

DOCUMENTACIÓN DE ARQUITECTURA Y DISEÑO DE SOFTWARE

Índice:

1. Integrantes
2. Enunciado del proyecto
3. Restricciones
4. Casos de uso
5. Atributos de calidad
 - Desempeño
 - Escalabilidad
 - Disponibilidad
 - Mantenibilidad
 - Interoperabilidad
 - Usabilidad
 - Autenticación
 - Confidencialidad
 - Integridad de datos
6. Resumen de métricas
7. Vistas arquitecturales
 - Vista de componentes
 - Vista de contexto
 - Vista de desarrollo
 - Vista de despliegue
 - Vista de información
8. Capturas de usabilidad
 - Aplicación móvil
9. Capturas de usabilidad
 - Aplicación web
10. Resultados obtenidos

1. Integrantes

Nombre	Código	Correo
Lady Jeraldynne Pinzon Capera	201513489	lj.pinzon12
Sebastián Martínez Castaño	201416607	s.martinez15
Laura Natalia Bello Jiménez	201512763	ln.bello10
Camilo Zambrano Votto	201515438	c.zambrano10

2. Enunciado del proyecto

CONTEXTO

El hospital Cardiológico de Santa Fe es una institución privada en Bogotá con más de 30 años de experiencia en el diagnóstico, cuidado y manejo de enfermedades cardíacas. El hospital es muy eficiente en el área de emergencias y cirugías, su equipo ha tratado cientos de casos complejos y es referente a nivel nacional en el área de cardiología.

PROBLEMA

Con toda su experiencia, las directivas desean mejorar su división de prevención, rehabilitación y cuidados externos. Actualmente, el hospital sólo dispone de 500 camas para pacientes en diversos grados de salud y atiende alrededor de 1000 pacientes al mes, los cuales pueden estar internados o seguir algún tratamiento posoperatorio o de cuidados ambulatorios desde sus hogares.

DESCRIPCIÓN DEL SISTEMA A CONSTRUIR

Los pacientes en postoperatorio o rehabilitación deben ir al hospital una vez por semana para realizarse un conjunto de exámenes de acuerdo a su cardiopatía y así saber el estado de salud actual y ajustar el tratamiento de ser necesario.

El hospital ve este proceso como algo dispendioso y complicado ya que cerca del 50 % de sus pacientes tiene que cambiar el horario de sus exámenes durante la semana y casi un 80 % de ellos cancela una cita por lo menos una vez al mes. La inasistencia afecta el plan de rehabilitación de los pacientes y los pone en riesgo de recaídas de salud o nuevas condiciones producto de la desatención.

El hospital, se ha aliado con Thing Worx S.A.S. para darle a sus pacientes en rehabilitación unos brazaletes inteligentes que recopilan información cardíaca relevante para las diferentes cardiopatías (i.e., frecuencia cardíaca, presión sanguínea, nivel de estrés). Desde un punto de vista operacional los componentes del sistema interactúan como sigue:

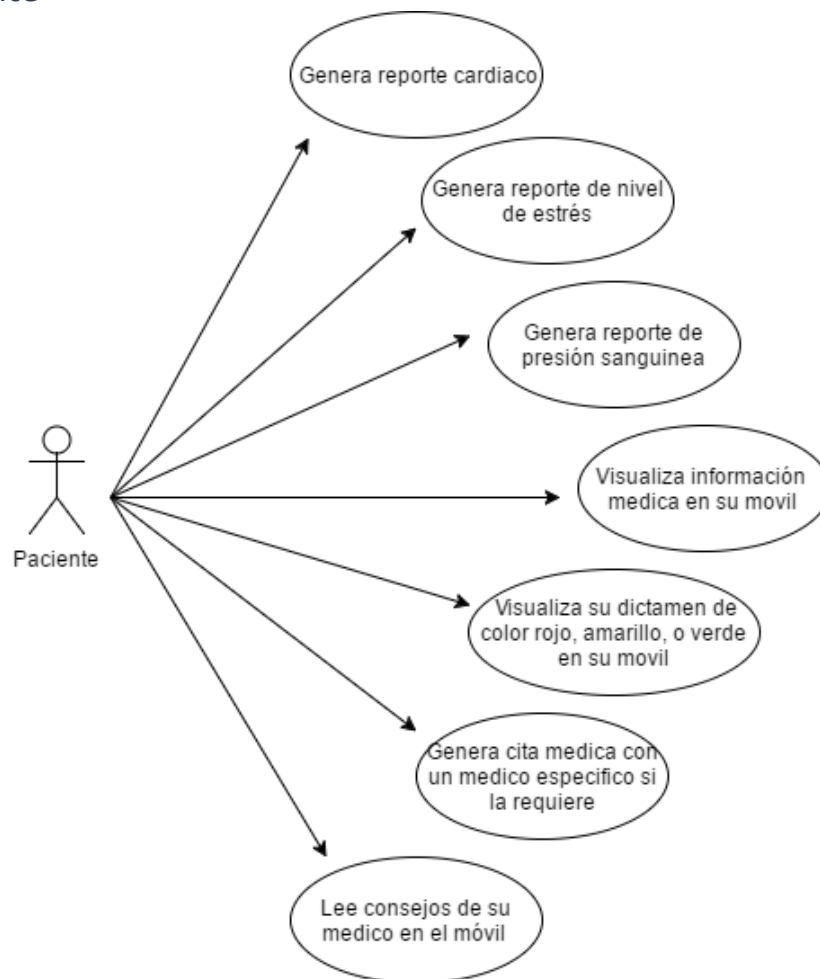
- El dispositivo le recuerda al paciente que debe medir la información cardíaca
- Una vez hecha la medición, el brazalete se sincroniza con un dispositivo móvil del paciente con el fin de:
 - Mostrar al paciente dicha información y un dictamen que se encuentra en cualquiera de 3 niveles: rojo (requiere atención médica de urgencia ya que es un paciente de alto riesgo, i.e., recientemente infartado o con cirugía mayor), amarillo (debe darse un consejo relacionado a la higiene de vida, e.g., relajarse y alejarse de una situación de estrés cuando se detecte un ascenso en la tensión) y verde (todo está bien, siga con su higiene de vida). De igual manera, debe ser posible ver la información cardíaca reportada en un período de tiempo a través de gráficas simples en el dispositivo.
 - Reportar al hospital la información cardíaca del paciente periódicamente.
 - Cuando la condición de salud se encuentra en nivel rojo, el dispositivo debe notificar a la ambulancia, los servicios de emergencia del hospital y al médico tratante de la ubicación (GPS) y condición actual del paciente para coordinar la atención y tomar el menor de tiempo posible para llegar a donde el paciente y atenderlo. La emergencia debe quedar guardada en el sistema.
- A partir de la información cardíaca reportada, los médicos pueden enviar a los dispositivos consejos de cuidado relacionados a dieta, actividad física, toma de medicamentos, asistencia a citas médicas. Sólo los médicos especialistas autorizados pueden reconfigurar los parámetros de los marcapasos para regular el ritmo cardíaco de un paciente, primero se reporta al dispositivo inteligente y luego de éste al marcapasos.
- Cada paciente puede tener uno o más de un médico asignado en cualquier momento de su tratamiento y que cada una de las decisiones, sea tratamientos, exámenes o diagnósticos deben estar completamente documentados dentro de la historia clínica digital, la cual se utiliza como sustento para el pago y notificación de procedimientos con las EPS.

3. Restricciones

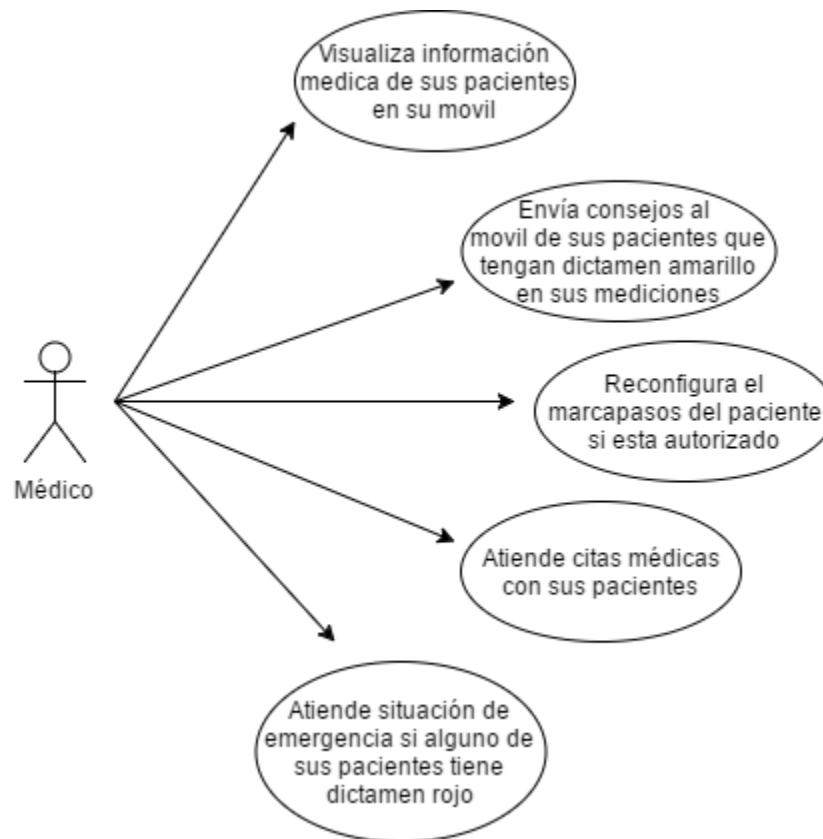
Identificador	Tipo	Descripción
RE-01	Negocio	El proyecto deber ser desarrollado en un tiempo límite.
RE-02	Tecnología	La aplicación debe poder recibir información de sensores por medio del uso de tarjetas wiring.

4. Casos de uso

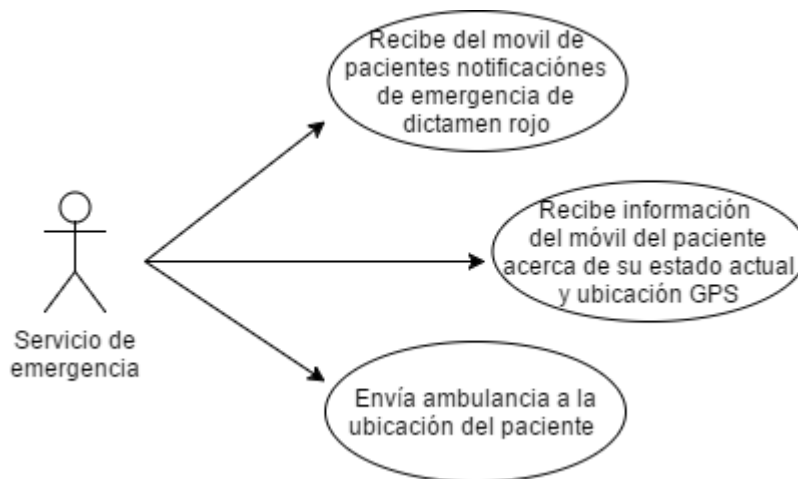
- Paciente



- Médico



- Servicio de emergencia



5. Atributos de calidad

- Desempeño

Escenario	3
Identificador	EC3
Prioridad	Alta
Atributo	Desempeño
Fuente	Sensores
Ambiente	Normal
Estímulo	Carga de datos sobre el sistema
Medida de respuesta	Se recibe información hasta de 1000 notificaciones de sensores en una ventana de tiempo de 1 segundo

- Escalabilidad

Escenario	5
Identificador	EC5
Prioridad	Alta
Atributo	Escalabilidad
Fuente	Pacientes inscritos en la aplicación
Ambiente	Saturado, post y durante desarrollo
Estímulo	Uso del servicio login de la aplicación medicAndes
Medida de respuesta	Se espera que el sistema permita realizar por lo menos 1000 "login" de usuarios y recupere correctamente su información en menos de 30 segundos cada solicitud

- Disponibilidad

Escenario	1
Identificador	EC1
Prioridad	Media
Atributo	Disponibilidad
Fuente	Condiciones externas
Ambiente	Normal
Estímulo	Todos los casos de uso
Medida de respuesta	Tiempo de recuperación ante fallas eléctricas en menos de un segundo

Escenario	9
Identificador	EC9
Prioridad	Media
Atributo	Disponibilidad
Fuente	Paciente
Ambiente	Normal
Estímulo	Solicitud de generar mediciones
Medida de respuesta	El sistema debe poder resolver el 98% de las solicitudes realizadas a lo largo del año.

- Mantenibilidad

Escenario	2
Identificador	EC2
Prioridad	Baja
Atributo	Mantenibilidad

Fuente	Equipo de desarrollo
Ambiente	Ambiente de pruebas
Estimulo	Detector de errores en el código
Medida de respuesta	La cobertura es mayor o igual al 90% en todo el proceso de desarrollo

- Usabilidad

Escenario	7.1
Identificador	EC71
Prioridad	Alta
Atributo	Usabilidad
Fuente	Paciente
Ambiente	Normal
Estímulo	Dictamen del paciente de color amarillo
Medida de respuesta	El médico afirma encontrar y recordar el procedimiento para recibir la notificación de dictamen y responder con el mensaje medico al paciente

Escenario	7.2
Identificador	EC72
Prioridad	Alta
Atributo	Usabilidad
Fuente	Paciente
Ambiente	Normal

Estímulo	Dictamen del paciente de color rojo
Medida de respuesta	El médico afirma encontrar y recordar el procedimiento para recibir la notificación de dictamen y responder que asistirá a la emergencia

Escenario	7.3
Identificador	EC73
Prioridad	Alta
Atributo	Usabilidad
Fuente	Sensores
Ambiente	Normal
Estímulo	La toma de mediciones de frecuencia cardiaca, nivel de estrés y presión
Medida de respuesta	El paciente afirma encontrar y recordar el procedimiento para visualizar su información médica de cada medición

- Autenticación

Escenario	8.1
Identificador	EC81
Prioridad	Alta
Atributo	Autenticación
Fuente	Paciente
Ambiente	Normal

Estímulo	Ingresar a la aplicación medicAndes
Medida de respuesta	Se requirió de al menos una autenticación previa del paciente con usuario y contraseña para poder usar la aplicación

Escenario	8.2
Identificador	EC82
Prioridad	Alta
Atributo	Autenticación
Fuente	Medico
Ambiente	Normal
Estímulo	Ingresar a la aplicación medicAndes
Medida de respuesta	Se requirió de al menos una autenticación previa del médico con usuario y contraseña para poder usar la aplicación

- Confidencialidad

Escenario	6
Identificador	EC6
Prioridad	Alta
Atributo	Confidencialidad
Fuente	Paciente
Ambiente	Normal
Estímulo	La frecuencia del marcapasos ya

	no es funcional para el paciente
Medida de respuesta	MedicAndes debe implementar un sistema de autenticación y autorización el cual solo le permite al médico especialista cambiar la frecuencia del marcapasos del paciente

- Integridad de datos

Escenario	4
Identificador	EC4
Prioridad	Alta
Atributo	Integridad de datos
Fuente	Sensores
Ambiente	Normal
Estímulo	La toma de mediciones de frecuencia cardiaca, nivel de estrés y presión
Medida de respuesta	El 100% de los datos entre los sensores y la aplicación son encriptados

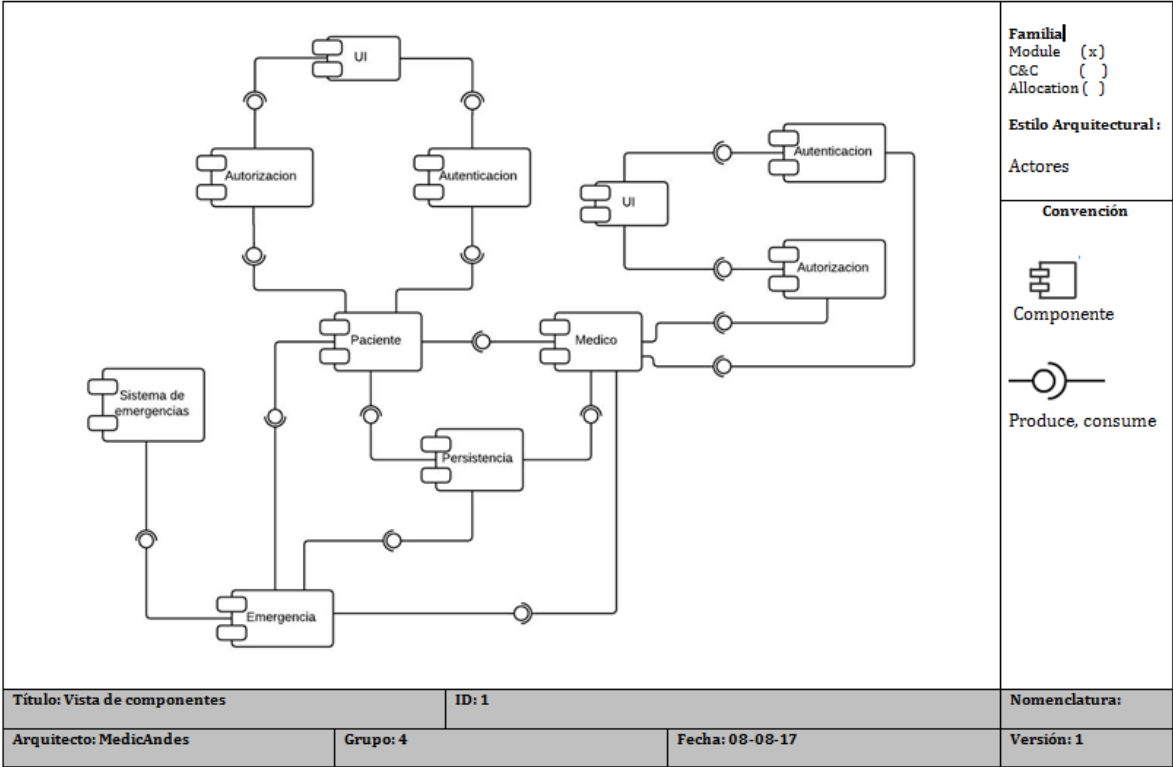
6. Resumen de métricas

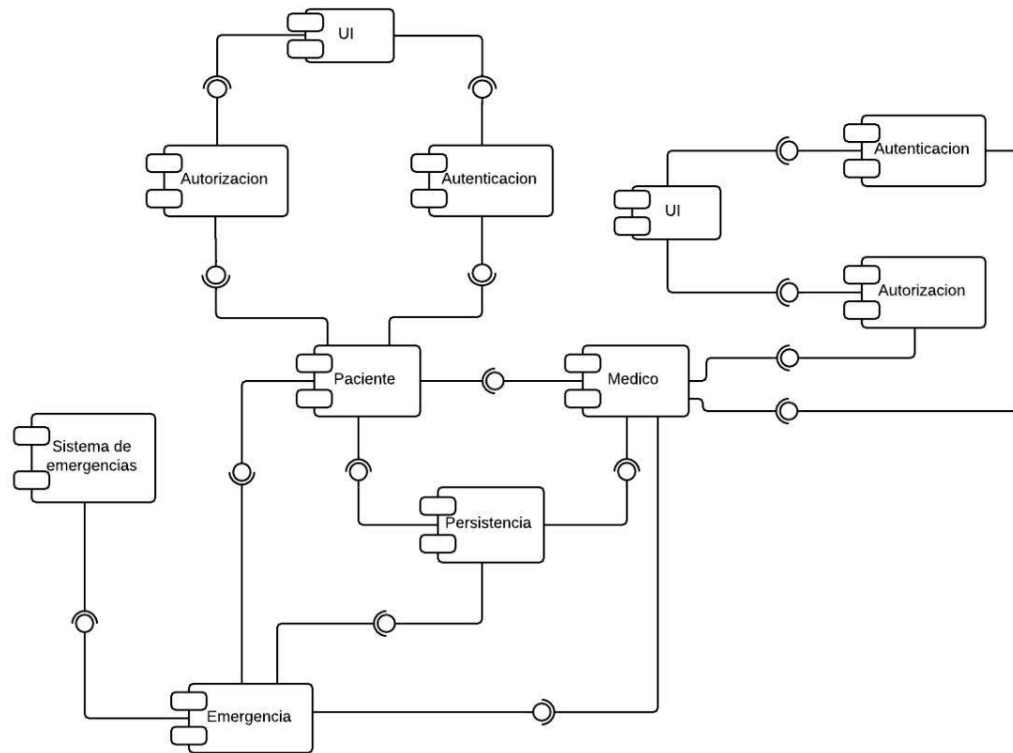
Escenario de calidad	Métrica	Resultado esperado	Resultado obtenido
EC3	Latencia	Menor o igual a 1000 milisegundos.	1940 milisegundos.
EC5	#Num usuarios exitosamente logueados en 1 segundos	1000	1000

EC1	Grado de Disponibilidad (porcentual-/ anual)	98%	No es medible en este curso
EC4	Porcentaje de mensajes encriptados	100%	No aplica todavía.
EC71,EC72, EC73	Porcentaje de usuarios que afirman ubicar y recordar las funcionalidades	90%	No aplica todavía.
EC2	Porcentaje de cobertura en todo el proceso de desarrollo.	90%	No aplica todavía

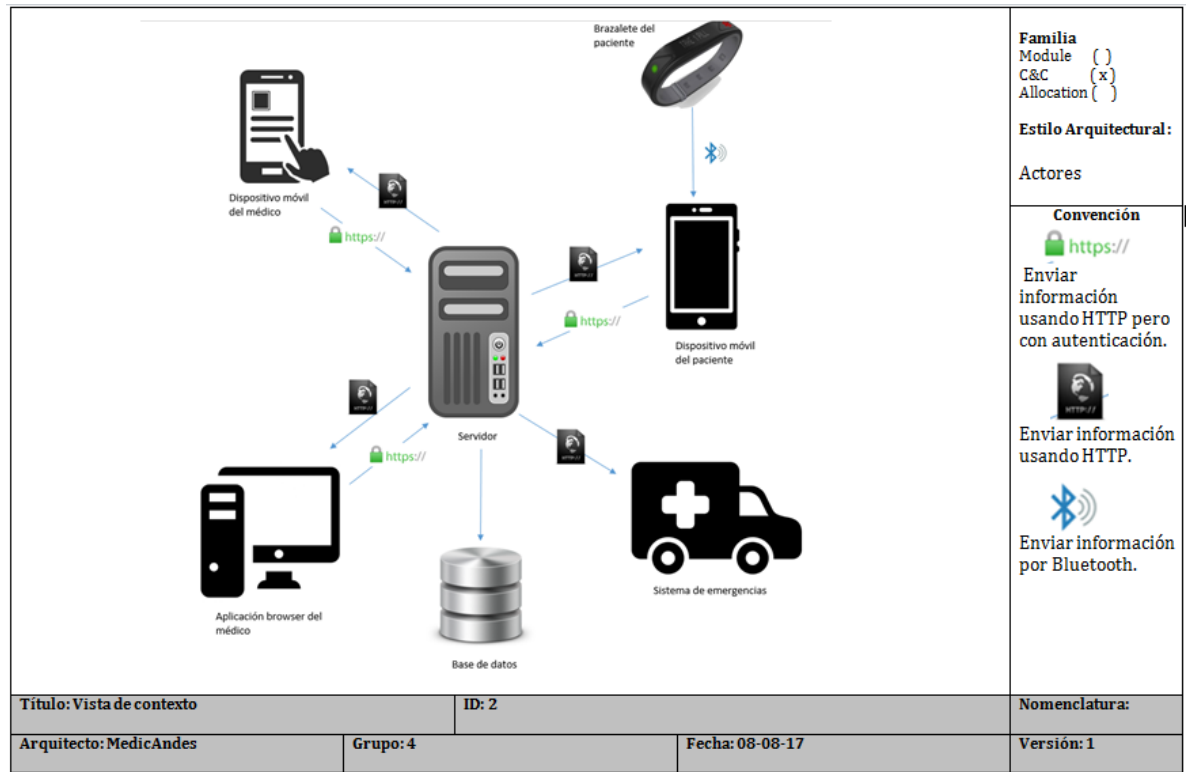
7. Vistas arquitecturales

- Vista de componentes



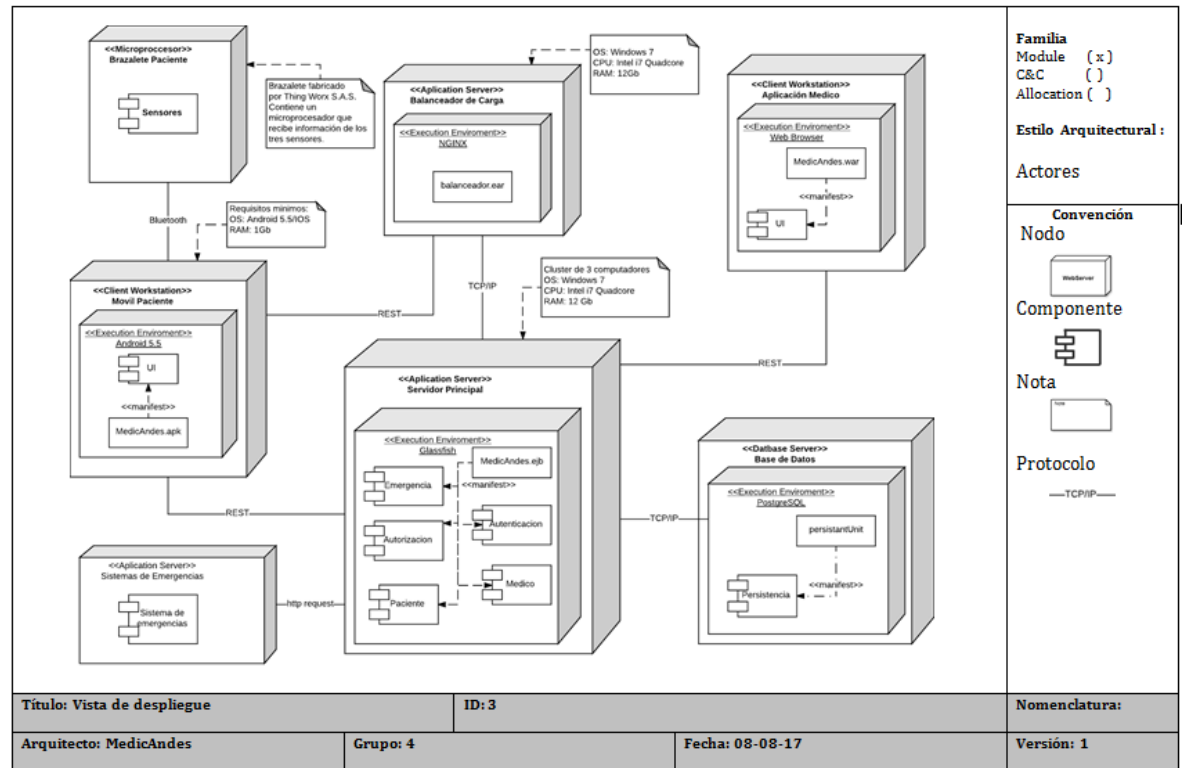


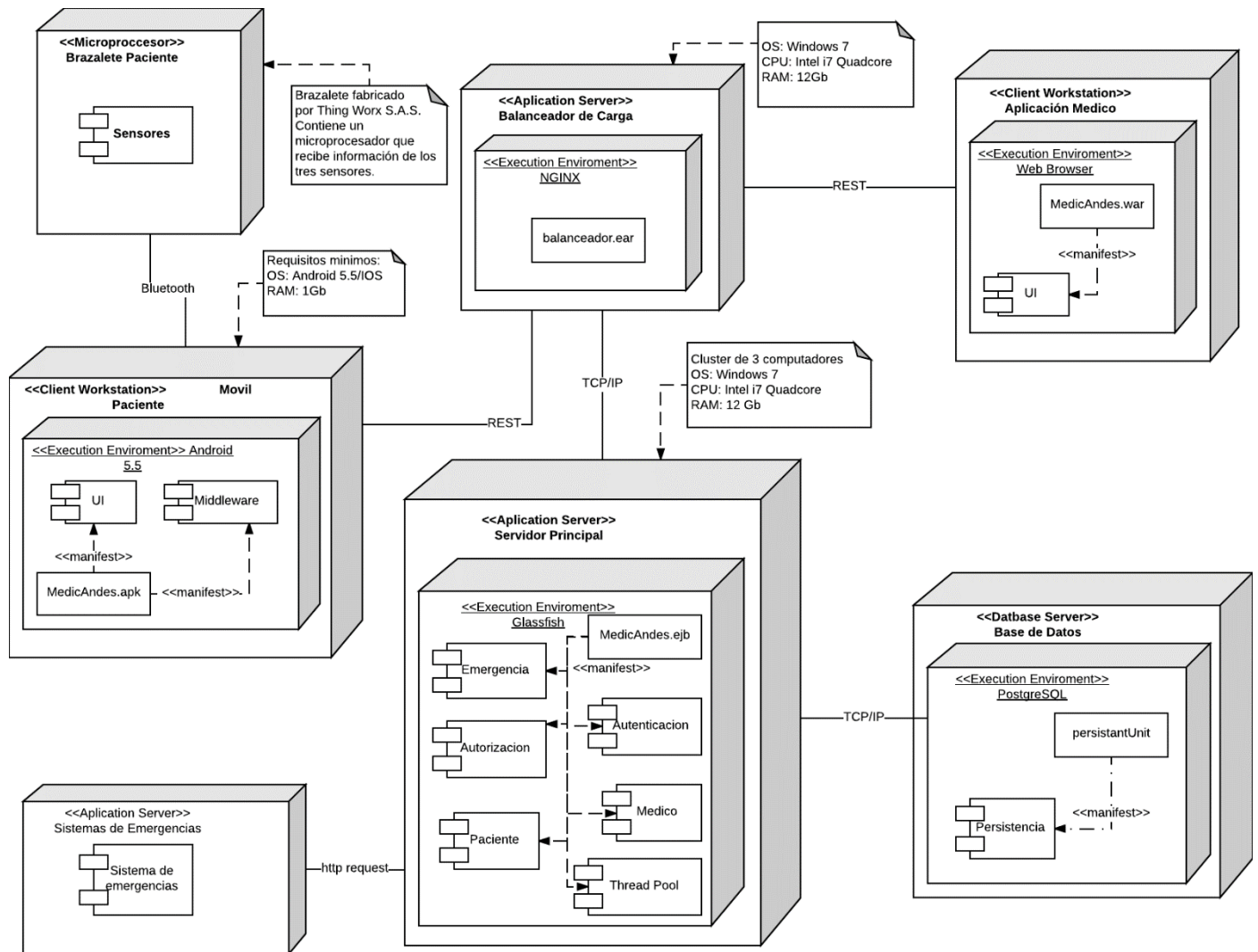
- Vista de contexto



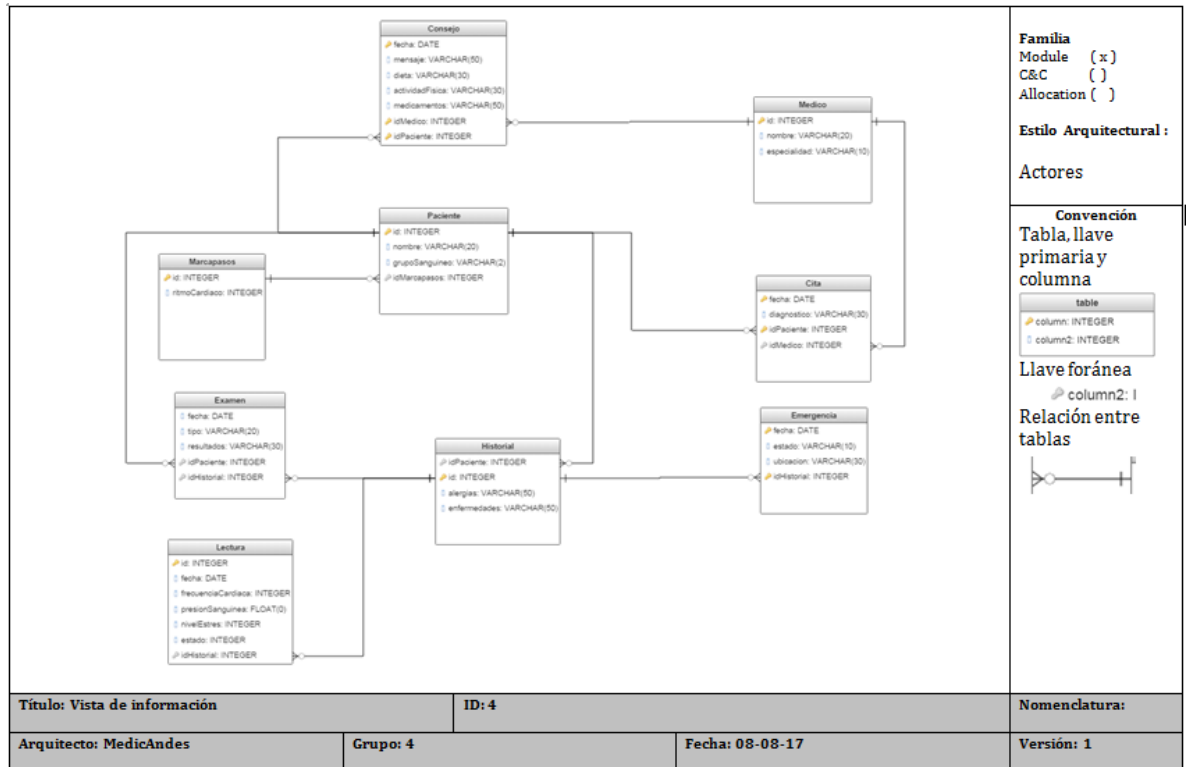


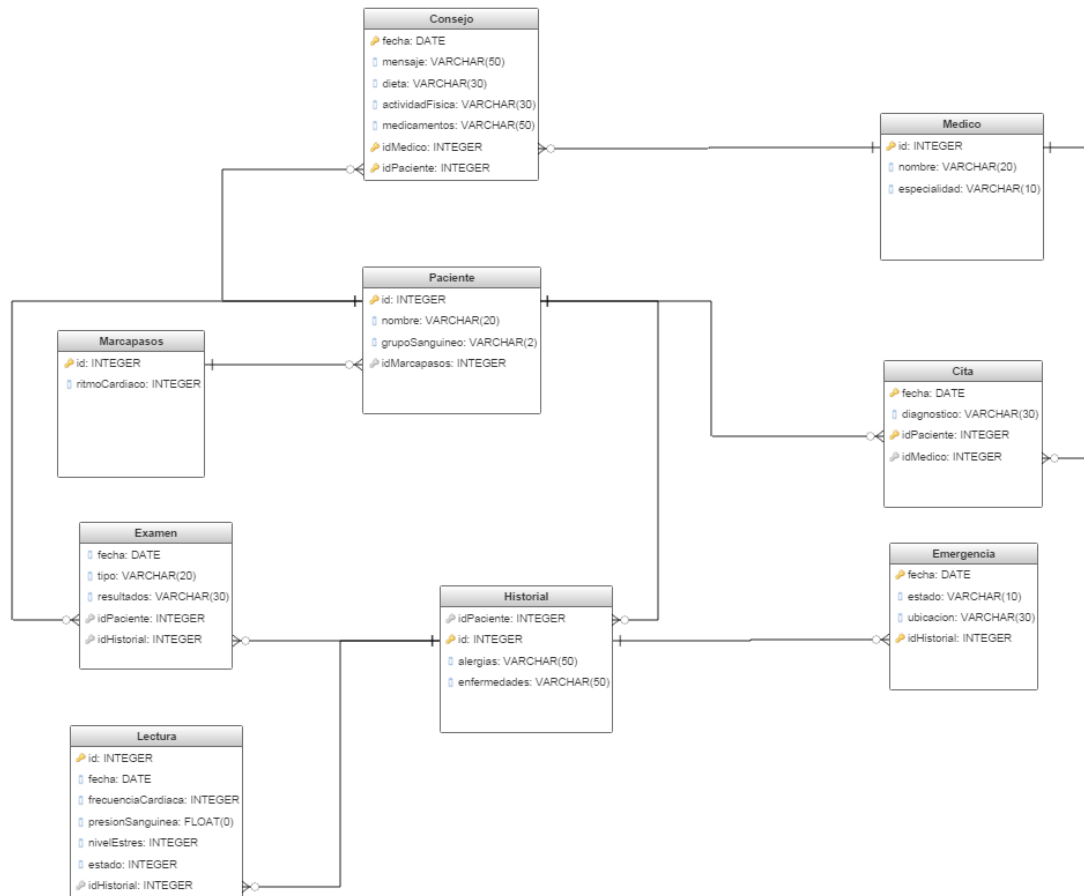
• Vista de despliegue



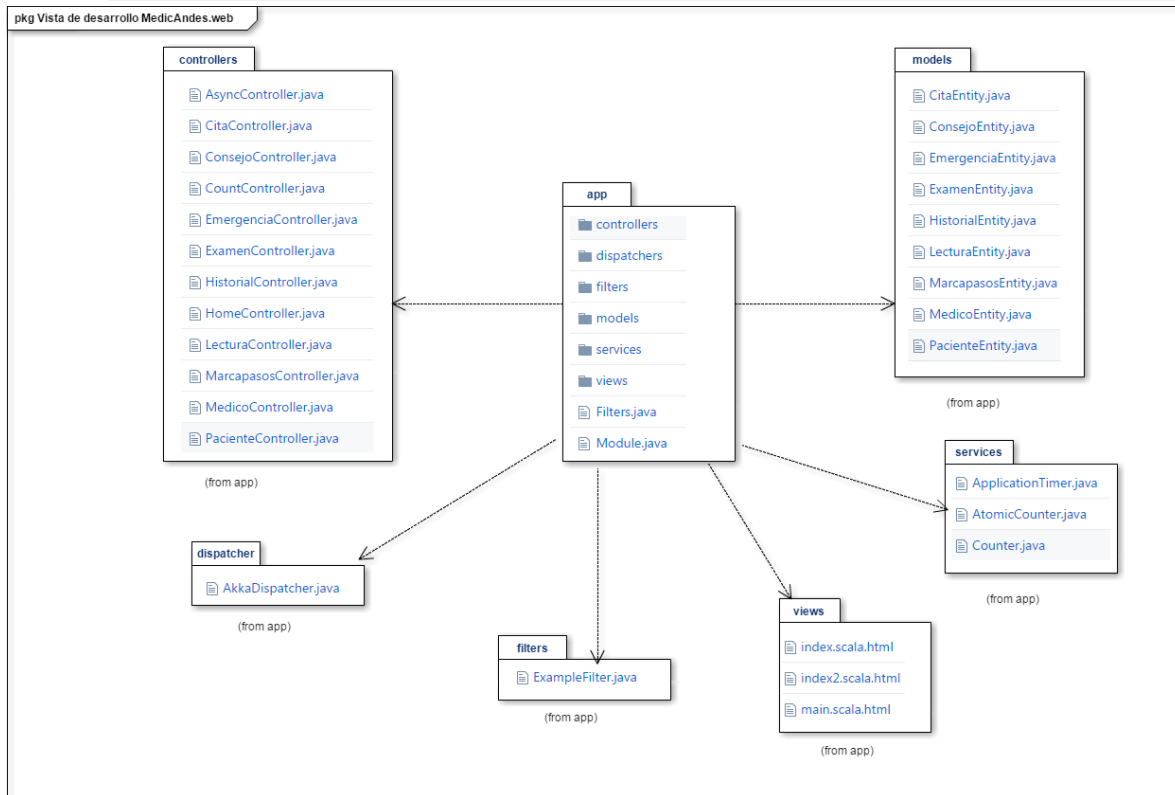
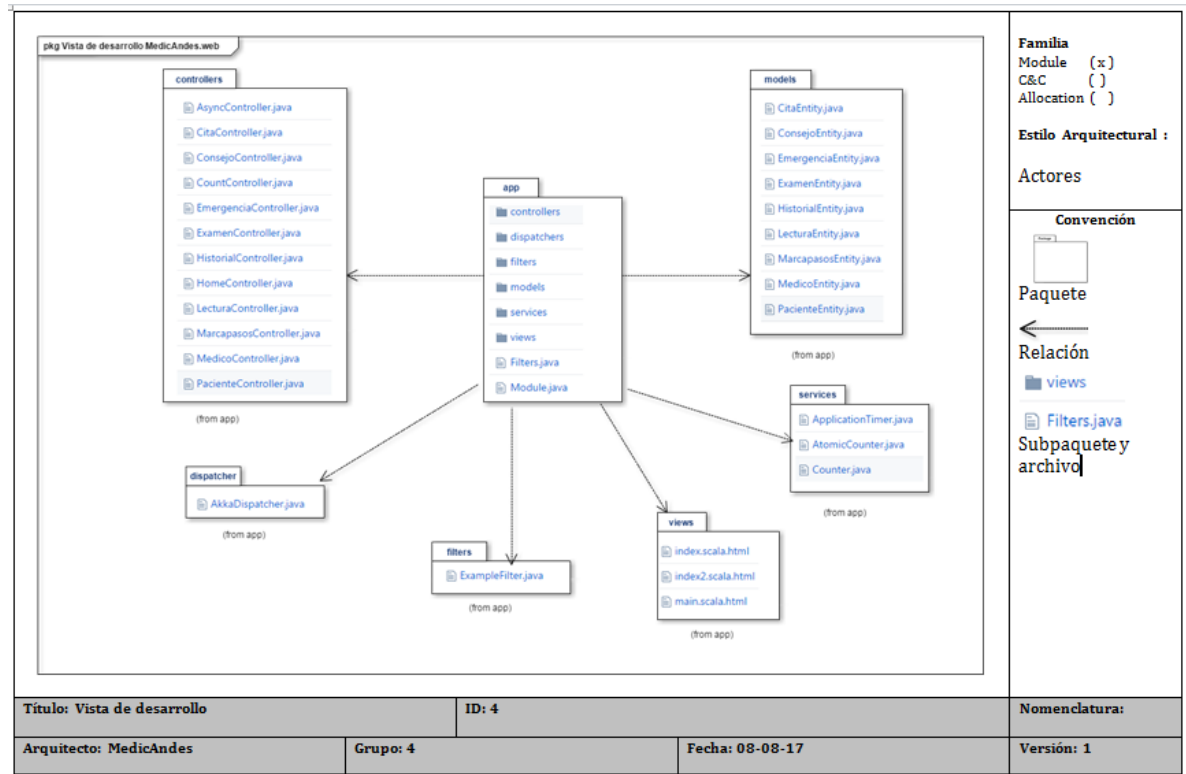


• Vista de información

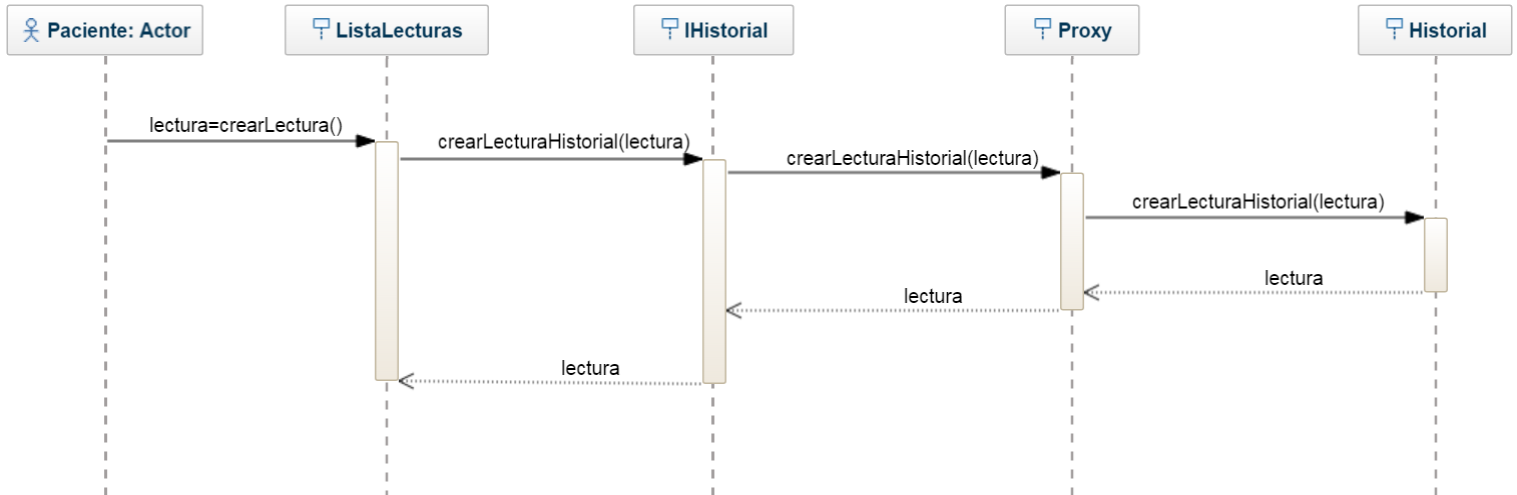




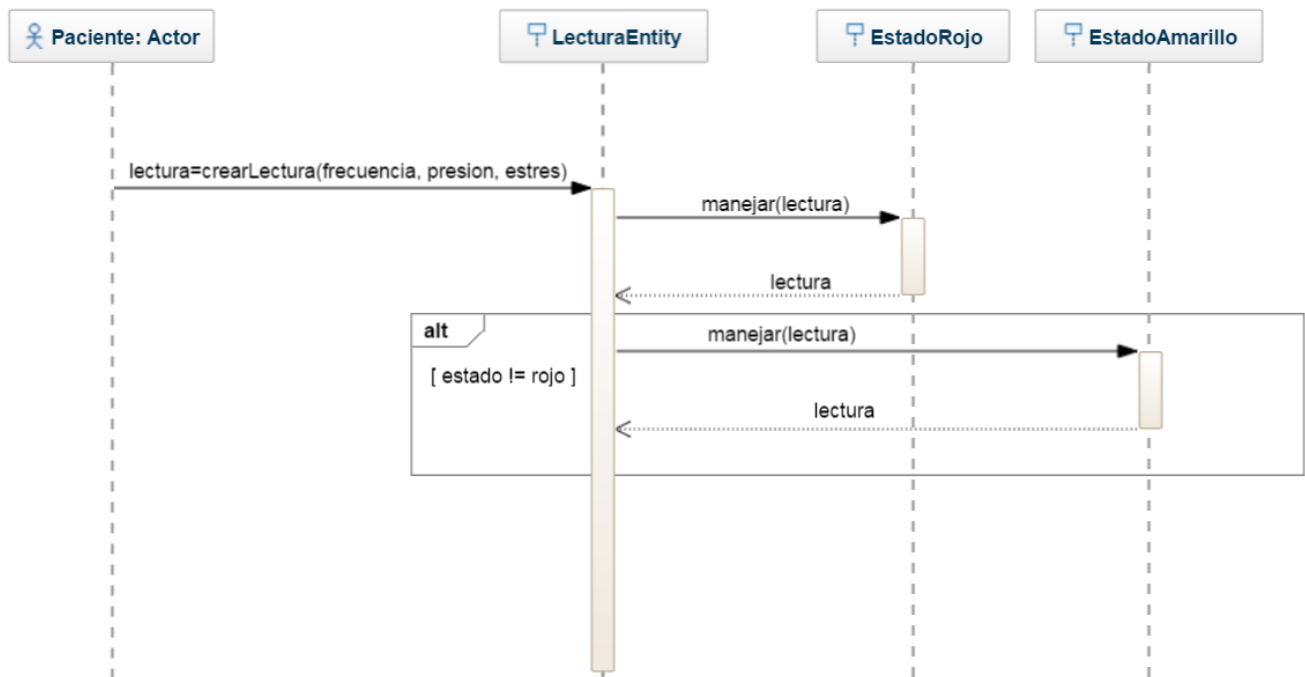
- Vista de desarrollo



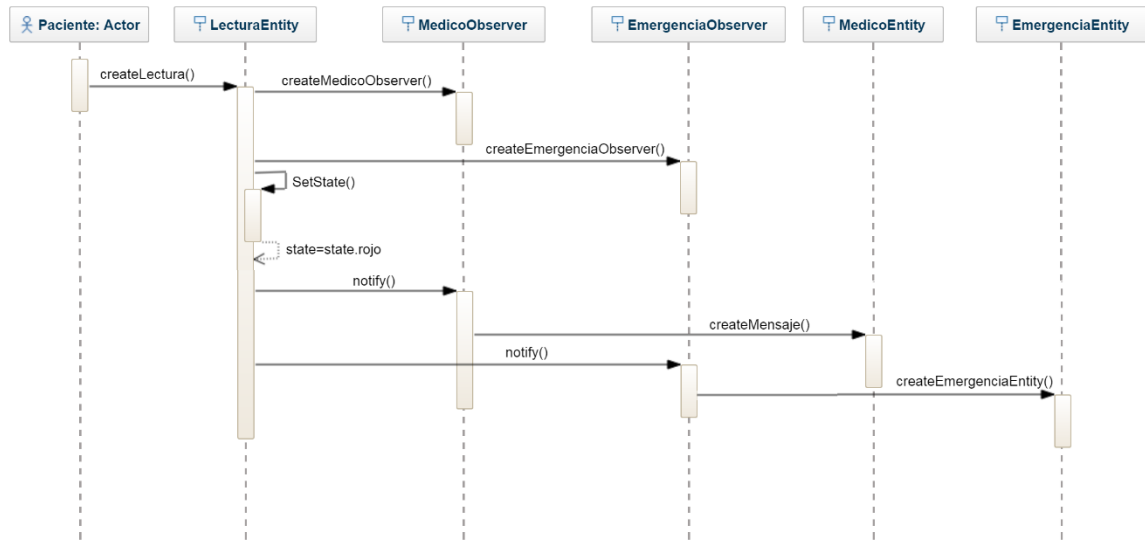
- Vista de secuencia
 - Modificabilidad
- Patrón Proxy



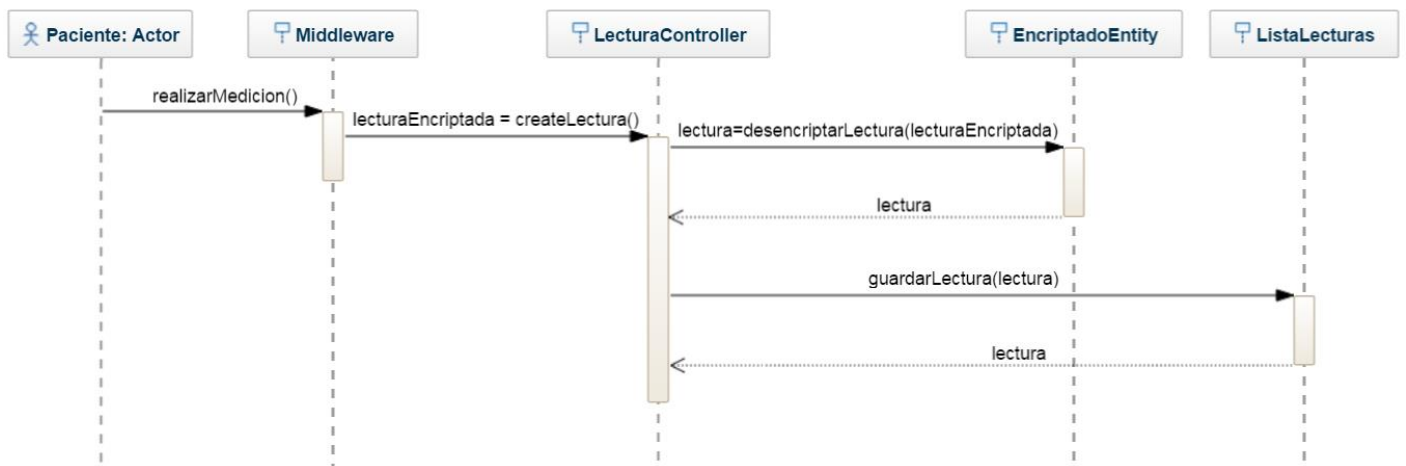
Patrón Estado



Patrón Observer



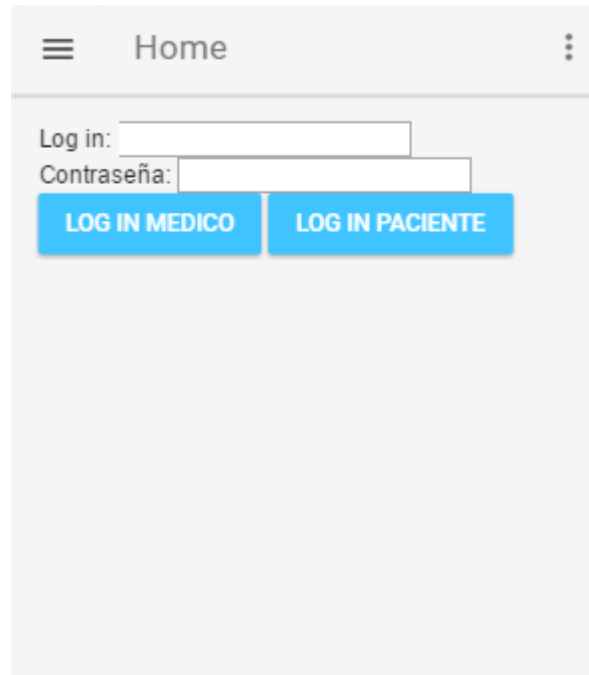
- Seguridad



8. Capturas de usabilidad: desarrollo móvil

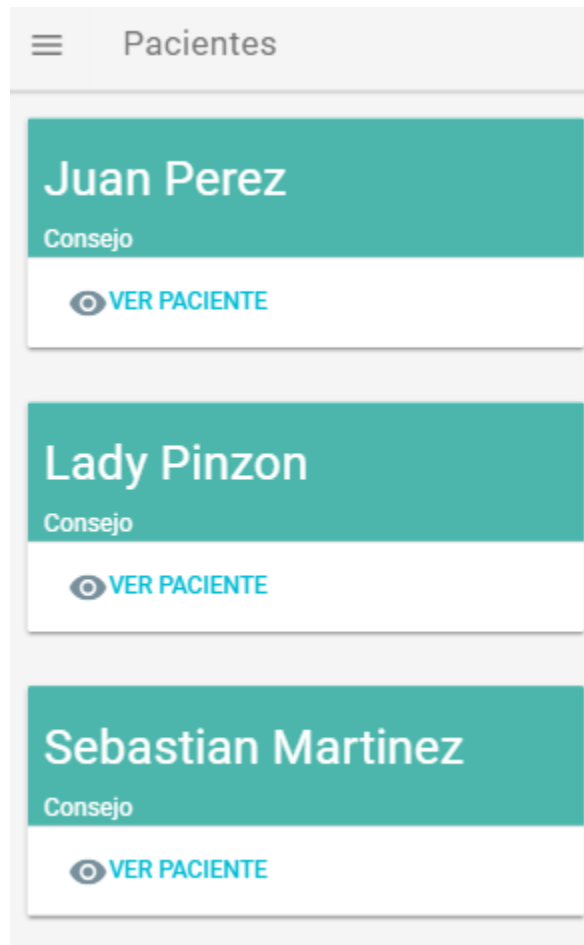
Capturas de aplicación móvil

Captura del log in:

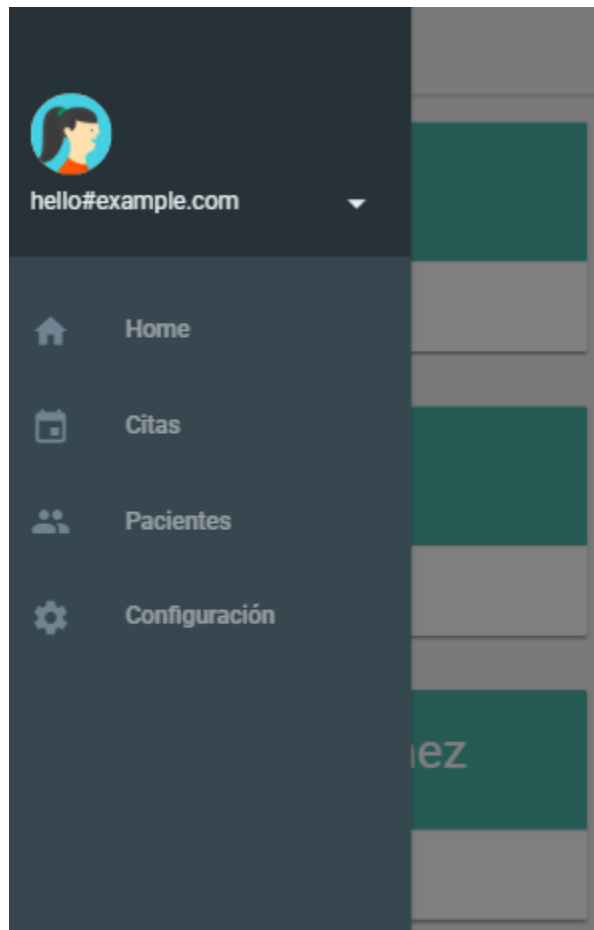


The screenshot shows a mobile application interface for a login screen. At the top, there is a header bar with a hamburger menu icon on the left, the word "Home" in the center, and a vertical ellipsis icon on the right. Below the header, the login form consists of two text input fields: "Log in:" and "Contraseña:". Below these fields are two blue buttons with white text: "LOG IN MEDICO" and "LOG IN PACIENTE".


Captura de la vista de pacientes de un médico:



Captura de la barra de información en la izquierda:



Captura de la información que un médico puede ver de un paciente:

 Juan Perez

[< REGRESAR](#) [CONSEJO](#)

Cantidad emergencias:

1

Alergias:

ninguna

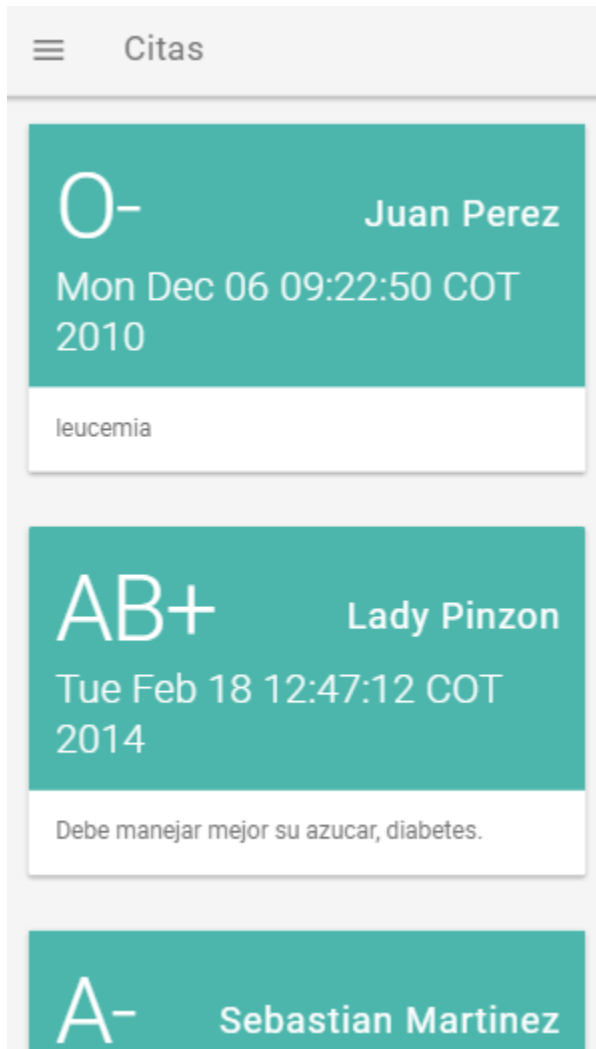
Enfermedades:

Leucemia

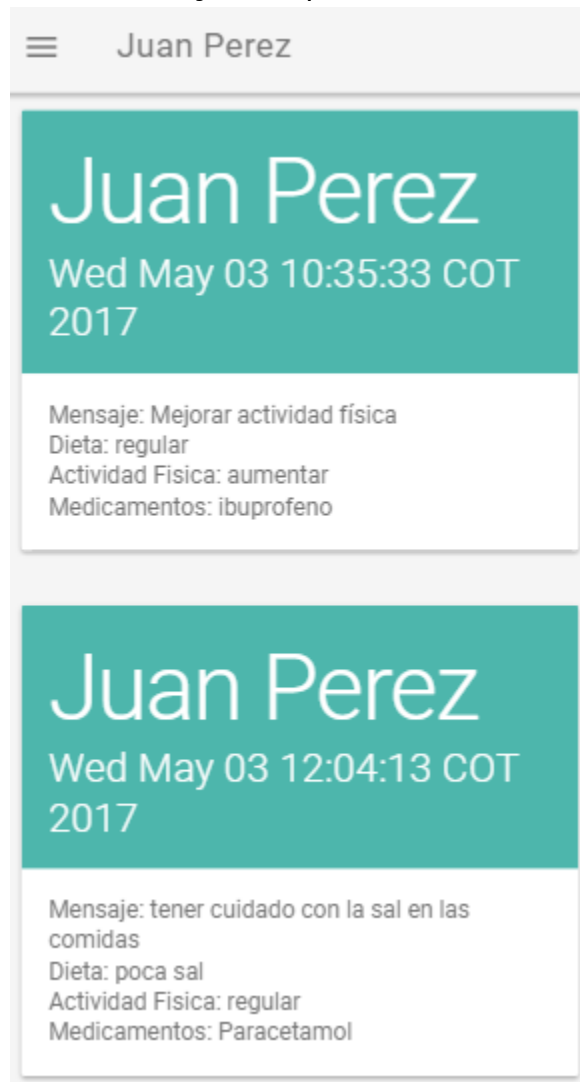
Captura del envío de un mensaje al paciente por un médico:



Captura del historial de citas de un médico:

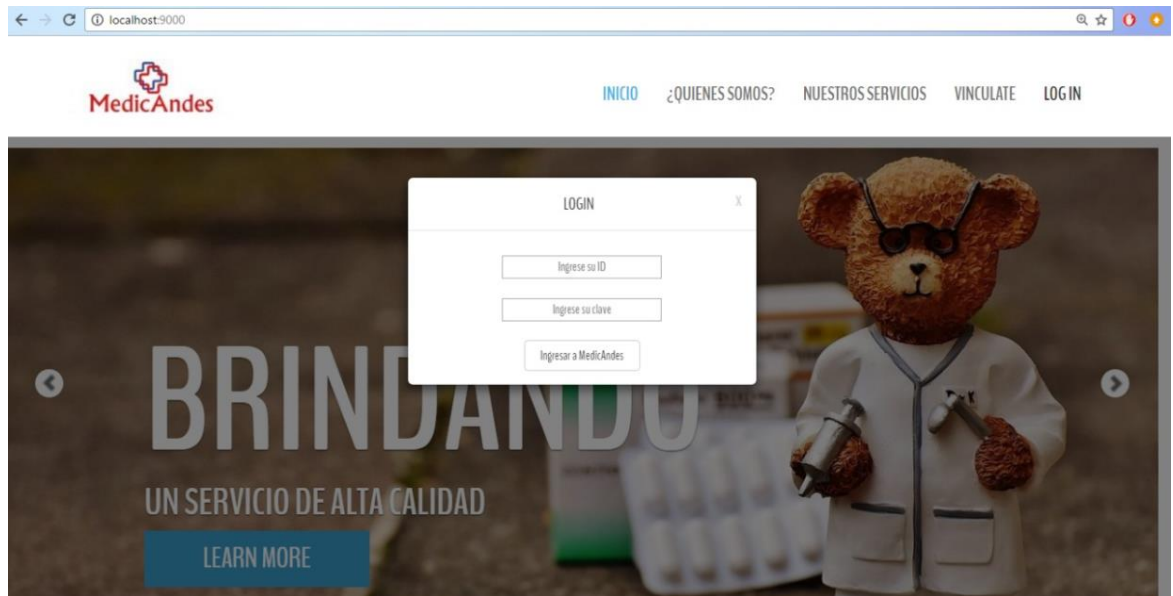


Captura del historial de consejos del paciente

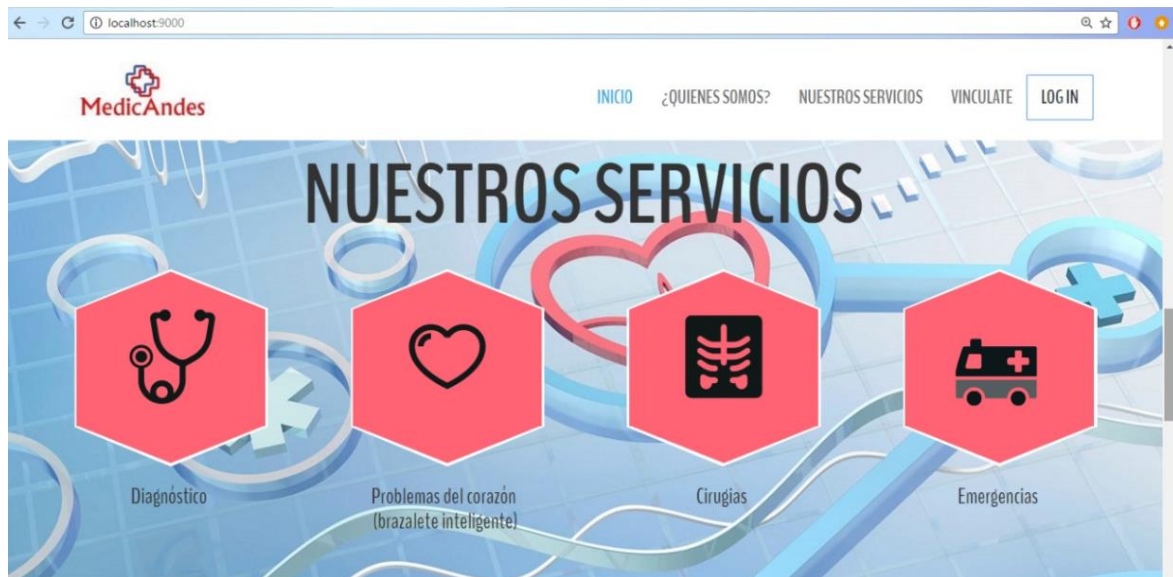


9. Capturas de usabilidad: desarrollo móvil

- Página de bienvenida donde el usuario medico puede loguearse



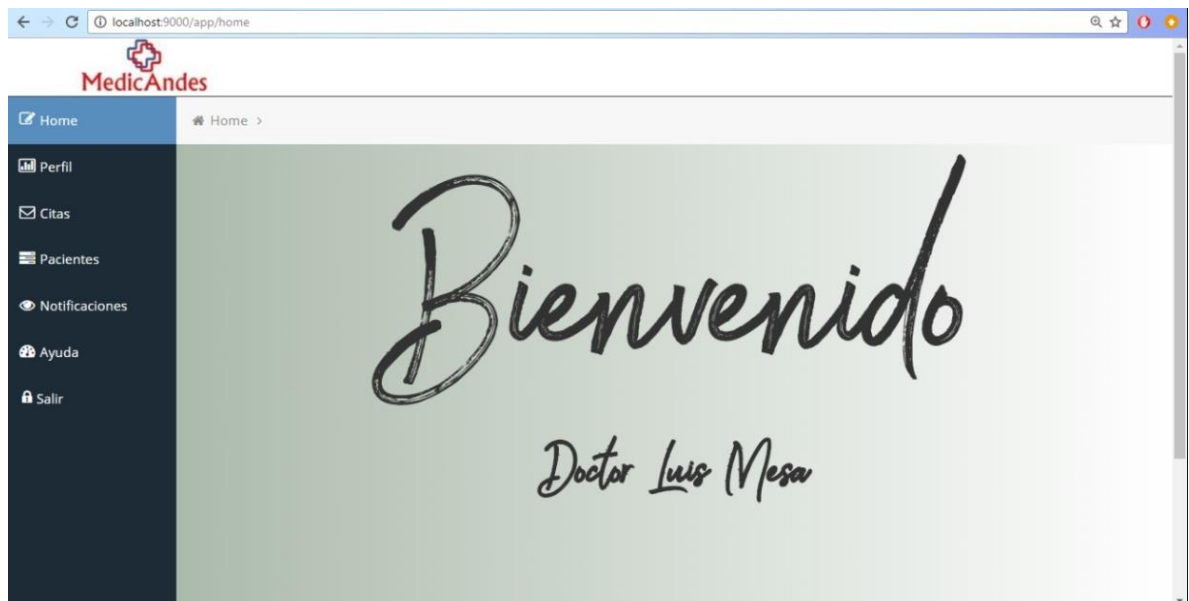
- Información básica de la entidad de medicina



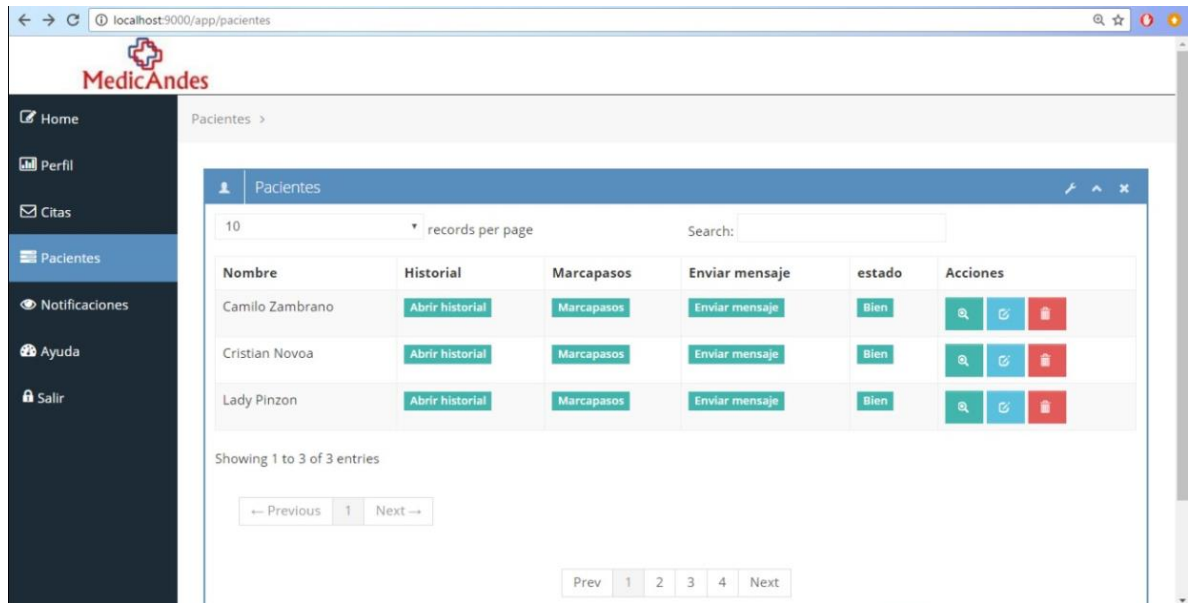
- Información básica de la entidad de medicina



- Página de bienvenida al médico al loguearse



- Visualización de pacientes de ese médico



10 records per page Search:

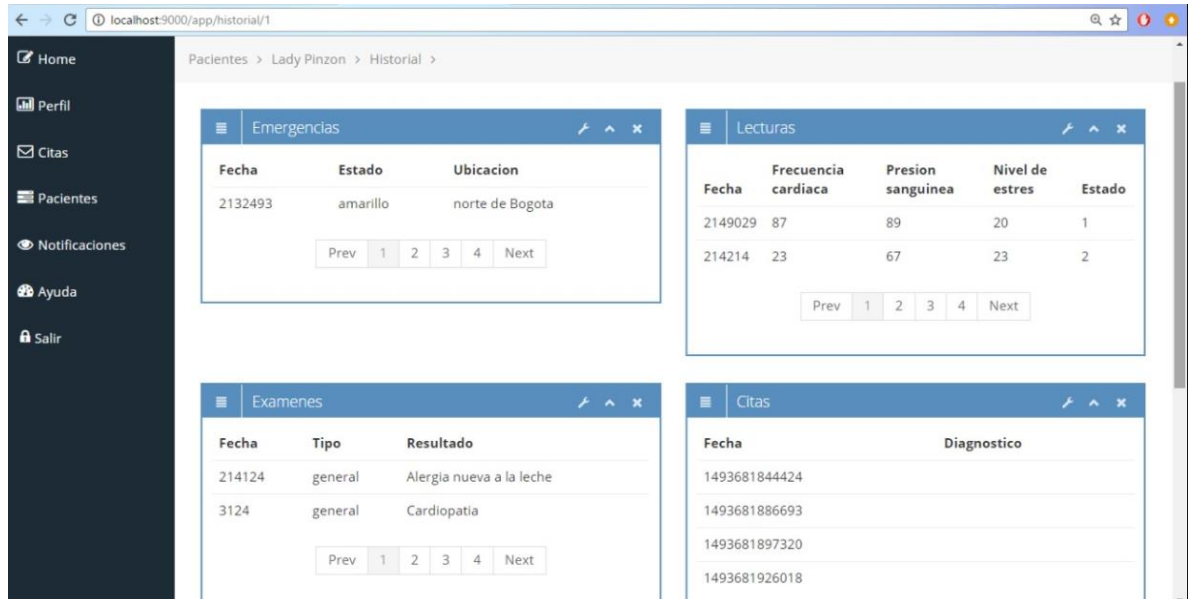
Nombre	Historial	Marcapasos	Enviar mensaje	estado	Acciones
Camilo Zambrano	Abrir historial	Marcapasos	Enviar mensaje	Bien	Buscar Actualizar Eliminar
Cristian Novoa	Abrir historial	Marcapasos	Enviar mensaje	Bien	Buscar Actualizar Eliminar
Lady Pinzon	Abrir historial	Marcapasos	Enviar mensaje	Bien	Buscar Actualizar Eliminar

Showing 1 to 3 of 3 entries

← Previous 1 Next →

Prev 1 2 3 4 Next

- Visualización del historial del paciente seleccionado por el médico



Pacientes > Lady Pinzon > Historial >

Emergencias

Fecha	Estado	Ubicacion
2132493	amarillo	norte de Bogota

Prev 1 2 3 4 Next

Lecturas

Fecha	Frecuencia cardiaca	Presion sanguinea	Nivel de estres	Estado
2149029	87	89	20	1
214214	23	67	23	2

Prev 1 2 3 4 Next

Exámenes

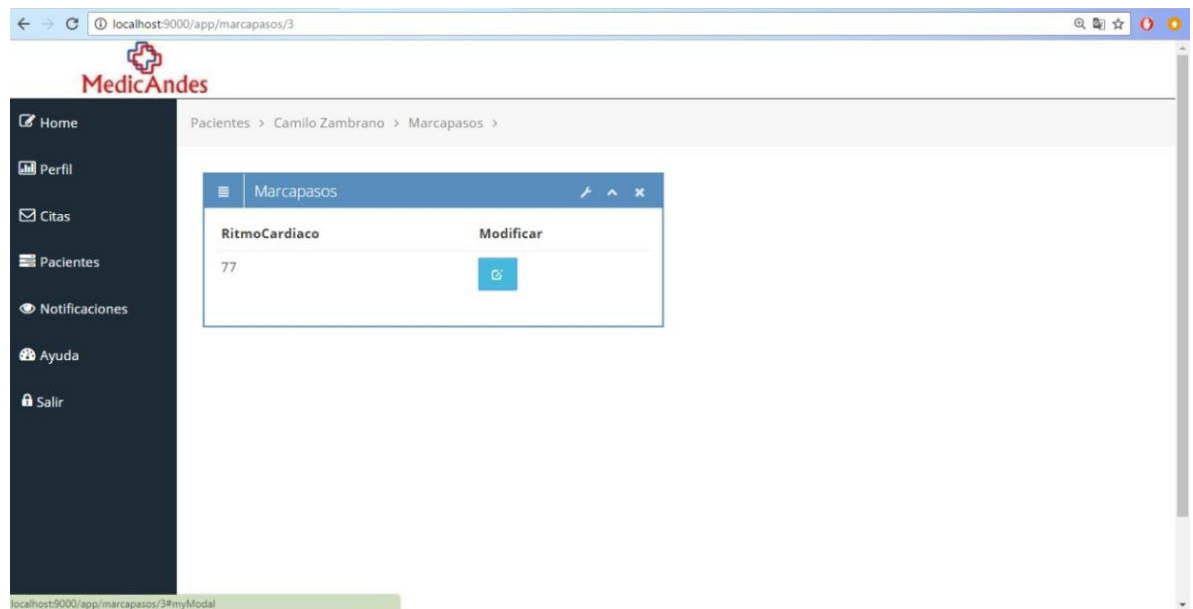
Fecha	Tipo	Resultado
214124	general	Alergia nueva a la leche
3124	general	Cardiopatía

Prev 1 2 3 4 Next

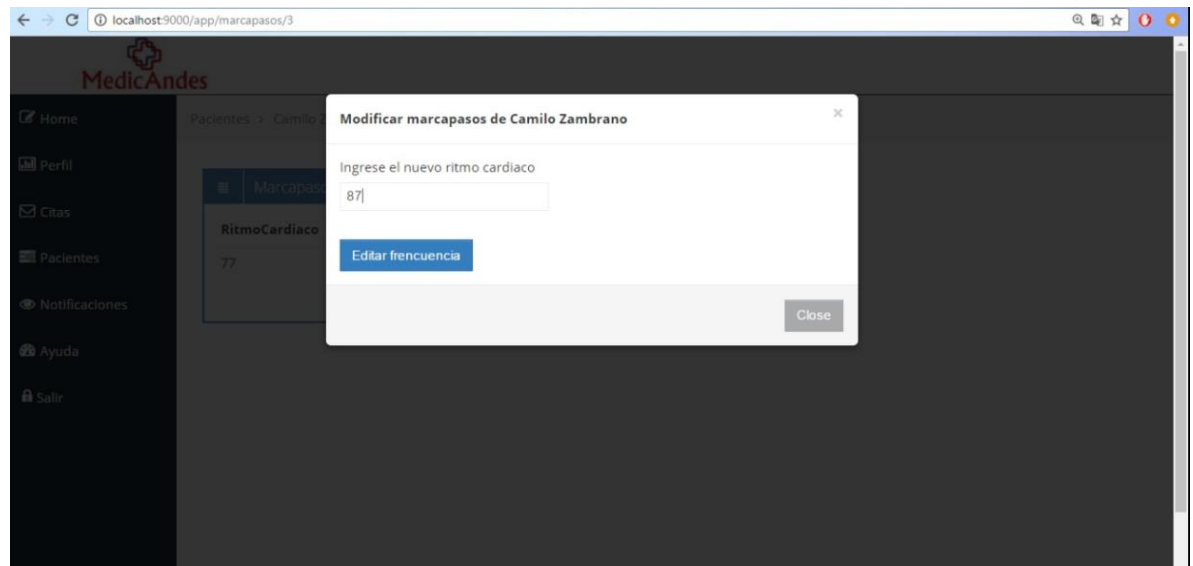
Citas

Fecha	Diagnostico
1493681844424	
1493681886693	
1493681897320	
1493681926018	

- Visualización del ritmo del marcapasos y modificación del mismo por parte del médico



- Modificar el ritmo del marcapasos



- Enviar mensaje al paciente

localhost:9000/app/mensajes/3

MedicAndes

Pacientes > Camilo Zambrano > Mensajes >

Mensajes

Enviar mensaje a Camilo Zambrano

Mensaje:

Dieta:

Actividad física:

Medicamentos:

Enviar mensaje

Close

- Filtra por el nombre del paciente buscado

localhost:9000/app/pacientes

MedicAndes

Pacientes >

Pacientes

10 records per page Search:

Nombre	Historial	Marcapasos	Enviar mensaje	estado	Acciones
Lady Pinzon	Abrir historial	Marcapasos	Enviar mensaje	Bien	

Showing 1 to 1 of 1 entries (filtered from 3 total entries)

[← Previous](#)
[1](#)
[Next →](#)

[Prev](#)
[1](#)
[2](#)
[3](#)
[4](#)
[Next](#)

- Visualización de las citas del medico

localhost:9000/app/citas

MedicAndes

Home > Calendar

Calendar

May 2017

Draggable Events

- Descanso
- Cita general
- Examen
- Cita con especialidad
- Cita de emergencia
- Other

☐ remove after drop

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	1	2	3	4	5	6
	Cita general					
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

- Agendar cita

localhost:9000/app/citas

Home > Calendar

Calendar

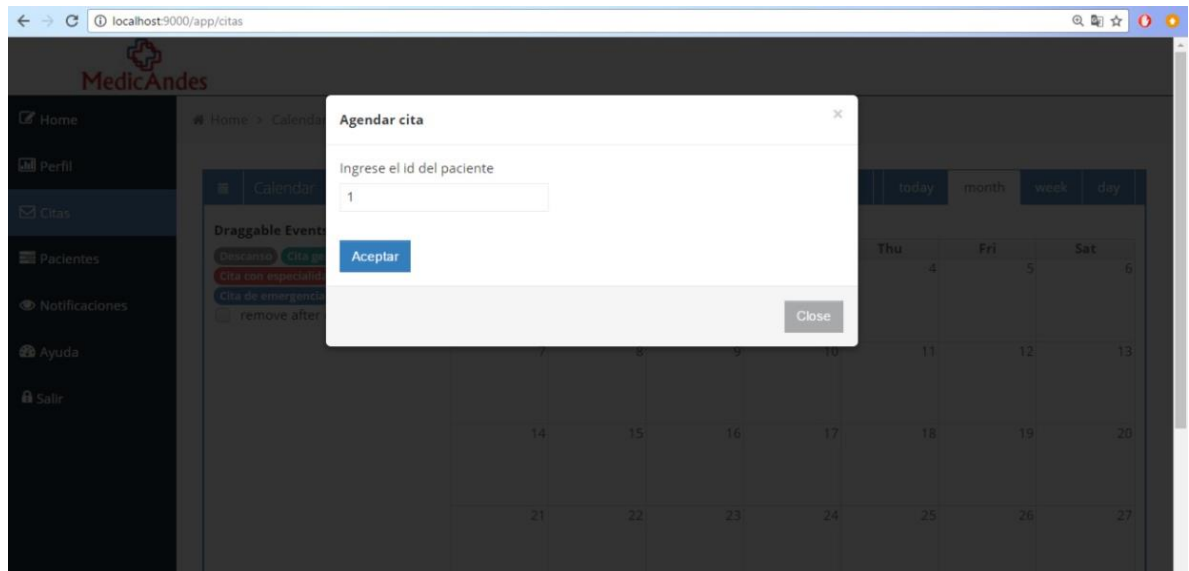
May 2017

Draggable Events

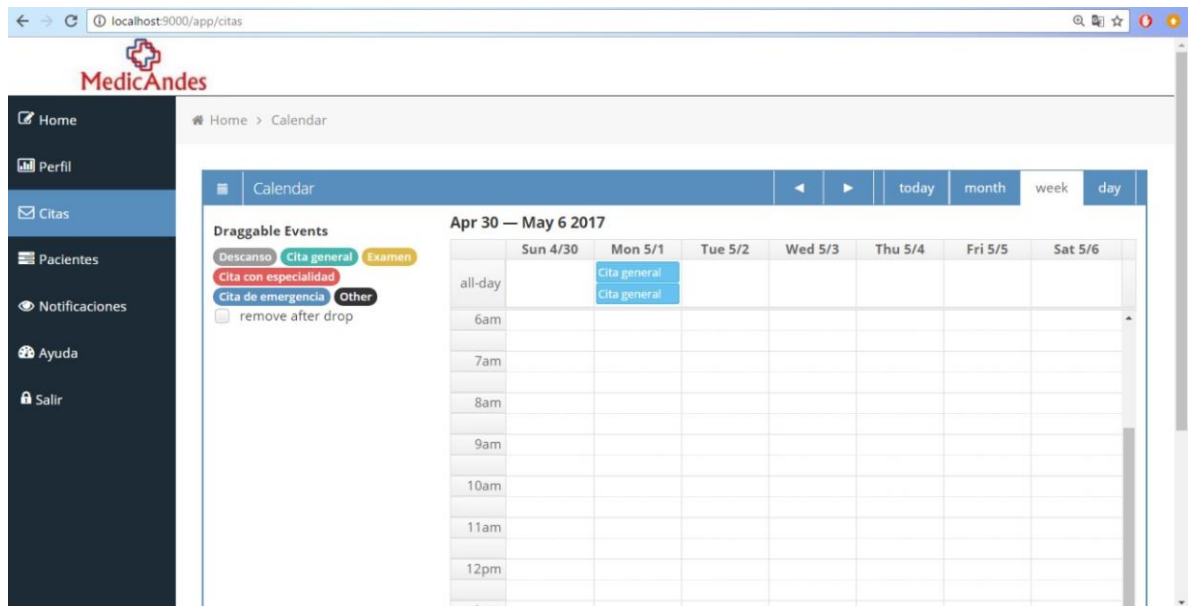
- Descanso
- Examen
- Cita con especialidad
- Cita de emergencia
- Other

☐ remove after drop

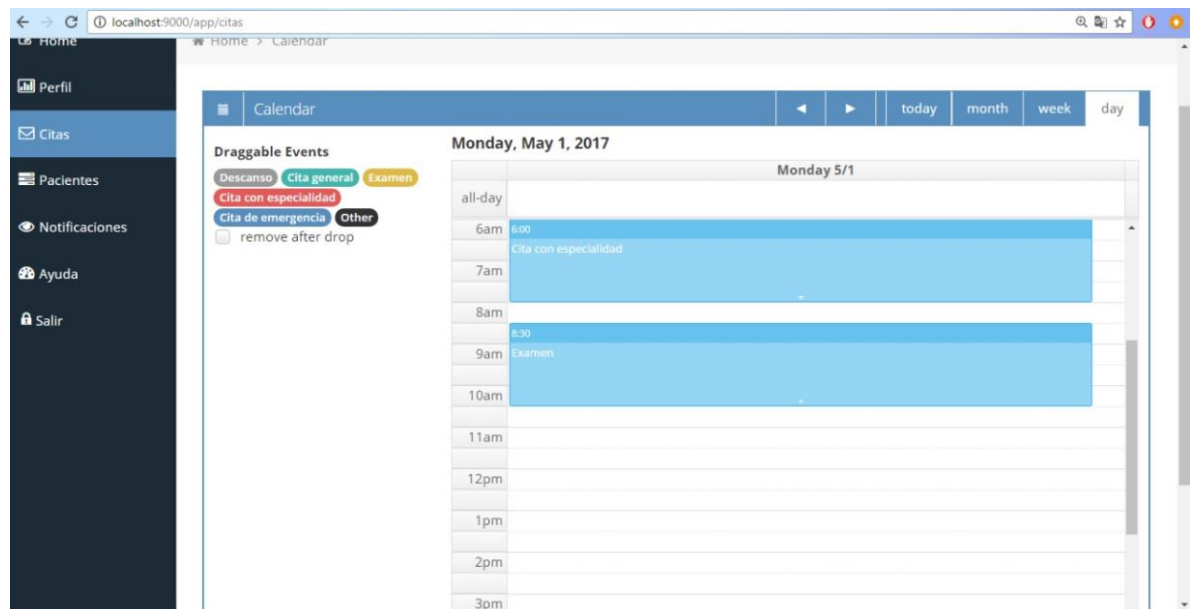
Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	1	2	3	4	5	6
	6a Cita con especialidad	6:30a Examen				
	8:30a Examen	7a Cita con especialidad				
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3



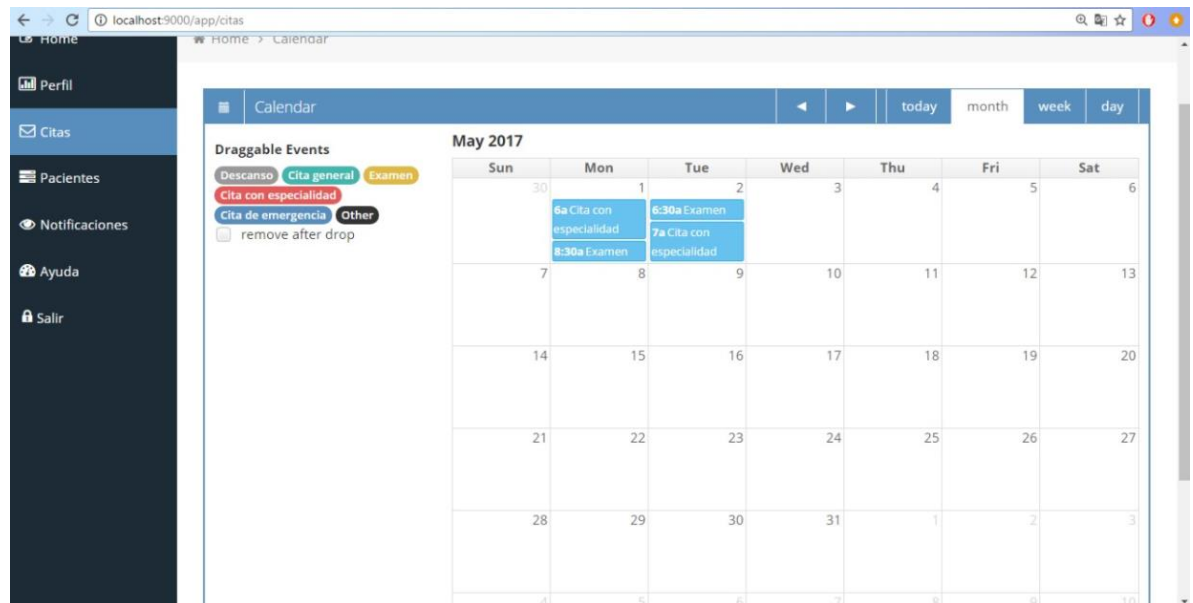
- Visualización de citas por semana



- Visualización de citas por día



- Visualización de citas con hora exacta en mes



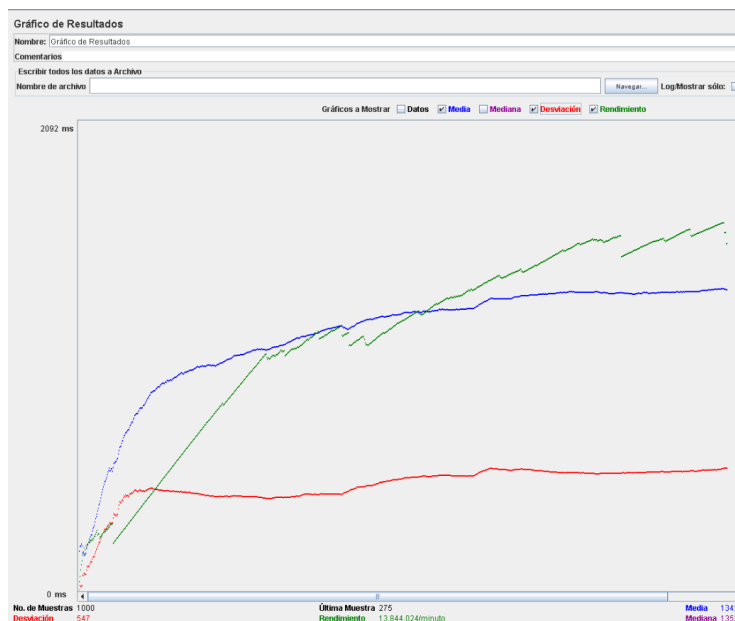
10. Resultados Obtenidos

Post-experimentación

Resultados obtenidos

Reporte resumen											
Nombre: Reporte resumen											
Comentarios											
Escribir todos los datos a Archivo											
Nombre de archivo								Navegar...	Log/Mostrar sólo:	<input type="checkbox"/> Escribir en Log Sólo Errores	<input type="checkbox"/> Éxitos
Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes	
crearLecturas	1000	1342	10	2867	547,72	0,00%	230,7/sec	55,43	0,00	248,0	
Total	1000	1342	10	2867	547,72	0,00%	230,7/sec	55,43	0,00	248,0	

Los resultados obtenidos se pueden evidenciar a través de la tabla anterior, en la cual se puede observar que la media de respuesta es de 1342 milisegundos, con un rendimiento de 230,7 threads ejecutados por segundo con un 0% de error, lo que indica que todas las peticiones fueron atendidas exitosamente.



Adicionalmente, se puede ver mejor en la gráfica que el rendimiento fue mejorando cuando la cantidad de threads aumentaba hasta llegar a mil peticiones, además la desviación es baja lo que indica que la distancia entre los diferentes tiempos de respuesta no fue tan grande.

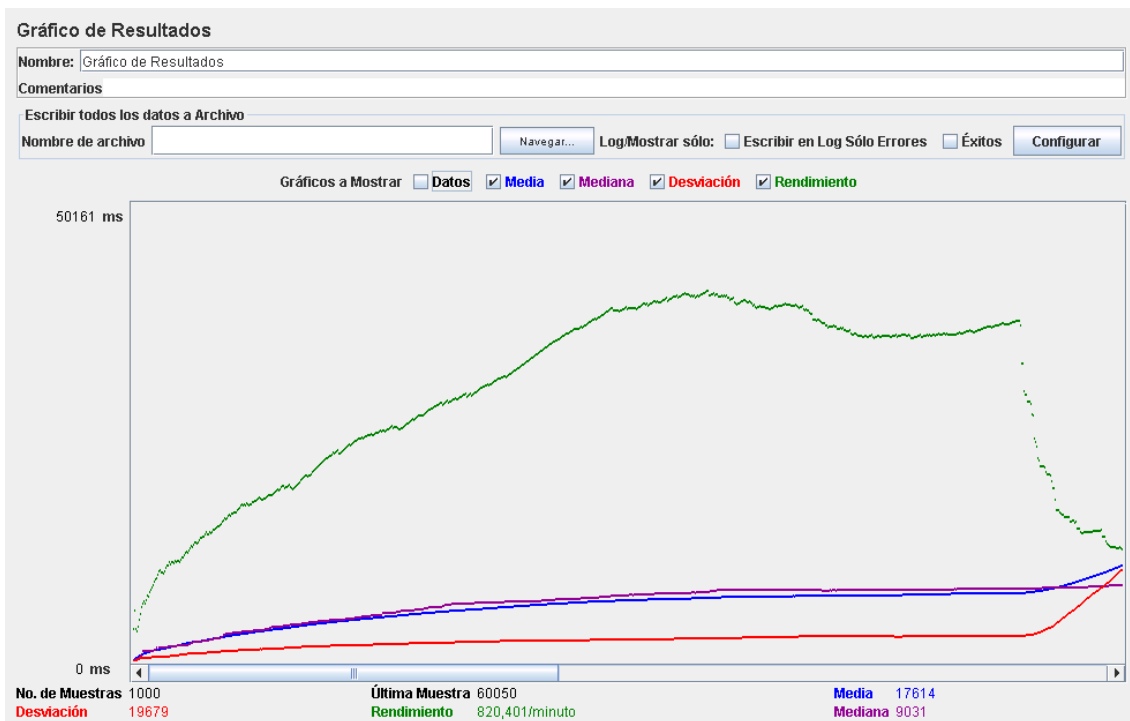
Resultados obtenidos con el balanceador de carga

Los resultados obtenidos al implementar el balanceador de carga, con el método Round Robin, se pueden observar a continuación:

Reporte resumen										
Nombre: Reporte resumen										
Comentarios										
Escribir todos los datos a Archivo										
Nombre de archivo										
Navegar...										
Log/Mostrar sólo: <input type="checkbox"/> Escribir en Log Sólo Errores <input type="checkbox"/> Éxitos										
Configurar										
Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
crearLecturas	1000	17614	250	62458	19679,89	0,00%	13,7/sec	3,88	0,00	290,7
Total	1000	17614	250	62458	19679,89	0,00%	13,7/sec	3,88	0,00	290,7

En esta se puede observar que la media fue de 17614 milisegundos, es decir, 17 segundos, con un rendimiento de 13,7 peticiones atendidas por segundo. Además, se observa que el porcentaje de error fue cero, por lo cual se sabe que todas las solicitudes enviadas fueron procesadas correctamente.

Adicionalmente, en la gráfica que se expone a continuación podemos ver la relación existente entre el tiempo y la cantidad de muestras. La línea verde muestra el rendimiento, se observa que a medida que aumenta la cantidad de muestras, en el tiempo más de ellas se van procesando. La línea azul nos indica la media, es decir la cantidad de tiempo que toma para responder con el número de muestras correspondientes, este tiempo va aumentando, pero no en gran medida, aumenta logarítmicamente. Es por esto que la línea roja, que expone la desviación estándar no aumente mucho, porque la diferencia de la media en cada punto no es tan grande.



Resultados obtenidos con seguridad

Los resultados obtenidos al implementar algoritmos para asegurar confidencialidad e integridad se pueden observar a continuación:

Reporte resumen

Nombre: Reporte resumen

Comentarios

Escribir todos los datos a Archivo

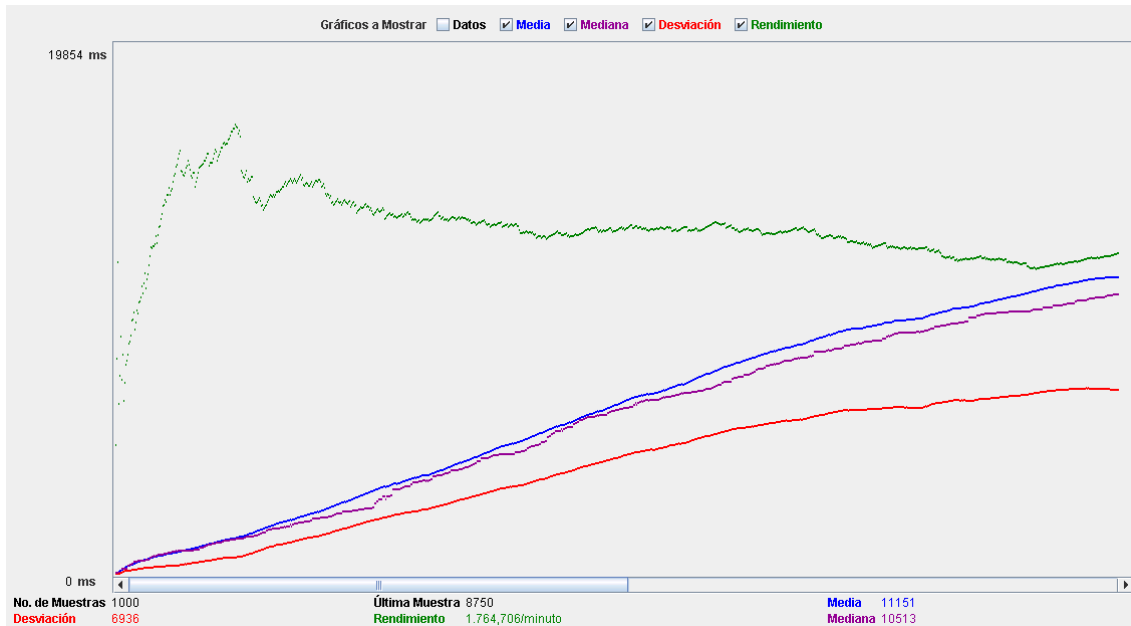
Nombre de archivo Navegar... Log/Mostrar sólo: ☐ Escribir en Log Sólo Errores ☐ Éxitos

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
crearLecturas	1000	11151	84	28348	6936,04	0,00%	29,4/sec	6,89	0,00	240,0
Total	1000	11151	84	28348	6936,04	0,00%	29,4/sec	6,89	0,00	240,0

En esta se puede observar que la media fue de 11151 milisegundos, es decir 11 segundos, con un rendimiento de 29,4 peticiones atendidas por segundo. Además, se observa que el porcentaje de error fue cero, por lo cual se sabe que todas las solicitudes enviadas fueron procesadas correctamente.

Adicionalmente, en la gráfica que se expone a continuación podemos ver la relación existente entre el tiempo y la cantidad de muestras. La línea verde muestra el rendimiento, se observa que a medida que aumenta la cantidad de muestras, se mantiene constante. La línea azul nos indica la media, es decir la cantidad de tiempo que toma para responder con el número de muestras

correspondientes, este tiempo va aumentando.



Duración real

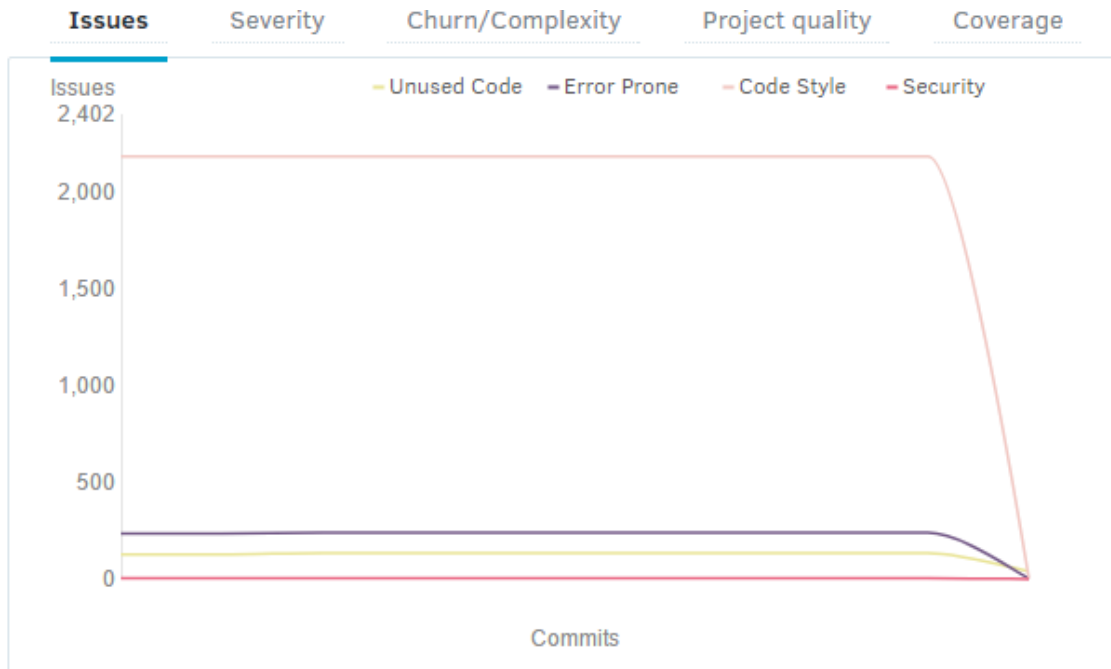
El tiempo teórico de análisis del enunciado del proyecto fue una hora y el tiempo experimental fue también de una hora. Por otro lado, en la segunda etapa el modelo conceptual y la arquitectura a utilizar se demoró 2:30 horas cuando el tiempo teórico fue de 3 horas. Además, la tercera etapa es la de construcción de la lógica de negocio que tomó 4 días cuando teóricamente se decidió que tomaría tres días. Finalmente, la cuarta etapa de pruebas nos llevó aproximadamente un día, cuando habíamos estipulado 3 horas.

Modificabilidad

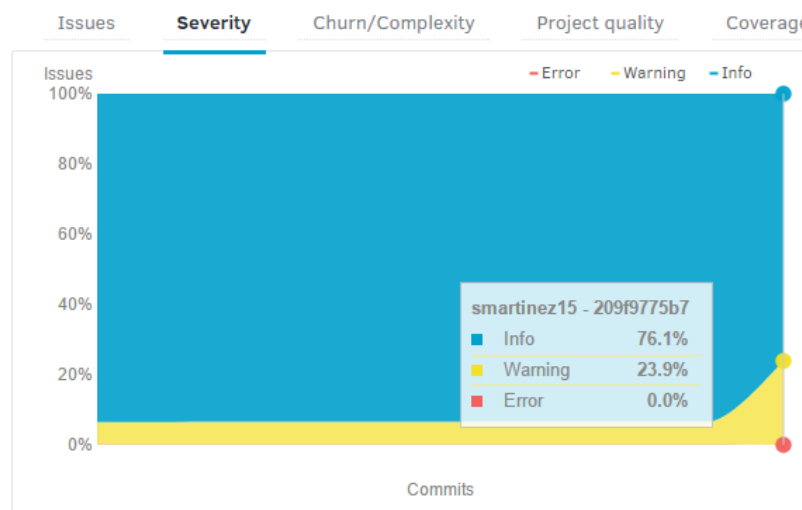
Según las métricas obtenidas en Codacy, antes de implementar los patrones de modificabilidad la clase que tenía mayor complejidad ciclomática, 25, era PacienteEntity. Dicha complejidad indica la complejidad lógica de un programa y ayuda a estimar el riesgo, costo y estabilidad. En este caso, la clase es compleja y es una clase de alto riesgo debido a que va a ser difícil probarla, entenderla y modificarla. Por otra parte, la clase que tiene más issues es UserController, con 8 issues, las cuales en su mayoría se deben a librerías

importadas que no se usaron.

Antes de implementar modificabilidad se puede observar que la cantidad de issues disminuyó a medida que aumentaban los commits, esto dice que se fueron mejorando errores



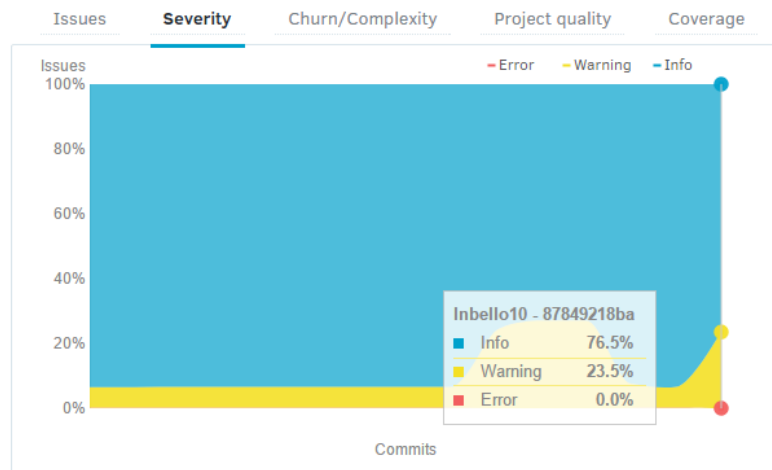
Además, en la severidad se observa el % de errores, warnings(el % de warnings y errores indica que tan difícil será la mantenibilidad y si hay bugs escondidos) e info que se encontraron en el código:



Después de la modificabilidad



Los errores tuvieron picos, pero se logró arreglar.



Además, el % de warning disminuyó, lo cual evidencia que la modificabilidad ayudó.

En lo que respecta a otras métricas, se observa la comparación antes y después de implementar la modificabilidad:

	Antes	Después
Code Style	2	2
Error prone	5	7
Unused code	39	42
Total issues	46	51

La métrica code style hace referencia a las líneas de código que atentan contra las buenas prácticas, en este caso hubo dos clases a las cuáles no se les definió un paquete al que pertenecen.

Error prone dice que tan propenso es a errores, después de la modificabilidad aumentó en 2 líneas debido a que hay algunos métodos que pueden causar errores si no se saben usar, por ejemplo, usar un if...then...else para un método que retorna boolean.

Unused code da señales de cuantas líneas de código no fueron utilizadas en ningún momento, esto se debe en su mayoría a librerías que fueron importadas pero que no fueron usadas en ningún momento.

Se puede observar como el número de issues aumentó, no es un aumento significativo, pero da cuenta de que se pudo observar mejor cuales librerías eran realmente las necesarias.

Finalmente, el factor de acoplamiento es 1 antes ya que no había relaciones de herencia, pero luego de implementar la modificabilidad se añaden tres relaciones de herencia por lo que este factor disminuye. Las tres nuevas relaciones de herencia se deben al proxy del historial y a las dos nuevas clases que se crearon para los estados rojo y amarillo de la emergencia.

Patrones implementados

Proxy: Se implementó proxy debido a que el historial es un elemento que guarda muchas cosas y resulta costoso buscarlo en la base de datos cada vez que se quiere realizar una consulta, es por eso que se crea un proxy que guarda las últimas 15 lecturas en caso de que se quiera ver el historial de las lecturas. Adicionalmente, si alguien desea conocer el historial completo (citas, exámenes,

lecturas, emergencias pasadas, alergias) este patrón ayuda a que solo se cree una vez el historial y se crea cuando es estrictamente necesario.

State: Este patrón dice que se cambia el comportamiento de un elemento cuando cambia su estado, es por esto que se decidió implementarlo para las lecturas. Una lectura recibe tres entradas, nivel de estrés, ritmo cardiaco y presión sanguínea, dependiendo de los valores que reciba la misma lectura va a cambiar su estado a verde, amarillo o rojo, es decir, hace una cosa diferente dependiendo de lo que lea de los sensores. Esto ayuda a que la lectura sepa cómo manejar la situación en tiempo de ejecución.

Observer: Cuando se implementa observer se asegura que un grupo de observadores están pendientes de lo que ocurre para comportarse de alguna forma en particular, es por esto que se implementó para la lectura. Al cambiar esta de estado, a amarillo, rojo o verde, se le notifica, dependiendo del caso, al médico y al sistema de emergencias. Ellos son los observadores de la lectura y los que van a tener algún tipo de comportamiento en especial, cuando se vuelve estado amarillo, el médico (que es un observador) es notificado y el comportamiento de él es enviar un consejo; cuando el estado es rojo se le notifica al sistema de emergencias y al médico para que vayan a recoger al paciente; finalmente cuando el estado es verde, no se realiza ninguna acción. Este patrón es diferente de state porque notifica a otras clases para que hagan algo, asigna responsabilidades, mientras que state cambia el comportamiento dentro de la misma clase.

Artefactos construidos

En nuestro proyecto implementamos el estilo de arquitectura por actores a través de Play, por esto manejamos dos paquetes principales, uno de controladores, y el otro de modelos (Entities). Estos paquetes se construyeron para manejar la lógica del negocio de manera que se acoplara bien con la tecnología de Play, además se hizo una conexión con la base de datos Postgres para manejar la persistencia y poder probar el funcionamiento del proyecto. No se implementó ninguna otra arquitectura ni artefacto, dado que solamente con

estos paquetes se seguía el lineamiento de Play, manejo de concurrencia por medio de dispatchers.

Análisis

Uno de los motivos por los que el tiempo medio no fue mejor se puede deber a la conexión a la base de datos, dado a que toma mucho tiempo acceder a la memoria secundaria a recuperar los registros, y Play ya tiene la información en la RAM. Por otro lado, se usaron dos tácticas, una es la introducción a la concurrencia ya que se procesa en paralelo los eventos con threads, y el otro, mantener múltiples copias de computaciones, es decir, la introducción de dispatcher que controlan la división de la carga. De igual forma usamos FIFO en la táctica de Scheduling Policy para así, tratar las solicitudes como iguales, sin problema de que alguna tenga más prioridad que otra. Uno de los principales problemas es que en la prueba de carga se logra crear todos los threads en un segundo, pero toma más tiempo responder a las solicitudes ya que son solicitudes de POST, por lo tanto, la base de datos tiene que hacer revisión de restricciones, candados, entre otros.

Análisis comparativo

En los resultados obtenidos del experimento 1, se pudo observar que al ejecutar las pruebas desde una sola máquina la media era más baja que al ejecutarla cuando se implementó el balanceador de carga.

Con respecto a la diferencia de rendimiento, en el balanceador de carga se utilizó el método Round Robin, ya que es necesario repartir los POST de lectura entre todos los servidores de forma equitativa. En este método se le da a cada proceso un pequeño periodo de tiempo y luego se suspende para darle la oportunidad a otros procesos. Es por esto que se observa que el rendimiento es menor cuando se tiene el balanceador de carga. No se procesa totalmente una gran cantidad de solicitudes, sino que por segundo se van haciendo los procesos de a pocos. En el experimento 1 el rendimiento fue de 230,7/sec mientras que en el experimento 2 es de 13,7/sec.

Con respecto a la diferencia de medias, es probable que sea más alta luego de la implementación del balanceador debido a que algún servidor se pudo haber sobrecargado y como las solicitudes se envían de forma equitativa, el retraso de ese servidor afecta el tiempo total de respuesta a las solicitudes. Además, existe un tiempo adicional al momento de procesar las lecturas ya que estas se envían desde máquinas separadas, entonces eso toma un poco más de tiempo que cuando las lecturas se envían desde la misma máquina. La media en el experimento 1 fue de 1,3 segundos y en el experimento 2 de 17 segundos.

Por otra parte, los KB/sec es mayor cuando se ejecutaron las pruebas recibiendo solicitudes desde una sola máquina, eso quiere decir que un mayor número de información fue enviado. Mientras que con el balanceador este número es pequeño, se puede deber a que el tiempo destinado a los procesos es limitado, por lo tanto, la cantidad de datos que se envían también es limitado. En el experimento 1 fue de 55,43 y en el experimento 2 de 3,88.

Finalmente, en ambos experimentos el porcentaje de error fue cero, por lo que se puede concluir que en ambos el total de las solicitudes fueron atendidas, de forma diferente, pero todas se procesaron en su totalidad.

Análisis comparativo seguridad

Se cifró el envío de lecturas de manera que haya confidencialidad e integridad, al ir comparando con el hash obtenido localmente y el enviado en el mensaje.

Se ve una diferencia de medias, ya que seguridad es un trade off de desempeño. La media en el experimento 1 fue de 1,3 segundos y en el experimento 2 de 11 segundos. Lo cual es mejor que cuando se implementó el balanceador de carga. Se demora más cuando hay seguridad porque hay que descifrar la lectura que nos enviaron y eso toma un poco más de tiempo.

El rendimiento también se ve afecta, en este experimento fue de 29,4 y en el anterior fue de 230, esto se debe a que el descifrado retrasa un poco la creación de lecturas.

Se implementó autenticación y autorización, existen usuarios que tienen login,

password y un permiso, el cual indica que tienen permitido hacer. Además, para poder acceder o modificar algo deben ingresar un token que se les asignará cuando hagan login, si este login no coincide con el usuario y la clave se lanzará un error, o si el token no es el mismo de la tabla, se lanza un error de autenticación.

Finalmente, en ambos experimentos el porcentaje de error fue cero, por lo que se puede concluir que en ambos el total de las solicitudes fueron atendidas, de forma diferente, pero todas se procesaron en su totalidad.

Conclusiones

En conclusión, la hipótesis planteada fue puesta a prueba y se obtuvieron resultados positivos, es decir, se comprobó que la cantidad de usuarios y solicitudes que recibía el sistema no tuvieron una alta influencia en la latencia (tiempo de respuesta entre el momento que se recibe la solicitud y en que se responde). Algunas recomendaciones son disminuir el acceso a la base de datos lo más que sea posible, también, es importante diseñar el modelo relacional muy bien desde el principio, para evitar que el nivel de acoplamiento sea alto.