

Universidad Tecnológica Nacional  
Facultad Regional Santa Fe



Proyecto Final de Carrera  
Ingeniería en Sistemas de Información

**“Desarrollo e implementación de un sistema de integración de la información de pacientes para profesionales que atienden la discapacidad”**

Autores:  
Daniel Solís  
Juan José Barrionuevo Sastre

Director:  
Ing. Lisandro Vrancken

Noviembre 2017

## Contenido

Capítulo 1: Introducción.....	4
1.1 Contexto .....	4
1.2 Delimitación del problema .....	5
1.3 Objetivo general .....	5
1.4 Objetivos específicos .....	5
1.5 Ámbito que abarca.....	6
1.6 Fundamentación.....	6
1.7 Estructura del informe .....	7
Capítulo 2: Proceso de desarrollo.....	8
2.1 Concepto y terminologías .....	8
2.2 Metodología.....	10
2.3 Introducción al desarrollo iterativo e incremental .....	11
2.4 Planificación .....	11
2.4.1 Fase 1: Planificación .....	12
2.4.2 Fase 2: Desarrollo .....	13
2.4.3 Fase 3: Prueba y depuración .....	13
2.5 Herramientas.....	14
Capítulo 3: Análisis y diseño .....	17
3.1 Requerimientos funcionales .....	17
3.2 Requerimientos no funcionales .....	17
3.3 Actores.....	18
3.4 Casos de uso .....	19
3.5 Modelo de Dominio.....	22
3.6 Modelo Entidad-Relación.....	23
Capítulo 4: Funcionalidades.....	24

4.1 Portada .....	24
4.2 Búsqueda de pacientes y profesionales .....	25
4.2.1 Grupos .....	25
4.2.2 Notificaciones .....	26
4.3 Perfil Paciente .....	27
4.3.1 Creación de entradas .....	28
4.3.2 Historial Clínico .....	28
4.3.3 Multimedia .....	29
4.3.4 Tratamientos .....	31
4.3.5 Profesionales .....	32
4.4 Perfil Profesional .....	32
4.5 Administración de mensajes .....	33
Capítulo 5: Arquitectura .....	35
5.1 Supuestos y consideraciones .....	35
5.2 Arquitectura física del sistema .....	35
5.3 Vista lógica de alto nivel .....	37
5.4 Backend .....	37
5.5 Frontend: .....	38
5.6 Patrones de diseño .....	39
5.7 Tecnologías y criterios de diseño .....	41
5.7.1 Capa de presentación .....	41
5.7.2 Capa de negocio .....	41
5.7.3 Capa de acceso a datos .....	41
5.8 Arquitectura tecnológica .....	42
Capítulo 6: Conclusiones .....	43
6.1 Futuro de la aplicación .....	44

Anexos.....	45
Bibliografía.....	46

## Capítulo 1: Introducción

---

“Sistema de Integración de la Información del Paciente para profesionales que atienden la Discapacidad” (SIIPD) es la plataforma web que nace como resultado de este proyecto final de carrera. La misma fue desarrollada con el asesoramiento de un experto en educación especial, el cual se dedica hace más de treinta años a la atención de chicos con capacidades diferentes en distintas escuelas especiales, centros de salud y de forma particular. En su labor de acompañar a los pacientes en la alfabetización y desarrollo, se encontró con la necesidad de compartir información y mantener una comunicación con el resto de los profesionales. Esta necesidad, pobremente satisfecha hasta el momento, dio lugar a la idea de contar con una herramienta informática que atienda la problemática y es justamente la que termina siendo el puntapié inicial para el presente proyecto final de carrera.

Este capítulo presenta el contexto general, análisis del problema, alcances, objetivos y fundamentos del mismo, para finalizar con la estructura de este informe.

### 1.1 Contexto

En la actualidad, los niños y jóvenes con capacidades diferentes son atendidos por profesionales de distintas disciplinas (docentes especiales, terapeutas ocupacionales, psicólogos, fonoaudiólogos y médicos) los cuales en su conjunto conforman el grupo profesional que asiste al paciente, y trabajan en forma colaborativa independientemente de su locación física. Estos, además de brindar el enfoque propio de su profesión, deben contar con una fluida comunicación entre ellos para estar atentos e identificar posibles problemas u oportunidades de mejoras y así llevar adelante un mejor acompañamiento del individuo.

Esta necesidad de comunicación, a lo largo del tiempo, fue resuelta ineficientemente por un cuaderno de hojas el cual acompaña al paciente y en el que el profesional acota notas y observaciones que realizó en su visita. Este elemento, además de resultar poco práctico, tiene disponibilidad y posibilidades limitadas debido a que no siempre está presente para el profesional e incluso el contenido que se puede volcar en el mismo está limitado a texto manuscrito.

Por otro lado, existen reuniones mensuales en las que participan los profesionales que atienden a un paciente en común en donde se acuerda un avance colectivo y coordinado, pero

estas reuniones raras veces se llegan a concretar debido a las complicaciones que les conlleva a cada uno asistir, resultando en una pobre comunicación entre el grupo de profesionales que lo atiende e incurriendo en una ineficiente disponibilidad de información para el tratamiento de los pacientes.

## *1.2 Delimitación del problema*

En base a lo descripto anteriormente se desarrolló un sistema web que mejora la disponibilidad y accesibilidad de la información relacionada a un paciente para con el grupo de profesionales que lo atiende, al mismo tiempo que brinda herramientas de comunicación entre profesionales, con el objetivo de incrementar la eficiencia en la detección de problemas propios de la índole de cada profesión y su posterior tratamiento en conjunto con el resto de los profesionales.

A continuación se describen los objetivos que se plantearon durante la planificación del proyecto.

## *1.3 Objetivo general*

Desarrollar e implementar un sistema que permita la integración y comunicación de la información referente a un paciente entre el grupo de profesionales que lo atienden.

## *1.4 Objetivos específicos*

- Facilitar la gestión de la información relevante para los profesionales a través de una plataforma que permite integrar la misma en una red interactiva de profesionales.
- Permitir a los profesionales coordinar e intercambiar la información referente a pacientes, patologías o tratamientos, manteniendo un registro de todos los pacientes atendidos por un grupo de profesionales.
- Permitir a los profesionales publicar y compartir diferentes formatos multimedia como imágenes y videos.
- Permitir la comunicación en tiempo real de los profesionales pertenecientes a un mismo grupo.

## *1.5 Ámbito que abarca*

El proyecto se llevó adelante en el marco del desarrollo de un sistema a medida guiado en cuanto a sus requerimientos por un experto en el dominio, quien cuenta con bastos años de experiencia en lo que se refiere a educación especial y trabaja en numerosos grupos interdisciplinarios de atención a pacientes con capacidades diferentes. Estos grupos operan de forma local, en un mismo centro de salud o de forma distribuida en distintas locaciones teniendo siempre en común la atención de un mismo paciente. Bajo el contexto descripto hasta el momento, se definió un alcance inicial para SIIPD de operar en un entorno local, pero con la proyección de darle soporte para trabajar en internet, cuestión que no resulta un detalle menor porque afectó de forma directa las decisiones arquitectónicas tomadas para el sistema y que hacen que esté preparado para trabajar en una red abierta.

Finalmente se optó por poner a disposición un conjunto acotado de funcionalidades necesarias para dar soporte al intercambio, almacenamiento e integración de la información con la que trabajan los profesionales en sus sesiones con los pacientes, sumado a los mecanismos de comunicación requeridos para dar soporte al intercambio de mensajes entre los miembros de un grupo.

## *1.6 Fundamentación*

En la actualidad, tanto los profesionales como los grupos de profesionales que se dedican al tratamiento de pacientes con discapacidades no disponen de herramientas informáticas adecuadas para llevar adelante tratamientos de forma coordinada e integrada. Además, tampoco disponen de herramientas que les permitan intercambiar valiosa información recabada en las sesiones y tratamientos que realizan a los pacientes, los cuales son propios de cada profesión. La difusión de dicha información para con el resto de los profesionales de un grupo que atiende a un mismo paciente es de suma importancia y ayuda a la evolución del mismo.

Hoy en día, gracias al gran avance de las Tecnologías de Información y Telecomunicación, se hace posible la comunicación en todo momento de los grupos de profesionales que atienden a pacientes con discapacidad. Sin embargo, al no contar con una herramienta especializada en el almacenamiento, distribución y administración de la información referente a este tipo de actividades, los profesionales no llegan a coordinar de

manera eficiente, dando como resultado tratamientos que no llegan a tener la efectividad esperada.

Llevar a cabo este proyecto nos permitiría entonces hacer frente a las carencias actuales en aquellos aspectos relacionados a todo tipo de tratamientos para personas con discapacidad, los cuales claramente tienen una relación de dependencia unos con los otros. Al mismo tiempo, se facilitaría el tratamiento a pacientes con discapacidades similares, ya que la información estaría disponible en todo momento a modo de biblioteca informática.

### *1.7 Estructura del informe*

El presente informe final está estructurado de la siguiente manera:

Capítulo 2 – Proceso de desarrollo: introduce los conceptos básicos y terminologías del sistema, detalla las herramientas que fueron utilizadas para el desarrollo e indica una descripción sobre la metodología utilizada.

Capítulo 3 - Análisis y diseño: se exponen los requerimientos funcionales y no funcionales relevados, al mismo tiempo que se detallan las características más importantes para el análisis base y diseño del proyecto.

Capítulo 4 - Funcionalidades: se detallan las principales funcionalidades que ofrece el sistema a través de un recorrido por las pantallas de la plataforma.

Capítulo 5 – Arquitectura del sistema: presenta una descripción completa de todos los componentes que hacen al sistema con una breve explicación de sus funcionalidades y fundamentos de su elección.

Capítulo 6 - Conclusiones: presentación de las conclusiones a partir de la experiencia obtenida en el proceso de desarrollo del proyecto, así como las futuras extensiones que pueden realizarse para este proyecto.



## Capítulo 2: Proceso de desarrollo

---

En el presente capítulo se expone una introducción a las características y descripción de los conceptos relacionados con el sistema, para luego realizar un detalle sobre las herramientas utilizadas en el desarrollo y finalmente se expondrá la metodología utilizada con sus fundamentaciones de aplicación.

### 2.1 Concepto y terminologías

A continuación, se detallan los principales conceptos y terminologías que revisten importancia en relación al contexto del dominio del sistema, así como también conceptos propios del proceso de desarrollo utilizado.

- **SIIPD:** Sistema de Integración de la Información del Paciente para profesionales que atienden la Discapacidad.
- **Experto:** Es aquella persona dueña de la idea que da concepción al sistema. Es el referente en cuanto a lo que se espera del sistema y es el encargado de validar y aceptar las diferentes iteraciones en el proyecto.
- **Paciente:** En el contexto de SIIPD es todo individuo que padece determinada patología o discapacidad específica y que ha sido registrado en el sistema como tal.
- **Profesional:** Docente, médico o cualquier otro tipo de profesional que acompaña o atiende al paciente en una determinada disciplina o especialidad.
- **Usuario:** Profesional registrado en SIIPD con permisos para operar dentro del sistema como tal.
- **Usuario administrador:** Encargado de la administración del sistema. Tiene permisos especiales para ingresar a la sección de administración del sistema y habilitar o bloquear usuarios, editar pacientes o profesionales.
- **Grupo:** Profesionales que atienden a un mismo paciente y cuentan con la necesidad de intercambiar información y estar comunicados.
- **Miembro de grupo:** Usuario que forma parte del grupo de profesionales que atiende a un paciente. Tiene permiso para agregar cualquier tipo de entrada y acceder a la información del paciente.

- **Administrador de grupo:** Es un miembro de grupo que además cuenta con permisos para editar el perfil del paciente y administrar los miembros del grupo e información que se carga.
- **Entrada:** Es la información relacionada al paciente que el profesional considera suficientemente relevante para comunicarla al resto de los miembros del grupo. Se carga en el sistema desde el perfil del paciente y el resto de los miembros del grupo pueden verla en dicho perfil o en la pantalla principal del sistema. Además de la entrada que permite cargar información solo como texto plano, existen los siguientes tipos específicos de entrada:
  - **Entrada multimedia:** Es una entrada cuyo contenido es video o imagen e incluye una descripción del contenido en formato de texto.
  - **Historia clínica:** Información relacionada a las patologías del paciente que generalmente es cargada por un profesional especialista en el área correspondiente.
  - **Tratamiento:** Permite llevar el registro de una práctica médica en curso o finalizada, realizada por el paciente a cargo del profesional, registrando fechas de inicio, fin y el resultado de la misma.
- **Api:** La interfaz de programación de aplicaciones (del inglés API: *Application Programming Interface*), es un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizada por otro software como una capa de abstracción.
- **Api REST:** Interfaz entre sistemas que trabaja sobre el protocolo HTTP para obtener y manipular datos, en cualquier formato (XML, HTML, JSON, entre otros). Por lo general se utilizan los diferentes verbos o métodos HTTP para indicar determinadas acciones como son: GET para obtener datos desde el servidor, POST para crear enviar datos al servidor y persistirlos en determinada tabla o DELETE para borrar datos.
- **Repositorio:** Almacenamiento centralizado donde se guardan y mantienen archivos informáticos. Se utilizan entre otros fines, para versionado de archivos y código fuente en proyectos de desarrollo de software.
- **Servicio:** En este contexto se entiende por servicio a toda aquella clase que encapsula la lógica del dominio, con el fin de desentender el resto de los controladores de dicha

lógica y mejorar el desacoplamiento entre módulos. Éste es inyectado en los controladores y servicios donde sea requerido mediante inyección de dependencia.

- **WebSockets:** es una tecnología avanzada que hace posible abrir una sesión de comunicación interactiva entre el navegador del usuario y un servidor. Con esta API, puede enviar mensajes a un servidor y recibir respuestas controladas por eventos sin tener que consultar al servidor para una respuesta.
- **Framework:** entorno de trabajo o marco de trabajo es un conjunto estandarizado de conceptos, buenas prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.
- **ORM:** El Mapeo Objeto-Relacional (Object-Relational Mapping en inglés) es un modelo de programación que consiste en la transformación (“mapeo”) de las tablas de una base de datos relacional, en su correspondiente/s clase/s en un determinado lenguaje de programación. El objetivo es simplificar y generalizar las operaciones de acceso a la base de datos, de modo que el desarrollador no necesite escribir, en la mayoría de los casos, código SQL nativo.
- **Backend:** En el marco del presente proyecto se hace referencia a Backend cuando nos referimos al sistema Symfony2 que se ejecuta en el servidor web. Es la parte del sistema que implementa la mayor parte de la lógica de negocio y acceso a datos.
- **Frontend:** Parte del sistema que se ejecuta completamente en el navegador del usuario final o que está directamente relacionada a la interfaz de usuario. En SIIPD el Frontend consiste en un proyecto AngularJS que interactúa con el Backend a través de la API Rest que este último expone.
- **JSON:** Acrónimo de JavaScript Object Notation, es un formato de texto ligero para el intercambio de datos. Es un subconjunto de la notación literal de objetos JavaScript.

## 2.2 Metodología

Para el desarrollo del proyecto se siguió una metodología de desarrollo de software iterativo e incremental. Esta elección se fundamentó en las características propias del proyecto como son la disponibilidad total del cliente y su compromiso en el mismo, como así también la necesidad de tener prototipos funcionales para su validación y de esta forma reducir los

posibles riesgos, al mismo tiempo que administramos las expectativas del usuario final y llevamos acciones para corregir los posibles desfasajes entre iteraciones.

### *2.3 Introducción al desarrollo iterativo e incremental*

El desarrollo iterativo e incremental es una metodología de desarrollo de software que se basa en la construcción de distintos bloques temporales que se le denominan iteración. En una iteración se repite un determinado proceso de trabajo que brinda un resultado más completo para un producto final, de forma que quien lo utilice reciba beneficios de este proyecto de manera creciente.

Para lograr esto, cada requerimiento debe tener un completo desarrollo en una única iteración que debe de incluir pruebas y documentación para que el equipo pueda cumplir con todos los objetivos definidos y así estar listo para ser entregado al cliente. Con este curso de acciones se mitigan posibles riesgos que atenten contra el cumplimiento del cronograma del proyecto.

Lo que se busca es que en cada iteración los componentes logren evolucionar el producto dependiendo de los componentes completados en las iteraciones antecesoras, agregando más opciones de requisitos y logrando así un mejoramiento mucho más completo.

Una manera primordial para dirigir al proceso iterativo incremental es la de priorizar los objetivos y requerimientos en función del valor que ofrece el cliente.

### *2.4 Planificación*

Antes de comenzar con el diseño e implementación del sistema, se realizó una fase inicial de investigación preliminar. El objetivo de dicha fase fue la de evacuar la mayor cantidad de dudas referidas a los requerimientos del sistema y poder contar desde un principio con un relevamiento exhaustivo de los mismos. Además se llevó adelante un análisis en donde se generaron los diagramas de diseño requeridos a los fines de documentar el relevamiento y análisis. Asimismo, el alcance del proyecto se distribuyó en etapas claramente diferenciadas para poder realizar una aplicación efectiva de la metodología y seguimiento del proyecto acorde al contexto que supone un proyecto final de carrera.

Para el desarrollo del proyecto se aplicó una metodología incremental de cuatro etapas de ocho semanas de duración. Cada una de estas etapas comparte un formato similar entre ellas a excepción de la primera en donde se destina más tiempo al análisis inicial del dominio, identificación de casos de uso, definición de alcance, preparación del entorno y planificación global del proyecto entre otras cuestiones (ver Anexo C para una lista más detallada de las actividades). Las etapas se organizaron en tres fases de la siguiente forma:

#### 2.4.1 Fase 1: Planificación

En esta primera fase se llevó adelante una revisión del análisis y diseño, además de generar la planificación para el resto del sprint y ajustar los cambios necesarios en el diseño en base a iteraciones anteriores. Entre las actividades incluidas se destacan:

- Se llevaron adelante las reuniones con el experto en dominio.
- Se validó el desarrollo de la etapa anterior.
- Se ajustaron las planificaciones y prioridades.
- Se profundizó en temas referentes al dominio que se iban a estar desarrollando en la presente etapa.
- Se desarrollaron los documentos y diagramas de diseño.
- Se revisó el conjunto de casos de uso a ser desarrollados.
- Se tomaron decisiones referentes al diseño de la arquitectura del sistema.
- Se realizó la división y asignación de tareas entre los miembros del equipo para lo que se desarrollaría en la etapa actual.

A excepción de la primera etapa, donde algunas actividades se extendieron hasta la cuarta semana por tratarse del inicio del proyecto y necesitarse una mayor dedicación al relevamiento de requerimientos, en el resto de las etapas de esta fase se tuvo una dedicación exclusiva de una semana. Además, mencionar que para la etapa 1, si bien se realizaron las actividades antes citadas, estas no correspondían a modificaciones o adaptaciones de etapas anteriores sino más bien a primeras instancias que dieron como producto las versiones iniciales de las planificaciones y proyecciones de actividades.

Dentro de las actividades que conforman esta primera fase se destaca el relevamiento de requerimientos en conjunto con el experto en dominio, la especificación de los casos de uso, el diseño de las plantillas de casos de uso, test de aceptación y test funcionales, elaboración del

cronograma de actividades, plan de gestión de la configuración y reuniones periódicas de trabajo.

Finamente mencionar que todas las actividades listadas para esta fase se encuentran documentadas en los anexos correspondientes y en las subsiguientes secciones del presente informe.

#### 2.4.2 Fase 2: Desarrollo

En esta fase se procede a desarrollar lo planificado y se le dio una dedicación de cuatro semanas en las que se destacan las siguientes actividades:

- Implementación de entidades lógicas y repositorios.
- Implementación de controladores y servicios.
- Implementación de pantallas y componentes de la interfaz gráfica.

A su vez, el desarrollo de estas tareas se llevó delante de forma conjunta y paralela, guiada por la aplicación de la arquitectura elegida (ver Capítulo 4: Arquitectura).

Por otro lado, luego de asignar las actividades a realizar en la presente fase, se organizaron las tareas de estas actividades para ser desarrolladas de forma remota o presencial, apoyándonos en el uso de herramientas de comunicación como Skype para compartir escritorios o mantener llamadas, mientras que la gestión de la configuración del código fuente se llevó adelante mediante la plataforma Bitbucket donde se creó un proyecto GIT privado para versionar el código y gestionar los incidentes. Finalmente, toda documentación y recursos en formato de texto, presentaciones y planillas de cálculo fueron almacenadas en una cuenta de Google Drive, la cual nos permite versionar estos recursos a la vez que habilita su edición en forma remota y simultánea.

#### 2.4.3 Fase 3: Prueba y depuración

En este estadio de la etapa se procede a probar y depurar lo desarrollado en la fase actual con el objetivo de validar técnica y funcionalmente los requerimientos implementados. Es así que se confeccionan las pruebas funcionales, mientras que en la medida que avanzan las etapas se realizan pruebas de integración para garantizar el estado funcional de la aplicación en forma global. Finalmente se mantienen reuniones con el cliente a fines de validar lo implementado.

De esta forma, a partir de la aplicación de la metodología, se obtuvo al final de cada etapa una versión funcionalmente verificada y validada. Además, el aporte de la retroalimentación del cliente al final de cada etapa, permitió solventar los problemas relacionados al dominio del negocio y así poder hacer una mejor administración de los riesgos del proyecto, siendo éste uno de los puntos fuertes de la metodología que nos facilitó cumplir con los objetivos y tiempos del proyecto.

Las pruebas funcionales realizadas en esta fase fueron guiadas por una serie de planillas que fueron definidas en conjunto con el experto en el dominio, las cuales especificaban los casos de aceptación esperados para determinados casos de uso. Estas plantillas y casos de aceptación se elaboraron en la fase uno de la etapa inicial del proyecto, pero luego fueron evolucionando como resultado de las sucesivas reuniones y etapas de la metodología.

Por otro lado, se utilizó la herramienta Jira provista por Bitbucket para facilitar el seguimiento de los errores surgidos de las pruebas realizadas por el equipo de desarrollo durante la fase 2 y 3. Esta herramienta funciona como una especie de pizarra virtual donde un miembro del proyecto puede cargar incidencias y reportarlas a cualquier miembro del proyecto, además de hacer un seguimiento del estado de las mismas e incluir o requerir información adicional.

## 2.5 Herramientas

En esa sección se presenta una breve descripción de las herramientas utilizadas durante el desarrollo del proyecto, así como también sus ventajas y principales características.



**PhpStorm:** Entorno Integrado de Desarrollo para aplicaciones web PHP comercializado por JetBrains. Posee soporte para diferentes plugins gratuitos provistos por la empresa y desarrolladores particulares, que agregan soporte a numerosos frameworks y librerías como Symfony2 o Doctrine.



**Git:** Sistema de control de versiones distribuido utilizado para versionar el código fuente y documentación de proyectos de software.

---



**Bitbucket:** ofrece un repositorio Git privado y gratuito para proyectos de hasta cinco miembros.



**Composer:** Software utilizado para la administración de librerías y paquetes en proyectos PHP. Se configura utilizando archivos JSON y se apoya en proyectos almacenados en [packagist.org](http://packagist.org).



**Bower:** Software utilizado para la administración de librerías y paquetes en proyectos JavaScript. La mayoría de los proyectos están disponibles a través de GitHub.



**Visual Paradigm:** Software para creación de diferentes diagramas UML. Se vende bajo licencia, pero existe una versión gratuita disponible.



**Balsamiq Mockups:** Herramienta de maquetado que permite representar una aproximación de las pantallas de un proyecto de software, de manera sencilla facilitando el relevamiento de requerimientos funcionales.



**JIRA:** Sistema que permite gestionar eficientemente diferente tipos de tareas en un proyecto de software, como por ejemplo el reporte de errores y seguimiento de los mismos hasta su resolución. Bitbucket ofrece JIRA junto con sus repositorios GIT.



**Google Docs:** Herramientas de oficina orientadas al uso colaborativo en línea, utilizando el almacenamiento provisto por Google Drive. Provee un editor de textos, planillas de cálculos y herramientas para presentaciones entre otros.





**Google Drive:** Sistema de almacenamiento en la nube, que entre tantas funcionalidades permite compartir y colaborar en la redacción de documentos entre múltiples usuarios de Google en tiempo real.



**Skype:** Software de mensajería instantánea que permite chat, llamadas de voz, video llamadas, compartir escritorios remotos y transferencia de archivos.

---

## Capítulo 3: Análisis y diseño

---

El siguiente capítulo detalla los requerimientos funcionales y no funcionales relevados así como las características más importantes necesarias para el análisis y diseño del sistema.

### 3.1 *Requerimientos funcionales*

En esta sección se resumen los requerimientos funcionales que guiaron la toma de decisiones al momento de diseñar la plataforma.

Funcionalidades exclusivas para usuarios profesionales:

- Registro y administración de pacientes, así como también autorizar a otros profesionales a trabajar en el mismo grupo.
- Gestor de permisos para miembros del grupo.
- Carga y edición de información de perfil personal y profesional.
- Publicación y edición de nuevas entradas de texto, imágenes y video en los grupos a los que el usuario está vinculado.
- Carga de historial clínico y Tratamientos en los grupos donde el usuario tiene acceso.
- Comentar en entradas, historia clínica y tratamientos en los grupos donde el usuario tiene acceso.
- Envío de mensajes a otro profesional dentro de la plataforma.
- Sistema de notificaciones para solicitud de ingresos a grupos, aprobaciones de ingresos a grupo o invitaciones a grupos.

Funcionalidades para usuarios administradores:

- Gestión de los usuarios profesionales y grupos de profesionales de la plataforma.
- Autorizar las altas de registraciones de usuarios para operar en el sistema.

### 3.2 *Requerimientos no funcionales*

Los requerimientos no funcionales, son aquellos requerimientos que describen los aspectos del sistema que son apreciables por el usuario, pero que no incluyen una relación

directa con la funcionalidad del sistema. Hacen referencia, en general, a los atributos de calidad del mismo.

Los requerimientos no funcionales identificados se listan a continuación:

- **Usabilidad:** A raíz de la escasa experticia del usuario en el uso de sistemas informáticos, se realizó especial hincapié en la usabilidad del sistema, tanto en sus interfaces graficas como en los mecanismos que brindan las funcionalidades. Para esto se respetaron los estándares en lo que refiere a interfaces gráficas, al mismo tiempo que se diseñaron los mecanismos de interacción para que fuesen lo más parecido posible a las plataformas habituales a los usuarios.
- **Seguridad:** Entendiendo que gran parte de la información que se administra es de carácter privado, la seguridad es otro foco en el desarrollo del proyecto. Para cubrir esta necesidad se utilizó el módulo de seguridad provisto por el framework Symfony2, el cual es ampliamente usado y cuenta con una comunidad que le brinda soporte constantemente.
- **Extensibilidad:** Una de las premisas del proyecto es partir de un sistema funcional pero con la expectativa de ir ampliando sus funcionalidades en la medida que los usuarios lo vayan requiriendo. A razón de esto es que desde la concepción del diseño se tuvo en cuenta este apartado, proyectando su arquitectura en módulos bien definidos con el objetivo de facilitar su extensibilidad.

### 3.3 Actores

El sistema admite dos tipos de usuarios, en primera instancia los profesionales que integran los grupos que atienden a los niños con discapacidad. Si bien estos usuarios pertenecen a distintas ramas profesionales, todos comparten un factor en común, la falta de experiencia en el uso de sistemas informáticos complejos. Este detalle cobra especial importancia cuando se entiende que son los propios usuarios los que van a nutrir de información al sistema, por lo que la amigabilidad y usabilidad fue puesta como foco en etapas tempranas del análisis. Teniendo esto en cuenta, se diseñó la aplicación para que fuese lo más familiar al usuario promedio, no solo respetando los estándares de diseño de pantallas sino también los mecanismos que se utilizan para dar funcionalidad al sistema, procurando que estos sean lo más parecido a los empleados por las plataformas sociales que se utilizan en la actualidad.

El otro tipo de usuario son los administradores de la plataforma, estos solo disponen de acceso a un panel de administración con funcionalidades acotadas a la gestión de usuarios y pacientes.

### 3.4 Casos de uso

A partir de los requerimientos relevados se especificaron los siguientes casos de uso con el objetivo de cubrir las necesidades funcionales identificadas:

- **CU001 – Registro de Profesionales:** El sistema permite a un profesional, no registrado en el sistema, hacerlo completando un mínimo de datos requeridos.
- **CU002 – Login:** El sistema permite la identificación del usuario por medio de un nombre de usuario y clave, para autorizar su ingreso al sistema. Además se audita el tiempo y el host desde el que se accede.
- **CU003 – Logout:** Egreso de un usuario autenticado del sistema y liberación de cualquier recurso asociado al mismo.
- **CU004 – Restablecer Contraseña:** Permite que un usuario que ha olvidado su contraseña, pueda restablecerla por medio de su cuenta de email registrada en el CU001.
- **CU005 – Editar Perfil de Profesional:** El sistema muestra los datos del profesional y su cuenta de usuario permitiendo la edición de los mismos.
- **CU007 – Ver Perfil Paciente:** El sistema muestra la pantalla del perfil de un paciente en la que se muestra el menú del grupo en caso que el profesional autenticado sea miembro o administrador y un menú por defecto en el caso que no lo sea.
- **CU008 – Ver Perfil Profesional:** El sistema muestra la pantalla del perfil de un profesional, en la que se muestra un menú con diferentes acciones.
- **CU009 – Portada:** El sistema muestra la pantalla principal, la cual se compone de la sección principal con el resumen de la últimas entradas de todos los grupos del profesional, un menú de navegación a la izquierda y una barra de notificaciones en la parte superior.
- **CU010 – Crear Editar Borrar Entrada:** Permitir al usuario crear una entrada, la cual puede ser de diferentes tipos y se encuentra asociada a un único paciente. También se permite la administración de las entradas creadas por el profesional autenticado.

- **CU11 – Crear Editar Borrar Comentario Entrada:** Permite al usuario crear un comentario sobre una entrada propia o sobre la de un autor con el cual comparte grupo. También se permite la edición y eliminación de los comentarios creados por el profesional autenticado.
- **CU012 – Crear Editar Borrar Historial Clínico Paciente:** Permite al usuario crear una entrada de tipo Historial Clínico que se encuentra asociada a un único paciente. También se permite la edición y eliminación de los Historiales clínicos creados por el profesional autenticado.
- **CU013 – Crear Editar Borrar Tratamiento Paciente:** Permite al usuario crear una entrada de tipo Tratamiento que se encuentra asociada a un único paciente. También se permite la edición y eliminación de los Tratamientos creados por el profesional autenticado.
- **CU014 – Ver Grupo de Profesionales del Paciente:** Habilita a cualquier profesional a consultar el grupo de profesionales que se encuentran actualmente vinculados al paciente, es decir, listar todos los profesionales que están tratando actualmente a un paciente determinado.
- **CU015 – Registrar Paciente:** Permite la creación de un paciente que no se encuentra cargado en el sistema.
- **CU016 – Editar Paciente:** Permite la edición de un paciente que fue registrado en el sistema previamente, por el usuario que pretende editarlo, siempre y cuando el mismo sea administrador del grupo.
- **CU017 – Asignar Profesional a Paciente:** Permite vincular a un paciente con un profesional. Se agrega al profesional al grupo de profesionales del paciente, pudiendo un profesional pertenecer a más de un grupo.
- **CU018 – Eliminar Profesional Vinculado a Paciente:** Permite al administrador del grupo eliminar a cualquier profesional vinculado al paciente. Se mantendrán todas las entradas, comentarios, historial clínico y tratamientos que el profesional haya registrado aunque éste ya no tendrá acceso.
- **CU019 – Administrar Grupos del Profesional:** Permite al profesional gestionar los grupos a los que pertenece.
- **CU020 – Solicitar Ingreso a Grupo:** Habilita a cualquier profesional a solicitar al administrador del paciente ingresar al grupo relacionado al mismo.

- **CU021 – Sugerir Ingreso a Grupo:** Permite que cualquier profesional que pertenece a un grupo, solicitar el ingreso de otro profesional.
- **CU022 – Abandonar Grupo:** Permite que un profesional abandone un grupo.
- **CU023 – Enviar Mensaje a Profesional:** Permite al usuario el envío de mensajes en tiempo diferido a cualquier profesional registrado en el sistema.
- **CU024 – Administrar Mensajes:** Permite que un profesional administre todos los mensajes recibidos y enviados a otro/s profesional/es.
- **CU025 – CRUD Profesionales:** Funcionalidad accesible solo por administradores del sistema que permite administrar los profesionales del sistema.
- **CU026 – CRUD Pacientes:** Funcionalidad accesible solo por administradores del sistema la cual permite administrar los pacientes del sistema.

A continuación se presenta el diagrama de casos de uso.

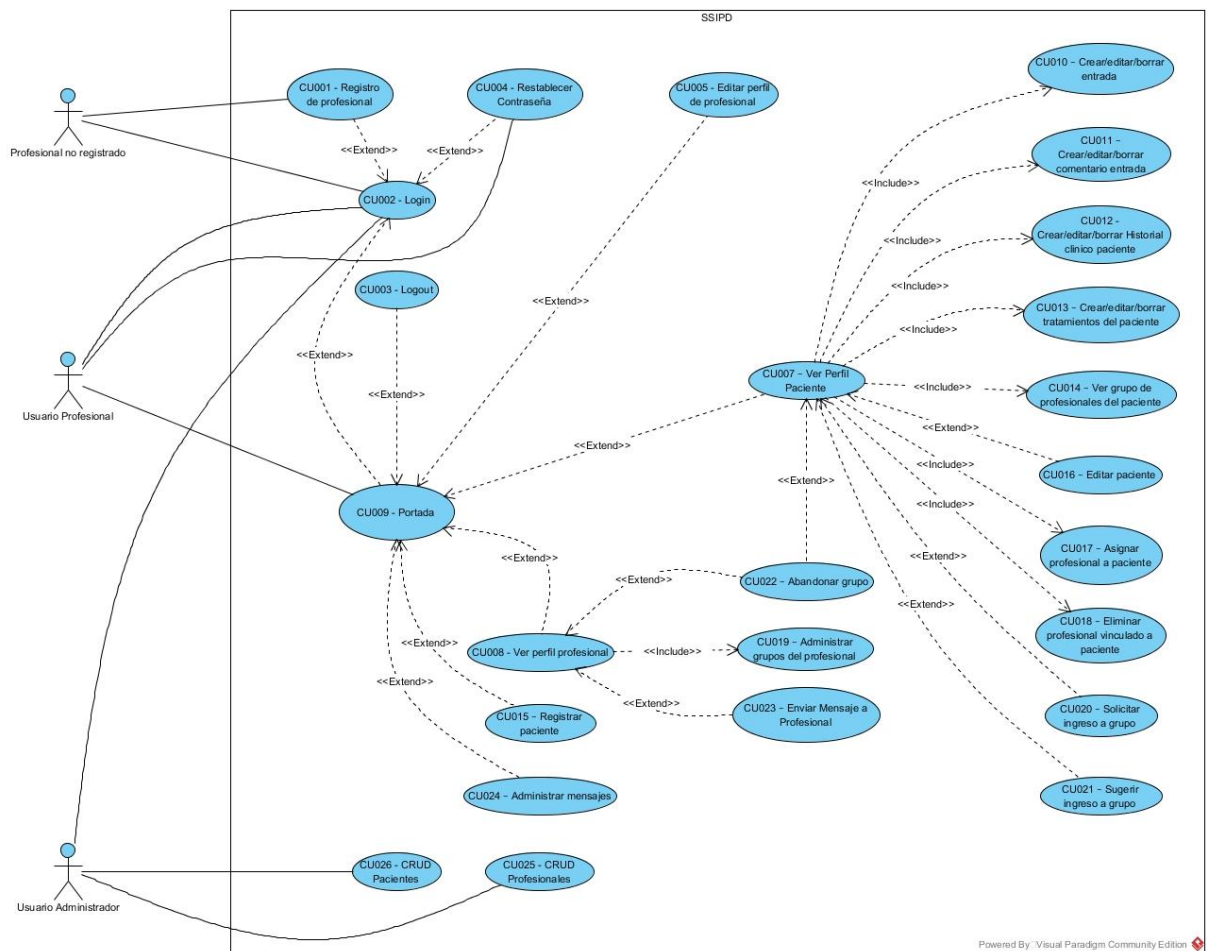
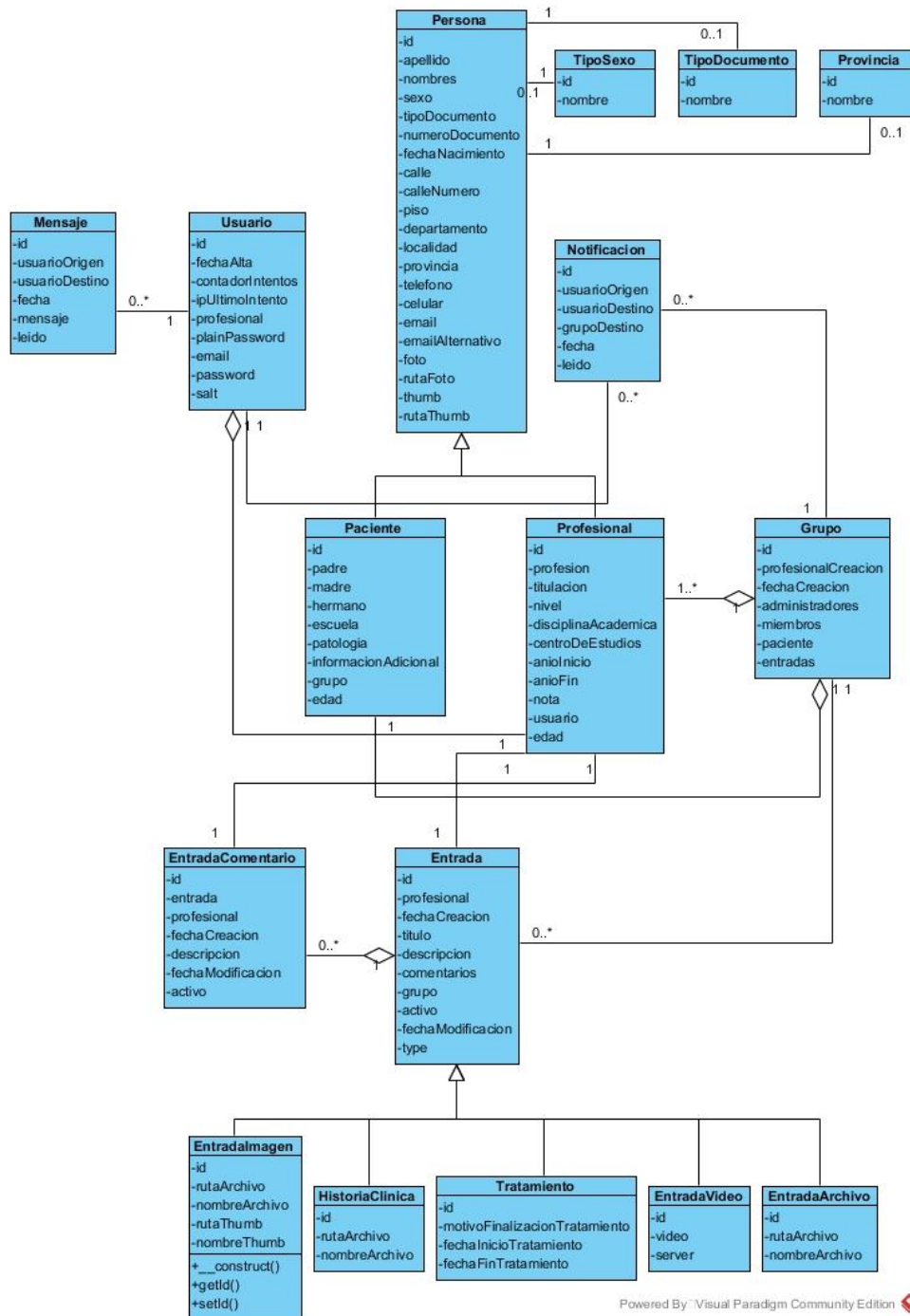


Fig. 1: Diagrama de casos de uso

El detalle de cada caso de uso se puede encontrar en el **Anexo E: Casos de uso**.

### 3.5 Modelo de Dominio

A continuación se presenta el diagrama de clases que representa el dominio de la aplicación.



Powered By Visual Paradigm Community Edition

Fig. 2: Diagrama de clases

### 3.6 Modelo Entidad-Relación

En la presente sección se presenta el diagrama E-R, el cual refleja las entidades conceptuales más importantes dentro del dominio del sistema.

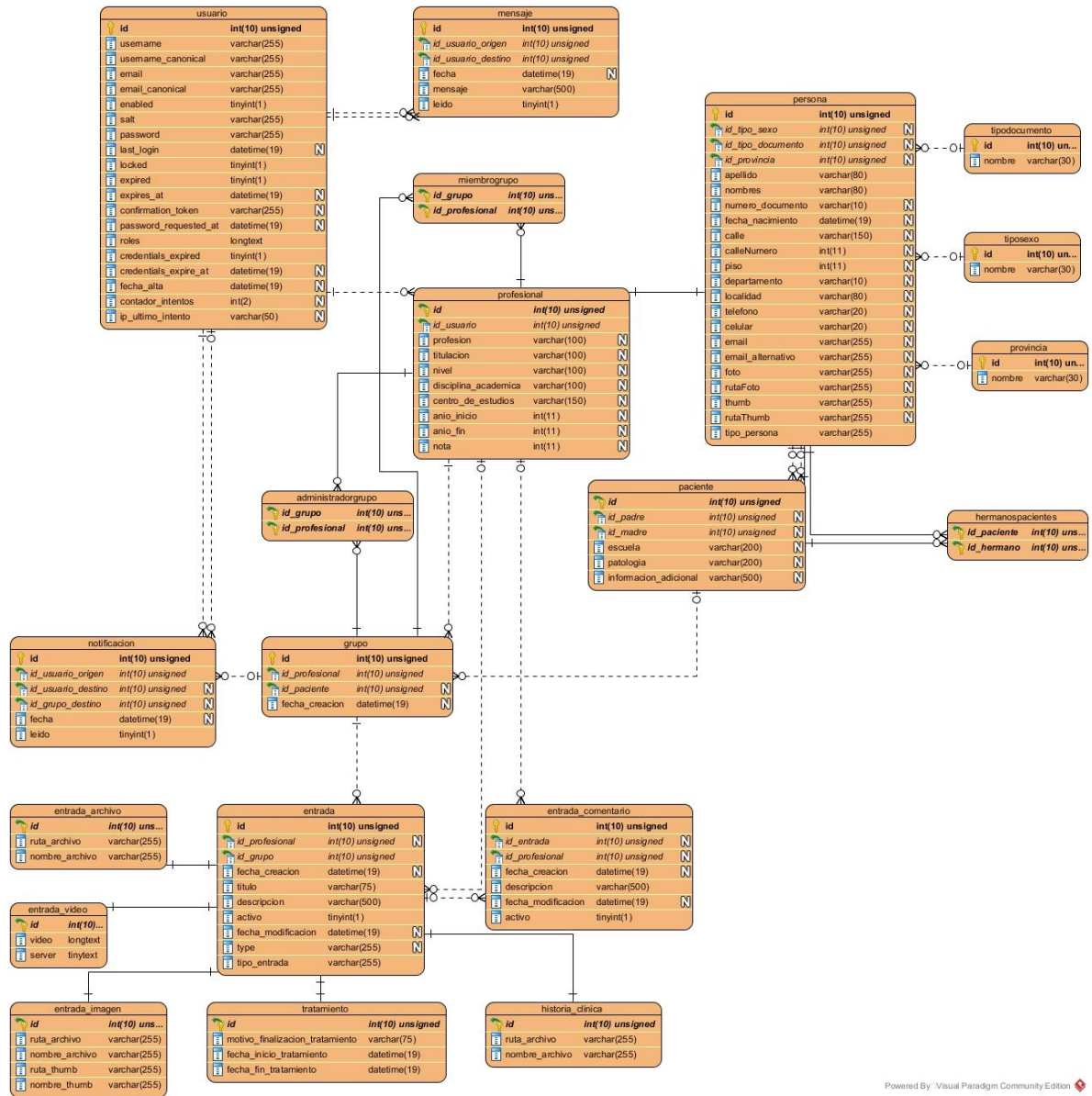


Fig. 3: Diagrama Entidad Relación



## Capítulo 4: Funcionalidades

A continuación se repasarán las pantallas principales de la aplicación junto con imágenes de estas, explicando las funcionalidades a las que dan soporte.

### 4.1 Portada

Una vez que el usuario inicio sesión se le presenta la pantalla principal, conocida como “portada”. Esta reúne la información más reciente vinculada al profesional, mostrando las últimas entradas de los grupos a los que pertenece el usuario.

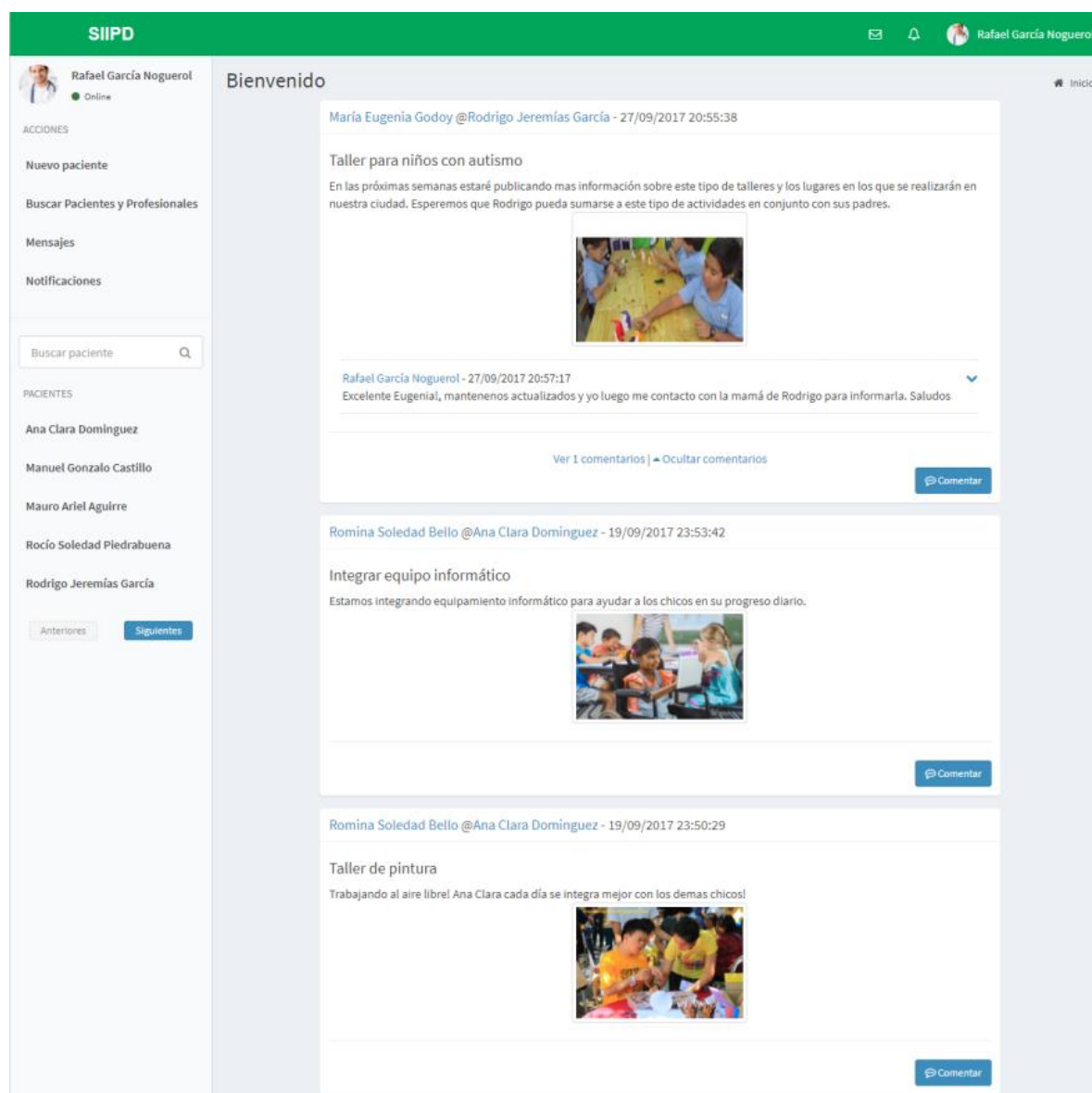


Ilustración 1 - Portada

Además se presentan los accesos a otras funcionalidades como son la creación de pacientes, búsqueda de grupos a los cuales está vinculado el usuario autenticado, búsqueda de profesionales, administración de mensajes, barra de notificaciones y perfil del profesional autenticado entre otras que se detallaran más adelante en este capítulo.

## 4.2 Búsqueda de pacientes y profesionales

Al hacer clic en la opción “Buscar Pacientes y Profesionales” del menú principal, se muestra una ventana modal en la que se pueden realizar búsquedas de profesionales y pacientes indicando el nombre parcial o completo. El sistema realizará una búsqueda unificada, dando los resultados en el mismo listado y mostrando diferentes opciones de acción para cada resultado.

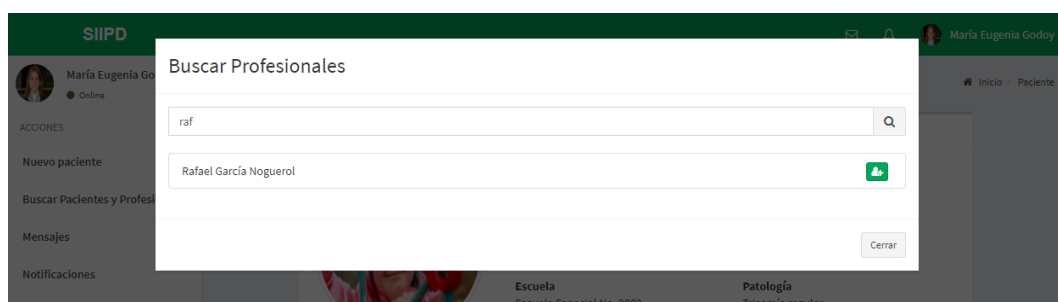


Ilustración 2 - Búsqueda de profesionales

### 4.2.1 Grupos

Esta pantalla reúne las funcionalidades de administración de los grupos a los que el usuario pertenece. En ésta el usuario puede desvincularse de un grupo como así invitar a otros profesionales a unirse a grupos a los que el administra.

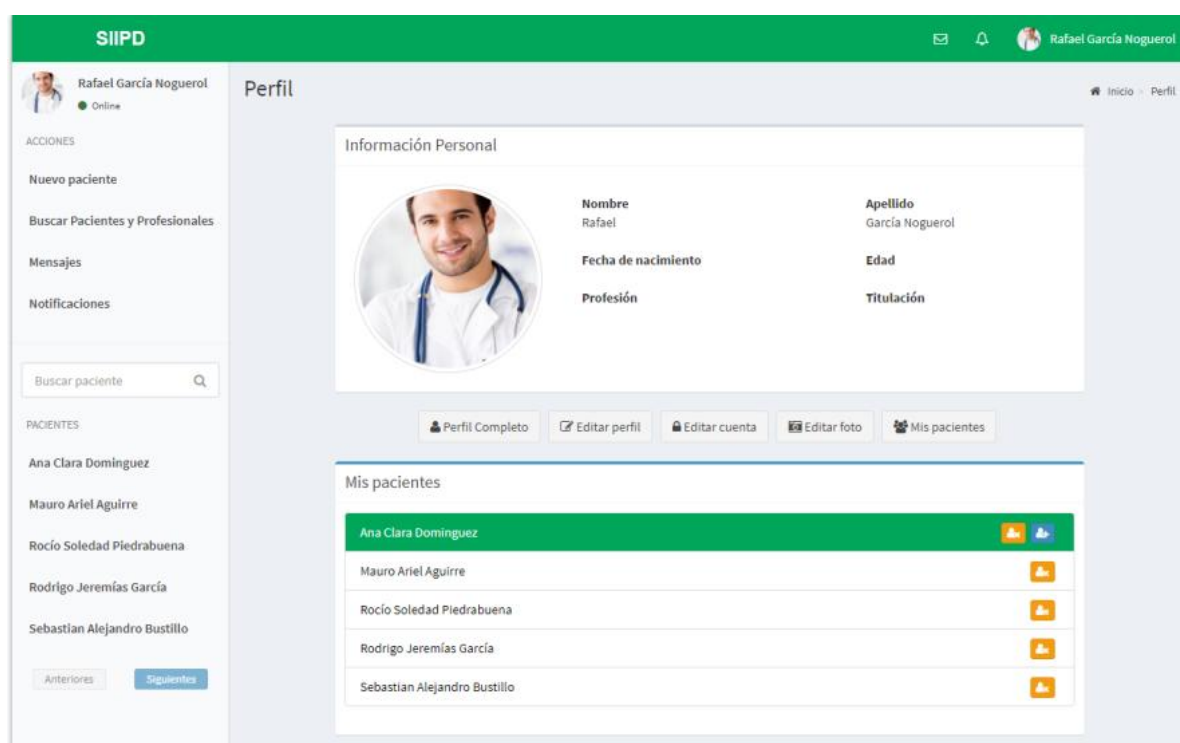


Ilustración 3 - Sección grupos profesionales

El listado de grupos es accesible desde el perfil del propio profesional autenticado y se muestra una vez que se presiona el botón “Mis pacientes”.

#### 4.2.2 Notificaciones

Situado arriba a la derecha, sobre la barra principal se encuentra el icono de notificaciones. Éste avisa al usuario cuando algún profesional solicitó el ingreso a alguno de los grupos que administra, cuando el propio usuario fue aceptado en un grupo al cual pidió el ingreso o cuando le llega una invitación para formar parte de un grupo al que no pertenece.

Este tipo de notificaciones funciona en tiempo real ya que fue implementada mediante un servidor WebSocket que crea una conexión bidireccional y permite este tipo de alertas.

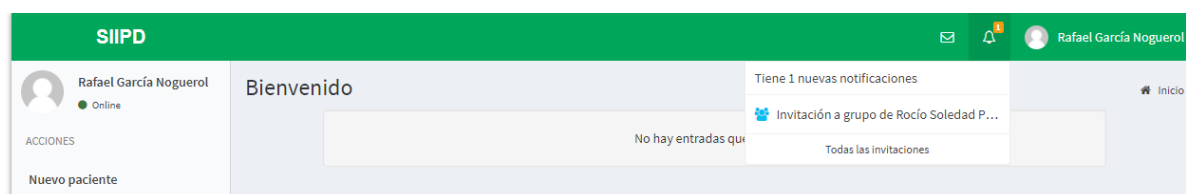


Ilustración 4 - Notificaciones

### 4.3 Perfil Paciente

En la pantalla de perfil del paciente se listan todas las entradas relacionadas, exclusivamente a este paciente, ordenadas cronológicamente. A esta solo pueden acceder, de forma completa, los profesionales que fueron previamente aceptados por el administrador del grupo.

The screenshot shows the 'Perfil' page for a patient named Rodrigo Jeremías. The interface is divided into a left sidebar and a main content area. The sidebar contains navigation options such as 'Nuevo paciente', 'Buscar Pacientes y Profesionales', 'Mensajes', and 'Notificaciones'. The main content area displays the patient's personal information, including a profile picture, name, date of birth, and pathology. Below the personal information, there are tabs for 'Inicio', 'Historial Clínico', 'Tratamientos', 'Multimedia', 'Profesionales', and 'Perfil Completo'. The 'Entrada' tab is active, showing a form to create a new entry with fields for 'Escribe el título de la entrada' and 'Escribe el texto de la entrada'. A recent entry titled 'Taller para niños con autismo' is displayed, featuring a photo of children at a table and a 'Comentar' button.

Ilustración 5 - Perfil Paciente

Además se pueden crear nuevas entradas, comentar entradas existentes, administrar el grupo, invitar a profesionales a unirse y filtrar los distintos tipos de entradas como se muestran más adelante.

Si el profesional que ingresa a ver el perfil del paciente no es miembro del grupo, entonces se muestra solo la sección “Información personal” e inmediatamente debajo el botón “Solicitar ingreso”

#### 4.3.1 Creación de entradas

La creación de nuevas entradas se realiza desde la pantalla principal del grupo para los casos de entradas de texto o multimedia, mientras que para las de tratamientos e historial clínico se deben crear desde las respectivas secciones. Para el caso general se solicita el ingreso de un título y descripción, mientras que para las entradas multimedia se pide adicionalmente el link del video o imagen desde el sistema de archivos.

Al cargar una entrada, el sistema valida que los campos completados tengan la longitud adecuada y que la imagen no supere el tamaño máximo soportado. En caso de producirse un error de validación, se informa al usuario mediante una notificación flotante en la parte superior derecha de la pantalla.

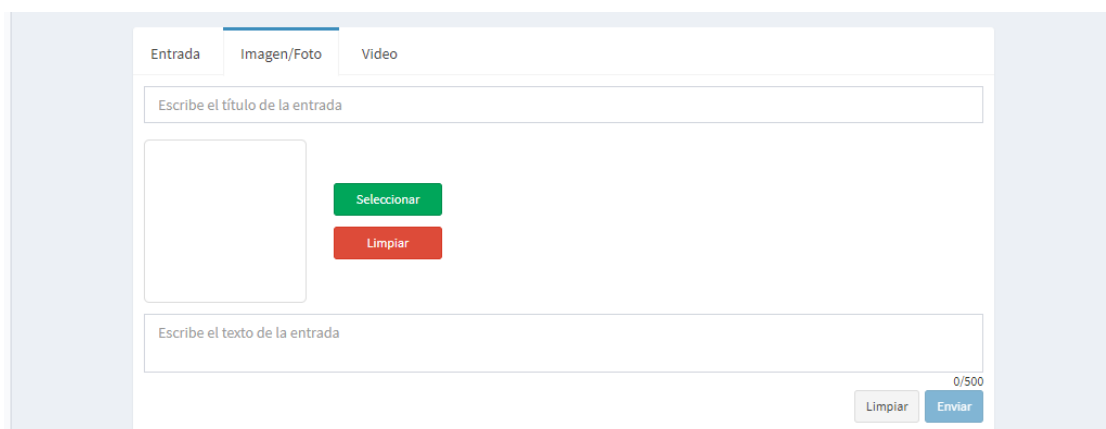


Ilustración 6 - Creación de entradas

#### 4.3.2 Historial Clínico

La pantalla de historial clínico filtra todas las entradas para solo mostrar las que se relacionan con los casos que derivaron en atenciones clínicas. Este tipo de entradas además de un título y descripción, poseen de manera adicional un link a un archivo alojado en el servidor de SIIPD que fue subido por el profesional que lo cargó a fin de proporcionar información adicional.

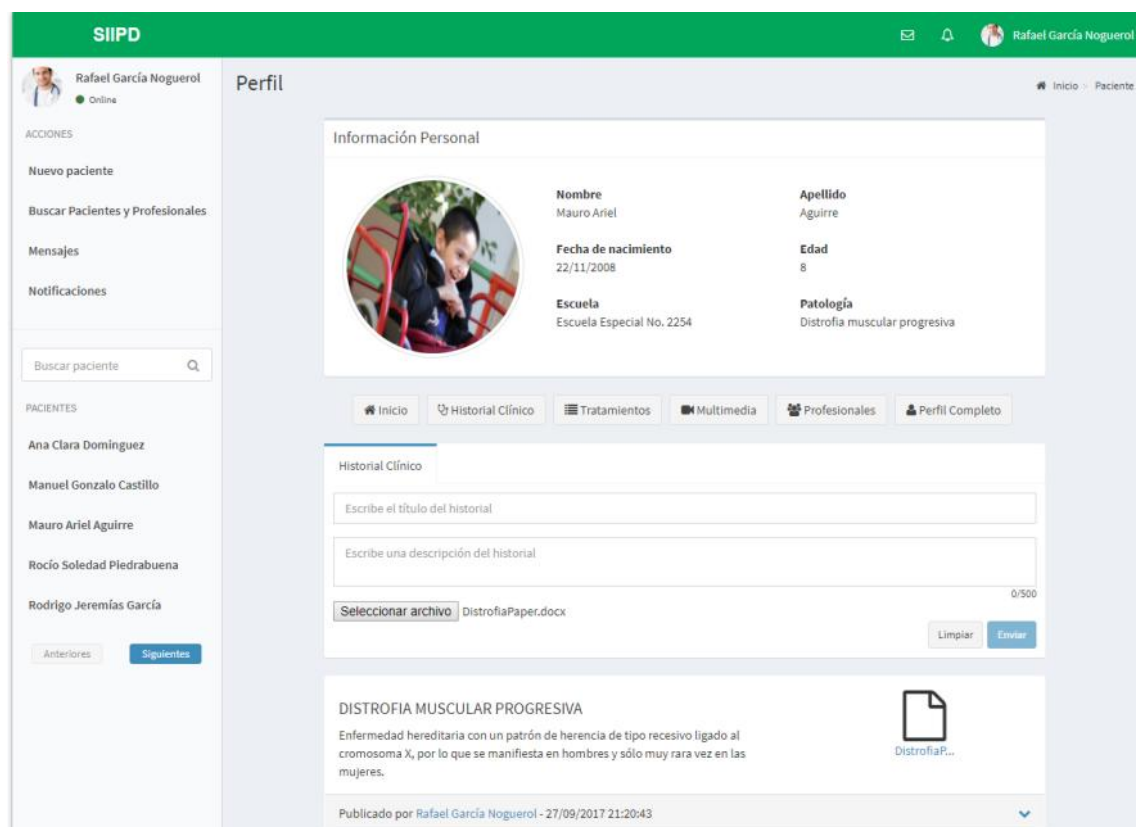


Ilustración 7 - Sección Historial Clínico

### 4.3.3 Multimedia

Aquí se filtran todas las entradas que contienen material audio visual a fines de poder organizar y encontrar fácilmente la información.

La versión actual de SIIPD soporta los tipos de imágenes más comunes como son, gif, jpg, png, bmp y realiza una optimización de los mismos modificando su tamaño y calidad antes de subirlos al servidor.

Por otra parte, para el soporte de archivos de video se decidió utilizar YouTube, ya que la gestión de este tipo de archivos es sumamente compleja y SIIPD solo se limita a validar que la URL cargada al momento de crear la entrada de video sea válida.

SIIPD



 María Eugenia Godoy

**María Eugenia Godoy**  
● Online

ACCIONES

[Nuevo paciente](#)

[Buscar Pacientes y Profesionales](#)

[Mensajes](#)

[Notificaciones](#)

---

Buscar paciente

---

PACIENTES

[Ana Clara Dominguez](#)

[Mauro Ariel Aguirre](#)

[Rocío Soledad Piedrabuena](#)

[Rodrigo Jeremías García](#)

[Sebastian Alejandro Bustillo](#)

[Anteriores](#)
[Siguientes](#)

## Perfil

**Información Personal**

**Nombre**  
Rodrigo Jeremías

**Apellido**  
García

**Fecha de nacimiento**  
29/05/2010

**Edad**  
7

**Escuela**  
Escuela Especial No. 2083

**Patología**  
TRASTORNO GENERALIZADO DEL DESARROLLO - NO ESPECIALIZADO

[Inicio](#)
[Historial Clínico](#)
[Tratamientos](#)
[Multimedia](#)
[Profesionales](#)
[Perfil Completo](#)

Imagen/Foto    Video

Escribe el título de la entrada

Escribe el texto de la entrada

0/500  
[Limpiar](#) [Enviar](#)

María Eugenia Godoy - 27/09/2017 20:55:38

**Taller para niños con autismo**

En las próximas semanas estaré publicando mas información sobre este tipo de talleres y los lugares en los que se realizarán en nuestra ciudad. Esperemos que Rodrigo pueda sumarse a este tipo de actividades en conjunto con sus padres.

Ver 2 comentarios

[Comentar](#)

María Eugenia Godoy - 19/09/2017 23:03:36

**Progreso en el aula**

En el video se ve el progreso de Rodrigo en el aula. De a poco se va integrando al grupo.

[Comentar](#)

Copyright © 2017 Daniel Solís - Juan José Barrionuevo Sastre. Todos los derechos reservados.
Arriba

Ilustración 8 - Sección Multimedia

#### 4.3.4 Tratamientos

En esta sección se presenta la funcionalidad para la carga de nuevos tratamientos los cuales requiere del ingreso de un título, fechas de inicio y fin como así como también una descripción del mismo. Además al acceder a esta se filtran las entradas para mostrar solo las pertenecientes a los tratamientos previamente cargados.

The screenshot displays the 'Tratamientos' (Treatments) section of the SIIPD system. The top navigation bar shows the user 'Rafael García Noguero' and the system name 'SIIPD'. The left sidebar contains navigation options like 'Nuevo paciente', 'Buscar Pacientes y Profesionales', and a list of patients including 'Ana Clara Dominguez', 'Manuel Gonzalo Castillo', 'Mauro Ariel Aguirre', 'Rocio Soledad Piedrabuena', and 'Rodrigo Jeremias Garcia'. The main content area shows the profile of 'Mauro Ariel Aguirre' with personal information: Name (Mauro Ariel), Surname (Aguirre), Birth Date (22/11/2008), School (Escuela Especial No. 2254), and Pathology (Distrofia muscular progresiva). Below the profile, there are tabs for 'Inicio', 'Historial Clínico', 'Tratamientos', 'Multimedia', 'Profesionales', and 'Perfil Completo'. The 'Tratamiento' form includes a text input for the treatment name, two date pickers for start and end dates, and a text area for description. Below the form, there are two existing treatment entries: 'MEDICAMENTOS - ESTEROIDES' and 'FISIOTERAPIA', each with a description and dates. The bottom of the page shows a copyright notice for 2017 and the name 'Daniel Solís - Juan José Barrionuevo Sastre'.

Ilustración 9 - Sección Tratamientos

Al igual que con otros tipos de entradas, el sistema valida que los datos cargados sean consistentes, por ejemplo para las fechas se valida que la de inicio sea siempre menor que la de fin y en caso de no cumplirse, no deja guardar el tratamiento.



### 4.3.5 Profesionales

En esta sección se listan todos los profesionales vinculados al paciente. En esta se detalla el nombre y apellido junto con una descripción de la profesión que este ejerce. A su vez, los usuarios administradores del grupo cuentan con la posibilidad de agregar nuevos miembros a la vez que brinda las herramientas para desvincular miembros activos del grupo.

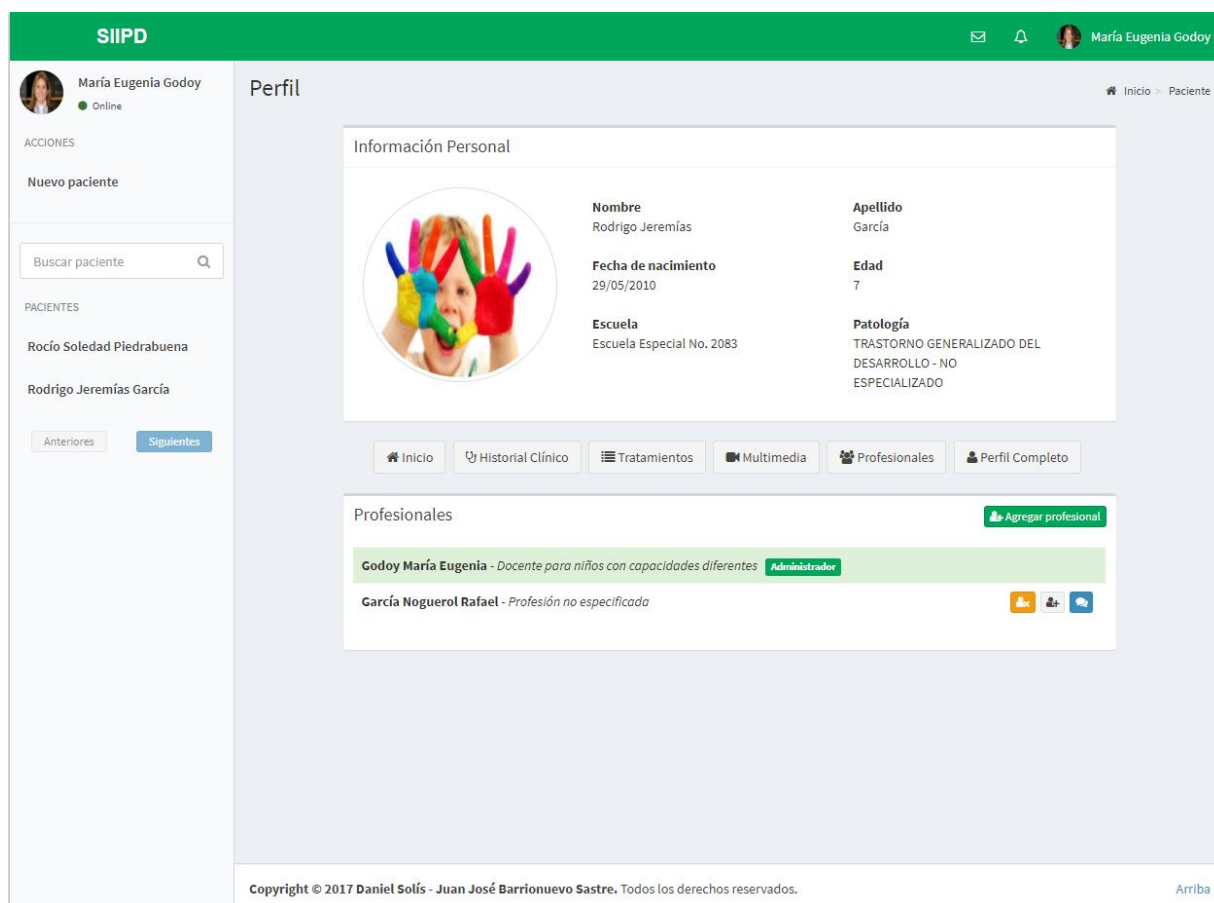


Ilustración 10 - Sección Profesionales

### 4.4 Perfil Profesional

En la pantalla del perfil del profesional se muestra los datos de este junto con su información personal, siendo potestad del propio usuario editar y administrar sus datos. A su vez, en la sección “Mis Pacientes” se listan los grupos a los que se encuentra vinculado con la opción de desvincularse de estos.

The screenshot displays the professional profile of María Eugenia Godoy on the SIIPD platform. The interface includes a navigation sidebar on the left with options like 'Nuevo paciente', 'Buscar Pacientes y Profesionales', 'Mensajes', and 'Notificaciones'. The main content area is titled 'Perfil' and is divided into several sections:

- Información Personal:** Includes a profile picture, name (María Eugenia), surname (Godoy), birth date (17/01/1989), age (28), profession (Docente para niños con capacidades diferentes), and degree (Profesor de Educación Especial en Discapacitados Intelectuales).
- Mi perfil:** A detailed section with sub-sections:
  - Datos de cuenta:** Username (megodoy), email (megodoy@mail.com), registration date (18/09/2017), and last connection date (18/09/2017).
  - Información personal:** Name (María Eugenia), surname (Godoy), birth date (17/01/1989), sex (Femenino), and document type/number (DNI 77722201).
  - Ubicación y datos de contacto:** Province (Santa Fe), locality (Santa Fe), address (Vieytes 2355), and phone number (034246976544).
  - Información profesional y académica:** Profession (Docente para niños con capacidades diferentes), degree (Profesor de Educación Especial en Discapacitados Intelectuales), discipline (Profesorado de educación especial en discapacitados intelectuales), center (I.S.P. N° 8 "Almte. Guillermo Brown"), start year (2007), and end year (2011).

At the bottom of the page, there is a copyright notice: 'Copyright © 2017 Daniel Solís - Juan José Barrionuevo Sastre. Todos los derechos reservados.' and the word 'Arriba' in the bottom right corner.

Ilustración 11 - Perfil Profesional

## 4.5 Administración de mensajes

En la siguiente imagen se visualiza la bandeja de administración de mensajes, en donde el usuario podrá gestionar y contestar sus mensajes.

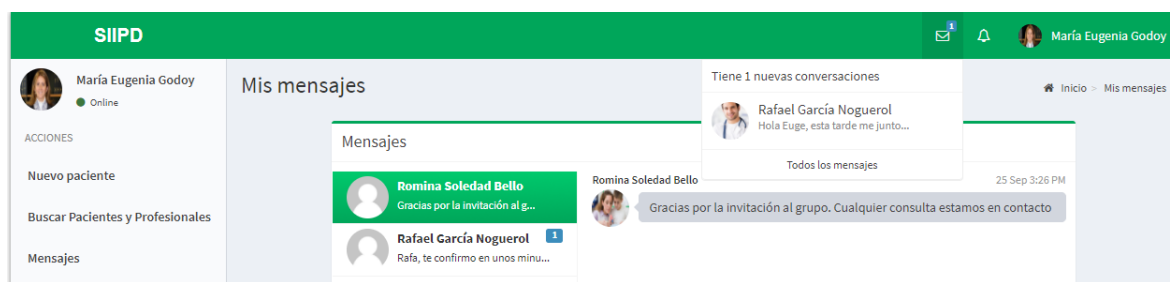


Ilustración 12 - Administrador de mensajes

## Capítulo 5: Arquitectura

---

En este capítulo se detalla la arquitectura de software diseñada para la implementación del sistema, así como también diferentes consideraciones arquitectónicas, patrones de diseño utilizados, buenas prácticas de desarrollo y consideraciones para el despliegue del sistema.

### *5.1 Supuestos y consideraciones*

El Sistema de Integración de la Información del Paciente – SIIPD – debe cumplir con determinados atributos de diseño que permitan que el mismo pueda ser desarrollado en sucesivas etapas incrementales, fácilmente integrables y a su vez sentar una base para el desarrollo de mejoras y nuevas funcionalidades con el menor impacto posible. Además se hizo foco en el desarrollo de un sistema multiplataforma, es decir, abarcar el mayor número de sistemas operativos y hardware para su puesta en producción, considerando que se planea proyectar un futuro despliegue en línea con el mínimo esfuerzo posible.

Teniendo en cuenta estas bases son necesarias las siguientes propiedades para la arquitectura definida:

- Exhibir un diseño basado en componentes de propósito claro y concreto, con alto grado de cohesión.
- Disminuir el esfuerzo requerido para el mantenimiento, actualización y mejoras del sistema mediante un diseño desacoplado de los componentes.
- Proveer mecanismos de seguridad utilizando componentes altamente probados.
- Facilitar la extensión de la plataforma a dispositivos móviles y mantener una independencia del framework del frontend mediante la exposición de una API con interfaces bien definidas.

### *5.2 Arquitectura física del sistema*

A continuación se presenta el diagrama de despliegue donde se detalla la distribución física del sistema.

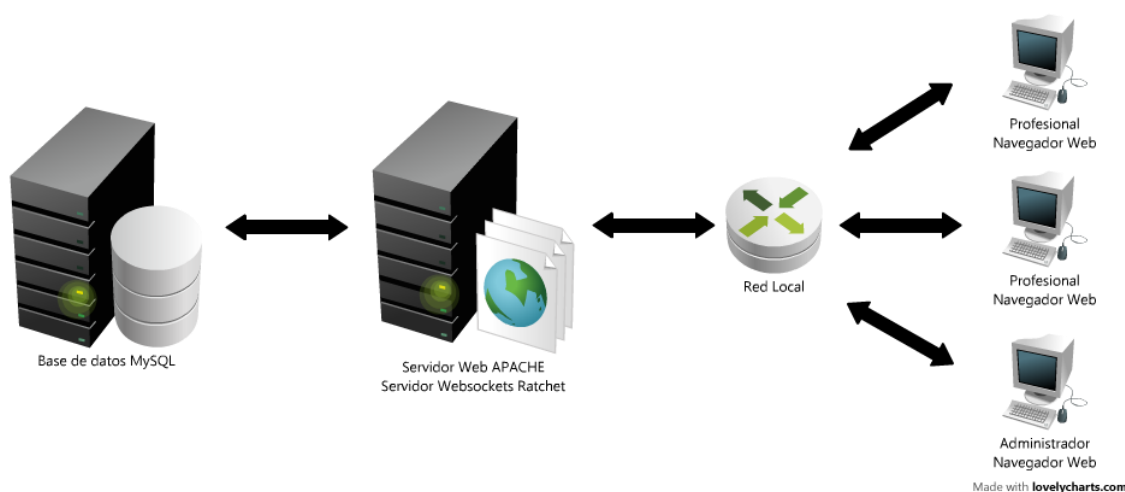


Fig. 4: Diagrama de despliegue

- **Base de datos:** MySQL fue seleccionado como el motor de base de datos para almacenar la información del sistema. Fundamentamos esta decisión de acuerdo a los tres aspectos claves como son la gran cantidad de plataformas que soportan MySQL, su licencia open source y su gran comunidad que provee un soporte continuo. Inicialmente el servidor de base de datos estará funcionando en el mismo servidor físico en donde se encuentra el servidor web, aunque este se planea distribuir en dos servidores físicos cuando en un futuro el tráfico lo amerite.
- **Servidor Web:** Este equipo es el que procesará todas las peticiones que lleguen desde los clientes (profesionales y administradores). Los fundamentos de utilizar un servidor Apache radica, como en el caso de la base de datos, en que podemos tener instalaciones estables en gran cantidad de plataformas y además posee licencia open source. Por otro lado, considerando la decisión de implementar SIIPD utilizando Symfony2 que se ejecuta sobre PHP, el soporte en línea es completo y hay una gran comunidad a lo largo del mundo que lo respalda.
- **Navegador Web:** Este se encontrará instalado en la pc del usuario final y será el encargado de interpretar el metalenguaje HTML descargado del servidor web y además de ejecutar el código JavaScript. Por tal motivo es requerido que los clientes utilicen un navegador web moderno, con soporte para HTML5 y Ecma5 en adelante, además de una resolución no menor a 1024x768px para una visualización óptima del sistema.

### 5.3 Vista lógica de alto nivel

En la siguiente figura se puede observar una representación del sistema completo integrado por el backend (Symfony2) y el frontend (AngularJs) que se ejecutan en el hardware del servidor y cliente respectivamente. Además se muestran los puntos de interacción entre la API Rest expuesta por el backend y los controllers AngularJS.

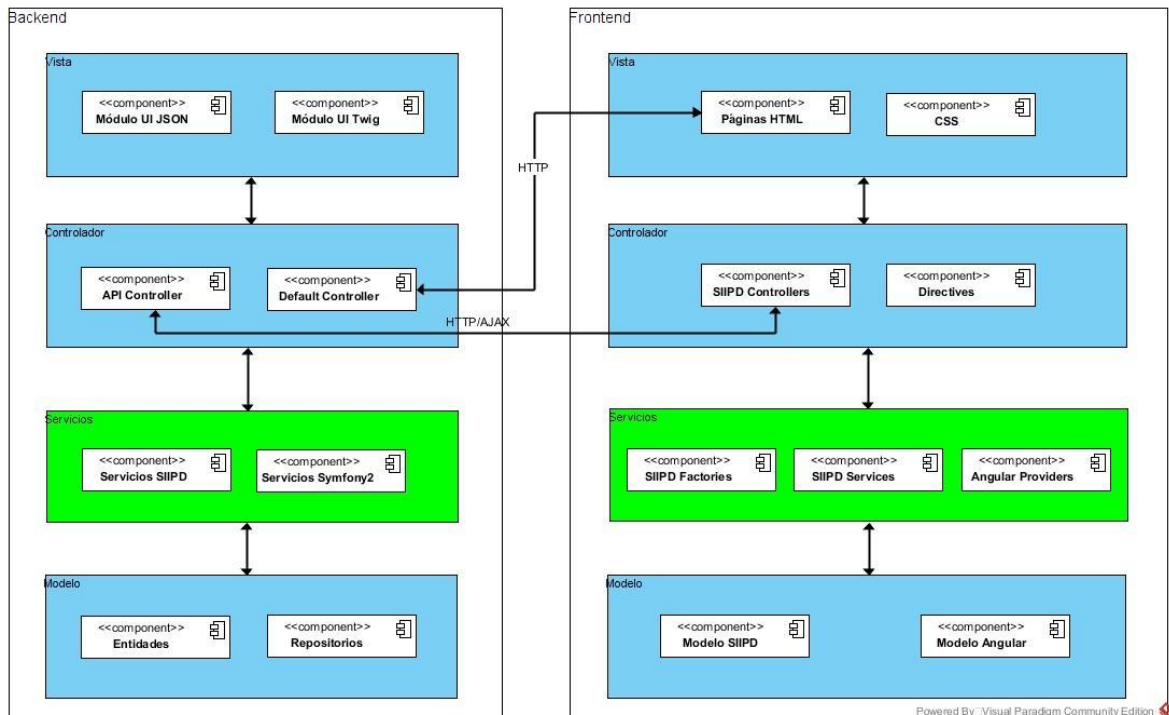


Fig. 5: Diagrama de distribución por capas

### 5.4 Backend

El backend consiste en una aplicación Symfony2 MVC, con el agregado de una capa de servicios bien definida aprovechando el contenedor de servicios provisto por el framework.

**Vista:** Compuesta por el conjunto de plantillas Twig, que son retornadas por los controladores al cliente en la respuesta HTTP como un archivo HTML convencional. Esta capa procesa tanto los HTML como los objetos JSON que son retornados en la respuesta HTTP a las invocaciones realizadas a la API Rest. Esto se logra por medio de 2 librerías soportadas por el proyecto Symfony2 que son FOSRest Bundle y JMS Serializer.

**Controlador:** SIIPD contará con dos controladores principales, el API Controller, encargado de atender las peticiones recibidas desde la ruta /api y el Default Controller,

responsable de recibir todas las peticiones HTTP que no tengan como requerido el content type JSON, es decir aquellas que requieran una respuesta cuyo contenido sea HTML, como por ejemplo el index del sitio.

**Servicios:** Su funcionalidad es la de mantener los controladores lo más simple posibles encargándose de la lógica del negocio. Utilizando inyección de dependencias, son los encargados de interactuar con otros servicios a fin de implementar la lógica del sistema.

**Modelo:** Está compuesto por entidades que representan el modelo de datos de la aplicación, mapeando la base de datos. Estas entidades a su vez poseen un repositorio que funciona como Objetos de Acceso a Datos y facilita las operaciones de la aplicación con la base de datos. Para cumplir con dicho objetivo se utiliza el ORM Doctrine incluido en todo proyecto Symfony2 por defecto.

## 5.5 Frontend:

Por su parte, el frontend del sistema se implementa utilizando AngularJS, HTML 5 y CSS. Esta decisión está fundada en la premisa que AngularJS permite desarrollar diferentes módulos JavaScript bien definidos además de proveer excelentes mecanismos para interacción con API Rest y con el DOM HTML, entre otras.

**Vista:** Esta es la capa visible por el usuario final de la aplicación y está integrada por todas las páginas HTML y los archivos CSS que son utilizados por el navegador web para mostrar al usuario los elementos de la interfaz de usuario. Esta capa se realizó íntegramente utilizando como base la plantilla AdminLTE de Abdullah Almsaeed la cual fue adaptada para el sistema y que está diseñada a partir del uso de la librería CSS Bootstrap v3.

**Controlador:** Los controladores AngularJS son los encargados de interactuar con la API Rest del backend para luego impactar en el Modelo y finalmente actualizar la Vista, todo esto es posible gracias al mecanismo de enlace de dos vías (two-way binding) brindado por el framework.

Por otra parte, todos aquellos componentes que deban ser reutilizados dentro del sistema se implementaron mediante directivas, ya que es uno de los medios principales provistos por AngularJS para la separación bien definida en componentes.

**Servicios:** AngularJS provee un contenedor de servicios que permite implementar tres tipos diferentes de servicios (services, providers y factories). En SIIPD se desarrollaron por medio de factories e invocaciones AJAX con el fin de ser reutilizadas en todos los controladores del frontend según fue necesario.

**Modelo:** En AngularJS el modelo está compuesto principalmente por un objeto llamado \$Scope el cual permite pasar a la vista los resultados obtenidos de las diferentes invocaciones realizadas a la API Rest desde el controlador.

### 5.6 Patrones de diseño

**Patrón MVC:** El uso de ambos frameworks nos facilita organizar el código de acuerdo a las convenciones establecidas por estos.

La selección de ambos frameworks (Symfony2 y AngularJS) está fundada en que estos utilizan el patrón y permiten tener una clara separación de las diferentes capas, lo que facilita el desarrollo y mantenimiento a su vez que potencia el desarrollo de mejoras a futuro.

A continuación se presentan dos diagramas que en los que se muestra la aplicación del patrón para Symfony2 y AngularJS respectivamente.

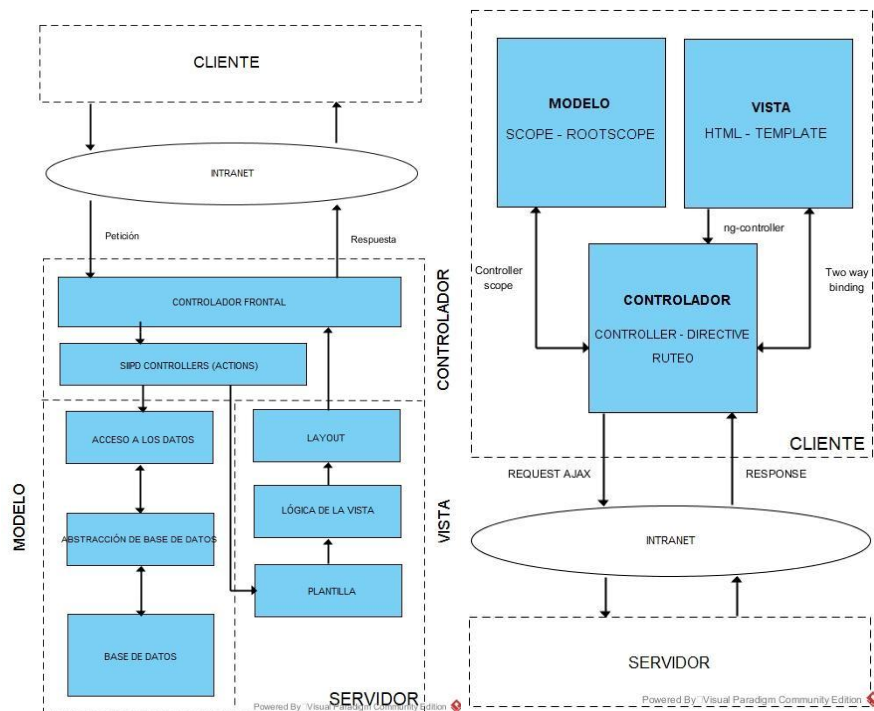


Fig. 6: Diagrama de componentes



**Inyección de dependencia:** Como se mencionó anteriormente se añade una capa extra al modelo MVC, que consiste en un conjunto de servicios que permite delegar responsabilidades, a la vez de reutilizar y modularizar el código. Un servicio es un objeto que se registra en el contenedor de servicios de Symfony2. Utilizando el contenedor de servicios es posible inyectar dichos servicios en los controladores o en otros servicios, en lugar que estos sean quienes creen instancias de los mismos.

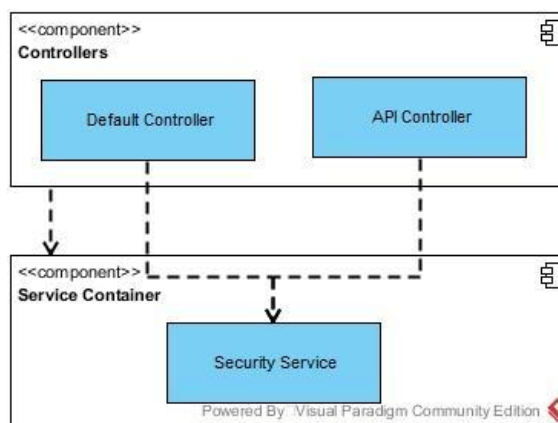


Fig. 7: Diagrama Controladores-Contenedor de servicios

Otros patrones:

Debido que se utilizan diferentes frameworks y librerías, es natural utilizar diferentes patrones, que se nombran a continuación:

- **Singleton:** Utilizado por ejemplo en el ruteo de Symfony2. El método `getInstance` de la clase `sfRouting` es utilizado por el front controller del framework y se encarga de direccionar todas las peticiones que se hagan al backend.
- **Registry:** Este patrón es utilizado tanto en Symfony2 como en Angular para compartir de manera eficiente datos y objetos de la aplicación evitando la necesidad de crear variables globales.
- **Factory:** Se utiliza en el frontend para agrupar todas las llamadas Ajax a la API Rest, de forma que se permita reutilizarlas donde sea necesario y tener una agrupación lógica de las mismas.
- **Proxy y Repository:** Ambos patrones de diseño son utilizados en Doctrine, el ORM provisto por Symfony2 y que se utilizará en SIIPD para el acceso a los datos.

## 5.7 Tecnologías y criterios de diseño

### 5.7.1 Capa de presentación

**Backend:** La capa de presentación del backend, al pertenecer a una API Rest no se considerarán vistas, sino que se consideran recursos que son solicitados en diferente formato utilizando el atributo Content-Type del encabezado HTTP de la petición. En este caso, el único tipo de recurso soportado por la API de SIIPD es Application/Json

**Frontend:** En este caso, la capa de presentación está formada por un conjunto de páginas HTML 5, cuyo estilo está especificado en diferentes archivos de estilo cascada CSS2. En esta capa es muy importante tener en cuenta un atributo de calidad mencionado anteriormente como la usabilidad.

### 5.7.2 Capa de negocio

**Backend y Frontend:** Como fue indicado anteriormente, esta capa se apoya en una serie de servicios inyectados a los controladores, los cuales realizan la mayor parte de la lógica y validaciones. Además los servicios son inyectados en otros servicios para poder cumplir dichos objetivos. Los controladores por su parte son los encargados de direccionar todo el flujo entre las diferentes capas, por ejemplo, un servicio nunca podrá retornar una respuesta a una petición invocando a la capa de presentación; esta tarea es responsabilidad del controlador.

### 5.7.3 Capa de acceso a datos

**Backend:** El acceso a los datos se realiza por medio del ORM Doctrine 2, mapeando las tablas a objetos PHP que se denominan Entidades. A su vez, todas las operaciones de creación, obtención, actualización y eliminación de datos se delegan a una clase repositorio para cada una de las entidades definidas.

**Frontend:** El acceso a los datos se realiza por medio de AJAX utilizando el provider \$http de AngularJS y luego las respuestas JSON recibidas desde la API Rest son manipuladas como objetos JavaScript.

## 5.8 Arquitectura tecnológica

- **Sistema Operativo:** Los servidores web y de base de datos (en caso de encontrarse en equipos de hardware diferentes o en entornos virtuales diferentes) podrán ser Unix, GNU/Linux, Windows o cualquier otro sistema que soporte la instalación de un servidor Web con PHP 5.5+. Para una implementación inicial, recomendamos Windows Server 2012 o Debian 8.
- **Motor de base de datos:** Considerando lo detallado en los requerimientos no funcionales dentro de este mismo capítulo, la base de datos seleccionada fue MySQL 5.5+, ya que es considerada la base de datos open source con mayor soporte a nivel mundial y se ajusta completamente a todos los requerimientos funcionales.
- **Servidor de aplicación:** Debido a la facilidad de instalación de sistemas Symfony2 en servidores APACHE, se decidió utilizar como servidor Web, la versión 2 de APACHE Server. Además esta decisión está fundada en que la integración con PHP es relativamente sencilla y además permite configuraciones muy estables y seguras en los sistemas operativos sugeridos. Otra característica importante de APACHE es su gran soporte y comunidad para resolución de posibles conflictos.
- **Almacenamiento de documentos:** Para todas aquellas operaciones del sistema que requieran almacenamiento de datos, se optó por utilizar el sistema de archivos del servidor de aplicaciones, por sobre la base de datos. Esto se debe a que se desea mantener una base de datos limpia y de tamaño reducido.

## Capítulo 6: Conclusiones

---

Las carencias en las comunicaciones entre los profesionales que atienden la discapacidad traen consigo una pérdida de oportunidades en la detección de problemas, con su posterior impacto en la eficacia del tratamiento conjunto del paciente.

Si bien el sistema de información generado como parte de este proyecto tuvo un alcance acotado a la comunidad profesional que atiende la discapacidad, los aportes esperados no son menores entendiendo que, hasta el momento, no se disponía de ninguna herramienta para el registro de las actividades del paciente e integración de este tipo de información.

Poner a disposición estos recursos apunta a mejorar la cantidad y calidad de la información a la que pueden acceder colaborativamente, lo cual mejora la comunicación con un impacto directo en la eficacia del tratamiento y acompañamiento del paciente, siendo este precisamente el objetivo general del proyecto. Bajo esta concepción, podemos concluir que el proyecto fue ampliamente satisfactorio desde el punto de vista del cumplimiento de objetivos, más aún, considerando la finalidad del producto generado.

Desde el plano personal, el proyecto llega como el cierre ideal de una etapa en donde pudimos aportar el aprendizaje recolectado en todos los años de estudio y aplicarlo en la práctica real, resultando en una experiencia enriquecedora. El proyecto nos permitió participar en el proceso completo del ciclo del desarrollo, desde el entendimiento de las necesidades del cliente, el relevamiento de los requerimientos, el diseño de la arquitectura y la implementación de la solución en sí. Todo esto, se dio en el marco de un acompañamiento del cliente, dado que no solo participó de la etapa inicial sino que más bien se involucró en el propio desarrollo del proyecto, participando activamente de las reuniones para validar funcionalidades y aportar su valiosa retroalimentación. Esto hizo posible implementar la metodología elegida de la mejor forma posible al mismo tiempo que la adaptamos al contexto que supone un proyecto final de carrera.

Consideramos que el éxito del proyecto ha significado una contribución a la sociedad y un gran aporte profesional, que simboliza la transición a una nueva etapa profesional, que nos plantea nuevos objetivos y metas.

## 6.1 Futuro de la aplicación

El proyecto se enfocó desde el inicio bajo la idea de servir como punto de partida para el desarrollo de una plataforma extensible que se pueda ir ampliando en cuanto a funcionalidades en la medida que los usuarios lo requieran. Es por esto, que se mantuvo el foco en la extensibilidad, como un requerimiento no funcional fuertemente buscado, algo que se vio reflejado en la arquitectura y en el desarrollo de la implementación para poder soportar, en un futuro, los siguientes requerimientos:

- **Exposición del sistema a redes públicas:** La plataforma fue pensada para ser usada, en un principio, en una intranet con un número reducido de usuarios. Pero conforme vaya evolucionando se extenderá para poder ser accedida desde cualquier punto, permitiendo a los usuarios hacer seguimientos fuera de los centros de salud especializados en discapacidad.
- **Incorporación de los centros de salud especializados como actores dentro del sistema:** Como próximo paso importante dentro del modelo lógico es el de incorporar a los centros de salud que agrupan a estos profesionales, para posterior brindarle acceso como un tipo distinto de usuarios permitiendo así el poder administrar y tener visibilidad de las actividades que realizan los profesionales que integran estos centros.
- **Integrar la información de obras sociales:** Se planea modelar e incorporar al dominio la información referente a las obras sociales, entendiendo que estas forman parte del proceso administrativo en la gran mayoría de los casos.
- **Desarrollo de herramientas que den soporte a las actividades del dominio:** Una vez desarrollada las funcionalidades principales que hacen al núcleo de la plataforma se procederá a incorporar otras herramientas y funciones que son utilizadas por los usuarios del sistema, con el fin de integrar y centralizar la información así como también las herramientas que se utilizan en la operatoria habitual como pueden ser una agenda de actividades y un sistema de reportes que hagan uso de los datos ya existentes.

## Anexos

---

En esta sección se detallan los anexos adjuntos al informe final, los cuales facilitan y ayudan a tener un mayor entendimiento del proyecto.

- **Anexo A, Cronograma de avance:** Diagrama de Gantt con la planificación de las actividades a lo largo del calendario de proyecto.
- **Anexo B, Estimaciones de esfuerzo:** Ponderación de esfuerzo y prioridad para posterior cálculo de estimación de esfuerzo en base a los Casos de uso relevados.
- **Anexo C, Actividades realizadas:** Lista de todas las actividades desarrolladas en el proyecto.
- **Anexo D, Plan de gestión de riesgos:** Informe con el detalle de los riesgos relevados y plan de contingencia.
- **Anexo E, Casos de uso:** La totalidad de los casos de uso en detalle.

## Bibliografía

---

1. E. Kendall, K., E. Kendall, J.: Análisis y diseño de sistemas, 6ta edición. Pearson Educación, México (2005).
2. Martin, Robert C.: CleanCode, 1ra edición. Prentice Hall. Estados Unidos (2009).
3. Peralta, Mario: Introducción a UML (1995).
4. Pfleeger, Shari Lawrence: Ingeniería del Software, 3era edición (2002).
5. Sommerville I.: Ingeniería del software, 7ma edición. Pearson Educación. S.A.. Madrid (2005).
6. Bootstrap 3.3, Guía de referencia. <https://getbootstrap.com/docs/3.3/css/>
7. Symfony 2.8, Guía de referencia. <https://symfony.com/doc/2.8/reference/index.html>
8. AngularJS, Guía de referencia. <https://docs.angularjs.org/api>