

# YesDoc: Asistente médico personal

*Universidad Tecnológica Nacional, Facultad Regional Mendoza*

## Abstract

Este trabajo presenta a *YesDoc*, una aplicación de salud cuyo objetivo principal es empoderar a las personas en el cuidado de su salud, generando un rol participativo y de compromiso. *YesDoc* mejora la atención médica, tanto en un plano de atención como de prevención, cambiando el paradigma actual donde la información la gestiona el médico, por uno en que el usuario es el propietario capaz de gestionarla según sus necesidades. *YesDoc* unifica los datos médicos, situando al usuario como eje principal. Permite llevar la información a donde más la necesite y disminuye los tiempos inter-consulta, ya que ofrece una comunicación fluida médico-paciente a través de su plataforma. Brinda la unificación, centralización y disposición de información relacionada con el bienestar y la salud, sin importar la ubicación. Permite gestionar las autorizaciones hacia terceros expertos o responsables/comprometidos, y brinda la seguridad de que esta información no será manipulada por una persona fuera del entorno de confianza. Permite el ingreso ágil, simple e intuitivo, y la presentación resumida, siempre comprensible, de diversos tipos de datos como mediciones, estudios e imágenes entre otros. El trabajo consiste en el desarrollo de una aplicación web puesta a disposición de cualquier persona, diseñada para presentar una vista óptima sobre un amplio rango de dispositivos electrónicos tales como tablets, notebooks, computadoras personales y dispositivos móviles. Esta aplicación consume servicios web dispuestos como recursos, tanto para *YesDoc* como para cualquier otra aplicación de salud que requiera su uso. Para garantizar la interoperabilidad con otros sistemas de salud, se tienen en cuenta los estándares informáticos mundialmente utilizados en el campo de la salud.

## Palabras Clave

salud, registro personal de salud, phr, carpeta de salud, paciente, api, rest, responsive.

## Introducción

La idea de este trabajo surgió de los problemas que genera tener información médica en un formato físico, como puede ser papel, pa-

ra diferentes estudios o imágenes, ya que la probabilidad de extraviar esta información es alta. El traslado de los mismos es incómodo, y ocurre que en muchas de las consultas en la que es útil contar con los estudios médicos realizados previamente, el paciente no ha concurrido con éstos. Relevando la situación actual identificamos que el proceso de interacción paciente-médico que plantea la estructura de salud actual, fomenta consultas innecesarias. Esto provoca que los médicos tengan agendas cargadas, pero poco productivas.

Al investigar profundamente acerca de los sistemas de salud actuales, nos encontramos con que en la provincia de Mendoza no existen desarrollos acordes a las necesidades del usuario y, sobre todo, adecuados a los estándares de salud, siendo incapaces de permitir la interoperabilidad.

En cuanto a lo que refiere al aspecto legal de la salud en nuestro país, no existen aún leyes que contemplen los problemas previamente enunciados, aunque se está trabajando en una reglamentación que acompañe la digitalización de la información de salud en Argentina. Concluimos que la estructura y tecnología presentes no cubren la necesidad tanto de médicos como pacientes de contar con datos de salud históricos, unificados, completos y disponibles de acuerdo con las facilidades que brindan las tecnologías. Las leyes, tanto en nuestro país como en el mundo, no han logrado abordar estos conflictos. Por lo tanto, es imperioso encararlos desde otra perspectiva: la del **paciente**, quien es el principal interesado en su salud.

Un ejemplo de solución a este problema es la conocida historia clínica universal. El Reino Unido plantea, como un plan del departamento de salud denominado “Cuidado y Salud personalizado”, lograr la integración de los registros de paciente para 2020. Es-

pañá, por otro lado, había planteado hace 9 años implantar una historia clínica electrónica universal incorporada en la tarjeta sanitaria para todos los usuarios del Sistema Nacional de Salud.

Sin embargo, lo que ocurre es que hoy estas soluciones se encuentran con el problema de la integración y las trabas del sistema legal de salud de los propios países. Para enfrentar este problema y lograr un resumen histórico de salud de la persona, se encara la situación desde otra perspectiva, centrándonos en el paciente, dejando así de lado la burocracia de instituciones médicas y de los mismos médicos, y creando una herramienta que le permita al paciente controlar su salud, que lo empodere y fomente su participación activa. Sirve además a los médicos, ya que pueden consultarla con los permisos correspondientes, y a las instituciones médicas para mostrar al paciente los datos médicos de su propiedad.

Como se observa, la tendencia es hacia la integración, pero el problema es que los sistemas no han tenido esto en cuenta. Por esto, se desarrolla una aplicación fiel a los estándares mundialmente utilizados, permitiendo así la interoperabilidad con sistemas actuales, y con sistemas futuros que cumplan con los mismos.

Además de esta característica primordial, el sistema se basa también en otros cuatro pilares fundamentales:

#### 1. Disponibilidad total

*YesDoc* brinda a toda persona la unificación y centralización de la totalidad de su información relacionada con el bienestar y la salud, poniéndola a disposición de la misma y de terceros autorizados por ella en cualquier lugar del mundo. El paciente concurrirá a la institución de salud solo en los casos en los que actualmente se considera necesario, se evitarán consultas solo para llevar resultados, también que el paciente repita estudios de forma innecesaria.

#### 2. Universalidad

*YesDoc* garantiza que, sin importar la institución a la que concurras para ser atendido, la información sobre la salud del paciente sigue una línea temporal precisa. Las instituciones médicas no solicitarán reiterativamente sus datos personales.

#### 3. Interacción

*YesDoc* establece un nuevo paradigma de comunicación paciente-médico, que lleva el control del lado del paciente en una primera etapa de comunicación y luego deja en manos del médico la decisión sobre como proceder con la misma. Además permite establecer relaciones de confianza con personas allegadas o a cargo.

#### 4. Confidencialidad

*YesDoc* brinda la discreción y seguridad necesarias para que la información no pueda ser accedida por personas o empresas que no correspondan.

### Elementos del Trabajo y metodología

Para llevar adelante la aplicación será necesario utilizar:

**API REST**, “Application Programming Interface, REpresentational State Transfer”, es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP (“HyperText Transfer Protocol”). REST permite crear servicios y aplicaciones que pueden ser usadas por cualquier dispositivo o cliente que entienda HTTP, brindando la posibilidad de que otras aplicaciones puedan comunicarse con el sistema. Para su diseño debemos tener en cuenta que lo que presenta una API REST a las aplicaciones son **recursos**, éstos pueden ser utilizados por la aplicación a través de **URIs** (“Identificador de Recurso Único”), esto es, las URIs representan los medios por los cuales podemos llegar a un determinado recurso desde un navegador. Por otro lado tenemos las **representaciones**, éstas determinan como manipular los recursos en una arquitectura API RESTful,

en general representan parte del estado del recurso que es transferido entre cliente y servidor. Por último, nuestra API RESTful presenta **servicios**, estos indican que es lo que puede obtener una aplicación de nuestros recursos haciendo uso de los métodos que define el protocolo HTTP. Entonces, a modo de resumen y ejemplo, nuestra API presenta un recurso, digamos una persona, a través del método GET de HTTP brinda el servicio de entregar información de la persona como puede ser el nombre, dirección, edad entre otras, contenida en determinada representación, digamos JSON (“JavaScript Object Notation”). Para que una API sea considerada REST debe cumplir 6 restricciones [20]:

1. Interfaz uniforme

Usar los verbos HTTP como las acciones que vamos a tomar sobre los recursos.

2. Sin estado Cada mensaje entre el cliente y el servidor es completamente descriptivo lo que hace que no sea necesario almacenar información entre mensajes.

3. Cliente-Servidor

Se asume un funcionamiento en una arquitectura de sistemas desconectados.

4. Cacheable

Las respuestas del servidor pueden ser almacenadas en una memoria temporal intermedia de forma explícita, implícita, o previa negociación.

5. Sistema en capas

Existencia de software o hardware intermedio entre el cliente y el servidor.

6. Código bajo demanda

Esta restricción, a diferencia de las primeras cinco, es opcional y establece que parte de la lógica alojada en el servidor puede extenderse al cliente para que éste sea quien la ejecute.

**La arquitectura MVC**, (modelo–vista–controlador) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello se propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Esto presenta ventajas en la reutilización de código y la separación de conceptos, lo que facilita tanto el desarrollo como el mantenimiento de las aplicaciones.[18]

**Open Source**, es el término con el que se conoce al software distribuido y desarrollado libremente. Nuestro proyecto se adapta a esta corriente de desarrollo para aprovechar algunas de sus ventajas como:

- Mejora de la calidad y la innovación.
- Reducción de costos mediante el ahorro del pago de licencias.
- Reducción de dependencias de fabricantes de software.

Entre las dependencias Open Source que utilizamos podemos destacar a AngularJS, Plantuml [[wiki:plantUML](#) ], GIT y Flask. Además nuestro código se encuentra disponible en GitHub.com bajo el nombre de la organización **YesDoc**.

**GIT como herramienta de control de versionado** que nos permite el mantenimiento de versiones del código de forma eficiente y confiable, permitiéndonos llevar un registro histórico de los cambios realizados

**UML**, “Unified Modeling Language” es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema, este lenguaje nos guiará a lo largo del desarrollo para poder comprender el funcionamiento del sistema desde distintas perspectivas y poder documentarlo en su totalidad. Para poder exportar UML utilizaremos los siguiente dia-

gramas que nos muestran diferentes aspectos.

- Diagrama de secuencia: muestra las interacciones entre los actores y el sistema, y entre los componentes del sistema.
- Diagrama de clase: muestra las distintas clases en el sistema y su relación.

**Metodología ágil**, envuelve un nuevo enfoque para la toma de decisiones en los proyectos de software, refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto, así el trabajo es realizado mediante la colaboración de equipos auto-organizados y multidisciplinarios, inmersos en un proceso de toma de decisiones a corto plazo compartido.[4]

Dado que YesDoc es un proyecto que abarca un espectro amplio de usuarios con necesidades diferentes y cambiantes, se utiliza esta metodología flexible e iterativa.

Dentro del abanico de marcos de desarrollo ágiles, utilizamos Scrum.

Este se encuentra caracterizado por adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto; por basar la calidad del resultado más en el conocimiento tácito de las personas en equipos autoorganizados, que en la calidad de los procesos empleados; y por el solapamiento de las diferentes fases del desarrollo, en lugar de realizar una tras otra en un ciclo secuencial o de cascada.[21]

## Resultados

Como resultado del trabajo de relevamiento hemos producido una lista completa de **historias de usuario** según indica la metodología que usamos para guiar el desarrollo.

A partir del marco brindado por las historias de usuario se ha producido una evolución iterativa de un **diagrama de clases** que representa a todas las entidades que participan del sistema *YesDoc*. Aunque, destacamos que dicho diagrama aún puede cambiar.

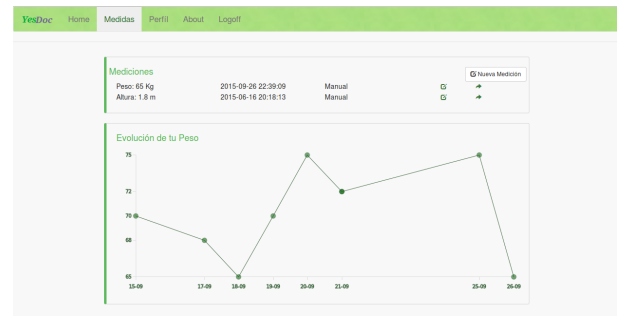


Figura 1: Interfaz de usuario

Como mencionamos en la sección anterior, la aplicación web se encuentra versionada en un repositorio de GitHub, ubicado en la dirección <https://github.com/yesdoc/web>. Éste presenta, en su rama principal la versión estable del proyecto. Para poder probar el proyecto, la aplicación web se encuentra desplegada en Heroku, una plataforma que brinda servicios de computación en la nube. Puede ser accedida desde la dirección <http://yesdoc.herokuapp.com>.

La versión estable del desarrollo de la API REST se encuentra versionada en la rama principal de otro repositorio, también alojado en GitHub, en la dirección <https://github.com/yesdoc/api>. La documentación completa de la API puede consultarse en <https://yesdoc-api.herokuapp.com/api/spec.html#!/spec>.

En la **Figura 1** se muestra la interfaz de la aplicación, que permite observar las últimas mediciones y la gráfica de evolución del peso de un paciente.

## Discusión

Nuestro trabajo se somete a discusión pública a través de la liberación de código en GitHub, para que todos aquellos interesados en el proyecto puedan realizar su aporte y transmitir sus inquietudes. Uno de los mayores problemas a afrontar es la de conseguir universalizar los tipos de mediciones ya que actualmente no existe un estándar universal seguido por todas las instituciones, para resolver este problema fue necesario implemen-

tar el sistema de una forma totalmente flexible, capaz de adaptarse a los cambios y a las necesidades de cada una de las instituciones responsables de realizar la carga de los análisis.

El mayor impacto en la comunidad que ofrece YesDoc es la posibilidad de recibir los resultados de los análisis de cualquier institución en su propia plataforma.

## Conclusión

YesDoc busca facilitarle la vida a todas aquellas personas que forman parte de las actividades relacionadas al cuidado de su salud, permitiéndole una comunicación fluida y cercana medico-paciente, un control detallado de su historial médico y principalmente la posibilidad de poder recibir los resultado de sus análisis en su cuenta de YesDoc desde su celular o su computadora.

## Referencias

- [1] Hospital Italiano de Bs As. *Historia Clínica Electrónica*. 2007. URL: <https://www.hospitalitaliano.org.ar>.
- [2] Ministerio de Salud. *Ley 26.529 (Texto reducido) - Derechos del Paciente en su relación con los Profesionales e Instituciones de la salud*. 2009. URL: <http://www.ms.gba.gov.ar/sitios/srpr/files/2014/09/Derecho-paciente-profesion.pdf>.
- [3] HL7. *Producto CCD — HL7*. Internet; descargado 10-julio-2015. 2010. URL: [http://wiki.hl7.org/index.php?title=Product\\_CCD](http://wiki.hl7.org/index.php?title=Product_CCD).
- [4] Intelligence to Business. *5 beneficios de aplicar metodologías ágiles, Intelligence to Business*. [Internet; descargado 19-septiembre-2015]. 2015. URL: <http://www.i2btech.com/blog-i2b/tech-deployment/5-beneficios-de-aplicar-metodologias-agiles-en-el-desarrollo-de-software/%7D>.
- [5] Open Health. *Plataforma libre para el desarrollo de sistemas de salud*. 2015. URL: <http://www.openehr.org/>.
- [6] Microsoft. *HealthVault*. Internet; descargado 9-mayo-2015. 2015. URL: <https://www.healthvault.com/ar/es>.
- [7] OpenEHR. *Modelo de referencia para la interoperabilidad y computabilidad de e-health*. 2015. URL: [http://www.openehr.org/what\\_is\\_openehr](http://www.openehr.org/what_is_openehr).
- [8] PlantUML. 2015. URL: <http://plantuml.com/%7D>.
- [9] Wikipedia. *Carpeta de Salud — Wikipedia, La enciclopedia libre*. Internet; descargado 9-mayo-2015. 2015. URL: [http://es.wikipedia.org/w/index.php?title=Carpeta\\_de\\_Salud&oldid=82184105](http://es.wikipedia.org/w/index.php?title=Carpeta_de_Salud&oldid=82184105).
- [10] Wikipedia. *CIE-10*. 2015. URL: <https://es.wikipedia.org/wiki/CIE-10>.
- [11] Wikipedia. *Clasificación internacional de atención primaria*. 2015. URL: [https://es.wikipedia.org/wiki/Clasificaci%C3%83%C2%B3n\\_Internacional\\_de\\_Atenci%C3%83%C2%B3n\\_Primaria](https://es.wikipedia.org/wiki/Clasificaci%C3%83%C2%B3n_Internacional_de_Atenci%C3%83%C2%B3n_Primaria).
- [12] Wikipedia. *Clinical Document Architecture*. 2015. URL: [https://en.wikipedia.org/wiki/Clinical\\_Document\\_Architecture](https://en.wikipedia.org/wiki/Clinical_Document_Architecture).
- [13] Wikipedia. *Código ATC*. 2015. URL: [https://es.wikipedia.org/wiki/C%C3%83%C2%B3digo\\_ATC](https://es.wikipedia.org/wiki/C%C3%83%C2%B3digo_ATC).

- [14] Wikipedia. *Computerized physician order entry*. 2015. URL: [https://en.wikipedia.org/wiki/Computerized\\_physician\\_order\\_entry](https://en.wikipedia.org/wiki/Computerized_physician_order_entry).
- [15] Wikipedia. *Continuidad del Registro del Cuidado*. 2015. URL: [https://es.wikipedia.org/wiki/Continuidad\\_del\\_Registro\\_del\\_Cuidado](https://es.wikipedia.org/wiki/Continuidad_del_Registro_del_Cuidado).
- [16] Wikipedia. *Historia médica*. 2015. URL: [https://en.wikipedia.org/wiki/Medical\\_history](https://en.wikipedia.org/wiki/Medical_history).
- [17] Wikipedia. *HL7 v3*. 2015. URL: [https://es.wikipedia.org/wiki/HL7\\_v3](https://es.wikipedia.org/wiki/HL7_v3).
- [18] Wikipedia. *Modelo-vista-controlador* — Wikipedia, La enciclopedia libre. [Internet; descargado 19-septiembre-2015]. 2015. URL: <https://es.wikipedia.org/w/index.php?title=Modelo%3%A2%C2%80%C2%93vista%3%A2%C2%80%C2%93controlador&oldid=85182091%7D>.
- [19] Wikipedia. *Registro de salud electrónico*. 2015. URL: [https://en.wikipedia.org/wiki/Electronic\\_health\\_record](https://en.wikipedia.org/wiki/Electronic_health_record).
- [20] Wikipedia. *Representational state transfer* — Wikipedia, The Free Encyclopedia. [Online; accessed 19-September-2015]. 2015. URL: [https://en.wikipedia.org/w/index.php?title=Representational\\_state\\_transfer&oldid=681289038%7D](https://en.wikipedia.org/w/index.php?title=Representational_state_transfer&oldid=681289038%7D).
- [21] Wikipedia. *Scrum* — Wikipedia, La enciclopedia libre. [Internet; descargado 19-septiembre-2015]. 2015. URL: <https://es.wikipedia.org/w/index.php?title=Scrum&oldid=85036921%7D>.