

---

# Documento de diseño de software

## Procesos de Ingeniería de Software

Presentado por: Juan Camilo Ariza  
Cardona  
Esteban Cardona Gil  
Jhan Carlos del Rio Muñoz

Pontificia Universidad Javeriana Cali

4 de mayo de 2020

# Índice general

<b>1</b>	<b>Introducción</b>	<b>3</b>
1.1	Propósitos . . . . .	3
1.2	Alcance . . . . .	3
1.3	Definiciones, acrónimos y abreviaciones . . . . .	3
1.4	Referencias . . . . .	4
1.4.1	Descripción de la metodología usada . . . . .	4
1.4.2	Referencias bibliográficas . . . . .	4
<b>2</b>	<b>Representación arquitectónica</b>	<b>6</b>
<b>3</b>	<b>Objetivos y restricciones de la arquitectura</b>	<b>9</b>
<b>4</b>	<b>Vista de casos de uso</b>	<b>10</b>
<b>5</b>	<b>Vista lógica</b>	<b>11</b>
5.1	Diagrama de secuencias . . . . .	11
5.2	Modelo entidad-relación . . . . .	11
5.3	Modelo relacional . . . . .	18
<b>6</b>	<b>Vista de procesos</b>	<b>21</b>
6.1	Diagrama de actividades . . . . .	21
<b>7</b>	<b>Vista física</b>	<b>22</b>

# 1 Introducción

## 1.1. Propósitos

El fin de este documento es mostrar a los diferentes stakeholders el funcionamiento del software en su totalidad, la descripción de cada uno de los alcances y restricciones que poseerá cada rol descrito anteriormente, además de eso se mostrarán y explicarán los diferentes diagramas que ilustran el comportamiento del sistema en sus diferentes ámbitos o áreas que estos abarcan. Un ejemplo de esto puede verse reflejado en el diagrama de la arquitectura de software cuya funcionalidad es indicar la estructura, funcionamiento e interacción entre las partes del software para su correcto funcionamiento y en el diagrama de casos de uso cuyo fin es representar la forma en como un cliente opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan. Así mismo se explicarán los métodos de selección entre los diferentes modelos de los diagramas que se expondrán a continuación. Además se tendrán más ilustraciones en formas de diagramas lógicos, secuenciales, de procesos y de despliegue, todo con la finalidad de que el cliente o las partes interesadas tengan una noción de como funcionara el producto a entregar.

Este documento es fuertemente influenciado por los requisitos funcionales y no funcionales descritos en el documento de especificación de requerimientos de software adjunto en la raíz de la carpeta: «*SRS.pdf*».

## 1.2. Alcance

El diseño del aplicativo DA Clinic implementa todo lo necesario para gestionar las historias clínicas de nutrición dentro del sistema de acuerdo con las especificaciones establecidas por los stakeholders, con el fin de suplir sus necesidades de la mejor manera; Además este permitirá agilizar el tiempo de consultas de un paciente y el nutricionista. Esto será logrado usando el framework de Laravel, haciendo uso de una base de datos la cual permite manipular y almacenar la información de los usuarios de una manera eficiente.

## 1.3. Definiciones, acrónimos y abreviaciones

- **BD:** Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

- **Laravel:** Es un framework que permite desarrollar aplicaciones web de forma sencilla, de una manera expresiva y elegante. está especialmente guiado para desarrollo web con PHP.
- **Framework:** Es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.
- **Stakeholders.** Hace referencia a una persona, organización o empresa que tiene interés en una empresa u organización dada.
- **SRS.** El estándar IEEE 830-1998 para el SRS (en inglés) o ERS (Especificación de requerimientos de software) es un conjunto de recomendaciones para la especificación de los requerimiento o requisitos de software el cual tiene como producto final la documentación de los acuerdos entre el cliente y el grupo de desarrollo para así cumplir con la totalidad de exigencias estipuladas.
- **AWS:** Amazon Web Services es una colección de servicios de computación en la nube pública que en conjunto forman una plataforma de computación en la nube, ofrecidas a través de Internet por *Amazon.com*.

## 1.4. Referencias

### 1.4.1. Descripción de la metodología usada

Para la elaboración del documento de diseño de software del aplicativo web DA Clinic, se utilizaron varios trabajos externos que corresponden a vídeos, material visto en clase y plantillas de latex.

En el caso de las plantillas y el material visto en clase, este aplica globalmente, debido a que esta más orientado a la estructura del documento que al contenido específico de este.

### 1.4.2. Referencias bibliográficas

- [1] J. P. Eisenbarth. 2014. *SRS-Tex*[Online]. Disponible en: <https://bit.ly/2KUA3Qj>
- [2] *Diseño de software*. Notas de clase de 1813 Departamento de Electrónica y Ciencias de la Computación, Pontificia Universidad Javeriana Cali, 2020.
- [3] *Diseño de software II*. Notas de clase de 1813 Departamento de Electrónica y Ciencias de la Computación, Pontificia Universidad Javeriana Cali, 2020.
- [4] R. A. Jaramillo. "Diseño de arquitectura de software para una empresa de tecnología de apoyo a negocios de juegos de azar", Tesis maestría. Pontificia Universidad Javeriana, Cali, 2019.

[5] W. C. Huaman. 2018. *Los 10 patrones comunes de arquitectura de software*[Online]. Disponible en: <https://bit.ly/2xzWhEk>

[6] C. A. Dorantes. 2015. *Arquitectura de una aplicación en Laravel*[Online]. Disponible en: <https://bit.ly/35r3N0E>

[7] A. Garrido. 2017. *Arquitectura del software multicapa*[Online]. Disponible en: <https://bit.ly/35r4kQh>

[8] H. L. Guevara. 2020. *Modelo de Arquitectura de Software C4 — Ejemplo práctico*[Online]. Disponible en: <https://bit.ly/2KZsrfe>

[9] N. A. Landa. 2018. *Diagrama de Despliegue - Tutorial UML en español*[Online]. Disponible en: <https://bit.ly/3aYJpFs>

## 2 Representación arquitectónica

A continuación se muestran los patrones de arquitectura candidatos, de acuerdo con las decisiones tomadas respecto al diseño y desarrollo del aplicativo web DA Clinic usando el framework de Laravel con PHP.

Para esta selección es importante resaltar que para seleccionarlos se tuvo en cuenta cuales podían estar cerca o estaban alineados con los estándares de calidad propuestos.

- **Arquitectura 3 capas** (Características, Ventajas, Desventajas).
- **Arquitectura N-Capas** (Características, Ventajas, Desventajas).
- **Arquitectura Cliente-servidor** (Características, Ventajas, Desventajas).

Para definir la arquitectura que se implementara se hizo una evaluación entre 4 criterios esenciales que debe contemplar la arquitectura a realizar, dichos criterios son:

- Desempeño.
- Flexibilidad.
- Disponibilidad.
- Seguridad.

Dichos criterios se evaluaron del 1 al 5 siendo 1 no cumple y 5 cumple absolutamente. Los resultados obtenidos a continuación:

	<b>Arquitectura 3 Capas</b>	<b>Arquitectura N-Capas</b>	<b>Arquitectura cliente-servidor</b>
Desempeño	5	5	3
Flexibilidad	5	5	
Disponibilidad	5	5	3
Seguridad	3	3	
<b>Total</b>	18	18	

Dado que la arquitectura propuesta de 3 capas obtuvo la misma calificación de la segunda arquitectura propuesta decidimos implementar la de 3 capas, un patrón de arquitectura de 3 capas suple los requerimientos y los estándares de calidad propuestos.

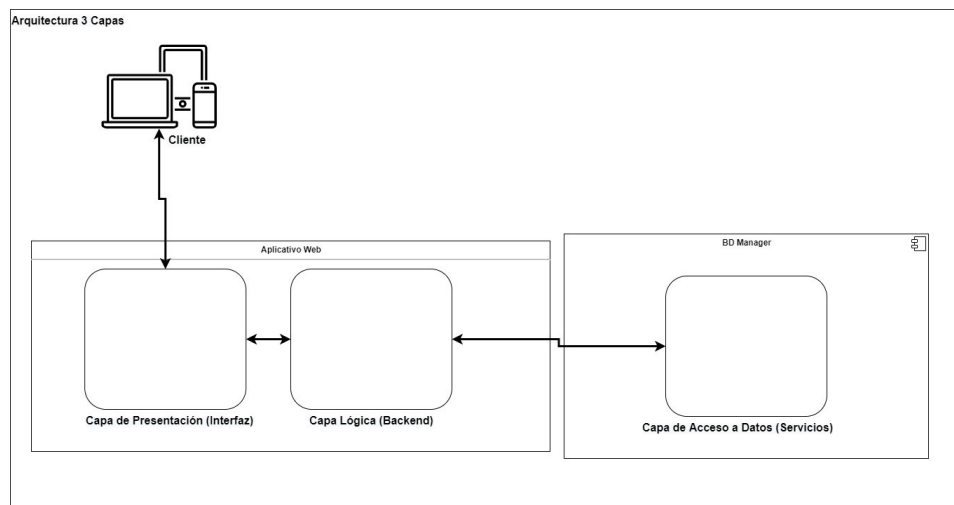


Figura 2.1: **Nivel 0:** Diseño exclusivo para el cliente, debe saber que el aplicativo web implementado se conectara a la BD autorizada para realizar las peticiones de consultas, actualizaciones enviadas desde el aplicativo



Figura 2.2: **Nivel 1:** Zoom del componente Aplicativo Web

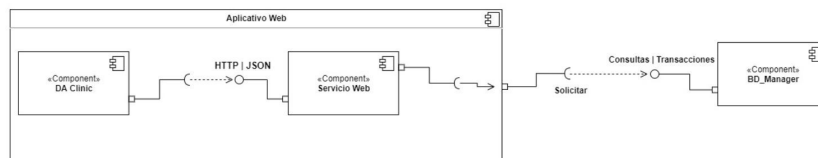


Figura 2.3: **Nivel 2:** Zoom del Aplicativo web detallado en un Diagrama de Paquetes.

A continuación se exponen cada uno de los niveles de nuestro modelo de la arquitectura 3 capas:

En el paquete Interfaz observamos dos paquetes contenidos, estos paquetes representan las ventanas de comunicación que tiene el sistema con el usuario. El primero es el de paquete Login, que permite al usuario hacer su respectivo ingreso o registro (dependiendo si es Administrador, Nutricionista, Paciente registrado o sin registrar). El segundo

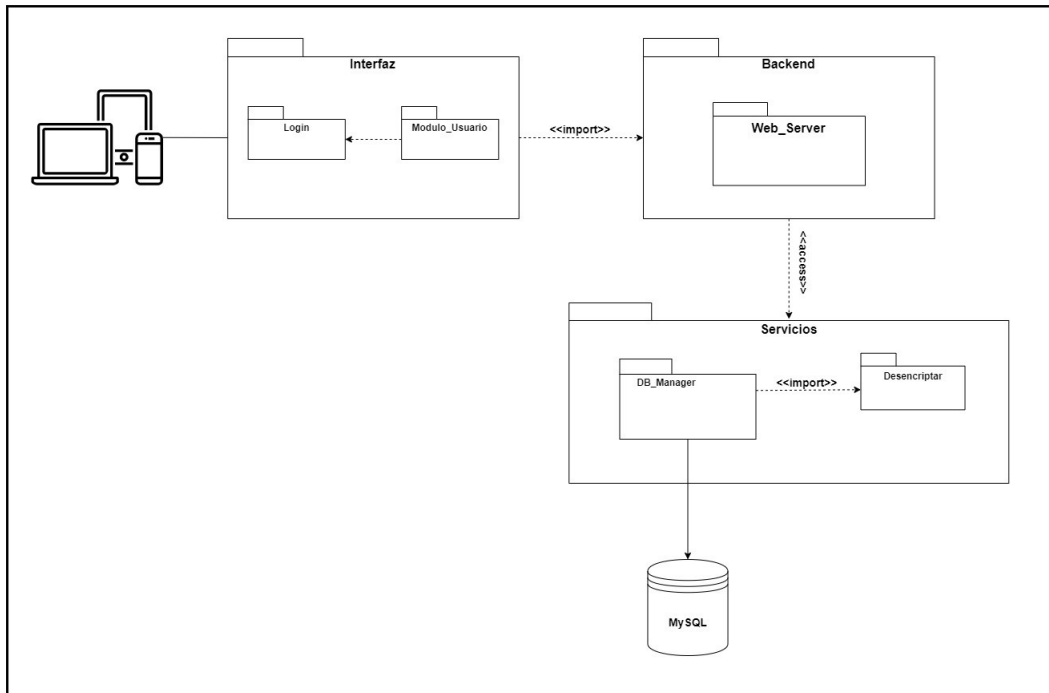


Figura 2.4: **Zoom del Sistema completo detallado en un Diagrama de Paquetes.**

paquete depende del Login y estará disponible después de identificar el usuario; por esta razón hay una dependencia del paquete Login al paquete Modulo\_Usuario. Cabe aclarar que el paquete Modulo\_Usuario interactúa con el usuario de acuerdo con las acciones que tenga permitidas; ya que hay módulos exclusivos para administradores, otros para nutricionistas y otros para pacientes.

Luego, en el paquete Backend, están todos los métodos y funciones que permiten esta interacción del sistema con el usuario en la Interfaz. En este paquete estará la lógica del aplicativo web. Este modelo funciona gracias a que el paquete Backend consume el servicio web que directamente interactúa con la BD. Es por esta razón que el paquete Backend tendrá el paquete Web\_Server contenido.

Este último paquete es la representación del Servicio Web que expone los métodos que posteriormente permite al paquete Backend interactuar con la base de datos. En este paquete es donde se realizan las consultas y transacciones en la Base de Datos.



### **3 Objetivos y restricciones de la arquitectura**

El objetivo de la arquitectura será el de mostrar a los stakeholders como se conectara cada uno de los componentes internos del software a implementar, además la arquitectura está diseñada para que sea lo más confiable en cuanto a la precisión de procesos a implementar y sus resultados exactos, al rendimiento debido a que estará en el mismo servidor ahorrando tiempo de respuesta debido a que no se mantiene conexiones o dependencias con otros servidores, en la usabilidad se adapta a las características de estética y consistencia en las interfaces gráficas, en la usabilidad física ya que el software deberá trabajar sobre cualquier computador que cuente con los requerimientos mínimos descritos anteriormente en el SRS Como restricción es que el sistema se encuentra sujeto a errores geográficos, problemas de disponibilidad debido a que se cuenta con recursos muy limitados porque se cuenta con un solo servidor, escalabilidad por el diseño del sistema está implementado con un solo nodo y solo soporta un pequeño flujo pequeño de usuarios y es limitado por el ancho de banda. Esas son las restricciones y objetivos con las que se cuenta en el sistema en desarrollo con la arquitectura implementada.

## **4 Vista de casos de uso**

## 5 Vista lógica

### 5.1. Diagrama de secuencias

Con este diagrama se busca que los desarrolladores adquieran un entendimiento básico de la manera en la que funcionan algunos de los procesos, con el fin de que puedan saber cómo debe realizarse la implementación y el llamado entre los componentes del sistema; lo anterior se verá descrito mediante un diagrama de secuencia que muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo, este se basa en el diagrama de clases para poder modelar aquello que se va a realizar en el sistema siguiendo intervalos de tiempo.

### 5.2. Modelo entidad-relación

Para el almacenamiento de la información de este proyecto se usará el sistema de gestión de bases de datos MySQL. Se proponen las siguientes entidades para el modelo entidad-relación:

- Clínica.
- Usuario.
- Dirección.
- Paciente.
- Nutricionista.
- Administrador
- Registros.
- Historia clínica general.
- Historia clínica.
- Síntomas.
- Calificaciones.
- Restablecimiento.

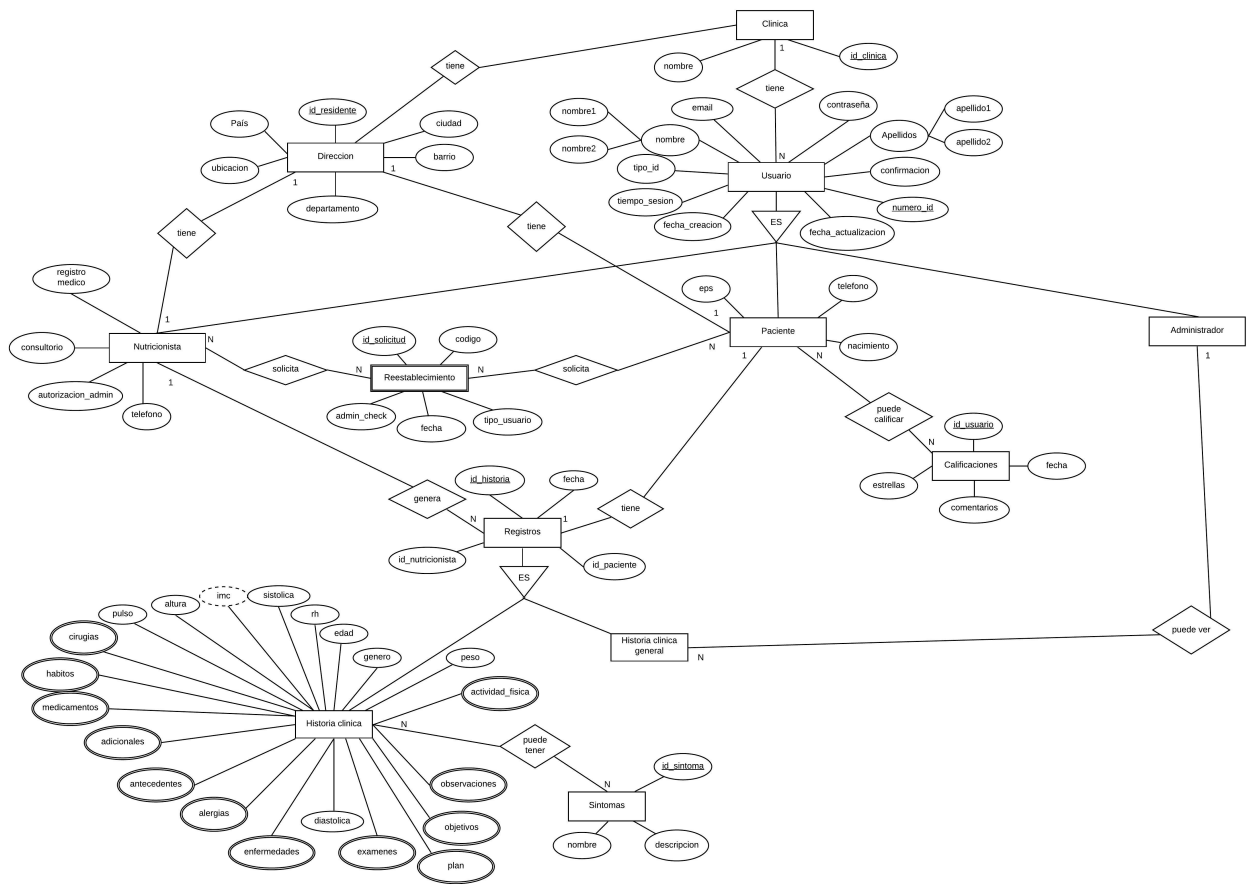


Figura 5.1: Modelo entidad relación de DA Clinic

## Clínica

La entidad «clínica» está considerada en este proyecto para obtener escalabilidad en el diseño del aplicativo web, de tal forma que no sólo se limite a la administración de una sola clínica sino que pueda ser aplicable a distintas clínicas. No debe confundirse con la opción de que cada sede de una misma clínica puede ser considerada como una clínica distinta en el sistema.

Esta entidad cuenta con dos atributos; uno funciona como un identificador único y el otro para distinguir las clínicas a través de un nombre.

Cabe mencionar que el administrador no necesita algún otro campo necesario, los campos correspondientes reflejan el modelo también de la entidad administrador.

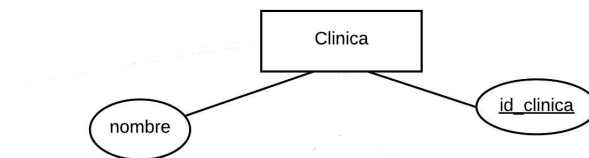


Figura 5.2: Entidad: clinica

## Usuario

Dentro del sistema es necesario considerar distintos usuarios. Es por esto que se propone la creación de una superclase que contenga todos los atributos comunes para el formulario de registro y datos importantes de los pacientes, los nutricionistas y el administrador.

Esta entidad consiste de otros atributos agregados para obtener información que permita garantizar aspectos de seguridad tales como el registro de la actividad del usuario, la confirmación del correo electrónico una vez creada la cuenta y la última vez que el usuario actualizó sus datos. Además, cuenta con un identificador único que corresponde a un documento de identificación, como la cédula, pasaporte o tarjeta de identidad.

## Dirección

Esta entidad es considerada como tal para garantizar una mejor precisión. Además, este diseño garantiza que una sola persona pueda tener sólo una dirección asociada. Esta restricción está considerada por propósitos prácticos.

Los atributos correspondientes a esta entidad son el documento de identificación (Esta puede estar asociada a un establecimiento, a un paciente o a un nutricionista), un país,

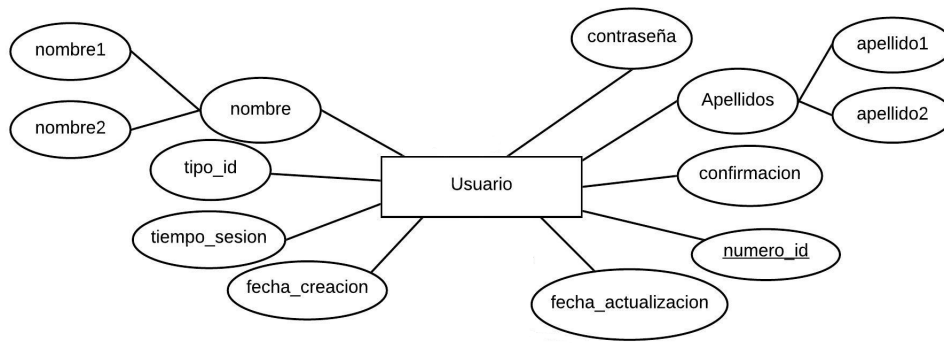


Figura 5.3: Entidad: usuario

un departamento, una ciudad, un barrio y una ubicación. La ubicación representa la típica dirección en la que se especifica a través de una calle, carrera o avenida.

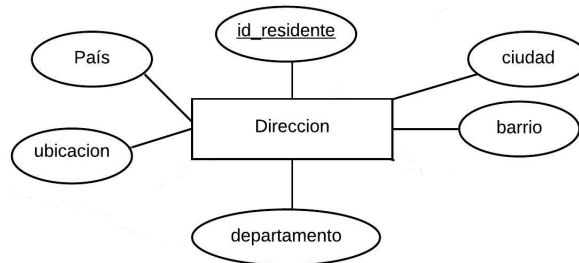


Figura 5.4: Entidad: direccion

## Paciente

La funcionalidad de esta distinción entre otro tipo de usuario y un paciente consiste en el enfoque que trae en la forma que afecta a una historia clínica. Es por esto que esta clase modela las limitaciones y el comportamiento de un paciente dentro del sistema propuesto.

Los atributos que distinguen a un paciente en la lógica de nuestro modelo propuesto son la entidad promotora de salud o eps, un teléfono y la fecha de nacimiento.

## Nutricionista

Dentro de nuestro sistema consideramos que un nutricionista pueda tener acceso a las distintas historias clínicas y pueda modificarlas. Esta es la razón principal de la distinción de esta entidad y de la entidad usuario. Por esto, limitamos el acceso que un nutricionista tiene al modelo respetando la seguridad de los otros usuarios del sistema.

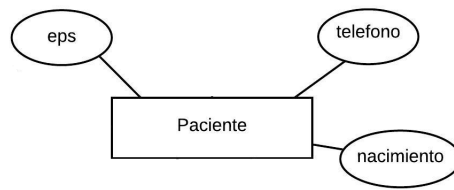


Figura 5.5: Entidad: paciente

Actualmente, esta entidad tiene pensada implementarse con los atributos de un usuario y con un teléfono, un consultorio, un registro médico que rectifique la autenticidad de este como profesional, y un campo booleano que determina si el administrador le permitió el acceso al usuario tener los permisos de un nutricionista. Este permiso tiene la peculiaridad que restringe todo el comportamiento de un nutricionista hasta que tome un valor positivo.

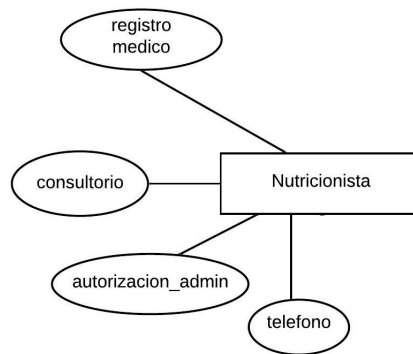


Figura 5.6: Entidad: nutricionista

## Administrador

Como habíamos mencionado anteriormente, no se hace distinción entre los campos de un administrador y un usuario. Sin embargo, el número de identificación de registrado a este se considera especial y le permite obtener los permisos de administrador dentro del sistema.

## Registros

Esta entidad esta creada con el propósito de restringir la información disponible para un nutricionista o un paciente y el administrador. Es por esto que no existe una diferencia entre esta y la entidad «Historia clinica general».

Así pues, esta entidad cuenta con un identificador de la historia clínico que corresponde a una versión de una historia clínica modificada por un nutricionista específico para determinado paciente. Esto con el fin de tener un registro de los cambios hechos en cada actualización de una historia clínica. Además, cuenta con la fecha de la actualización y el número del documento identificación del nutricionista y del paciente.

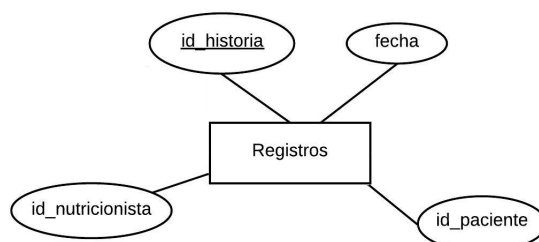


Figura 5.7: Entidad: registros

## Historia clínica

Esta entidad cuenta con mucho más detalle que la entidad registros, pues es la que vamos a usar desde el modelo para que los nutricionistas lo puedan modificar y los pacientes puedan visualizar. Por esta razón, los campos que distinguen a esta entidad de la de registros están relacionados con información sensible del paciente.

Dichos campos son la altura, el peso, la presión, el pulso registrados en la cita que se generó la historia clínica, el tipo de sangre, el género, el índice de masa corporal o IMC, la edad, cirugías, hábitos, medicamentos que usa, antecedentes, alergias, enfermedades padecidas, resultados de exámenes médicos, un plan de seguimiento recomendado para el paciente y las observaciones.

Notese que también cuenta con un atributo llamado «adicionales». Este corresponde a los campos adicionales que los nutricionistas agregaran a la historia clínica, modificaran o eliminaran a lo largo del tiempo.

## Síntomas

La razón por la cual definimos esta entidad como tal y no como un atributo de la historia clínica es porque también queremos usar esta información para el módulo de los síntomas disponible al público general.

Esta entidad cuenta con atributos que garanticen la unicidad indispensable en cualquier base de datos relacional, un nombre y una breve descripción.



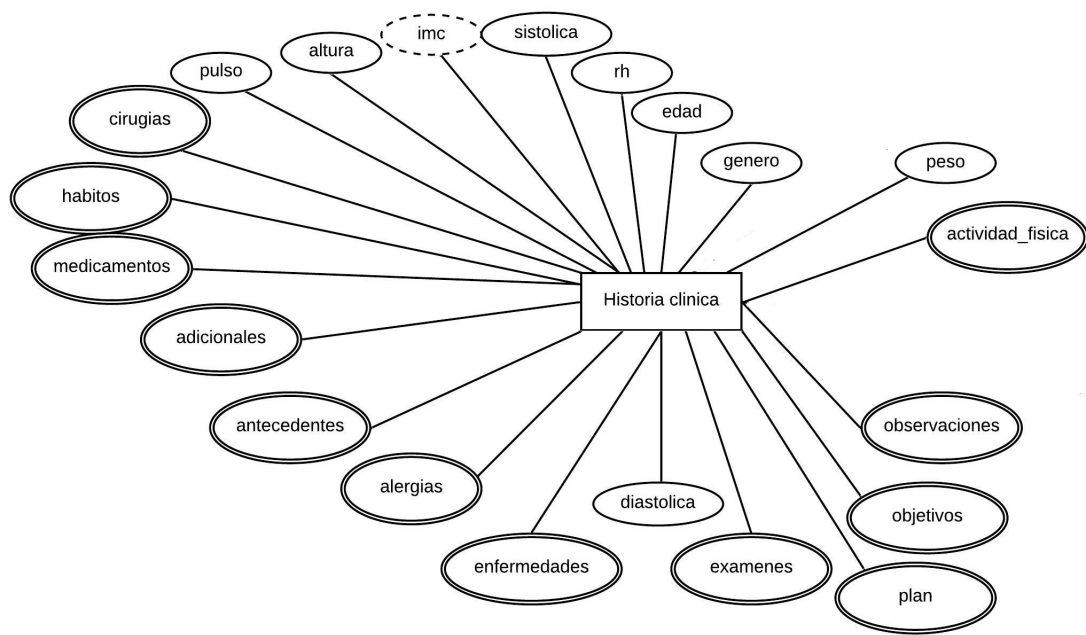


Figura 5.8: Entidad: Historia clínica

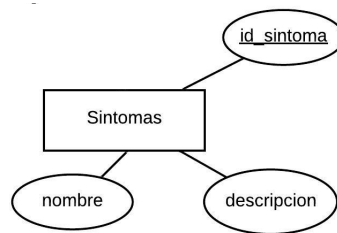


Figura 5.9: Entidad: Síntomas

### Calificaciones

Esta entidad tiene la funcionalidad de permitir almacenar las calificaciones dadas por los pacientes acerca del aplicativo web, su nivel de satisfacción y sus sugerencias.

Cuenta con atributos que permiten guardar varios registros de calificaciones de un paciente, por lo cual cada uno consta de una fecha, una cantidad de estrellas, unos comentarios y un identificador para la calificación.

### Restablecimiento

Dentro de los requisitos elicitados en el documento de especificación de requisitos,

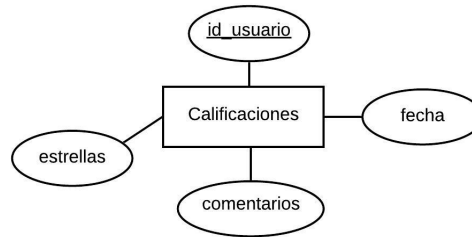


Figura 5.10: Entidad: Calificaciones

contemplábamos funcionalidades que tenían que ver con la posibilidad de que un nutricionista o un paciente pudiera solicitar una contraseña nueva en caso de olvidar la contraseña actual. Esta entidad permite implementar esta funcionalidad a través de atributos que determinen el tipo de usuario que solicita el restablecimiento de la contraseña, una autorización del administrador que sólo tendrá importancia si se trata de un nutricionista, un código que se le enviará al correo electrónico ingresado, y el número de identificación del usuario.

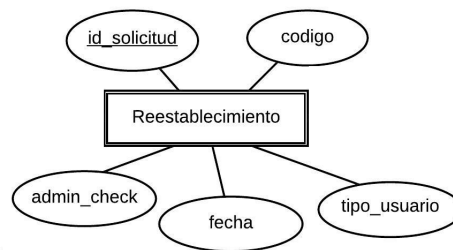


Figura 5.11: Entidad: Restablecimiento

### 5.3. Modelo relacional

Luego de considerar el diagrama entidad-relación, tratamos de convertirlo a un diagrama relacional. Estos fueron los resultados:

A continuación, procederemos a detallar los cambios entre la transición del modelo entidad-relación al modelo relacional:

Las entidades que no tuvieron ningún cambio en su estructura son Direccion, Clinica, Usuario, Paciente, Nutricionista, Administrador, Calificacion, Restablecimiento e Historia\_clinica\_general.



En cuanto a las relaciones, sólo prevalecieron las que en el modelo entidad-relación tuvieran una cardinalidad (1,1) o (1,n).

Para cada uno de los atributos multivaluados de la historia clínica se creara una entidad a parte que tendrá como llave primaria un identificador. Las entidades Alergia, Enfermedad, Habito, Medicamento, Cirugia, Sintomas y Actividad\_física tienen un atributo nombre. Las entidades Observacion, Examen, Plan\_seguimiento, Adicional, Objetivo, Antecedente, Observacion y Sintomas cuentan con un atributo de descripcion. Y la entidad de Examen también posee un atributo donde se esclarece el tipo de examen médico.

Uno de las modificaciones más notables es la creación de nuevas entidades intermedias asociadas a aquellas relaciones con cardinalidad (n,n) y atributos multivaluados. La razón de esta decisión es crear un modelo que no que sólo tenga relaciones de cardinalidad (1,n) o (1,1). Estas entidades son Paciente\_score, Restablecimiento\_paciente, Restablecimiento\_nutricionista, Observacion\_paciente, Alergias\_paciente, Examenes\_paciente, Plan\_paciente, Enfermedad\_paciente, Habitos\_paciente, Medicamentos\_paciente, Objetivos\_paciente, Cirugias\_paciente, Antecedentes\_paciente, Observaciones\_paciente, Sintomas\_paciente, Actividad\_paciente y Adicionales\_paciente.

## **6 Vista de procesos**

### **6.1. Diagrama de actividades**

se espera que el usuario tenga un entendimiento lógico de cómo se describen los procesos que se realizan en el sistema mediante un diagrama de actividades, dichos procesos se llevan a cabo teniendo en cuenta el diagrama de casos de uso descrito anteriormente. Este es considerado como un diagrama de comportamiento porque permite describir lo que debe suceder en el sistema que se está modelando.

## 7 Vista física

En el diagrama de arquitectura física, encontramos que el sistema puede ser accedido desde móvil y computador por medio del web browser, haciendo uso de los protocolos http del motor de búsqueda. El sistema cuenta con distintos recursos dependiendo del usuario autorizado que acceda. La mayoría de los servicios como el Login, funcionalidades del usuario hacen uso del web service que comunica con los servicios del sistema de la clínica para realizar autenticaciones, consultar información pertinente y actualizaciones. Este modelo se representa por medio del Diagrama de Despliegue visto a continuación.

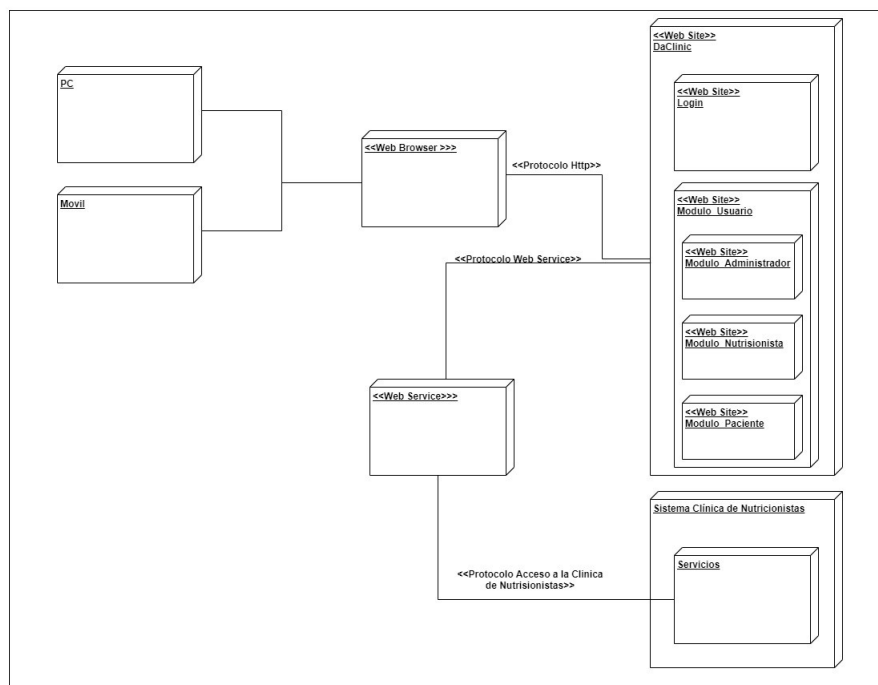


Figura 7.1: Diagrama de despliegue del sistema

En esta sección se quiere que el usuario tenga un entendimiento de la configuración de los elementos de hardware (nodos) y de cómo los elementos y artefactos del software que se trazan en los nodos.