IES NERVION

Dependant Care

1. **Estudio de Viabilidad**
   1. Objetivos
   2. Descripción general del sistema
      1. Diagrama de contexto del sistema
   3. Estudio de la situación actual
   4. Catálogo de requisitos previos
      1. Requisitos funcionales
      2. Requisitos de datos
      3. Requisitos de interfaz
   5. Alternativas de construcción
      1. Descripción de alternativas
      2. Valoración de alternativas propuestas
      3. Estudio de riesgos
   6. Solución propuesta
      1. Justificación
      2. Notas para la planificación
      3. Plan de contingencias
2. **Planificación**
   1. Selección de estrategia de desarrollo
   2. Establecimiento del calendario de hitos y entregas
3. **Análisis de Sistemas de Información**
   1. Definición del sistema
      1. Determinación del alcance del sistema
      2. Identificación de los usuarios participantes y finales
   2. Establecimiento de Requisitos
      1. Obtención de requisitos
      2. Especificación de Casos de Uso
      3. Análisis de requisitos
   3. Análisis de los Casos de Uso
   4. Análisis de Clases
   5. Definición de interfaces de usuario
   6. Análisis de consistencia y especificación de requisitos

1.-Estudio de Viabilidad

# 1.1 Objetivos

Crear una aplicación para realizar un seguimiento de una persona “dependiente”. A la persona que nos referimos la llamamos dependiente, pero no tiene porque ser necesariamente así.

La idea principal es que la aplicación admita anotaciones del seguimiento diario de alguien que toma medicación de forma regular, o que necesita anotar diariamente niveles como el azúcar o la tensión.

Desde dicha aplicación, un usuario puede registrar un perfil de “persona dependiente”, y sobre dicho perfil realizar anotaciones diarias sobre toma de medicación y niveles de azúcar/tensión. Además puede agregar la medicación que se toma, especificando o no el momento del día en que se la toma. Además, dicho usuario podrá agregar citas médicas, detallando todos los datos que puedan interesar.

También dicho usuario, designado como administrador al crear el perfil, tendrá la posibilidad de añadir otros usuarios, a los que se les asignará un rol de usuario normal o administrador.

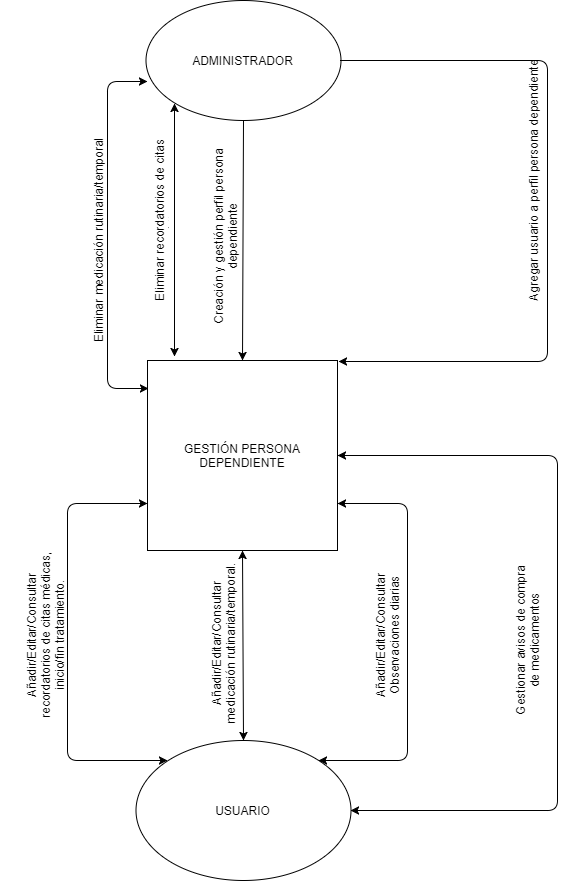
La aplicación debe diferenciar dichos roles cuando un usuario acceda a algún perfil, dando permisos totales a los administradores, y permisos de consulta y de gestión de las anotaciones del seguimiento diario.

# 1.2 Descripción general del sistema

Los requisitos mínimos serían:

* Crear/Gestionar perfil de persona dependiente.
* Gestionar perfiles de usuarios que serán asignados al perfil de la persona dependiente.
* Añadir/Editar/Eliminar/Consultar información sobre la persona dependiente (niveles de azúcar, tensión, estado general)
* Añadir/Editar/Eliminar/Consultar medicación rutinaria/temporal.
* Añadir/Editar/Eliminar/Consultar recordatorios de citas médicas, inicio/fin tratamiento.
* Gestionar avisos de compra de medicamentos, calculando la cantidad de medicamento por caja y la toma diaria.
* Añadir/Editar/Eliminar/Consultar observaciones diarias.

## 1.2.1 Diagrama de contexto del sistema



# 1.3 Descripción de la Situación Actual

En estos momentos no he encontrado ninguna aplicación que ofrezca lo propuesto en el proyecto, aunque si hay algunas que permiten el seguimiento de los niveles de azúcar ([Social Diabetes](https://play.google.com/store/apps/details?id=com.socialdiabetes.android)), otras que permiten hacer seguimiento de la tensión ([Presión Arterial](https://play.google.com/store/apps/details?id=com.szyk.myheart)).

Aparte de los sistemas existentes en plataformas como Google Play, existen sistemas que usan la administración de la sanidad pública que consiste en un programa precargado en una Tablet donde el usuario introduce todo los niveles de la persona dependiente y se envían directamente, pero no se tiene ningún seguimiento por parte de personas fuera de la sanidad pública.

El sistema que vamos a implantar va más enfocado a familiares de la persona dependiente, para mejorar los cuidados a una persona dependiente, donde podremos hacer un seguimiento completo en un entorno entendible por aquellos que usan el sistema, no solo para personal sanitario.

# 1.4 Catálogo de requisitos previos

## 1.4.1 Requisitos funcionales

|  |  |
| --- | --- |
| **Código** | **Descripción** |
| RQF1 | Control de registro de usuarios a la aplicación. |
| RQF2 | Permitir creación de perfiles dependientes por parte de usuario. |
| RQF3 | Control de asignación de usuarios normales a perfil de persona dependiente por parte de usuario administrador. |
| RQF4 | Consulta de información de perfil de persona dependiente. |
| RQF5 | Gestión de medicamentos asociados a un perfil de persona dependiente. |
| RQF6 | Gestión de citas médicas asociadas a un perfil de persona dependiente. |
| RQF7 | Gestión de registros diarios asociados a un perfil de persona dependiente. |

## 1.4.2 Requisitos de datos

|  |  |
| --- | --- |
| **Código** | **Descripción** |
| RQD1 | Almacenar datos de usuarios |
| RQD2 | Almacenar datos de perfiles de personas dependientes |
| RQD3 | Almacenar datos de medicación |
| RQD4 | Almacenar registros diarios (niveles azúcar/tensión, observaciones) |
| RQD5 | Almacenar datos de medicamentos |
| RQD6 | Almacenar dosis de medicamento específicas para un perfil concreto |
| RQD7 | Almacenar citas médicas |

## 1.4.3 Requisitos de interfaz

|  |  |  |  |
| --- | --- | --- | --- |
| **Código** | **Descripción** | **Prioridad** | **Fecha** |
| RQI1 | Mostrar opciones dependiendo de tipo de usuario |  |  |
| RQI2 | Mostrar perfiles asignados a usuario y opción de crearlos |  |  |
| RQI3 | Mostrar propiedades específicas de perfil general de persona dependiente |  |  |
| RQI4 | Mostrar medicamentos almacenados |  |  |
| RQI5 | Mostrar citas médicas |  |  |
| RQI6 | Mostrar registro históricos de perfil concreto |  |  |

# 1.5 Alternativas de construcción

## 1.5.1 Descripción de alternativas

Existen diferentes plataformas donde se pueden desarrollar este sistema:

* Aplicación Android (Dispositivo móvil)
* Aplicación IOS (Dispositivo móvil)
* Aplicación Windows Phone (Dispositivo móvil)
* Universal Windows Plataform (Multiplataforma con Windows 10)
* Aplicación Java (PC)
* Aplicación WPF (PC)

Además de la aplicación en sí, deberemos disponer de un servidor para almacenar los datos de todos los usuarios y perfiles.

También existen diferentes plataformas para desarrollar dicho servidor:

* Web Api de .NET con servidor Azure
* Api Rest en php sobre servidor
* Firebase(Base datos, Auth de usuarios… )

## 1.5.2 Valoración de alternativas propuestas

De las alternativas propuestas, las que no se usarían en dispositivos móviles las descartamos, pues la idea en sí del sistema es que el sistema informe en tiempo real del estado de la persona dependiente.

Por lo tanto, desarrollar una aplicación para dispositivos móviles sería más conveniente, por el hecho de que los usuarios usan más dichos dispositivos.

Basándonos en las gráficas de uso de cada tecnología móvil, sabemos que Android es la tecnología más usada (84,5%), por lo que realizar el proyecto para IOS o UWP/Windows Phone es menos eficaz a la hora de hacer llegar la aplicación a mayor cantidad de usuarios.

## 1.5.3 Estudio de riesgos

Un riesgo que debemos evitar es que los usuarios dejen de usar la aplicación, ya sea porque la interfaz no es adecuada, o por la complejidad de la aplicación misma.

Otro riesgo que debemos correr es el mantenimiento del servidor, ya que si la aplicación no genera ingresos, no sería posible el pago del mismo.

# 1.6 Solución propuesta

## 1.6.1 Justificación

La solución propuesta es desarrollar una aplicación Android que se apoye sobre una API Rest.

Como ya dijimos en el apartado anterior, Android es la plataforma más usada en dispositivos móviles, según un estudio realizado en 2017 🡪 [Link estudio realizado por The verge](https://www.muycomputer.com/2016/05/23/windows-phone-hundido-mercado/)

En dicho estudio se definen los porcentajes de uso:

* Android 84,5 %
* iOS 14,5 %
* Otros 1,0 %

Por lo tanto ofrece mayor “público” una aplicación en Android.

En cuanto de desarrollo del servidor, he optado por una ApiRest en PHP, ya que tengo habilitado un servidor, de momento gratuito, por lo que no se generaría ningún gasto al principio de la vida útil de la aplicación.

## 1.6.2 Notas para la planificación

Planificamos la aplicación para realizar un desarrollo mínimo para que sea “usable”.

## 1.6.3 Plan de contingencias

La planificación mínima debería desarrollarse en el tiempo estipulado.

Asignando un desarrollo modular, es posible realizar una aplicación simple y asequible en cuanto los tiempos de desarrollo se refieren. Además de favorecer la agregación de nuevos módulos a la misma.

2.-Planificacion

# 2.1 Selección de estrategia de desarrollo

Se ha elegido un ciclo de vida en espiral con dos iteraciones.

En la primera iteración deberá de especificarse la mayoría de actividades a realizar, y si es posible, se especificará las actividades que se dejan para la segunda iteración.

# 2.2 Establecimiento del calendario de hitos y entregas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id Tarea | Descripción tarea | Inicio | Fin | Duración |
| 1 | Estudio de viabilidad (EVS) | 09/10/2017 | 18/01/2018 | 101 días |
| 1.1 | Objetivos | 09/10/2017 | 09/10/2017 | 1h |
| 1.2 | Descripción general del sistema | 09/10/2017 | 19/10/1017 | 11 días |
| 1.3 | Descripción de la Situación Actual | 19/10/2017 | 20/10/2017 | 1 día |
| 1.4 | Catálogo de requisitos previos | 20/10/2017 | 14/12/2017 | 55 días |
| 1.5 | Alternativas de construcción | 14/12/2017 | 11/01/2018 | 28 días |
| 1.6 | Solución propuesta | 11/01/2017 | 18/01/2018 | 7 días |
| 2 | Planificación (GPI) | 18/01/2018 | - |  |
| 2.1 | Selección de estrategia de desarrollo | 18/01/2018 | 19/01/2018 | 1 día |
| 2.2 | Establecimiento del calendario de hitos y entregas | 19/01/2018 | 03/02/2018 | 14 días |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id Tarea | Descripción tarea | Inicio | Fin | Duración |
| 3 | Análisis de Sistemas de Información |  |  |  |
| 3.1 | Definición del sistema | 03/02/2018 | 26/02/2018 | 23 días |
| 3.2 | Establecimiento de requisitos | 27/02/2018 | 10/03/2018 | 11 días |
| 3.3 | Análisis de los Casos de Uso | 10/03/2018 | 19/03/2018 | 9 días |
| 3.4 | Análisis de Clases | 20/03/2018 | 25/03/2018 | 5 días |
| 3.5 | Definición de interfaces de usuario | 26/03/2018 | 03/04/2018 | 7 días |
| 3.6 | Análisis de consistencia y especificación de requisitos | 04/04/2018 | 15/04/2018 | 11 días |
| 4 | Diseño de sistemas de la información |  |  |  |
| 4.1 | Diseño de arquitectura | 16/04/2018 | 26/04/2018 | 10 días |
| 4.2 | Diseño de entidades | 26/04/2018 | 28/04/2018 | 2 días |
| 4.3 | Diseño de interfaces | 28/04/2018 | 03/05/2018 | 5 días |

3.- Análisis de Sistemas de Información

# 3.1 Definición del sistema

## 3.1.1 Determinación del alcance del sistema

En un principio, la aplicación debe ser capaz de aceptar el registro de usuarios, y la creación de perfiles de estos mismos, además de agregar, consultar y eliminar toda la información referente a los perfiles detallada anteriormente.

Como desarrollo pensado para la segunda iteración, se desarrollaría un sistema de gestión de avisos de citas médicas, implementando un servicio para notificar cuando la fecha de la cita se acerque, además de que la misma aplicación de la opción de abrir la ubicación de la cita mediante GoogleMaps u otra aplicación similar.

## 3.1.2 Identificación de los usuarios participantes y finales

|  |  |  |
| --- | --- | --- |
| **CODIGO** | **NOMBRE** | **DESCRIPCIÓN** |
| USU0 | Usuario general | Usuario general es aquel que accede a la aplicación pero aun no a accedido a ningún perfil. |
| USU1 | Usuario normal | Usuario general que accede a un perfil de persona dependiente que ha creado otro usuario y con permisos de consulta. Orientado a personal sanitario o personas ajenas al día a día de la persona dependiente. |
| USU2 | Usuario administrador | Usuario general que accede a un perfil de persona dependiente que ha creado el mismo, teniendo control total del perfil.  O que se le ha asignado como administrador a otro perfil.  Tiene privilegios de consulta, creación, edición y eliminación de los perfiles asignados y de los datos relacionados con él. |

# 3.2 Establecimiento de Requisitos

## 3.2.1 Especificación de Casos de Uso

CASO DE USO 1

|  |  |  |
| --- | --- | --- |
| **Caso de uso 1** | Registro de Usuario | |
| **Requisito** | RQF1 | |
| **Actor** | USU0 Usuario general | |
| **Descripción** | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario desea registrarse en nuestra aplicación. | |
| **Precondiciones** | Ninguna | |
| **Postcondiciones** | El USU1 se registrará como usuario de la aplicación | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | El usuario envía petición de registro al sistema |
| 2 | El sistema muestra el formulario de registro con los datos necesarios para registrarse |
| 3 | El usuario rellena el formulario y envía el resultado al sistema |
| 4 | El sistema registra el nuevo usuario |
| **Extensiones o flujos alternativos** | **Paso** | **Acción** |
| 4 | Si los datos proporcionados por el usuario en el paso 3 no son correctos, el sistema vuelve al paso 2 e informa de cual/es dato/s son incorrectos |

CASO DE USO 2

|  |  |  |
| --- | --- | --- |
| **Caso de uso** | Crear perfil de persona dependiente | |
| **Requisito** | RQF2 | |
| **Actor principal** | USU0 Usuario general | |
| **Descripción** | USU0 creará un perfil de persona dependiente | |
| **Precondiciones** | USU0 debe estar registrado en la aplicación | |
| **Postcondiciones** | USU0 creará un perfil de persona dependiente, donde será administrador (USU3) cuando acceda a él. | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | El usuario solicita la creación de perfil de persona dependiente |
| 2 | El sistema muestra el formulario de creación de persona dependiente |
| 3 | El usuario rellena el formulario y lo envía |
| 4 | El sistema registra el perfil |
| **Extensiones o flujos alternativos** | **Paso** | **Acción** |
| 4 | El sistema comprueba los datos, y no son correctos. En este caso se vuelve al paso 2 proporcionando información sobre los errores cometidos |

CASO DE USO 3

|  |  |  |
| --- | --- | --- |
| **Caso de uso** | Asignación de usuario a perfil de persona dependiente. | |
| **Requisito** | RQF3 | |
| **Actor principal** | USU2 Usuario administrador | |
| **Descripción** | El usuario asignará a otro usuario, especificando si normal o administrador, a un perfil de pendiente del que es administrador | |
| **Precondiciones** | El usuario está registrado y es administrador del perfil. | |
| **Postcondiciones** | El perfil tendrá asignado un nuevo usuario | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | USU2 pide acceso a asignación |
|  | 2 | USU2 añade nombre de usuario para búsqueda |
|  | 3 | El sistema consulta la lista de usuarios y realiza la asignación |
| **Extensiones o flujos alternativos** | **Paso** | **Acción** |
| 4 | El sistema no encuentra al usuario especificado, muestra información de que no es correcto el campo dado y no realiza asignación. |

CASO DE USO 4

|  |  |  |
| --- | --- | --- |
| **Caso de uso** | Consulta de información de perfil de persona dependiente. | |
| **Requisito** | RQF4 | |
| **Actor principal** | USU1 Usuario normal | |
| **Descripción** | El usuario accederá a la información de un perfil concreto, incluyendo registros de ese mismo día, registros históricos, citas médicas y medicamentos. | |
| **Precondiciones** | El usuario está registrado | |
| **Postcondiciones** | - | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Usuario selecciona un perfil |
|  | 2 | El sistema muestra el panel de información de un perfil. |

CASO DE USO 5

|  |  |  |
| --- | --- | --- |
| **Caso de uso** | Asignar usuario a perfil | |
| **Requisito** | RQF5 | |
| **Actor principal** | USU2 Usuario administrador | |
| **Descripción** | El usuario asignará al perfil otro usuario. | |
| **Precondiciones** | El usuario está registrado | |
| **Postcondiciones** | El perfil tendrá asignado un nuevo usuario que podrá acceder a él | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Usuario pide acceso a asignación de usuario |
|  | 2 | Usuario introduce Nick de usuario a buscar |
|  | 3 | El sistema comprueba datos y pide confirmación de asignación de usuario a perfil |
| **Extensiones o flujos alternativos** | **Paso** | **Acción** |
| 4 | El sistema comprueba los datos, y no encuentra el Nick de usuario. En este caso se vuelve al paso 2 proporcionando información sobre los errores cometidos |

CASO DE USO 5

|  |  |  |
| --- | --- | --- |
| **Caso de uso** | Gestión de medicamentos | |
| **Requisito** | RQF5 | |
| **Actor principal** | USU2 Usuario administrador | |
| **Descripción** | El usuario tendrá acceso a la gestión de los medicamentos asignados a un perfil de persona dependiente. | |
| **Precondiciones** | El usuario está registrado y es administrador del perfil al que accede | |
| **Postcondiciones** | La información será modificada, ya sea al crear, editar o eliminar un medicamento junto a la dosis asignada. | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Usuario pide acceso de administrador |
|  | 2 | El usuario selecciona un medicamento |
|  | 3 | El usuario elegirá la modificación que desea realizar |

CASO DE USO 6

|  |  |  |
| --- | --- | --- |
| **Caso de uso** | Gestión de citas médicas | |
| **Requisito** | RQF6 | |
| **Actor principal** | USU2 Usuario administrador | |
| **Descripción** | El usuario tendrá acceso a la gestión de las citas médicas asignadas a un perfil de persona dependiente. | |
| **Precondiciones** | El usuario está registrado y es administrador del perfil al que accede | |
| **Postcondiciones** | La información será modificada, ya sea al crear, editar o eliminar una cita médica concreta. | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Usuario pide acceso de administrador |
|  | 2 | El usuario selecciona una cita médica |
|  | 3 | El usuario elegirá la modificación que desea realizar |

CASO DE USO 7

|  |  |  |
| --- | --- | --- |
| **Caso de uso** | Gestión de registros diarios | |
| **Requisito** | RQF5 | |
| **Actor principal** | USU2 Usuario administrador | |
| **Descripción** | El usuario tendrá acceso a la gestión de los registros diarios asignados a un perfil de persona dependiente. | |
| **Precondiciones** | El usuario está registrado y es administrador del perfil al que accede | |
| **Postcondiciones** | La información será modificada, ya sea al crear, editar o eliminar un registro. | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Usuario pide acceso de administrador |
|  | 2 | El usuario selecciona un registro |
|  | 3 | El usuario elegirá la modificación que desea realizar |

# 3.3 Análisis de los Casos de Uso

Las conexiones con la ApiRest las gestionará un controlador, ApiClientManager, que llamará a las especificaciones de la llamadas definidas en una interfaz de servicio.

Los casos de usos serán “conducidos” por las vistas, que se encargaran de realizar llamadas para cualquiera de las funciones CRUD mediante su view model.

# 3.4 Análisis de Clases

## 3.4.1 Identificación de responsabilidades y atributos

Clase: DosisMedicamento

Engloba la relación de dosis y medicamento: nombre de medicamento (nombre comercial), nombre de fármaco, gramaje, dosis y descripción.

Clase: RegistroDiario

Representa a la información insertada por el usuario refiriéndose al estado de salud de la persona dependiente: nivel azúcar, nivel tensión, toma de pastillas y observaciones generales.

Clase: CitaMedica

Representa la información referente a una cita médica: especialista, localización, dirección, planta, consulta, fecha y hora.

Clase: Usuario

Contendrá la información necesaria para el acceso de aplicación y servidor.

Clase: Perfil

Representa a la persona dependiente a la que se le va a realizar el seguimiento. Contendrá información: nombre, apellidos, fecha nacimiento y rol de usuario que accede a él.

# 3.5 Definición de interfaces de usuario

Se especificarán varias interfaces de usuario, partiendo como base la interfaz de ingreso/registro, donde el usuario deberá cumplimentar los datos necesarios.

La siguiente interfaz será una lista de perfiles a los que está asignado el usuario.

Al seleccionar un perfil de la lista, nos llevará a la interfaz de Panel de Perfil, donde mostrará los registros de ese día y las opciones para gestionar registros, citas y medicamentos.

Existirá también una interfaz de creación para todas las entidades, mostrando los datos requeridos.

4.- Diseño de sistemas de la información

# 4.1 Diseño de la arquitectura

## 4.1.1 Descripción general del entorno tecnológico del sistema

El sistema se clasifica como cliente – servidor. Varios clientes (aplicaciones Android) serán capaces de conectarse al servidor donde se guardan los datos.

Este acceso será gestionado por una ApiRest, por lo que el cliente no accede directamente sobre el servidor.





4.1.2 Catálogo de requisitos de diseño

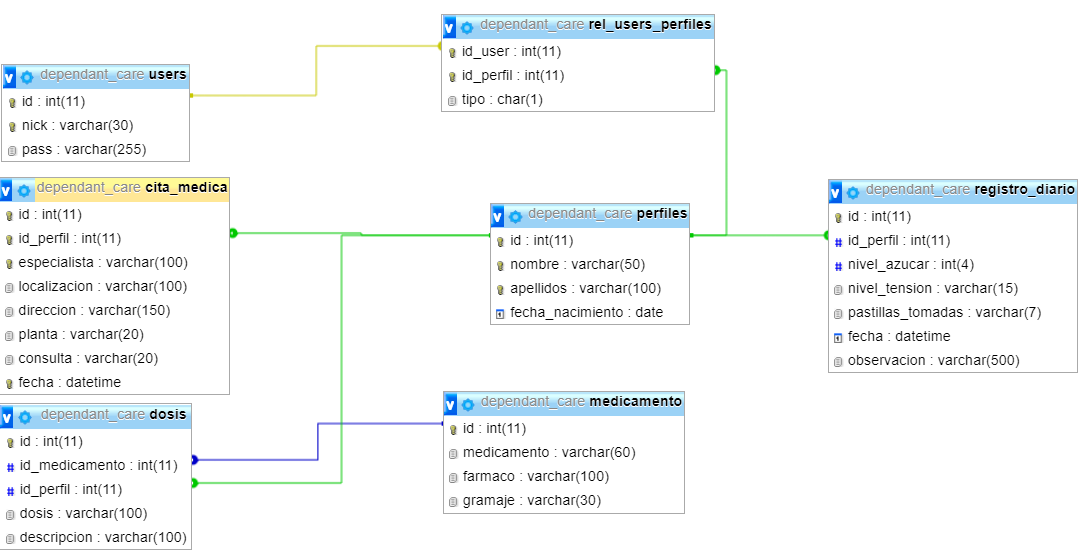
* Requisito de modelo de datos: se usará un modelo de datos relacional, usando para ello una base de datos SQL, cumpliendo con las 3 primeras Formas Normales.

El motor de la base de datos será mySql, y se impondrán privilegios a usuarios, de manera que solo los administradores puedan acceder a la gestión de la base de datos y el resto solo tenga permisos de lectura y escritura.

* Requisitos del servidor: se implementará el servidor en PHP versión 6. Esta implementación deberá de ser capaz de gestionar peticiones REST que vendrán dadas desde las aplicaciones clientes.
* Requisitos de vistas: las vistas deberán solicitar y dirigir la información dada por el usuario además de mostrar las funcionalidades necesarias para el uso de la aplicación.

Deberá, además, mostrar información al usuario de las acciones que puede realizar y el estado de las mismas, mostrando errores si fuera necesario.

## 4.3 Modelo entidad-relación



## 4.4 Diseño de la interfaz

