistema de forma modular y por lo tanto realizar modificaciones sin afectar drásticamente el mismo, se puede modificar entonces la representación de la información pero no su tratamiento, y gracias a eso no debemos preocuparnos de solicitar que las vistas se actualicen, pues el modelo de negocio de la aplicación permite que este proceso se realice de forma automática y además cuando se quieran aumentar métodos o datos, la modificación se haría sólo en el modelo de negocio y su respectiva interfaz gráfica y no en todo el sistema.

3. Objetivos y Restricciones de la Arquitectura

Objetivo	Descripción
----------	-------------



Sistema de Gestión de historias clínicas de nutrición

Pontificia Universidad Javeriana Cali

Procesos de Ingeniería De Software

Escalabilidad	Se debe poder definir de manera clara y sin cabida a la ambigüedad los componentes que hacen parte del sistema de gestión de historias clínicas de nutrición, dichos componentes deben realizar una tarea específica, sus interfaces de interacción deben ser de alta generalidad, de tal manera que cada componente sea lo más independiente posible, garantizando que el sistema sea escalable a razón de la facilidad de modificaciones y adición de nuevas funcionalidades.
Disponibilidad	Esta arquitectura debe permitir tener una disponibilidad del sistema de un 85% del tiempo
Integralidad	Con la implementación de esta arquitectura el sistema debe estar en capacidad para hacer que piezas de software desarrolladas separadamente trabajen correctamente juntas.

Restricciones

- Ninguna capa conoce la implementación de las otras, por ende deben interactuar mediante diferentes interfaces, lo cual puede aumentar el



Sistema de Gestión de historias clínicas de nutrición

Pontificia Universidad Javeriana Cali

Procesos de Ingeniería De Software

tiempo requerido para realizar algunas funcionalidades, por lo tanto contrarresta el hecho de que la arquitectura permite la reutilización de código.

8. Base de datos

A continuación se listan algunos de los tipos de bases de datos que podríamos considerar para implementar en el proyecto del sistema de gestión de historias clínicas, se hace una breve descripción de cada uno de ellos y se listan sus ventajas y desventajas [3]

Bases de datos relacionales

Las bases de datos relacionales son la arquitectura más utilizada hoy en día. Esta consiste en un conjunto de tablas formadas por filas (registros) y columnas (campos), tal como una hoja de cálculo.

Los registros representan cada una de las instancias descritas en la tabla y los campos los atributos de dichas instancias. Por su parte, las tablas comparten ciertos campos entre ellas, los cuales son llamados “campos clave”. La función de estos es establecer relaciones entre las tablas.

Ventajas

- Esta arquitectura lleva más de 50 años en el mercado, por lo que, hoy en día, existen gran cantidad de productos diseñados específicamente para gestionarla.
- Permite tener la información muy bien organizada y estructurada.



Sistema de Gestión de historias clínicas de nutrición

Pontificia Universidad Javeriana Cali

Procesos de Ingeniería De Software

- La atomicidad es una de sus principales propiedades. Esta se refiere a la condición de que cada operación o se realiza en su totalidad o se revierte, lo cual evita que las operaciones queden a medias.

Desventajas

Los datos abstractos o no estructurados como los del big data no son admitidos.

El mantenimiento es muy costoso y complicado cuando la base de datos crece a un gran tamaño.

Los tiempos de respuesta suelen ralentizarse a medida que la base de datos crece.

Bases de datos no relacionales (No SQL)

Las no relacionales son ejemplos de base de datos que, precisamente, no cumplen con el esquema de entidades y relaciones del modelo relacional, ni tampoco utilizan el formato de tablas para el almacenamiento de datos.

En cambio, este modelo se basa en formatos como clave-valor, mapeo de columnas o grafos para dicho almacenamiento.

Familia de columnas

Las bases de datos orientadas a columnas son un caso particular de la enorme familia de las bases de datos NoSQL. En este tipo de almacenes, en contraposición con el modelo relacional, la información se estructura en columnas en lugar de en filas.

Las comparativas de rendimiento entre los sistemas de gestión por filas (especialmente RDBMS) y los basados en columna, vienen principalmente de la mano de la eficiencia de los accesos que realizan a disco. Los accesos a posiciones consecutivas se producen en una cantidad de tiempo sensiblemente



Sistema de Gestión de historias clínicas de nutrición

Pontificia Universidad Javeriana Cali

Procesos de Ingeniería De Software

menor que los accesos que se producen a posiciones aleatorias a disco. Tener un sistema con multitud de accesos aleator

Ventajas

1. La principal ventaja de este tipo de sistemas es el rápido acceso a los datos: esto ya lo hemos demostrado con el modelo DSM el cual nos permite consultar rápidamente los datos columna a columna, al guardarse físicamente de manera contigua.
2. Un BBMS en una base de datos orientada a columnas, lee solo los valores de columnas necesarios para el procesamiento de una consulta determinada por lo cual las bases de datos orientadas a columnas tienen una mayor eficiencia en entornos de almacenes, donde las consultas, típicas incluyen los agregados realizados por un gran número de elementos de datos
3. Se comprime la información asignable de cada columna con el fin de mejorar el procesamiento desde el ancho de banda del acceso a disco
4. Cambios en el esquema tiene menor impacto y por lo tanto el coste de realizarlos es menor

Desventajas

1. No orientado a transacciones: este es el factor más débil de esta tecnología. El hecho de tener los datos guardados columna a columna nos permite retornar las filas más rápidamente, pero al insertar, actualizar o borrar un registro, se debería hacer en más de una ubicación (al tener que actualizar todos los pares



Sistema de Gestión de historias clínicas de nutrición

Pontificia Universidad Javeriana Cali

Procesos de Ingeniería De Software

clave-valor asociados a una relación). Por esta razón, este tipo de bases de datos no se recomienda para sistemas de tipo OLTP orientados a transacciones y alta concurrencia.

2. Reportes operacionales: también llamados reportes de seguimiento en los que se desea ver toda la información de una relación que puede contener muchas tuplas. En algunos casos esto puede resultar ineficiente comparado con los Row-Stores

3. No existe un modelo de datos que soporte teóricamente este modelo de base de datos

4. No existe un estándar que unifique los criterios de implementación de este modelo de base de datos.

Específicamente para el sistema gestor de historias clínicas de nutrición nos interesa un modelo que nos permita almacenar la información de manera estructurada y bien organizada, ya que tenemos demasiada información y queremos organizarla de la mejor manera posible para así poder brindar de manera natural todas las funcionalidades que el sistema requiere, también nos interesa que siempre se garantice la atomicidad, puesto que la historia clínica es un documento de muchísima importancia y por ningún motivo se puede permitir que los datos se guarden de manera errónea o incompleta. Por todo esto, consideramos que el modelo que mejor se adapta a nuestras necesidades es el modelo de base de datos relacional.

10. Anexos

Archivo	Ruta
---------	------



Pontificia Universidad
JAVERIANA
Cali

Sistema de Gestión de historias clínicas de nutrición

Pontificia Universidad Javeriana Cali

Procesos de Ingeniería De Software

Vista de Casos de Uso	\Entregafinal\CasosDeUso
Vista Lógica	\Entregafinal\VistaLogica
Vista de Procesos	\Entregafinal\VistaProcesos
Vista Física	\Entregafinal\VistaFisica
Vista de Despliegue	\Entregafinal\VistaDespliegue
MER y Modelo relacional	\Entregafinal\DiagramasBaseDatos