



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

IMPLEMENTACIÓN DE SOFTWARE PARA LA DETECCIÓN DE DAÑO RENAL AGUDO (AKI)

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN
COMPUTACIÓN

WALDO AARON URIBE FACHE

PROFESORA GUÍA:
MARÍA CECILIA BASTARRICA PIÑEYRO

MIEMBROS DE LA COMISIÓN:
PATRICIO INOSTROZA FAJARDIN
JOSÉ ALBERTO PINO URTUBIA

SANTIAGO DE CHILE

2016

RESUMEN DE LA MEMORIA PARA OPTAR AL

Título de: Ingeniero Civil en Computación

Por: Waldo Aaron Uribe Fache

Fecha: 18 de Abril, 2016

Profesora guía: María Cecilia Bastarrica Piñeyro

El Daño Renal Agudo consiste en una pérdida rápida de la función renal debido a un daño en los riñones. Dependiendo de su gravedad, se divide en 3 etapas. Es problema es común, serio y costoso en pacientes hospitalizados. Su detección temprana es fundamental para un óptimo tratamiento del paciente, pero en la práctica más del 80% de los casos no están siendo detectados, pese a que existen los exámenes para hacerlo.

En Chile, el Daño Renal Agudo nunca había sido caracterizado, no se tenía información sobre su incidencia.

En este trabajo se desarrolló un software, que permite la detección y caracterización del Daño Renal Agudo en el Hospital Clínico de la Universidad de Chile.

El software analizó exámenes históricos y detectó que el 6.02% de los pacientes analizados tuvo al menos un episodio de Daño Renal Agudo. De los pacientes diagnosticados, el 62.3% presentó al menos un episodio de Daño Renal Agudo en Etapa 1, 13.1% Etapa 2 y 24.6% Etapa 3.

Luego de obtener diagnósticos con el software, se compararon con aquellos registrados en las epicrisis del hospital, documento resumen de la hospitalización de un paciente. Se constató que el 83% de los pacientes no habían sido correctamente diagnosticados.

A raíz de este trabajo, se construyó una aplicación web capaz de analizar los exámenes de otros centros médicos. Actualmente se está trabajando en el levantamiento de fondos concursables del Gobierno de Chile, para dedicar tiempo completo a la implementación de esta nueva aplicación idealmente en todo Chile.

Los resultados obtenidos en este proyecto han sido expuestos en el International Student Congress of (bio)Medical Sciences ISCOMS (Holanda), en el XXXVI Congreso Chileno de Medicina Interna (Chile), y en el Congreso Conjunto de Nefrología e Hipertensión (Chile), ganando el premio al mejor poster en este último.

1 TABLA DE CONTENIDO

RESUMEN	2
1 TABLA DE CONTENIDO	3
1 INTRODUCCIÓN	4
1.1 CONTEXTO	4
1.2 PROBLEMA.....	5
1.3 SOLUCIÓN PROPUESTA.....	7
1.4 MOTIVACIÓN.....	8
1.5 OBJETIVOS.....	9
1.5.1 OBJETIVOS GENERALES.....	9
1.5.2 OBJETIVOS ESPECIFICOS.....	9
2 SOLUCIÓN.....	10
2.1 METODOLOGÍA	10
2.1.1 LEVANTAMIENTO GENERAL DE REQUISITOS.....	10
2.1.2 ESTUDIO DE FACTIBILIDAD.....	10
2.1.3 TOMA DETALLADA DE REQUERIMIENTOS.....	10
2.1.4 IMPLEMENTACIÓN.....	11
2.1.5 ANÁLISIS Y EXPOSICIÓN DE LOS RESULTADOS.....	11
2.2 PLANIFICACIÓN.....	11
3 TRABAJO REALIZADO.....	12
3.1 LEVANTAMIENTO GENERAL DE REQUISITOS	12
3.1.1 ESTUDIO DE FACTIBILIDAD.....	13
3.1.2 REQUISITOS DETALLADOS	17
3.1.3 SELECCIÓN DE TECNOLOGÍAS.....	18
3.1.4 DISEÑO DEL SISTEMA.....	22
3.1.5 PANTALLAS DE NAVEGACIÓN.....	27
3.1.6 ALGORITMO DE DETECCIÓN DE AKI.....	31
3.1.7 ESTUDIO DE USABILIDAD	35
4 VERIFICACIÓN Y VALIDACIÓN	49
4.1 VERIFICACIÓN.....	49
4.2 VALIDACIÓN.....	49
4.2.1 EXPERIMENTACIÓN.....	49
4.3 DISCUSIÓN DE LOS RESULTADOS.....	51
4.4 IMPLICANCIAS ÉTICAS	51
5 CONCLUSIONES.....	52
5.1 TRABAJO FUTURO	53
6 BIBLIOGRAFÍA	55

1 INTRODUCCIÓN

El objetivo de este trabajo fue implementar un software para la detección de Insuficiencia Renal Aguda¹ en el Hospital Clínico de la Universidad de Chile. A continuación se contextualizará el trabajo, explicando de qué se trata esta condición, cual es su gravedad, que problemas que se enfrentan en su diagnóstico y por qué es importante detectarla de manera temprana.

1.1 CONTEXTO

De acuerdo a cifras de la Organización Mundial de la Salud, la causa número uno de muerte a nivel mundial son las enfermedades cardiovasculares, produciendo casi 17,5 millones de muertes el 2012, que corresponden a 3 de cada 10 muertes¹. Por eso, resulta de vital importancia disminuir los factores de riesgo asociados a dichos episodios.

Dentro de los factores que aumentan los eventos cardiovasculares, se encuentra el Daño Renal Agudo o AKI, que consiste en una rápida pérdida de la función renal debido a una lesión en los riñones. De aquí se desprende la importancia para detectar de manera temprana la enfermedad. Además el daño renal agudo puede empeorar provocando daño renal crónico, que podría generar la necesidad de diálisis del paciente o incluso la muerte.

El diagnóstico de AKI se realiza a través de la medición de la concentración de creatinina en la sangre. La creatinina es un desecho que el metabolismo normal de los músculos genera de manera constante. Este desecho es filtrado por el riñón, y luego eliminado a través de la orina. Por lo tanto, la acumulación de creatinina, es evidencia de que el riñón no está funcionando correctamente.

Como consecuencia, el criterio AKINⁱⁱ establece que un alza en los niveles de creatinina de al menos 0.3 mg/dl en 48 horas o de un 50% en 7 días, determina la presencia de AKI.

La insuficiencia renal aguda, es un problema serio, costoso y común (afecta al 18% de los pacientes hospitalizados), por lo que su detección temprana es esencial para dar un manejo apropiado.

¹ Se hablará indistintamente de Daño Renal Agudo, Insuficiencia Renal Aguda, o AKI (Acute Kidney Injury)

En resumen:

- El Daño Renal Agudo es un mal funcionamiento de los riñones
- Se diagnostica midiendo la variación de la creatinina en la sangre
- Es un problema serio, común y costoso, por lo que su detección temprana es fundamental

1.2 PROBLEMA

Es común que el Daño Renal Agudo sea sub-diagnosticado pues su detección implica analizar varios exámenes, lo que toma bastante tiempo sin tener algún software de apoyo, y además requiere los conocimientos específicos de un nefrólogo.

Es el Jefe de Nefrología del Hospital Clínico de la Universidad de Chile quien señala que un alto porcentaje de los pacientes con Daño Renal Agudo no está siendo diagnosticado. Se describirá el contexto dónde se estudió el problema y porque ocurre.

En el Hospital Clínico de la Universidad de Chile (J. J. Aguirre), se realizan mediciones de creatinina a los pacientes al previo a alguna operación o cuando se les quiere administrar medicamentos. Su objetivo es conocer su función renal con el fin de evitar complicaciones relativas a su mal funcionamiento.

Sin embargo, según Rubén Torres, Jefe de la Unidad de Nefrología Intermedia del Hospital, sólo se presta atención a aquellos casos donde los niveles de creatinina son muy elevados, dejando el resto sin atender. Esto era es solo intuición, pues en Chile no existían datos acerca de la incidencia de la patología.

Es habitual que se diagnostique AKI si un paciente tiene creatinina superior a 2, ya que ese nivel se considera alto. Sin embargo, por definición, es necesario tener un examen basal para poder diagnosticar la enfermedad. Solo contrastando ambos exámenes se puede dar un diagnóstico. Se presenta un ejemplo para clarificar.

Ejemplo:

Dado un paciente con 2 exámenes de creatinina, donde el primero indica una densidad de 1.0 mg/dl de creatinina en la sangre y el segundo de 1.5 mg/dl.

Según el criterio AKIN, un alza en los niveles de creatinina de un 50% implica daño renal agudo, por lo tanto, este paciente debería ser diagnosticado. Sin embargo, es común que sólo se considere un examen al momento de evaluar, y como 1.5 mg/dl se considera bajo, no se diagnostica. Esto es un error pues siempre se debe considerar el nivel basal del paciente.

Por otro lado, en una persona sana, los niveles de creatinina son directamente proporcionales a la masa muscular, por lo tanto, es normal que una persona musculosa tenga altos niveles de creatinina, lo que no implica necesariamente que tiene problemas renales.

En un escenario real, el paciente tiene varios exámenes y es necesario tomar todos los pares que estén en un rango de 7 días, para ver si en alguno de ellos existe un aumento mayor al 50%.

El problema empeora cuando el paciente visita el hospital por un motivo distinto a su función renal. Por ejemplo, si es internado por una fractura, es atendido por un traumatólogo, quien ordena una batería de exámenes antes de operarlo. Como no está analizando específicamente su función renal y como el traumatólogo no es especialista en riñón, generalmente se pasa por alto.

1.3 SOLUCIÓN PROPUESTA

Se propone entonces, implementar un software que analice la base de datos del Hospital, con el fin de diagnosticar a aquellos pacientes que padecen de Insuficiencia Renal Aguda.

Una herramienta que analice los exámenes y sepa cuales son los criterios para detectar el Daño Renal Agudo, agilizaría el proceso de diagnóstico.

Dada la naturaleza del proyecto, que requiere de conocimientos médicos, acceso a datos sensibles e integración con el sistema informático propio del Hospital, las siguientes personas colaboraron con su desarrollo.

Rubén Torres: Médico Nefrólogo Internista, Jefe de la Unidad Nefrológica Intermedia del Hospital Clínico de la Universidad de Chile y Profesor de la Escuela de Medicina de la Universidad de Chile.

Néstor Saavedra: Jefe de Sistemas Informáticos del Hospital Clínico de la Universidad de Chile.

María Eugenia Pinto: Jefa del Servicio de Laboratorio Clínico del Hospital Clínico de Universidad de Chile.

José Miguel Vicente: Estudiante de Medicina de la Universidad de Chile.

Este trabajo cuenta con la aprobación de la Unidad de Nefrología y del comité de ética del Hospital Clínico. Además, el Hospital ha destinado recursos para comprar el hardware necesario para su ejecución.

1.4 MOTIVACIÓN

En otros lugares del mundo se han construido herramientas de detección de Insuficiencia Renal Aguda. Técnicamente el hospital chileno cuenta con lo necesario para implementar un software de detección automática. Estudios similares, factibilidad técnica y un problema que afecta directamente la salud de los pacientes motivó el desarrollo de este trabajo.

En el Hospital de la Universidad de Nottingham, se implementó un sistema de alertas automáticas para la detección temprana de AKI. El campo hospitalario mencionado es similar hospital chileno, en cuanto ambos están asociados a una Universidad y proporcionan espacios para la investigación y desarrollo por parte de alumnos y cuerpo docente. Lo anterior sienta un precedente con respecto a la factibilidad del proyecto. Es necesario mencionar que este trabajo sólo contempla la recolección y análisis de datos, mas no el envío de alertas automáticas.

El hospital chileno busca una solución desacoplada de su sistema actual, que use solamente los datos existentes y no estimaciones. Además, es necesario generar los diagnósticos estudiando variaciones de exámenes en intervalos de 7 días como máximo.

El software de Nottingham está incluido dentro del sistema de administración del hospital. Si un paciente no tiene suficientes exámenes para generar un diagnóstico el software estima niveles de creatinina. Por otro lado el sistema de Nottingham le da validez a pares de exámenes que están muy distantes en el tiempo. Estas definiciones impidieron la implementación directa del software en el hospital chileno.

El hospital chileno cuenta con un sistema informático que permite a doctores almacenar datos de sus pacientes, exámenes y procedimientos ejecutados. Sin embargo el sistema es usado sólo para organizar datos y no se realizan análisis sobre ellos.

Sería de gran ayuda crear un software que sea capaz de generar un diagnóstico tan pronto los exámenes sean tomados en el hospital. Existe la capacidad técnica para realizar estos análisis, además los exámenes ya están siendo almacenados y son de disposición inmediata, solo falta implementar el software necesario para su análisis.

Mejorar el diagnóstico del Daño Renal Agudo, una enfermedad común asociada significativamente con la mortalidad, morbilidad e incremento de la estadía hospitalaria ⁱⁱⁱ, presenta un aporte para la salud en Chile.

1.5 OBJETIVOS

Este trabajo tuvo como objetivo caracterizar o bien obtener estadísticas sobre la incidencia del Daño Renal Agudo. Se detallarán de manera general y específica estos objetivos.

1.5.1 OBJETIVOS GENERALES

Obtener estadísticas sobre la incidencia del Daño Renal Agudo a través de la construcción de un software que procese la información de la base de datos del hospital. Además, se buscó contrastar los diagnósticos generados con el software con aquellos encontrados por los doctores.

1.5.2 OBJETIVOS ESPECIFICOS

- Descargar datos relevantes desde la base de datos del Hospital.
- Implementar un algoritmo que permita diagnosticar AKI.
- Obtener perfiles de pacientes que contengan sus diagnósticos.
- Generar las siguientes estadísticas relativas a la incidencia del Daño Renal Agudo.
 - Porcentaje de pacientes con Insuficiencia Renal Aguda por etapa.
 - Porcentaje de pacientes que empeoraron su condición, es decir, que pasaron a una etapa más grave.
 - Porcentaje de pacientes que mejoraron su condición.
- Contrastar los diagnósticos obtenidos de manera automática con aquellos documentados por los doctores en la epicrisis (documento resumen con los diagnósticos y tratamientos de los pacientes hospitalizados).

2 SOLUCIÓN

La solución consiste en diseñar e implementar un software que sea capaz de analizar los datos del hospital. Para este desarrollo, se utilizó la metodología descrita a continuación.

2.1 METODOLOGÍA

El desarrollo del proyecto se dividió en 5 etapas.

- 1 Levantamiento general de requisitos
- 2 Estudio de factibilidad
- 3 Toma detallada de requisitos
- 4 Implementación del software
- 5 Análisis y exposición de los resultados

Estas etapas se dividen de la siguiente manera.

2.1.1 LEVANTAMIENTO GENERAL DE REQUISITOS

En esta etapa se buscó entender el problema y establecer cuáles son los requerimientos necesarios para dar una solución. Se consideró el nivel mínimo de detalle necesario para poder evaluar la factibilidad del software.

2.1.2 ESTUDIO DE FACTIBILIDAD

Antes de continuar con la ejecución del proyecto, se verificó la factibilidad técnica y legal del software. Los temas más importantes dentro de esta etapa fueron la integración con la base de datos del hospital y restricciones de privacidad de datos médicos, ya que al ser de carácter sensible, es necesario conseguir permisos especiales para su manipulación.

2.1.3 TOMA DETALLADA DE REQUERIMIENTOS

Una vez validada la factibilidad, se levantaron de manera detallada los requerimientos del software.

2.1.4 IMPLEMENTACIÓN

En esta etapa se realizó la implementación del software a través de ciclos periódicos de desarrollo. Cada ciclo duró 2 semanas y tuvo como objetivo mostrar partes funcionales del software, que el cliente validó para continuar con el siguiente ciclo. Esta etapa se dividió en 2 sub-etapas, la primera de diseño, donde se realizó un bosquejo del sistema, para luego construir un diseño detallado del mismo. La segunda consistió principalmente en desarrollar el código que sustenta el software.

2.1.5 ANÁLISIS Y EXPOSICIÓN DE LOS RESULTADOS

Una vez terminado el software, se determinó el cumplimiento de los objetivos planteados inicialmente. También se analizaron las estadísticas arrojadas por el sistema para ver si efectivamente la falla renal aguda está siendo sub-diagnosticada.

2.2 PLANIFICACIÓN

Para realizar el trabajo, se designó tiempo a cada una de las etapas usando una carta Gantt. La siguiente figura muestra la duración de cada una de las etapas.

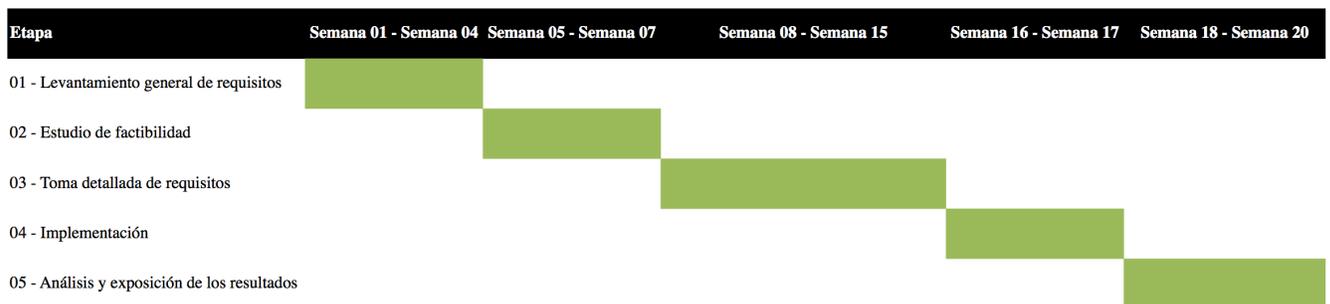


Figura 1: Carta Gantt

3 TRABAJO REALIZADO

Este capítulo describe el trabajo realizado en cada una de las etapas descritas en la carta Gantt.

3.1 LEVANTAMIENTO GENERAL DE REQUISITOS

A lo largo de reuniones con las personas involucradas en el proyecto, se establecieron de manera general requisitos para el sistema. En esta etapa se definieron usando lenguaje natural, para poder pasar rápidamente al estudio de factibilidad, sin perder tiempo en detalles que podrían ser irrelevantes en el caso de que el proyecto no sea factible.

Requisitos generales

1. Para acceder al sistema, se debe autenticar al usuario con su email y contraseña
2. Se debe poder cargar los datos de los exámenes, a partir de un archivo Excel. Estos datos contienen los resultados de un examen para un paciente en particular
3. Se debe mostrar un listado con todos los pacientes que tengan AKI
4. Debe ser posible buscar a un paciente en particular, para ver si tiene AKI y en qué etapa
5. Se debe poder ver todos los exámenes de creatinina realizados a un paciente en particular
6. Se debe poder ingresar de manera manual las epicrisis, documento que detalla los diagnósticos y procedimientos realizados a un paciente durante su estadía en el Hospital
7. Se debe poder contrastar diagnósticos de AKI dentro de las epicrisis, contra aquellos obtenidos de manera automática por el sistema
8. El sistema debe mostrar estadísticas con los siguientes datos:
 - a. Qué porcentaje de los pacientes tiene AKI
 - b. Qué porcentaje de los pacientes tiene AKI Etapa 1, Etapa 2 y Etapa 3
 - c. Mostrar el porcentaje de pacientes que presentó un cuadro de AKI, y figuraba con una enfermedad crónica en la epicrisis

3.1.1 ESTUDIO DE FACTIBILIDAD

El estudio de factibilidad se enfocó en 2 objetivos. El primero, determinar si era posible acceder a los exámenes de los pacientes, pues estos datos son propios del hospital y es necesario tener la aprobación del comité de ética para obtenerlos. El segundo, determinar si era técnicamente posible extraer los exámenes de la base de datos para su posterior análisis. A continuación se describe como se abordó cada objetivo.

3.1.1.1 APROBACIÓN DEL COMITÉ DE ÉTICA

Se envió una carta al comité de ética con un detalle del proyecto y señalando que estos datos serán de uso privado, es decir, no serán compartidos. En el caso de publicar, sólo se hará referencia a los resultados obtenidos sin incluir información que permita identificar a los pacientes.

El comité de ética autorizó la extracción de datos mediante la intermediación del Jefe de Sistemas Informáticos del Hospital, a quién se debe enviar una solicitud por email señalando los datos que se necesitan.

3.1.1.2 COMUNICACIÓN CON LA BASE DE DATOS DEL HOSPITAL

Al igual que varios softwares de Business Intelligence, para extraer datos los datos se usó un proceso de ETL (Extract, Transform, Load).

3.1.1.2.1 EXTRACT

Los datos solicitados al Departamento de Sistemas Informáticos son entregados en formato Excel. Por lo tanto, el proceso de extracción se limitó a solicitar dicho archivo.

Las filas del archivo hacen referencia a un examen. En la figura, se muestra una fila del archivo.

OT	RUT	NOMBRE	SERVICIO SOLICITANTE	FECHA SOLICITUD	FECHA VALIDACION	CREA VFG	OBSERVACIONES
2719547	1386991-K	EUGENIO SILVA MORENO	SERVICIO DE URGENCIA	17/01/2014 13:50:17.000	17/01/2014 14:53:06.370	1.3	55.09

Figura 2: Fila del archivo Excel

Para comprender la información contenida las filas, se detallará el contenido de cada una de las columnas.

Columna nº1: OT (Orden de Tratamiento)

Identificador único del examen

Columna nº2: RUT (Rol Único Tributario)

Identificador único del paciente

Columna nº3: Servicio Solicitante

Servicio o área del hospital que solicita la toma del examen

Columna nº4: Fecha Solicitud

Fecha en que se solicita la toma del examen

Columna nº4: Fecha Validación

Fecha en que se le toma el examen al paciente

Columna nº5: Crea (Creatinina)

Densidad de creatinina en la sangre del paciente. Medida en mg/dl.

Columna nº5: VFG (Velocidad de filtración glomerular)

Velocidad por la que pasa la sangre a través de los glomérulos. Medida en ml/min. Este dato no es relevante para este trabajo, pues este trabajo usa los niveles de creatinina para el diagnóstico de la enfermedad.

Columna nº6: Observaciones

Observaciones sobre el examen

Cabe destacar que estos datos son suficientes para diagnosticar insuficiencia renal aguda.

3.1.1.2.2 TRANSFORM

El proceso de transformación consiste en tomar cada una de las celdas del archivo Excel y transformar el dato contenido en ella en uno que el software pueda consumir.

Para el proceso de transformación, fue necesario modelar la estructura de datos que alojará los exámenes y crear una aplicación que formatee el archivo Excel, para que pueda ser insertado en la base de datos.

Para este efecto se implementó una base de datos postgresql y una aplicación en Ruby on Rails².

Al momento de analizar los datos entregados en archivos excel, se detectaron varios problemas que son detallados a continuación.

Problemas encontrados:

- Datos nulos en la base de datos del hospital figuraban como el string “[NULL]” en el Excel.
- Todos los exámenes de creatinina con nivel superior a 60, figuraban como el string “>60” en vez de un valor entero como era de esperar.
- Algunos de los valores señalados como obligatorios eran entregados vacíos.
- Los headers de las columnas en el archivo Excel, no son los mismos que los usados en la base de datos de nuestra aplicación.

Para solucionar estos problemas, se creó un método que va analizando cada una de las filas del Excel, para luego asignarles en el formato requerido por la base de datos. Es importante destacar la modularidad de cada una de las funciones involucradas en el proceso, dónde cada una tiene una responsabilidad clara. Esto fue fundamental para la detección de errores durante el proceso de formateo.

La función que formatea cada una de las filas del Excel se ve así:

```
def self.parse_row(row_hash)
  answer = parse_headers(row_hash)
  answer = parse_nulls(answer)
  answer = parse_blank_fields(answer)
  return parse_greater_than_60(answer)
end
```

² La decisión de usar estas tecnologías es discutida en la sección 3.1.3

Esta función permite transformar cada fila en algo que la base de datos puede consumir, dejando el paso libre a la siguiente etapa.

3.1.1.2.3 LOAD

La etapa de carga consiste en tomar los datos que fueron formateados en el paso anterior, para luego cargarlos en la aplicación.

Con la etapa de transformación resuelta el load es sencillo. Basta con leer y formatear las líneas del archivo Excel, para luego ingresarlas a la base de datos. Lo único importante es respetar el encoding de la base de datos, que en este caso es iso-8859-1: utf-8.

Las figura 3 muestra un extracto del archivo Excel proveniente del hospital y la figura 4 muestra su reflejo en la aplicación.

OT	RUT	NOMBRE	SERVICIO SOLICITANTE	FECHA SOLICITUD	FECHA VALIDACION	CREA	VFG	OBSERVACIONES
2719464	10849239-2	MARIA JOSE ANDONAEGUI ALVAREZ	SERVICIO DE URGENCIA	17/01/2014 12:22:22.000	17/01/2014 13:31:39.277	0.5	[NULL]	
2719547	1386991-K	EUGENIO SILVA MORENO	SERVICIO DE URGENCIA	17/01/2014 13:50:17.000	17/01/2014 14:53:06.370	1.3	55.09	
2719605	6399364-6	SANTIAGO AUGUSTO PAEZ MATURANA	SERVICIO DE URGENCIA	17/01/2014 15:51:23.000	17/01/2014 16:43:13.183	0.9	> 60	
2719666	14156022-0	JESSICA ALEJANDRA LEON MIRANDA	SERVICIO DE URGENCIA	17/01/2014 18:15:20.000	17/01/2014 19:33:22.040	0.6	> 60	
2719698	9251967-8	LUCIA EUGENIA CABRERA GALLEGUILLOS	SERVICIO DE URGENCIA	17/01/2014 20:25:16.000	17/01/2014 21:59:31.703	0.5	> 60	
2719707	6748957-8	ALFREDO SERGIO ARAYA VILLALOBOS	SERVICIO DE URGENCIA	17/01/2014 20:42:22.000	17/01/2014 22:00:05.897	0.8	> 60	
2719717	5721325-6	MARIA ISABEL QUINTEROS GUZMAN	SERVICIO DE URGENCIA	17/01/2014 20:58:25.000	17/01/2014 22:30:36.620	1.1	53.07	
2719730	4778700-9	REINALDO JOSE GUILLERMO BILLIG PAPPURELLO	SERVICIO DE URGENCIA	17/01/2014 21:48:35.000	17/01/2014 22:48:13.617	1.9	38.18	
2719735	5072873-0	TERESA CRISTINA GARCIA MEJIAS	SERVICIO DE URGENCIA	17/01/2014 21:54:02.000	17/01/2014 22:45:38.527	0.8	> 60	
2719736	2249772-3	GERARDO VALENZUELA SEPULVEDA	SERVICIO DE URGENCIA	17/01/2014 21:55:06.000	17/01/2014 22:45:15.770	1.7	40.7	
2719738	18468616-3	JEMIMA CESIA BELLO MORALES	SERVICIO DE URGENCIA	17/01/2014 22:12:11.000	17/01/2014 23:53:23.420	0.6	> 60	
2719755	5895333-4	ROSA DEL CARMEN GALLARDO VEAS	SERVICIO DE URGENCIA	17/01/2014 23:04:04.000	18/01/2014 1:06:07.870	0.3	> 60	
2719770	16388746-0	JORGE ANTONIO ALEGRIA ZULETA	SERVICIO DE URGENCIA	18/01/2014 0:03:32.000	18/01/2014 1:07:52.303	0.9	> 60	
2719792	9855655-9	SOLEDAD REYES REYES	SERVICIO DE URGENCIA	18/01/2014 1:54:24.000	18/01/2014 3:48:53.360	0.5	> 60	
2719795	3405864-4	JORGE JACINTO HILARION GALLEGUILLOS ALE	SERVICIO DE URGENCIA	18/01/2014 2:06:09.000	18/01/2014 3:46:04.487	3.3	19.4	
2719809	11373117-6	PALMENIA DEL CARMEN LARA NOVOA	SERVICIO DE URGENCIA	18/01/2014 4:06:49.000	18/01/2014 5:07:38.283	0.6	> 60	
2719814	11562273-0	RODRIGO GASPARD VALDES HENRIQUEZ	INTERMEDIO MEDICO	18/01/2014 4:41:00.000	18/01/2014 7:08:53.683	1	> 60	
2719815	13910426-9	HECTOR ANDRES MEDINA PINO	INTERMEDIO MEDICO	18/01/2014 4:42:00.000	18/01/2014 7:09:06.690	0.1	> 60	
2719816	7565597-5	FILOMENA DEL TRANSITO MORENO PIZARRO	INTERMEDIO MEDICO	18/01/2014 4:43:00.000	18/01/2014 7:08:08.383	0.7	> 60	
2719817	4746837-K	MARIA CRISTINA MOLINA ARREDONDO	INTERMEDIO MEDICO	18/01/2014 4:45:00.000	18/01/2014 7:07:47.133	0.5	> 60	
2719818	5523770-0	LEONOR DEL TRANSITO MIRANDA CORDOVA	INTERMEDIO MEDICO	18/01/2014 4:46:00.000	18/01/2014 7:07:04.573	0.8	> 60	
2719819	14287637-K	IVON ERNESTINA PEREZ CARRASCO	INTERMEDIO MEDICO	18/01/2014 4:48:00.000	18/01/2014 7:20:13.730	1	> 60	
2719820	3511435-1	ELBA MARIA FUENTES GONZALEZ	INTERMEDIO MEDICO	18/01/2014 4:50:00.000	18/01/2014 7:05:39.687	0.5	> 60	
2719824	8358678-8	ENRIQUE ANGEL RIQUELME TORRES	INTERMEDIO MEDICO	18/01/2014 4:52:00.000	18/01/2014 7:20:48.573	0.6	> 60	
2719826	7396285-4	JUAN ALBERTO ECHEVERRIA REYES	INTERMEDIO MEDICO	18/01/2014 4:54:00.000	18/01/2014 7:05:59.757	0.6	> 60	
2719827	8442876-0	JOSE DANIEL GUTIERREZ JARA	INTERMEDIO MEDICO	18/01/2014 4:56:00.000	18/01/2014 7:06:19.257	0.5	> 60	
2719828	20650984-8	ESCARLET CAMILA MONROY FUENTES	INTERMEDIO MEDICO	18/01/2014 4:57:00.000	18/01/2014 7:06:37.817	0.4	> 60	
2719858	4990441-K	ALEJANDRA DEL CARMEN MALDONADO LOBOS	MEDICINA FISICA Y REHABILITACION	18/01/2014 6:54:23.000	18/01/2014 8:52:15.923	0.9	> 60	
2719862	9971976-1	PAULA ALEJANDRA ELGUETA FUENZALIDA	SERVICIO DE URGENCIA	18/01/2014 7:34:03.000	18/01/2014 8:51:34.220	0.7	> 60	
2719744	17230448-6	JESUS RENE HERNANDEZ SILVA	CLINICA PSIQUIATRICA	18/01/2014 7:46:54.000	18/01/2014 9:35:24.227	0.9	> 60	
2719715	17530324-3	ANNETTE ALEJANDRA LA VIA BARRAZA	CLINICA PSIQUIATRICA	18/01/2014 7:50:09.000	18/01/2014 9:34:50.827	0.4	> 60	

Figura 3: Excel con exámenes

Validated at	Patient	Creatinine mg/dl	GFR
2014 Jan 01, 12:23 am	Hector Andres Medina Pino	0.7	60.0
2014 Jan 01, 06:14 am	Jose Daniel Gutierrez Jara	0.4	60.0
2014 Jan 01, 06:14 am	Ana Julia Campos Valdes	2.2	22.53
2014 Jan 01, 06:14 am	Jorge Eugenio Cisternas Concha	0.6	60.0
2014 Jan 01, 06:14 am	Victor Manuel Sedano Inostroza	1.3	60.0
2014 Jan 01, 06:19 am	Juana Teresa Alvarado Navarrete	0.7	60.0
2014 Jan 01, 07:27 am	Mirella Del Carmen Silva Viscarra	0.1	60.0
2014 Jan 01, 07:34 am	Maria Eliana Veliz Barrera	0.6	60.0
2014 Jan 01, 07:35 am	Humberto Del Carmen Araya Vega	0.9	60.0
2014 Jan 01, 07:37 am	Ana Loreto Mena Vidal	0.3	60.0
2014 Jan 01, 07:45 am	Armando Dussailant Benitez	2.9	22.89
2014 Jan 01, 07:46 am	Jimena Andrea Torres Elgueta	3.4	16.36
2014 Jan 01, 07:56 am	Rodrigo Gaspar Valdes Henriquez	1.8	44.06
2014 Jan 01, 08:32 am	Maria Ramona Bravo Lopez	1.2	47.15
2014 Jan 01, 08:53 am	Zunilda Mercedes Abarca Romero	1.7	32.01

Figura 4: Vista de los exámenes desde la aplicación

Las imágenes dan cuenta de un proceso de carga exitoso.

3.1.2 REQUISITOS DETALLADOS

Luego de levantar los requisitos en lenguaje natural, fue necesario dar una descripción más detallada que sirvió como guía al momento de implementar el sistema. Para este efecto, se creó una tabla que contiene el título del requisito, su descripción y qué es necesario para implementarlo.

Requisito	Descripción	Implementación
Autenticación de usuarios	Para acceder al sistema, se debe autenticar al usuario con su email y contraseña.	<ul style="list-style-type: none"> • Pantalla de signup • Pantalla de login • Validación de datos de entrada • Pantalla de editar usuario
Carga de datos desde Excel	Se deben poder cargar los datos de los exámenes, a través de archivos Excel provenientes de la base de datos del hospital.	<ul style="list-style-type: none"> • Pantalla de carga de datos • Lectura de tabla Excel • Formateo de datos • Ingresar datos en forma masiva

	Estos datos contienen los resultados de un examen para un paciente en particular.	
Algoritmo de diagnóstico	Debe existir un algoritmo que determine el diagnóstico de un paciente	<ul style="list-style-type: none"> • Creación del algoritmo • Testeo del algoritmo
Listado de pacientes con AKI	Se debe mostrar un listado con todos los pacientes que tengan AKI	<ul style="list-style-type: none"> • Vista con listado de pacientes
Búsqueda de pacientes	Debe ser posible buscar a un paciente en particular, para ver si tiene AKI y en qué etapa.	<ul style="list-style-type: none"> • Barra de búsqueda • Vista con resultados • Perfil de paciente
Ingreso de epicrisis	Se debe poder ingresar de manera manual las epicrisis.	<ul style="list-style-type: none"> • Formulario de ingreso de epicrisis
Contraste de diagnósticos automáticos vs epicrisis	Se debe poder contrastar diagnósticos de AKI dentro de las epicrisis, contra aquellos obtenidos de manera automática por el sistema.	<ul style="list-style-type: none"> • Vista de estadísticas

3.1.3 SELECCIÓN DE TECNOLOGÍAS

Se decidió crear una aplicación web para solucionar el problema por la familiaridad que tienen los usuarios con los sitios web, por la gran cantidad de componentes Open Source para la creación de aplicaciones y por experiencia previa con estas tecnologías.

Para la implementación del software es deseable que las tecnologías cumplan las siguientes características.

- Open Source
- Bien documentadas
- Rápidas para el desarrollo
- Baja utilización de recursos
- Probadas por grandes empresas de tecnología

Así se facilita el proceso de desarrollo y se construye sobre componentes validados.

Para su categorización, las tecnologías se pueden agrupar en 3 grupos.

1. Front End

Todo aquello que va de cara al usuario final. Por ejemplo, estilo del sitio web o animaciones que se ven dentro del sitio.

2. Back End

Todo aquello que el usuario final no ve. Por ejemplo, base de datos.

3. Arquitectura Física

Todo aquello relativo a las máquinas que soportan los servicios. Por ejemplo, servidores.

Para cada tecnología se dará una explicación técnica de la selección.

3.1.3.1 FRONT END

Para el front end de la aplicación se eligieron tecnologías que aceleran la velocidad de desarrollo y mejoran la interpretación del código, es decir, ayudan a que sea más fácil de leer para el desarrollador.

Todas las tecnologías que se presentan son una capa por sobre los componentes más comunes en construcción de sitios web, como por ejemplo HTML y CSS.

A continuación se describen las tecnologías seleccionadas.

HAML (HTML Abstraction Markup Language): Sintaxis reducida para HTML.

HTML es el estándar para escribir páginas web, sin embargo su sintaxis es redundante.

HAML es una sintaxis reducida de HTML que literalmente reduce a más de la mitad el código, aumentando consecuentemente la velocidad de escritura al doble, esto se logra eliminando los tags de cierre y usando indexación en su reemplazo (como la sintaxis de Python), además de tener shortcuts para los tags más usados.

Como los browsers consumen HTML, frameworks como Ruby on Rails finalmente traducen HAML a HTML al momento de enviar la respuesta al browser. Esto es transparente para el desarrollador, quien nunca verá código HTML. Para hacer esto más tangible se muestra algo de código escrito en HTML y su correspondiente en HAML.

HTML

```
<div class='list-container'>
  <ul>
    <li> First Element </li>
    <li> Second Element </li>
    <li> Third Element </li>
  </div>
```

HAML

```
.list-container
  %ul
    %li First Element
    %li Second Element
    %li Third Element
```

SASS (Syntactically Awesome Style Sheets): Sintaxis reducida para CSS.

CSS es el estándar para darle estilo a una página web, pero su sintaxis es redundante y además no cuenta con la capacidad de crear variables ni de reutilizar trozos de código, similar a lo que pasa con HTML y HAML.

SASS cuenta con variables, snippets, operadores algebraicos y una serie de complementos para dar estilo de manera más fácil.

Bootstrap-SASS: Framework HTML, CSS y JS

Bootstrap es el framework HTML, CSS y JS más popular. Creado por Twitter, Bootstrap. Provee extensa y clara documentación para elementos HTML, docenas de componentes personalizados y útiles plugins para jQuery. Este framework provee un marco de trabajo para darle estilo a un sitio web, con una estructura clara y modular.

Bootstrap-SASS es la versión SASS de Bootstrap.

CoffeeScript: Sintaxis reducida para javascript

Javascript es el estándar para ejecutar código del lado del cliente. CoffeeScript es una sintaxis reducida de javascript, análogo a HAML y HTML.

Jquery: Sintaxis reducida para javascript

Es una sintaxis reducida para algunos métodos de javascript (en este caso Coffeescript), si bien no agrega funcionalidades extra, es más claro para escribir y leer.

3.1.3.2 BACK END

Para el back end, se eligieron tecnologías que cumplen con lo señalado en el la sección 3.1.3. Son open source, están bien documentadas, son rápidas para el desarrollo, tienen baja utilización de recursos y fueron probadas por grandes empresas. Fue importante una buena selección porque en el back end se concentra la mayor cantidad de código y el más complejo.

Se detallan las tecnologías seleccionadas.

Ruby on Rails: Framework para la creación de aplicaciones web

Se elige este framework por ser muy rápido para desarrollar, contar con un extenso repositorio de librerías en github, ser gratis, y por tener como máxima hacer que el código sea muy entretenido de escribir^{iv}, sin dejar de lado aspectos como performance, seguridad y modularidad.

Una característica notable de Ruby on Rails son los scaffolds o generadores, que permiten crear de manera rápida el CRUD de una entidad, donde CRUD es todo lo necesario para (C) Crear, (R) Leer, (U) Actualizar y (D) borrar.

Por ejemplo, al ejecutar el comando `rails scaffold user name:string email:string`, se crearán todas las vistas necesarias para crear, ver, actualizar y borrar usuarios, además de la tabla en la base de datos necesaria para ese propósito.

Github.com un sitio que provee todas las herramientas para revisión y control de versiones de código fuente. Github aloja a más de 10 millones de proyectos de software. Esto es relevante pues el código fuente de github está escrito en Ruby on Rails, lo que le da prestigio al Framework y lo valida frente a profesionales con basto conocimiento sobre código.

Postgres: Base de datos SQL

Una base de datos relacional es capaz de responder a todos los requerimientos planteados para este proyecto.

En experiencias previas, trabajando en babytuto.com, un e-commerce con más de 50.000 visitas mensuales, se trabajó con una base de datos postgres que pesa 60Gb. La base de datos no ha generado problemas. El software que se desarrolló no supera los 10Gb de almacenamiento, por lo que empíricamente postgres es una buena opción.

3.1.4 DISEÑO DEL SISTEMA

El diseño del sistema se divide en arquitectura física, lógica, base de datos y las pantallas o interfaces que se usan para interactuar con el sitio. La arquitectura física hace referencia a los servidores que alojan la aplicación, mientras que la arquitectura lógica detalla la interacción de los componentes del software del sistema.

La base de datos busca ser minimal, en cuanto cada tabla es vital para el funcionamiento del sistema, es decir, no existen tablas que sobran.

Las pantallas del sistema buscan dar una manera sencilla de interacción entre el computador y el usuario. La simplicidad de uso se estudia más adelante con un estudio de usabilidad.

A continuación se detallan las partes del diseño del sistema.

3.1.4.1 ARQUITECTURA FÍSICA

El lugar físico dónde se alojará la aplicación se limita a un computador dentro del departamento de nefrología. Esta decisión responde a un requerimiento del hospital que señala que el software y los datos deben estar físicamente dentro del hospital.

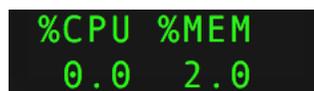
El computador seleccionado fue un DELL INSPIRON 14-3421 CORE I5 14 con las siguientes especificaciones:

- 500Gb en disco duro
- 4Gb de RAM
- Procesador Intel core i5 (2 núcleos de 1.7Gb c/u)

Se hicieron 2 pruebas de carga con la aplicación corriendo en el computador, para determinar el porcentaje de CPU y memoria RAM usa el computador.

Prueba 1

La primera prueba consiste en navegar por la aplicación y monitorear el uso de CPU y memoria RAM usando la terminal.



```
%CPU %MEM
0.0 2.0
```

Figura 5: Medición de CPU y Memoria RAM usando la terminal

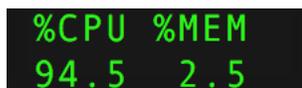
Mientras se navega el uso de recursos es bajo, usando un porcentaje de la CPU que no alcanza a ser visible y un 2.0% de la capacidad de la memoria RAM.

Se concluye que durante la navegación, la cantidad de recursos es baja y no es necesario mayor análisis.

Prueba 2

Se monitoreó el uso de CPU y memoria RAM usando la terminal, pero esta vez se midió mientras se cargaban datos al sistema a través de un archivo Excel. Como el archivo Excel tiene 15.000 líneas aproximadamente, es un momento donde se debiesen exigir intensivamente los recursos.

Durante la carga del Excel, el computador usó casi la totalidad de su capacidad de procesamiento, pero la utilización de memoria RAM sólo subió un 0.5% en comparación con la prueba anterior.



```
%CPU %MEM
94.5  2.5
```

Figura 6: Medición de CPU y Memoria RAM usando la terminal

Esto es esperable, muestra que el computador está utilizando su capacidad de procesamiento para realizar rápidamente el trabajo, sin aumentar significativamente el uso de la memoria RAM.

El uso intensivo de CPU podría ser preocupante, pues podría congelar a otras aplicaciones que estén corriendo simultáneamente, sin embargo, cuando se ejecutó otro proceso, el computador dividió los recursos de procesamiento.

Este comportamiento es esperable, ya que la CPU está diseñada para balancear la carga entre todos los procesos que están corriendo. Por lo tanto el funcionamiento normal del computador no se ve amenazado por la ejecución de la aplicación.

3.1.4.2 ARQUITECTURA LÓGICA

La arquitectura lógica está estrictamente basada en la arquitectura modelo, vista, controlador (MVC) utilizada por el framework Ruby on Rails. Existen componentes que si bien son propios de la aplicación, también se rigen por la arquitectura MVC. A continuación se explica la arquitectura MVC y se muestran los componentes propietarios.

MODELO VISTA CONTROLADOR (MVC)

El sitio se basa en la arquitectura modelo vista controlador (MVC), donde el modelo se encarga de encapsular la interacción con la base de datos. La vista corresponde a las páginas que el usuario ve y el controlador juega el rol intermediario entre el modelo y las vistas, está encargado de la lógica del sistema.

La arquitectura MVC es común entre los frameworks de desarrollo web. Lo anterior hace que el sistema sea fácil de entender no sólo para las personas en el proyecto, sino también a quienes se sumen en el futuro.

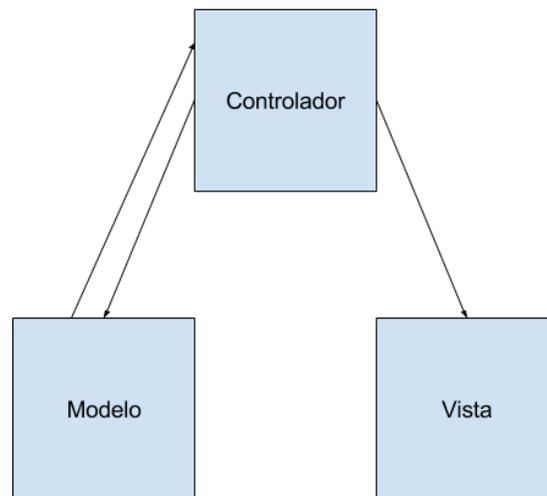


Figura 7: Diagrama de la arquitectura modelo vista controlador (MVC)

COMPONENTES PROPIOS

Existen funcionalidades propias de la aplicación, que no pueden ser resueltas con las componentes que provee Ruby on Rails. A continuación se muestra un diagrama con estos componentes, seguido de una breve descripción de cada uno.

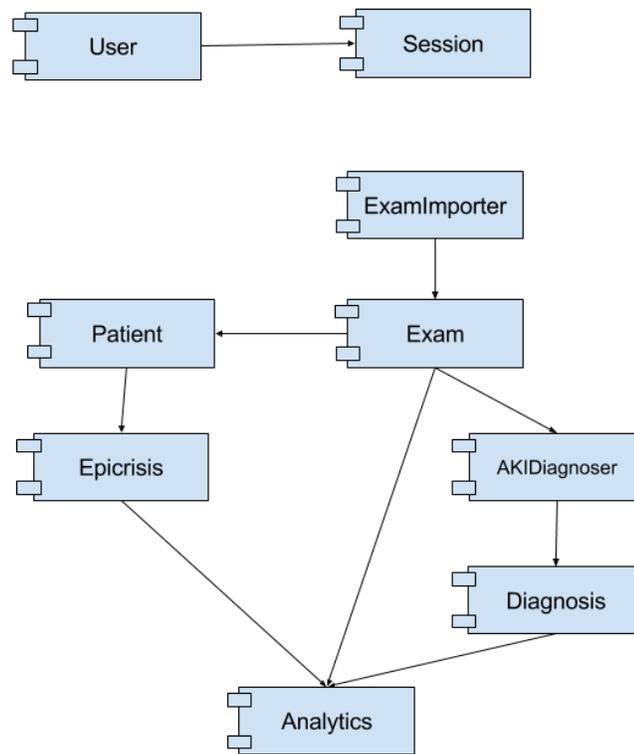


Figura 8: Diagrama de componentes

En la siguiente tabla se muestra cada componente con su responsabilidad, donde CRUD hace referencia a Create, Read, Update y Destroy.

Componente	Responsabilidad
User	CRUD de usuarios
Sesión	Autenticación de usuarios (login)
Exam	CRUD de exámenes
Exam Importer	Importar exámenes desde archivo Excel
Patient	CRUD de pacientes
Epicrisis	CRUD de epicrisis
AkiDiagnoser	Diagnosticar daño renal agudo
Diagnosis	CRUD de diagnósticos
Analytics	Análisis de datos

3.1.4.3 BASE DE DATOS

La base de datos es parte del core de la aplicación, su diseño debe ser mirado con especial cuidado. En un futuro será necesario extender el sistema, por lo que la base de datos debe ser capaz de adaptarse de manera sencilla a los cambios. Con esto en mente, se diseñó de manera minimal, es decir, cualquier cosa que se saque de la base de datos provoca que alguno de los requisitos se deje de cumplir. La base de datos debe ser tan simple que cualquier modificación debe ser trivial.

El siguiente esquema representa la estructura de la base de datos:

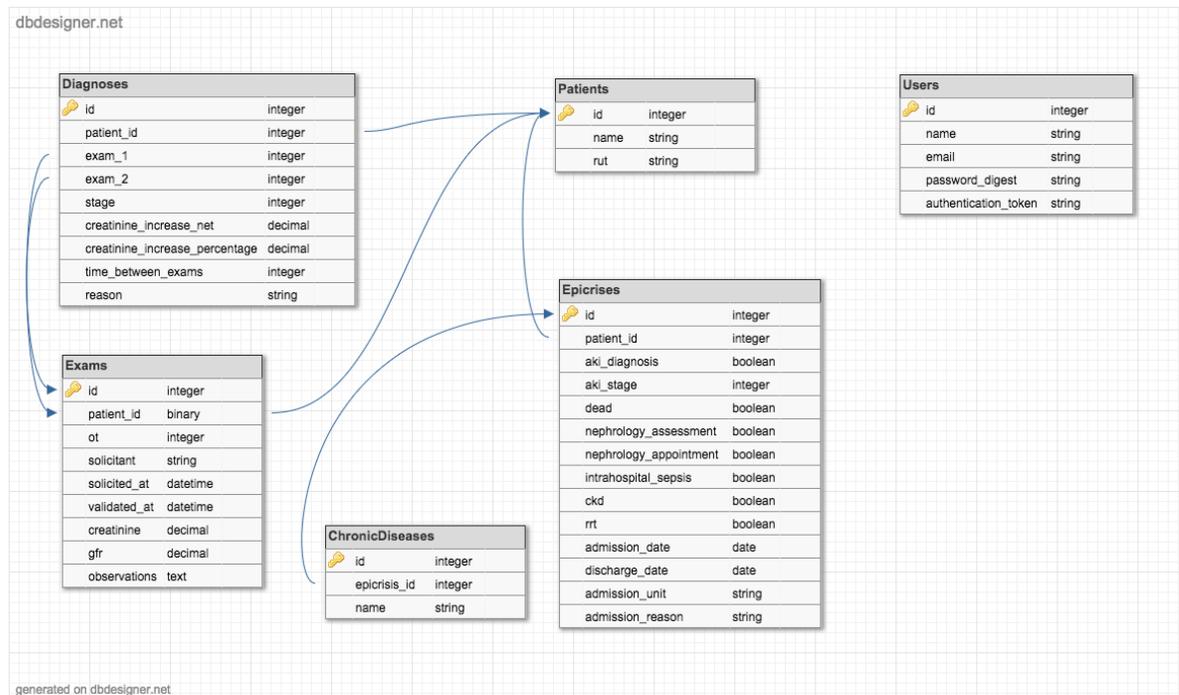


Figura 9: Diagrama de la base de datos

3.1.5 PANTALLAS DE NAVEGACIÓN

Como en cualquier aplicación web, la interacción con el usuario se hace a través de páginas web. Este capítulo describe las pantallas que visita el usuario durante la navegación en el sitio.

La primera vez que el usuario se enfrenta a la aplicación, se encontrará con la pantalla de inicio. Aquí el usuario puede registrarse e iniciar sesión. Su uso es bastante simple y no merece mayor discusión.

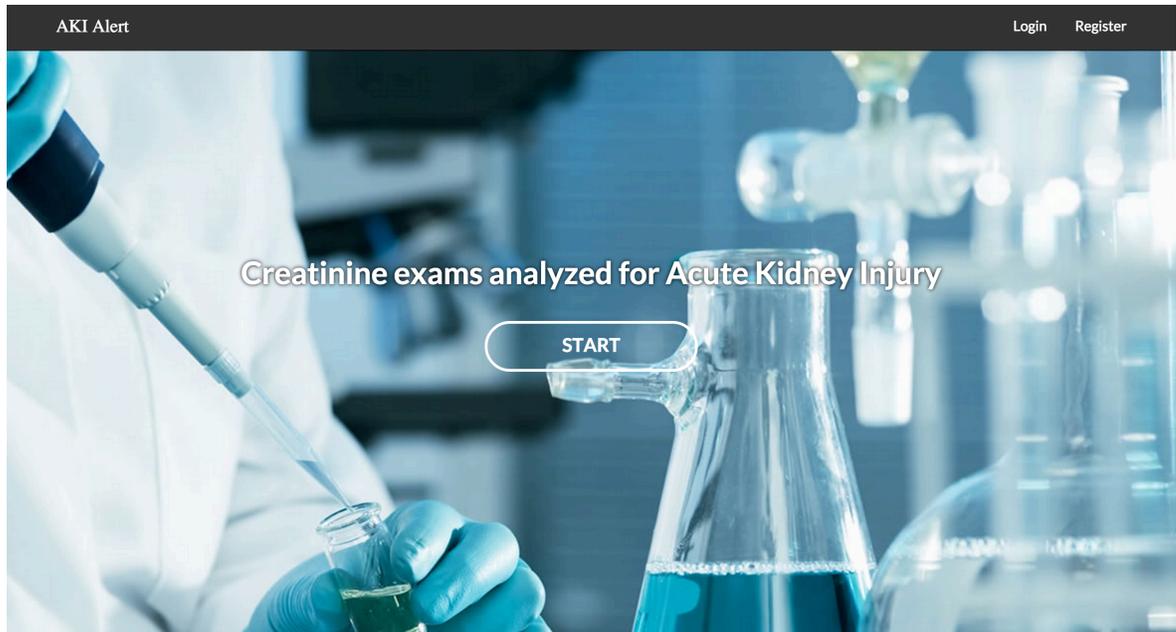


Figura 10: Pantalla de inicio

Una vez que el usuario ingresa verá la página de pacientes que tiene un listado con todos los pacientes ingresados en el sitio.

De aquí en adelante, todas las pantallas del sitio tendrán una barra de navegación superior, para acceder a las distintas secciones de la plataforma. Este estilo de navegación es uno de los más populares y debiese ser bastante familiar para cualquier usuario. Todas las pantallas fueron creadas usando componentes estándar de bootstrap, esto implica un diseño consistente y común (recordando que bootstrap es el framework CSS más popular).

Patients

RUT	Name		
4660401-6	Agustin Honorio Pavez Veas		
11622180-2	Aldo Jorge Low Low		
5194634-0	Alejandrina Del Carmen Valenzuela Orellana		
10907615-5	Alejandro Javier Veliz MuOz		
13876271-8	Alex Ivan De Los Angeles Ramos Valdivia		
4211437-5	Alfonso Andres Gonzalez San Martin		
2987216-3	Alfonso Jorge Galeb Daher		
4036408-0	Alicia Del Carmen Wilde Montes		
5526511-9	Alicia Iturra Valdivia		
3956998-1	Alicia Maria Berrios Alvarez		
7636421-4	Alicia Monica Perez Varas		
5276602-8	Alicia Valenzuela Ibarra		
6865698-2	Amelia De Las Mercedes Kohnenkamp Mondaca		
7987225-3	Ana Edelmira Herrera Carrasco		

Figura 11: Pantalla de pacientes

Otras pantallas relevantes son la de Diagnósticos, que muestra un listado con todos los diagnósticos encontrados y la pantalla de “perfil de paciente” que muestra los diagnósticos para un paciente en particular, los exámenes que se le han realizado y las epicrisis ingresadas.

Diagnoses

Patient	Exam 1	Exam 2	Creatinine net increase	Creatinine increase percentage	Time between exams	Stage	Reason
Rodrigo Gaspar Valdes Henriquez	0.7	1.1	0.4	57%	3 days	1	↑
Rodrigo Gaspar Valdes Henriquez	0.7	1.0	0.3	43%	1 day, 23 hours	1	↩
Hector Andres Medina Pino	0.4	0.6	0.2	50%	1 day, 5 hours	1	↑
Filomena Del Transito Moreno Pizarro	1.0	1.6	0.6	60%	1 day, 7 hours	1	↑
Filomena Del Transito Moreno Pizarro	1.0	1.6	0.6	60%	1 day, 7 hours	1	↩
Leonor Del Transito Miranda Cordova	0.4	0.6	0.2	50%	3 days, 11 hours	1	↑
Ivon Ernestina Perez Carrasco	0.9	1.5	0.6	67%	2 days, 22 hours	1	↑
Ivon Ernestina Perez Carrasco	0.6	1.5	0.9	150%	1 day, 23 hours	1	↩
Elba Maria Fuentes Gonzalez	0.2	0.7	0.5	250%	6 days, 7 hours	3	↑
Elba Maria Fuentes Gonzalez	0.2	0.7	0.5	250%	1 day, 23 hours	1	↩

Figura 12: Pantalla de diagnósticos

Search Patients Diagnoses Analytics Exams Epicrises Waldo Uribe

Teresa De Las Mercedes Crespo Toledo

Diagnostics

- Stage: 1
- Stage: 1

Exams

Date	Creatinine mg/dl
2014 Jan 24, 06:01	0.9
2014 Jan 23, 06:01	1.0
2014 Jan 22, 06:01	1.5
2014 Jan 21, 05:01	1.4
2014 Jan 20, 06:01	1.2
2014 Jan 19, 14:01	1.5
2014 Jan 18, 14:01	0.9

Epicrises

At stage: 1

Figura 13: Pantalla de un paciente

Finalmente, existe una vista que tiene las estadísticas obtenidas del análisis de los datos.

Search Patients Diagnoses Analytics Exams Epicrises Waldo Uribe

Analytics

AKI by stage

- Patients count: 393
- Patients with AKI: 393 (100.0%)
- Patients with AKI stage 1: 326 (82.95%)
- Patients with AKI stage 2: 64 (16.28%)
- Patients with AKI stage 3: 67 (17.05%)
- Patients with more than one episode: ???

Accuracy

- Epicrisis accuracy percentage: 17 %

Admission

- Medical Admission: 30.5 %
- Surgical Admission: 26.0 %
- ICU Admission: 43.5 %

Worsening of renal function

- Moved from AKI 1 to AKI 3: 20 patients [View Patients](#)
- Moved from AKI 1 to AKI 2: 30 patients [View Patients](#)
- Moved from AKI 2 to AKI 3: 0 patients [View Patients](#)

With Chronic Disease

Figura 14: Pantalla de estadísticas

3.1.6 ALGORITMO DE DETECCIÓN DE AKI

Como se ha discutido en capítulos anteriores, el software debe usar un criterio de detección de Insuficiencia Renal Aguda. Se hablará del criterio para luego explicar cómo fue implementado.

Criterio AKIN

Usando los niveles de creatinina, el criterio AKIN determina si un paciente tiene Insuficiencia Renal Aguda y con qué gravedad. El criterio divide la gravedad en 3 etapas, donde la 3ra es la más grave.

- **Etapas 1**

El paciente presenta un alza en sus niveles de creatinina de 0.3 [mg/dl] en 48 horas o de un 50% en 7 días o menos.

- **Etapas 2**

El paciente presenta un alza en sus niveles de creatinina de un 100% en 7 días o menos.

- **Etapas 3**

El paciente presenta un alza en sus niveles de creatinina de un 200% en 7 días o menos, o tiene más de 4.0 [mg/dl] y experimentó un alza de 0.5 [mg/dl] en menos de 48 horas.

Implementación

Para analizar los exámenes usando el criterio AKIN, se creó un algoritmo que se ejecuta cada vez que un examen es ingresado.

Una vez que es ingresado el examen de un paciente, el algoritmo compara ese examen con todos aquellos tomados en un intervalo de 7 días para ese paciente. El algoritmo determina si hubo un alza en los niveles de creatinina que gatille un diagnóstico.

Se habló de un algoritmo, sin embargo, una mejor aproximación para resolver este problema es crear objetos que engloben las distintas responsabilidades necesarias para diagnosticar. El algoritmo es sólo una parte dentro del proceso de diagnóstico. También es necesario obtener datos del paciente, estudiar exámenes dentro de un rango de tiempo y más lógica asociada. Una función no debiese asumir tanta responsabilidad.

Considerando lo anterior, fue necesario crear objetos que asuman distintas responsabilidades. Se explicarán los objetos³ más relevantes para la detección de AKIN y la relación entre ellos.

CreatinineTest

El objeto más simple es **CreatinineTest**, la representación de un examen de creatinina.

```
class CreatinineTest
  attributes :patient, :level, :date
  ...
end
```

El examen está asociado a un paciente, tiene un nivel de creatinina y la fecha en la que fue tomado.

Patient

Otro objeto relevante es **Patient**, la representación de un paciente.

```
class Patient
  attributes :id, :name, :rut, :age
  ...
end
```

³ En realidad lo que se presenta es una simplificación de los objetos.

TestPair

TestPair es la representación de un par de exámenes de creatinina, se usa para comparar fácilmente pares de exámenes.

```
class TestPair
  attributes :test01, test02
  ...
end
```

Akin

Aquí es cuando se empieza a poner interesante. Akin representa la evaluación de un par de exámenes usando el criterio AKIN. Se inicializa usando un objeto de la clase TestPair y tiene el método diagnosis, que retorna el peor diagnostico asociado a ese par de exámenes. Esta clase tiene una función privada que retorna el peor diagnóstico gatillado por un alza en menos de 48 horas y otro que hace lo mismo, pero considerando alzas en menos de 7 días.

Su implementación es la siguiente.

```
class Akin
  attributes :test_pair

  def initialize(a_test_pair)
    self.test_pair = a_test_pair
  end

  def diagnosis
    return max(diagnosis_48hrs, diagnosis_7days)
  end
end
```

```

def diagnosis_48hrs
  worst_stage = nil
  if test_pair.performed_in_range_of 48.hours
    if (creatinine_test_pair.level_raised? 0.5
        and test_pair.test_2.level >= 4)
      worst_stage = 3
    elsif (test_pair.level_raised? 0.3)
      worst_stage = 1
    else
      worst_stage = 0
    end
  end
end

  return worst_stage
end

def diagnosis_7days
  worst_stage = nil
  if (test_pair.performed_in_range_of 7.days)
    if (test_pair.level_percentage_raised? 200)
      worst_stage = 3
    elsif(test_pair.level_percentage_raised? 100)
      worst_stage = 2
    elsif(test_pair.level_percentage_raised? 50)
      worst_stage = 1
    else
      worst_stage = 0
    end
  end
end

  return worst_stage
end

```

Luego de revisar el código, se puede ver que el responsable final de entregar un diagnóstico es la función `diagnosis` llamada sobre una instancia de la clase `Akin`. Sin embargo, esa responsabilidad es repartida entre todos los otros objetos. Por ejemplo, el método **`performed_in_range_of(time_interval)`** de la clase `TestPair` se encarga de ver si un par de exámenes fueron tomados en un intervalo de tiempo determinado. Así hay otros métodos que asumen responsabilidades, como **`level_percentage_rised?(x_pct)`** que determina si para un par de exámenes el nivel de creatinina subió en `x_pct`. Al leer el código parece lenguaje natural, es fácil determinar que hace cada función y su implementación es obvia.

3.1.7 ESTUDIO DE USABILIDAD

Para determinar que la interfaz es fácil de usar, se desarrolló un estudio de usabilidad y se aplicó a 7 usuarios. Según Jacob Nielsen, referente en usabilidad, con 7 usuarios se encuentran más del 80% de los problemas^v, que para efectos de este trabajo es suficiente.

3.1.7.1 OBJETIVO

El objetivo del estudio viene dado por un requerimiento de los doctores, quienes declaran estar muy ocupados, por lo que el sitio debe ser capaz de cumplir un conjunto reducido de acciones de manera rápida.

3.1.7.2 USUARIOS

Existen 2 grupos de usuarios que usarán el software. El primer grupo corresponde a doctores de la Unidad de Nefrología del Hospital Clínico de la Universidad de Chile. Son nefrólogos de entre 35 y 55 años, quienes tienen experiencia con herramientas digitales para la gestión de pacientes, diagnósticos, exámenes, etc.

El segundo, estudiantes de medicina que cursan los 2 últimos años de la carrera. Ellos tienen facilidad para usar sitios web y también tienen experiencia con herramientas digitales médicas.

3.1.7.3 USO DEL SITIO

Los principales usos que se esperan del sitio son:

- Buscar pacientes
- Revisar los todos los diagnósticos generados por el sistema
- Ver estadísticas sobre la incidencia del daño renal agudo

3.1.7.4 CONTEXTO DE USO

El software es usado exclusivamente desde un computador ubicado en una sala asignada para el equipo de nefrología. En esta sala por lo general hay de 0 a 5 personas. Es un espacio cómodo para esa cantidad de gente, y se observa un ambiente tranquilo.

El sitio web es accedido con los siguientes objetivos:

- Para consultar si un paciente en particular fue diagnosticado con AKI
- Para ingresar las epicrisis
- Para ver estadísticas sobre la incidencia del daño renal agudo

3.1.7.5 METODOLOGÍA USADA

Para estudiar la usabilidad se utilizaron los siguientes métodos

- Observación + Thinking Aloud
- User Feedback
- Entrevista

A continuación, se muestra específicamente lo que se evaluó con cada método.

Con la Observación + Thinking Aloud se determinó qué tan fácil es ejecutar las tareas asignadas, además de entender el modelo mental del usuario. En esta evaluación se usó una escala de 3 niveles (No Logrado, Logrado con dificultad y Logrado) para analizar la realización de cada una de las tareas, además se documentaron los comentarios realizados por los usuarios al momento de ejecutarlas.

El User Feedback, rescata observaciones o ideas del usuario que permitan hacer la plataforma más fácil de usar.

En la Entrevista se evaluó cuál es la sensación que tiene el usuario. Esto se relacionará con la satisfacción del usuario al usar la plataforma. Se utilizaron 3 afirmaciones, sobre las que el usuario respondió qué tan de acuerdo se encuentra con dichas afirmaciones.

3.1.7.6 MUESTRA

Los usuarios involucrados en este estudio fueron:

- 3 Doctores
- 4 Estudiantes de medicina

3.1.7.7 INSTRUMENTOS (MÉTODOS)

Para el estudio se usaron 3 instrumentos o métodos. Observación + Thinking aloud, User Feedback y una entrevista. Los 2 primeros tienen como objetivo encontrar problemas de usabilidad y la entrevista está orientada a entender la sensación que tiene el usuario al usar el sitio. Se presentan los instrumentos y los resultados obtenidos.

3.1.7.7.1 OBSERVACIÓN + THINKING ALOUD

Este instrumento se aplica definiendo las tareas claves del sitio y se mide el desempeño de los usuarios para desarrollar dichas tareas.

Junto al equipo médico, se definieron 4 tareas clave que el sitio web debe realizar.

- Encontrar al paciente con nombre “Luis Onofre Barraza Gómez”
- Encontrar los exámenes del paciente “Ricardo Antonio Guardamagna Sanhueza”
- Encontrar los diagnósticos que fueron generados para “Ricardo Antonio Guardamagna Sanhueza”
- Ingresar una epicrisis para el paciente “Ricardo Antonio Guardamagna Sanhueza”, puede usar los datos que guste.

De acuerdo al desempeño del usuario, además se adjunta un campo para realizar observaciones.

La ejecución de estas tareas fue observada y evaluada por un examinador. Además, el usuario comentó lo que pensaba durante la ejecución. El examinador se encargó de registrar los comentarios relevantes.

3.1.7.7.2 USER FEEDBACK

User Feedback sirve para encontrar problemas de usabilidad a través de la comunicación con el usuario.

Los usuarios acordaron enviar emails con formato libre, pero cuyo contenido debe detallar problemas al momento de usar el sitio, e ideas para mejorarlo. Este ítem es importante, pues los doctores tienen la claridad sobre los conceptos que se deben agrupar de acuerdo a su significado, modo de utilizar herramientas similares y en general se busca entender el modelo mental de los usuarios.

3.1.7.7.3 ENTREVISTA

En la entrevista, se evaluó la reacción de los usuarios a 3 afirmaciones, y se registraron opiniones relativas al uso del sitio. Se buscó comprender cuál es la sensación que tienen al usar la plataforma, y entender qué tan satisfechos están con ella. El objetivo de la entrevista es rescatar información necesaria para mejorar la disposición de los contenidos en el sitio.

Afirmaciones

- La utilización del sitio fue agradable
- El sitio logró solucionar sus problemas de manera satisfactoria
- El sitio muestra los contenidos de manera apropiada

Procedimiento

Dado que el horario de los usuarios, en este caso doctores y estudiantes de medicina, es ajustado, el examinador se puso de acuerdo con ellos para establecer un horario común para realizar las evaluaciones de usabilidad.

Con cada usuario, se acordó un horario para realizar la Observación + Thinking Aloud, que dura máximo 15 minutos, seguido de la entrevista de no más de 5 minutos.

El User Feedback es enviado por los usuarios al mail del examinador, quien responde durante la semana la recepción de los comentarios.

Los resultados de las evaluaciones, se encuentran detallados con gráficos en la siguiente sección.

3.1.7.8 PRESENTACIÓN Y ANÁLISIS DE LOS RESULTADOS

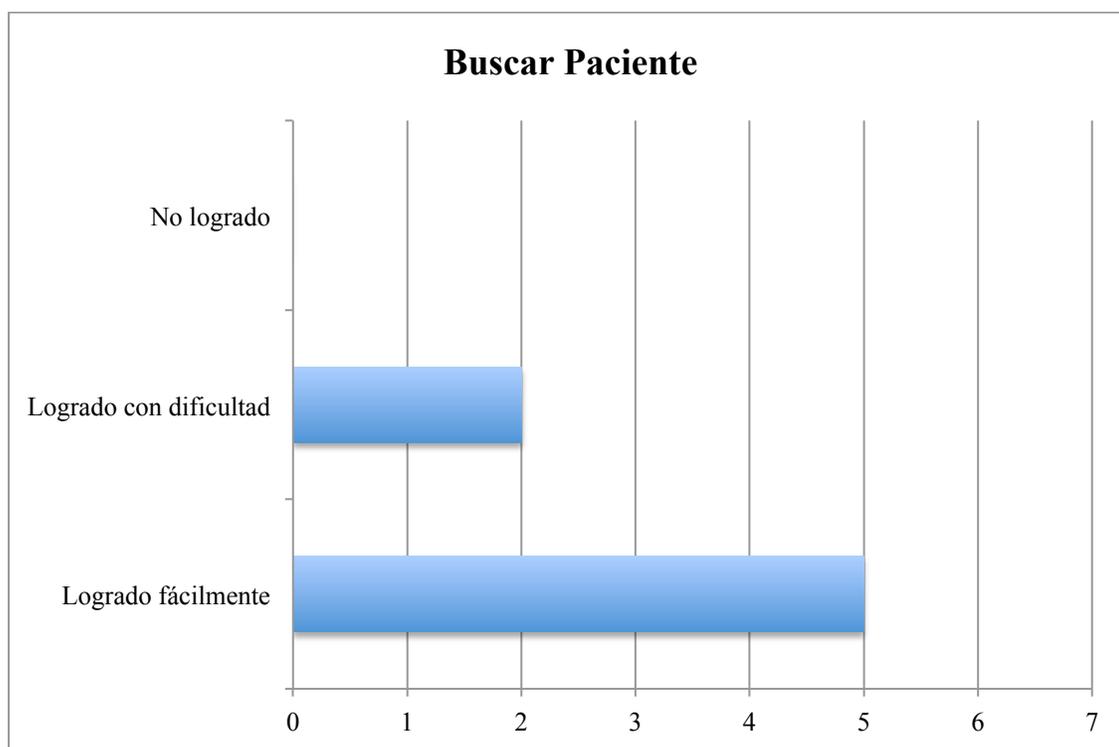
En las secciones siguientes se presentan los resultados obtenidos de la aplicación de los 3 instrumentos de estudio.

3.1.7.8.1 OBSERVACIÓN + THINKING ALOUD

A continuación se muestran los resultados en cada una de las tareas asignadas a los usuarios

Tarea #1

Encontrar al paciente con nombre “Luis Onofre Barraza Gómez”



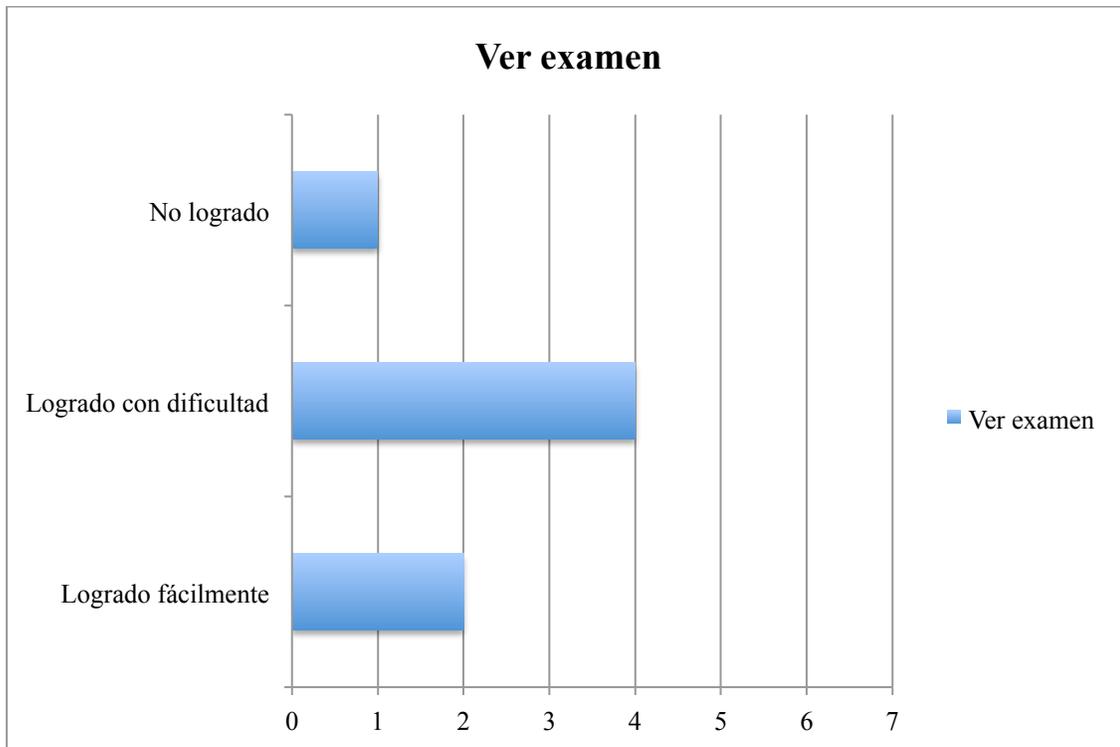
Observaciones

Los usuarios que no descubrieron el buscador de pacientes, sino que se fueron al listado de pacientes, se demoraron un poco más en realizar esta tarea.

Se señaló que la barra de búsqueda debería auto-completar lo que escriben, para así buscar más rápido.

Tarea #2

Encontrar los exámenes del paciente “Ricardo Antonio Guardamagna Sanhueza”



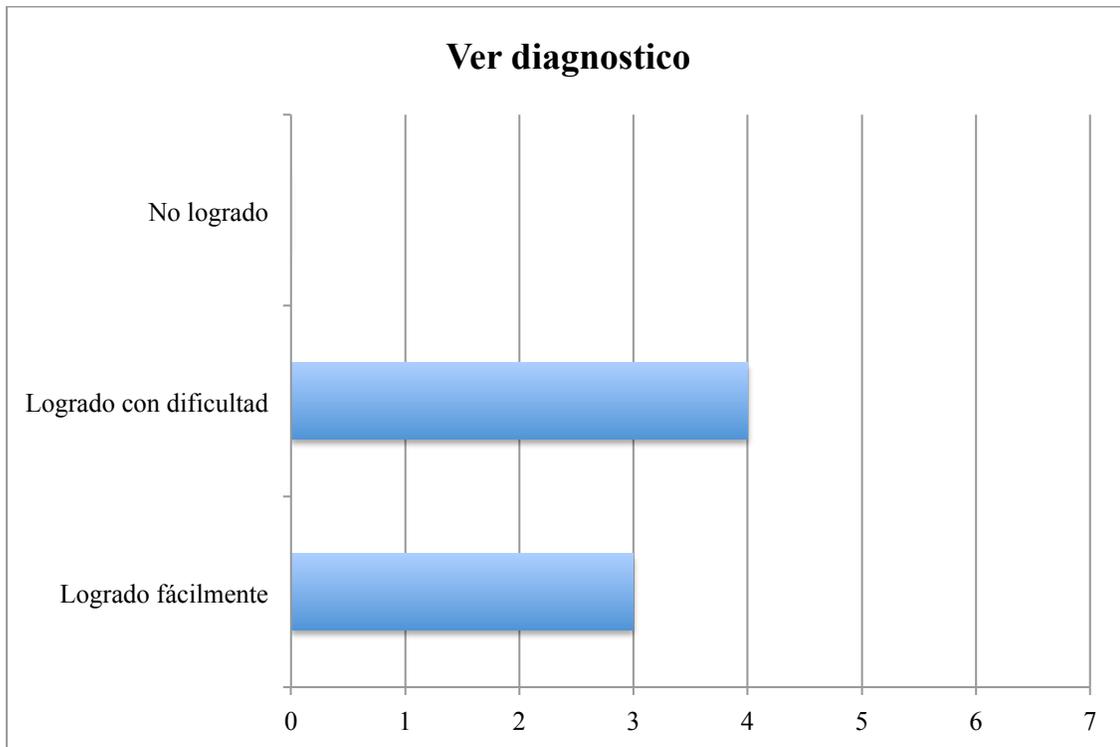
Observaciones

Algunos usuarios notaron que los exámenes se encontraban en el perfil del paciente, sin embargo, otros usuarios, aquellos que lograron la tarea con mayor dificultad, fueron a la pestaña “Exámenes”, en ese lugar resulta difícil encontrar los exámenes de un paciente determinado.

Los usuarios repetían frases del estilo: “No lo encuentro, mmm... ¿dónde está?”

Tarea #3

Encontrar los diagnósticos que fueron generados para “Ricardo Antonio Guardamagna Sanhueza”

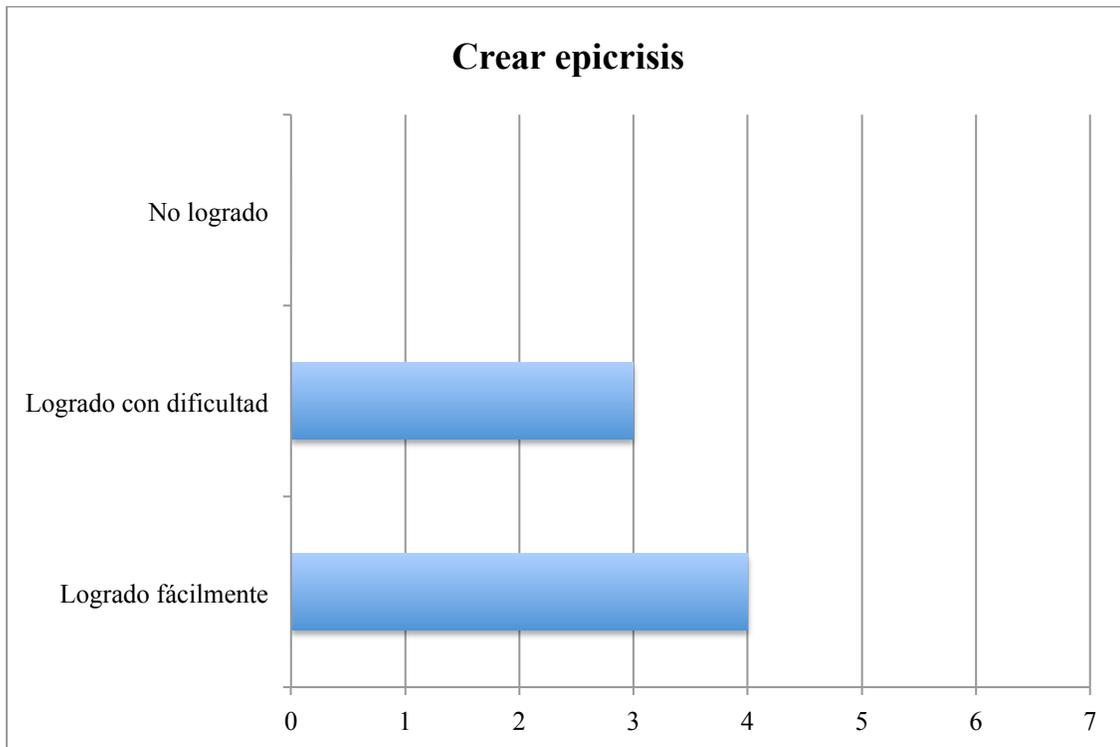


Observaciones

Existe un link en el encabezado de la página que dice “Diagnósticos”, en él, se encuentra un listado sin un orden particular de todos los diagnósticos. Por otro lado, en el perfil del paciente, se encuentra un listado con todos sus diagnósticos. Aquellos usuarios que pincharon en el link “Diagnósticos” tuvieron problemas para realizar la tarea.

Tarea #4

Ingresar una epicrisis para el paciente “Ricardo Antonio Guardamagna Sanhueva”, puede usar los datos que guste.



Observación

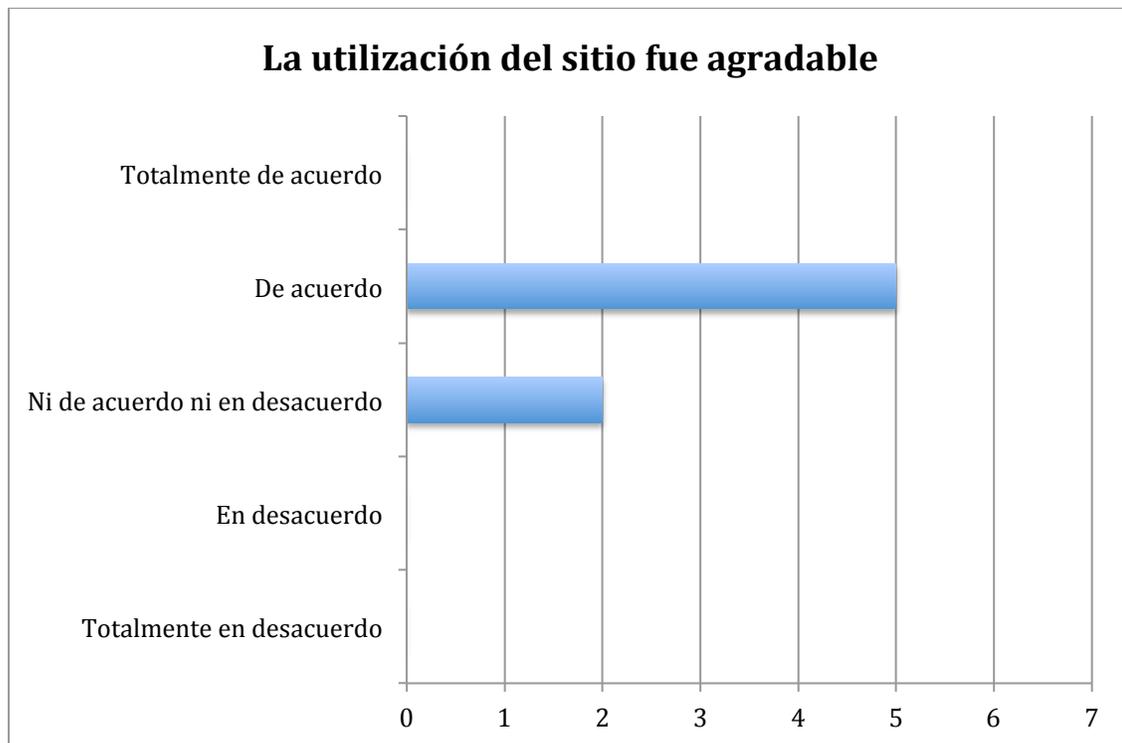
Los usuarios se demoraron en encontrar a los pacientes en el selector de pacientes. Algunos usuarios descubrieron que se podía crear epicrisis directamente desde el perfil del paciente, para ellos fue más fácil.

3.1.7.8.2 Entrevista

La entrevista se hizo inmediatamente después de la Observación + Thinking Aloud.

Los resultados se muestran a continuación.

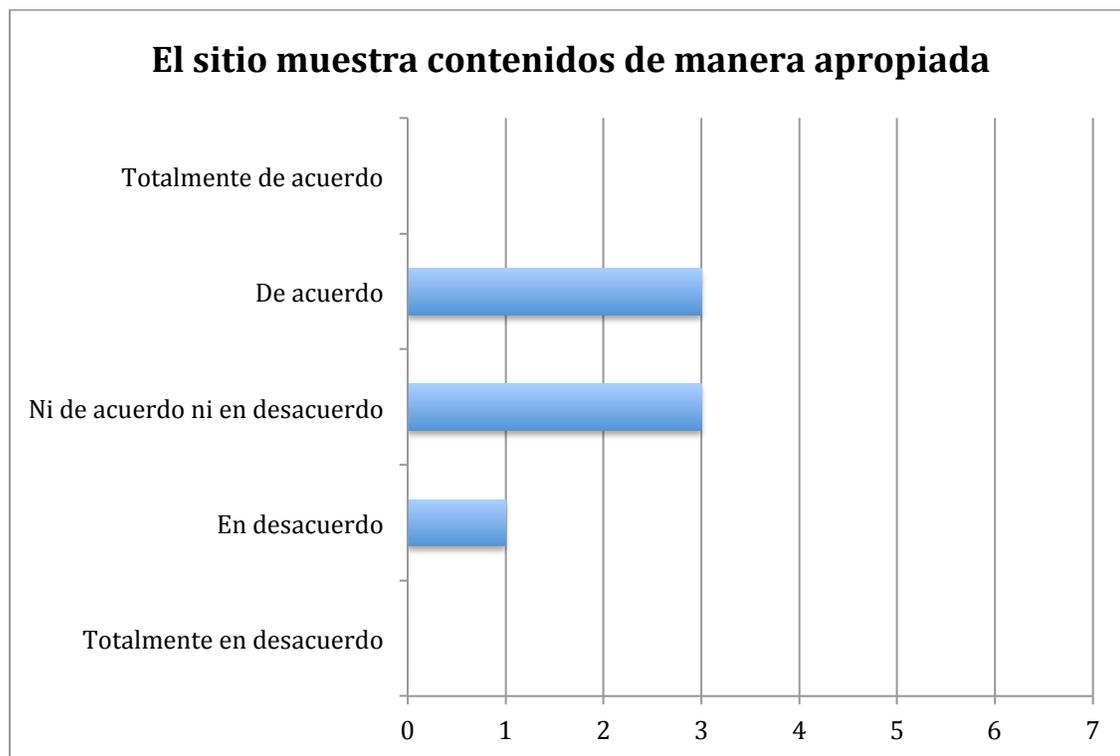
Pregunta #1: La utilización del sitio fue agradable



Pregunta #2: El sitio logró solucionar sus problemas de manera satisfactoria



Pregunta #3: El sitio muestra los contenidos de manera apropiada



Observaciones de la entrevista:

De la entrevista se puede destacar que el grado de satisfacción de los usuarios con el sistema es bueno. Sin embargo, con un mejor posicionamiento de los contenidos, se podrían facilitar algunas acciones dentro del sitio, haciendo de su uso una experiencia más agradable. En conversaciones posteriores a la entrevista, los usuarios manifestaron que la interfaz se ve limpia, en comparación a los sistemas del hospital y afirman que les es más sencillo usar una aplicación web, destacando la claridad en la que los datos son presentados.

3.1.7.8.3 USER FEEDBACK

En esta sección, se incluirá el feedback entregado por los usuarios.

Título	Importancia	Descripción	Ubicación
Ingresar epicrisis a un paciente	Media	Nos resultaría más fácil agregar las epicrisis desde el perfil del paciente.	/patients/<id>
Dg AKI	No especificada	Los usuarios requieren visualizar un campo llamado Dg AKI, dentro de la epicrisis. Ese campo no existe actualmente.	/epicrisis/new
Más datos en la epicrisis	Alta	Agregar los campos ckd y rrt a la epicrisis	/epicrisis/new
Cambio de nombre para Heart Disease	Media	La forma en la que los médicos se refieren a “Heart disease” es AMI.	/epicrisis/new
Cambio de nombre para EPOC	Media	Cambiar la sigla EPOC por COPD, su traducción a inglés.	/epicrisis/new
Cambio de nombre para DHC	Media	Cambiar la sigla DHC por CLD, su traducción a inglés.	/epicrisis/new

Agregar ESRD, RRT	Media	Agregar los campos ESRD y RRT a la epicrisis	/epicrisis/new
-------------------	-------	--	----------------

Observaciones user feedback:

Dentro del feedback entregado por los usuarios, lo que más se destaca es la necesidad por agregar más información al sistema, para que haga un mejor mapping con la realidad, es decir, con los datos que actualmente se usan para analizar a los pacientes. Por otro lado, se informa el deseo de mantener congruencia para hacer referencia a los términos médicos, en particular, usar siempre siglas en vez del nombre completo. Este último requisito, aunque para una persona común puede no ser evidente, para los usuarios finales, es vocabulario estándar.

3.1.7.9 CONCLUSIONES DEL ESTUDIO DE USABILIDAD

Recordemos que el objetivo del estudio de usabilidad es determinar qué tan fácil y eficiente era usar la plataforma.

En todas las pruebas realizadas, los usuarios lograron completar las tareas. Sin embargo, se pudo determinar de manera clara, que algunos links confundían a los usuarios llevándolos a lugares donde no se podían realizar las tareas.

Para este problema en particular, se recomienda mantener la información centralizada y no repartida en varios lugares en el sitio.

El estudio logró determinar cuál es el lugar correcto para posicionar la información, por ejemplo, al momento de buscar el diagnóstico de un paciente en particular, se detectó que el mejor lugar es dentro del perfil del paciente y no en un listado donde están todos los diagnósticos. El “perfil de paciente”, es algo que los usuarios mostraron comprender de buena manera, haciendo de este, un buen lugar para situar toda la información relativa al paciente.

Con respecto a la satisfacción del uso de la plataforma, se observa que los usuarios se manifiestan indiferentes al uso del sistema, con una inclinación a conformes con el desempeño. Se recomienda mejorar aquellos puntos donde el usuario se confunde, para lograr una mejor experiencia.

El user feedback mostró algunas inconsistencias en el sitio, ya que en algunos lugares se hacía referencia a condiciones usando siglas y en otras el nombre completo.

Las siglas parecen ser poco descriptivas, sin embargo, los usuarios finales señalaron que para ellos son comunes, de uso diario, y fáciles de reconocer.

Por otro lado, no existe una completa relación con los contenidos del sitio y el mundo real (mapping con el mundo real), pues faltaban datos que los doctores esperaban encontrar.

4 VERIFICACIÓN Y VALIDACIÓN

La verificación consiste en constatar si la aplicación funciona correctamente. La validación busca analizar si el software puede contrastar correctamente los diagnósticos generados por el software con aquellos generados por los doctores.

4.1 VERIFICACIÓN

La plataforma ha sido usada durante 6 meses por Nefrólogos del J.J. Aguirre y estudiantes de medicina de la Universidad de Chile. A la fecha no se han reportado problemas en el sistema, y el log de errores se encuentra vacío.

Técnicamente el proyecto fue bien ejecutado y más importante aún, su desarrollo sirvió para obtener las estadísticas que se buscaban.

4.2 VALIDACIÓN

La validación del software se hizo con un experimento que permitió probar el software.

4.2.1 EXPERIMENTACIÓN

El experimento consistió en utilizar el software para extraer y analizar los exámenes del hospital.

El sistema fue capaz de diagnosticar a los pacientes enfermos y de desplegar información relevante para la caracterización de la Insuficiencia Renal Aguda. Luego, para contrastar estos diagnósticos con las epicrisis, se realizó un experimento con los siguientes pasos.

Paso 1

Se extrajeron 15,547 exámenes de la base de datos del hospital. Estos exámenes están comprendidos entre el 17 de Enero del 2015 y el 17 de Junio del 2014, y corresponden a 5,418 pacientes.

Paso 2

Con el sistema, se analizaron los exámenes en búsqueda de pacientes con Daño Renal Agudo, se encontró que el 7% de los pacientes (393) padecían de la enfermedad.

Paso 3

De los pacientes diagnosticados, se eligieron 361. De ellos se traspasaron de manera manual, las epicrisis del sistema del hospital al software.

Paso 4

Una vez que las epicrisis se encontraban en el sistema, se contrastaron con los diagnósticos generados. Como resultado se obtuvieron las estadísticas buscadas.

De los pacientes diagnosticados por el software, sólo el 17% de ellos ya contaba con un diagnóstico previo en las epicrisis del hospital.

Los diagnósticos hallados se desglosan de la siguiente forma⁴.

- Pacientes con al menos un diagnóstico de AKI etapa 1
 - 326 pacientes (6.0% del total)
- Pacientes con al menos un diagnóstico de AKI etapa 2
 - 64 pacientes (1.2% del total)
- Pacientes con al menos un diagnóstico de AKI etapa 3
 - 67 pacientes (1.2% del total)
- Pacientes que empeoraron de AKI etapa 1 a AKI etapa 3
 - 20 pacientes (0.1% del total)
- Pacientes que empeoraron de AKI etapa 1 a AKI etapa 2
 - 30 pacientes (0.2% del total)
- Pacientes que empeoraron de AKI etapa 2 a AKI etapa 3:
 - 0 pacientes
- Pacientes con AKI 2 o 3 que bajaron a etapa 1 o 2:
 - 14 pacientes (0.0% del total)

Fue prioritario concentrarse en los pacientes que empeoraron. Aquellos que mejoraron no fueron foco de este estudio por el momento.

⁴ Es importante destacar que un paciente puede tener más de un episodio de AKI. Por ejemplo, si un paciente aumenta sus niveles de creatinina en un 50% es diagnosticado con AKI etapa 1. Si al día siguiente aumenta sus niveles en un 100%, será diagnosticado con AKI etapa 2.

4.3 DISCUSIÓN DE LOS RESULTADOS

Se corrobora y cuantifica la hipótesis planteada por el Jefe de la Unidad de Nefrología: El daño renal agudo es una enfermedad sub-diagnosticada en el Hospital Clínico de la Universidad de Chile.

Examinando a un mayor número de pacientes, el porcentaje con AKI podría ascender. Todos aquellos pacientes que sólo contaban con un examen no pudieron ser diagnosticados, pues no contaban con un examen basal. En particular la Universidad de Nottingham reporta un 18% de incidencia de AKI en la población.

4.4 IMPLICANCIAS ÉTICAS

Primero se debe destacar que los diagnósticos fueron realizados en base a la definición de la enfermedad según el criterio AKIN, por lo tanto, no hay lugar para falsos negativos. Falsos positivos podrían aparecer por fallas en la toma de los exámenes, pero luego de que estos son confirmados, la enfermedad debería ser diagnosticada. No diagnosticar a los pacientes si se tienen los exámenes para hacerlo, está al límite de ser una violación al artículo 22 del Código de Ética del Colegio Médico^{vi} que señala:

“Será negligente aquel profesional que poseyendo el conocimiento, las destrezas y los medios adecuados, no los haya aplicado. Actúa con imprudencia aquel médico que poseyendo los recursos y preparación necesarios para la atención de un paciente, los aplicare inoportuna o desproporcionadamente”.

Sin embargo, existe la salvedad que:

“La falta de recursos tecnológicos, cuya existencia no dependa del médico tratante, no acarrea responsabilidad alguna para el facultativo”.

En el centro hospitalario existe la tecnología para hacer el análisis, bastaría con acceder a la ficha del paciente e ir analizando todos los pares de exámenes que puedan gatillar el diagnóstico. Esto es impracticable, por lo que el software es necesario para resolver el problema.

5 CONCLUSIONES

Se construyó un software capaz de analizar la base de datos de hospital, para diagnosticar Daño Renal Agudo y obtener estadísticas sobre su incidencia.

En Chile, la Insuficiencia Renal Aguda nunca había sido caracterizada, no se tenía información sobre su incidencia, por eso ha levantado tanto interés.

Las tecnologías usadas en este proyecto son gratis y de fácil acceso, lo que no implica que sean de peor calidad que otras alternativas de pago, es más, estas son utilizadas por las empresas más grandes del mundo. Por ejemplo, Ruby on Rails fue utilizado en la creación de Twitter y es core de GitHub. Esto implica, que es posible realizar software de alto nivel, con pocos recursos y además en poco tiempo.

Resultó exitoso trabajar en conjunto con estudiantes de medicina y doctores con años de trayectoria, que tienen claridad sobre los problemas que deben ser resueltos. Esta colaboración interdisciplinaria, ayuda a resolver problemas que ninguna de las 2 áreas podrían resolver de manera independiente.

Para el caso particular del Hospital Clínico de la Universidad de Chile, el desarrollo de tecnología es lento, en consecuencia, el software está desactualizado. Sin embargo, la información está disponible, por lo que generar módulos que interactúen con dicha información es factible y deseable.

Por la importancia para la salud que podría tener la implementación de este software en otros recintos hospitalarios, se desarrolló una versión con mayor visibilidad (disponible en <http://www.akihunter.com>).

Esta versión permite a cualquier persona navegar por la plataforma para ver su funcionamiento, gráficamente es más atractiva que el software original y está diseñada para ser implementada otras instituciones.

Los resultados de este trabajo son interesantes para la comunidad científica y han sido presentados en las conferencias International Student Congress of (bio)Medical Sciences ISCOMS (Holanda), en el XXXVI Congreso Chileno de Medicina Interna (Chile), y en el Congreso Conjunto de Nefrología e Hipertensión (Chile), ganando el premio al mejor poster en la última.



Figura 15: Perfil de un paciente en la nueva versión del sitio

5.1 TRABAJO FUTURO

Luego de los descubrimientos encontrados en este trabajo, nace la responsabilidad de aterrizar el software a un proyecto que sea de utilidad para el paciente.

Esta responsabilidad será asumida por Rubén Torres, José Miguel Vicente y Waldo Uribe, quienes acordaron formar una empresa que permita continuar desarrollando software médico de manera económicamente sustentable.

La siguiente etapa empieza el 2016 con la postulación a Startup Chile, programa que entrega financiamiento por 20 millones de pesos y redes de contacto para el emprendimiento. El programa dura 6 meses y será utilizado para validar comercialmente el software de detección de Daño Renal Agudo.

Existen avances en el sector privado. Se concretó una reunión con Andres Bolstanksy, Profesor Asociado de la Universidad de los Andes y Jefe del Departamento de Nefrología de la Clínica Davila, quién está interesado en implementar el software en la clínica para monitorear la incidencia del Daño Renal Agudo en los pacientes hospitalizados.

En Chile la cantidad de pacientes dializados asciende a 17,000^{vii}. Para el gobierno, cada paciente dializado cuesta alrededor de CLP 800,000 mensuales, este tratamiento representa el mayor costo de la salud Chilena.

Existe una preocupación a nivel internacional por las muertes provocadas por AKI. La iniciativa 0by25^{viii}, busca reducir a 0 las muertes prevenibles por AKI el 2025.

Estos motivos son suficientes para dedicar tiempo y recursos en la ejecución de este proyecto, trabajar para que Chile cumpla 0by25 idealmente antes del plazo propuesto.

6 BIBLIOGRAFÍA

ⁱ Centro de prensa Organización mundial de la salud (2010). Las 10 causas principales de defunción en el mundo.

[en línea] <http://bit.ly/1WhkZxU>

ⁱⁱ José António Lopes y Sofia Jorge (2012). The RIFLE and AKIN classifications for acute kidney injury: a critical and comprehensive review.

[en línea] <http://bit.ly/1ZXDPs4>

ⁱⁱⁱ Renal and Transplant Unit, Nottingham University Hospitals NHS Trust, Nottingham, UK (2014). A real-time electronic alert to improve detection of acute kidney injury in a large teaching hospital.

[en línea] <http://1.usa.gov/1Y9xIiH>

^{iv} Randle Browning (2015). 13 Things you need to know about Ruby on Rails

[en línea] <http://bit.ly/1S0ddjV>

^v Jacob Nielsen (2000). Why You Only Need to Test with 5 Users.

[en línea] <http://bit.ly/1IbBiWV>

^{vi} Código Médico de Chile AG (2008). Código de ética.

[en línea]

<http://web.uchile.cl/archivos/derecho/CEDI/Normativa/C%F3digo%20de%20C9tica%20del%20Colegio%20M%E9dico%20de%20Chile.pdf>

^{vii} Universia Chile (2013). Crece el número de pacientes dializados en Chile.

[en línea] <http://bit.ly/233JUsh>