

CONCURSO DE TRABAJOS ESTUDIANTILES  
EST 2012

Categoría: Trabajos de Cátedra

**Sistema de reserva de turnos médicos on-line “Doc-net”**



UNIVERSIDAD  
**AUSTRAL**

Mayo 2012

Alumnos:

- Kevin Stessens
- Tomás Mateus

Docentes:

- Nicolás Damonte
- Ignacio Rodriguez
- Ignacio Cassol

## Sistema de reserva de turnos médicos on-line “Doc-net”

**Resumen:** Actualmente el proceso que el paciente debe realizar para obtener un turno con un médico, ya sea en hospitales o con un médico particular, es incómodo, tanto para él como para el médico que desea contactar. En base a esto surge la idea de Doc-Net, un sistema que permite al usuario eliminar la cartilla médica, evitar pasar horas en el teléfono intentando concretar una reserva y, al médico, obviar a los que no cumplen con sus reservas. Doc-Net optimiza un proceso que es ineficiente para ambas partes. El sistema incluye la gestión de historias clínicas.

El usuario dispone de un listado de médicos, que podrá consultar según sus preferencias y sacar turno con aquellos que seleccione, sin necesidad de una cartilla médica o de un llamado telefónico.

### 1. Introducción:

Hoy en día, un paciente que necesita sacar un turno debe pasar un proceso que es muy ineficiente. A la vez, los servicios y facilidades que ofrecen las redes sociales e internet, permiten minimizar dichas ineficiencias. Tras buscar un médico en una engorrosa cartilla, el interesado se ve obligado a contactarse ya sea vía telefónica, o presentándose personalmente en el consultorio. Estas pérdidas de tiempo innecesarias son las que soluciona Doc-Net de forma directa, ya que el trámite se realiza completo desde una misma aplicación.

Además, gracias a que cada paciente tiene un usuario, se logran evitar pérdidas o errores en la información, por ejemplo en las recetas, ya que en un futuro podría facilitarse la gestión de medicamentos incluyendo vademécums farmacológicos, o historias clínicas que se pierden en los ficheros. Fuera de esto, reduce el espacio físico necesario para el almacenamiento de las fichas de pacientes y sus historias clínicas.

En síntesis, el beneficio es amplio, y no solo aplica al sistema de turnos, sino a una mejora integral del servicio médico.

Cada paciente posee en el sistema un ranking que se genera en base al cumplimiento de los turnos solicitados, permitiendo que el médico restrinja su disponibilidad sólo a los pacientes que guarden cierta calificación.

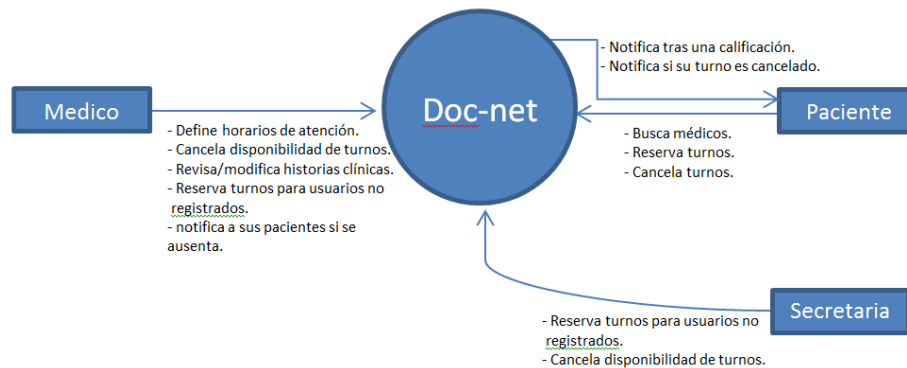
El sistema es novedoso también por unificar la gestión de turnos con la gestión de historias clínicas.

Este proyecto surge como trabajo final de una materia (Laboratorio I) de 3° año de Ingeniería Informática. Los objetivos académicos que nos llevaron a implementar este sistema son:

- Aplicación de los conocimientos de la programación Orientada a Objetos a un sistema concreto.
- Aplicación de conceptos vinculados a la Ingeniería del Software.
- Confección de un diagrama UML y codificación completa del mismo en Java.
- Aplicación, a un caso concreto, del ciclo de vida de un proyecto de software.
- Experiencia en el contacto y vinculación con *stakeholders*.
- Familiaridad con nuevas tecnologías.

## 2. Software similar existente:

Actualmente existen en el mercado sistemas de administración de consultorios médicos, tales como *Softmédico* y CITMED, pero los mismos se diferencian de nuestra propuesta ya que éstos apuntan sólo a los médicos y no al sistema completo que puede verse en la ilustración 1



1 - Diagrama de contexto Doc-Net

Por otro lado, existe un sistema llamado “turnos-médicos” que busca brindar el mismo servicio al que apunta Doc-Net, pero tras intentar utilizarlo, llegamos a la conclusión de que es difícil e incómodo de operar, ya que si bien es intuitivo al comienzo, se torna difícil la selección de médico, la cantidad de médicos afiliados es baja, y a la hora

de concretar la reserva de turnos el sistema falla. Muchas clínicas y hospitales cuentan con sistemas de turnos y/o con sistemas de gestión de fichas médicas. El Hospital Austral, por ejemplo, cuenta con un sistema integrado pero, el mismo está hecho a medida: no se trata de un producto existente en el mercado.

Sabemos que el sistema de reserva de turnos on-line es muy utilizado en el exterior, principalmente en Europa, donde países como España poseen sistemas de excelentísima calidad y cumpliendo las funciones que esperamos abarcar con Doc-Net. Al margen de lo dicho anteriormente, estos sistemas prácticamente no brindan al médico la posibilidad de mantener una historia clínica on-line, el cual es uno de los puntos mas fuertes de nuestro sistema. Ya que no solo brinda comodidad al medico sino que facilita las interconsultas y las migraciones de un medico a otro.

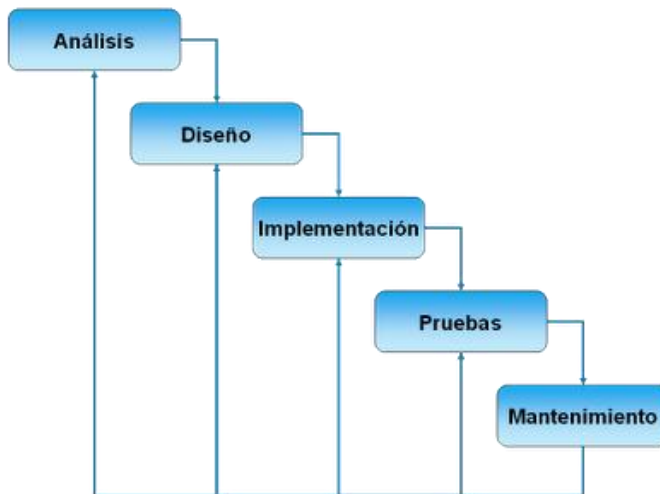
Debido a todo lo que se menciona en los párrafos anteriores, vemos viable la creación del sistema planteado, creyendo que el mismo servirá para mejorar los servicios ya existentes.

### **3. Metodología:**

Para desarrollar un sistema de información, es primordial definir el ciclo de vida que lo acompañará. Cuando hablamos de ciclo de vida, nos referimos a el modelo de actividades que se realizarán para la creación y el desarrollo de un sistema, con el orden en el que ocurrirán y la forma en la que estarán conectadas. Para el desarrollo de este tema nos hemos basado en [1]

El ciclo de vida elegido para el proyecto corresponde al modelo de cascada (Ilustración 2), donde se diferencian etapas marcadas y definidas, y donde un cambio en una de las etapas inferiores, resulta en una retroalimentación obligada hacia los niveles superiores/anteriores, para poder implementarla. Decidimos que en este tipo de proyecto era el que mejor se adecuaba a las necesidades, ya que nos permite visualizar y ordenar las ideas en un proceso determinado y ordenado, evitando confusiones o errores de diseño que serían aun más costosos que regresar etapas en el desarrollo.

De esta forma, en caso de detectarse un error en una etapa avanzada, se regresara hacia la etapa inicial, viendo como afecta en la misma dicho error y corrigiéndolo, así hasta regresar a la etapa donde lo habíamos hallado originalmente, y detallando en ella una solución final que ya ha sido respaldada en los niveles anteriores.



2 - Diseño en cascada

El proceso se llevó a cabo en el siguiente orden:

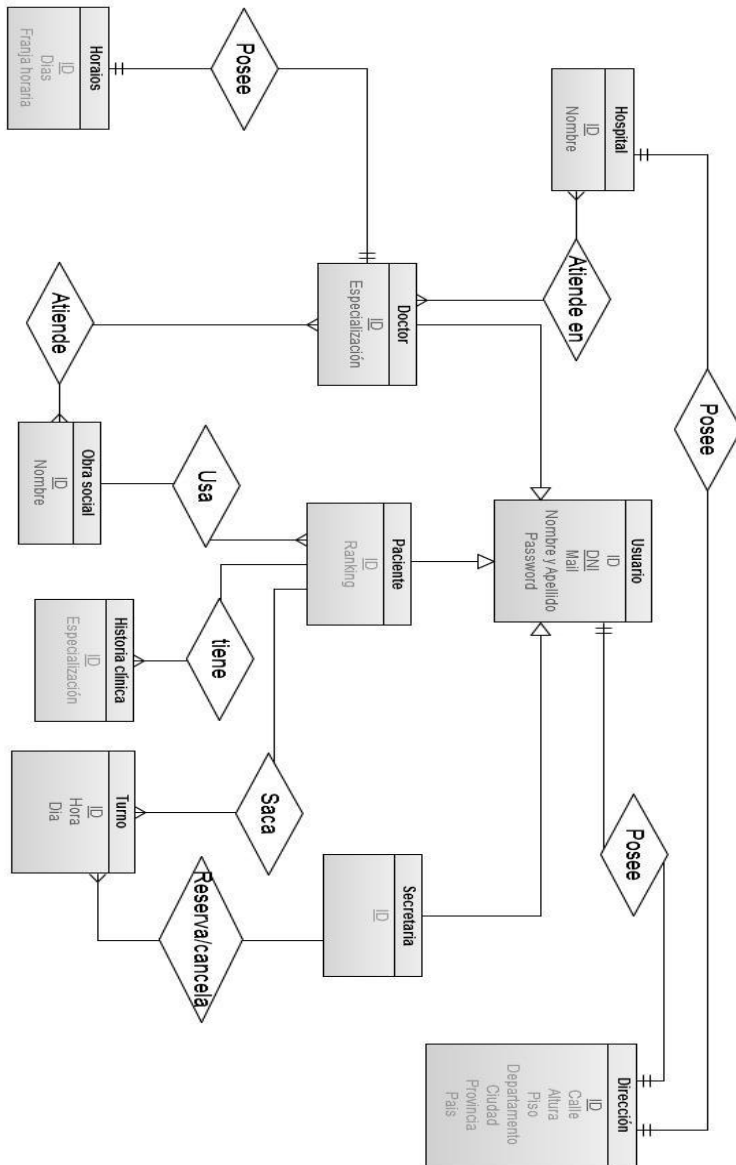
**a. Entrevistas:**

Se realizaron entrevistas para poder identificar las necesidades principales del proceso en cuestión, buscando delimitar los procesos principales y los que podían darle al sistema un valor agregado.

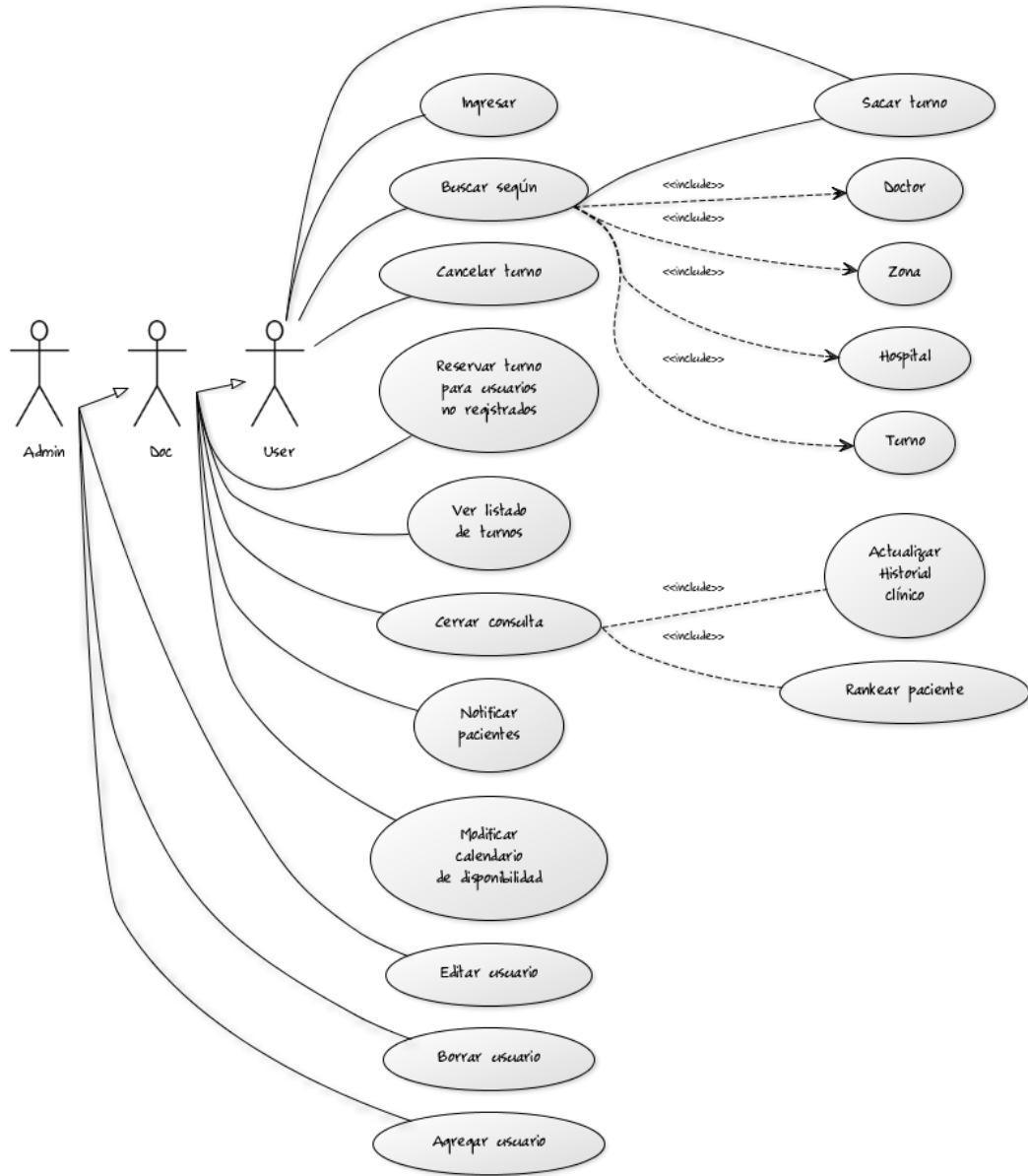
Durante las mismas, se abarcó un espectro amplio de personas, que fue desde profesionales del medio hasta pacientes particulares y de obra social. Afirmando así las funcionalidades que un usuario espera del sistema.

**b. Diagrama entidad-relación y UML:**

Se generó un diagrama UML (Ilustración 4) [2] [3] de casos de uso (que permitió delimitar los procesos) y uno de Entidad-Relación (Ilustración 3) [4] [5] (que delimita los campos que posee cada entidad).



3 - Diagrama Entidad-Relación



4 - Diagrama de casos de uso

### **c. Codificación:**

Para la codificación del proyecto, se decidió trabajar con un lenguaje orientado a objetos (Java, Jdk 7 [6]). Este lenguaje se utiliza tanto en el modelo, es decir, la codificación de las entidades con sus respectivos atributos, como en el control. Utilizamos una base de datos para persistir lo necesario de dichas entidades y poder operar sobre las mismas. Para facilitar la programación, asegurar la consistencia y generar documentación, utilizamos el entorno de programación IntelliJIDEA 10.5 [7]

Se eligió la codificación en un lenguaje orientado a objetos ya que es un sistema donde la reutilización de código es primordial, ya que gran parte de las acciones están entrelazadas y son similares. Además, los objetos permiten una programación más realista, donde se refleja la realidad del proceso con gran precisión.

Por otro lado, la programación orientada a objetos permite paralelizar el trabajo, permitiendo que el tiempo de trabajo sea más eficiente, y optimizando los resultados. Esto es extremadamente útil cuando se trabaja en equipo.

Finalmente, una razón determinante fue que este lenguaje permite mejorar el sistema y ampliarlo sin necesidad de modificarlo completamente, sino trabajando solo las áreas que se quieren optimizar o reescalar.

### **d. Testeo:**

Para comprobar el correcto funcionamiento, se realizaron pruebas de escritorio (unitarias y generales [8]), generando usuarios y simulando todas las interacciones posibles entre los mismos.

Para ello se utiliza un servidor local, mediante el sistema Tomcat [9], de Apache. De esta forma, se puede simular un servidor en el ordenador, permitiendo un testeo dinámico, donde los cambios realizados en el programa se pueden aplicar directamente sin necesidad de realizar *uploads* en un servidor web, agilizando enormemente el proceso de testeo y *debugging* del sistema.

Se utilizó también una base de datos local creada con HSQLDB [10], que de la misma forma que el servidor, se guarda y se ejecuta en el ordenador, pero posee las mismas funcionalidades que una base de datos que se encuentra en el servidor.

### **e. Documentación:**

Se realizó una documentación de cada clase del proyecto con sus respectivos métodos explicados.

El sistema IntelliJIdea 10.5, utilizado para realizar el proyecto, posee un sistema de documentación por clase, que puede estilizarse de forma cómoda para el desarrollador, y que más tarde permite imprimir o volcar a un archivo, sin el código.



Así se genera documentación didáctica, sencilla de leer y organizada por clase, sin necesidad de mantener un archivo paralelo.

Este sistema también despliega la documentación escrita dentro del ámbito de programación, permitiendo que cualquier programador lea la documentación de una clase rápidamente e in-situ.

#### **4. Propuesta de solución:**

Se propone una aplicación web en donde el paciente puede buscar turnos médicos según:

- Especialidad
- Obra Social
- Zona de atención
- Hospital
- Un médico en particular

Una vez reservado el turno de atención mediante un calendario interactivo, el médico es notificado con la fecha y horario de atención, los cuales puede revisar desde su panel de control, para cancelarlos y notificar al paciente en caso de que surgiese un imprevisto. De esta manera, ambos estarán informados sobre la cita.

Durante la consulta, se puede cargar o modificar la historia clínica del paciente. Una vez finalizada la consulta, el historial queda guardado y el médico confirma el cumplimiento del turno reservado para la calificación del cumplimiento de la asistencia del paciente. Marcando como no cumplidos los turnos anteriores no cerrados, modificando el ranking de los pacientes incumplidores. Cabe destacar que la historia clínica del paciente se divide según especialización, es decir que si el paciente cambiara de médico por alguna razón, el nuevo podría acceder a las entradas antiguas, evitando complicaciones.

Por otra parte, el sistema permite al médico comunicar en tiempo real a sus pacientes cuándo una franja horaria que normalmente está disponible, no lo está por alguna razón.

En caso de que algún usuario no se volcase al sistema informatizado de reserva, el doctor, o su secretaria, podrán reservar en el sistema turnos para usuarios no registrados, permitiéndole guardar historias clínicas y turnos en el sistema aunque el paciente no lo utilice por sí mismo.

#### **5. Bases tecnológicas del proyecto:**

El proyecto incluyó varias tecnologías en el proceso de desarrollo:

- La base de datos que guarda todos los registros se construyó a través de **HSQLDB** [10] e **Hibernate** [11], basándose en un modelo desarrollado en Java.

*Hibernate* es una librería de java que permite definir las entidades de la base de datos de forma sencilla, mediante anotaciones en los atributos de las clases que se desea persistir. *HyperSLQDataBase* es una base de datos diseñada íntegramente en Java, que posee la misma funcionalidad que *MySQL*, de manera *open source*.

- En cuanto a la parte visual, el *front-end* utilizó **JavaScript**, **HTML5** y **CSS**.
- El calendario interactivo, particularmente, también utilizó **Jquery** y **Json** para su desarrollo. En este caso tomamos un *Jquery week* [12] calendar *open source*, y lo adaptamos para nuestras necesidades.

## 6. Mejoras y nuevas versiones:

Una modificación a corto plazo incluirá la posibilidad de que los pacientes indiquen más de una obra social, aunque cabe destacar que el mismo tiene la opción de filtrar o no los doctores por obra social en caso de que quisiera realizar una consulta particular.

Inicialmente, el sistema busca abarcar a los médicos particulares, con posibilidad de ampliarse a clínicas u hospitales. En el futuro, el sistema podría abarcar las cartillas médicas de las obras sociales.

Un gran avance en cuanto a la expansión del proyecto es el plan de incorporar al mismo la posibilidad de emitir recetas, archivando en línea junto con la historia clínica del paciente, los medicamentos que le sean recetados. Esta modificación vendría acompañada de un vademécum farmacológico on-line que los médicos podrán consultar y agregar desde allí los medicamentos a la receta.

## 7. Bibliografía:

- [1] R. Schach Stephen, Ingeniería del software orientado a objetos.
- [2] I. Jacobson, G. Booch y J. Rambough, The Unified Software Development Process., Addison Wesley, 1999.
- [3] S. W. Ambler, The Elements of UML(TM) 2.0 Style..
- [4] R. Pressman, Ingeniería del Software. Un Enfoque Práctico. Cuarta Edición. 581 páginas., Mc Graw-Hill, 1998.
- [5] A. Mendelzon y J. Ale, Introducción a las Bases de Datos Relacionales.
- [6] S. Microsystems, «JDK 7 Project,» [En línea]. Available: <http://jdk7.java.net/>.
- [7] « IntelliJ IDEA 10.5,» [En línea]. Available: <http://www.jetbrains.com/idea/download/>.
- [8] E. Yourdon, Análisis Estructurado Moderno., Prentice-Hall Hispanoamericana, 1993.
- [9] «Tomcat en Wikipedia,» [En línea]. Available: <http://es.wikipedia.org/wiki/Tomcat>.
- [10] «HyperSQL,» [En línea]. Available: <http://hsqldb.org/>.
- [11] «Hibernate,» [En línea]. Available: <http://www.hibernate.org/>.
- [12] «Jquery week calendar,» [En línea]. Available: <https://github.com/robmonie/jquery-week-calendar/wiki/>.