



*Universidad Nacional de La Matanza*

**DEPARTAMENTO DE INGENIERÍA E INVESTIGACIONES  
TECNOLÓGICAS**

**DISEÑO DE SISTEMAS - CONSTRUCCIÓN DE SISTEMAS II**

**TRABAJO PRÁCTICO INTEGRADOR**

Sistema de Seguimiento de Pacientes Crónicos

**PROFESOR JEFE DE CÁTEDRA**

**LIC. DOMINGO DONADELLO**

**PROFESOR ADJUNTO**

**ING. ARIEL LOMONACO TORRES**

**PROFESORES AYUDANTES:**

**ING. HERNAN BOLOTCHI**

**ING. DANIEL GIANNOTTI**

**ING. VIVIANA LEDESMA**

**ING. ANALÍA MARÍN**

**ING. ROSA SANABRIA**

**AÑO 2013 – 2DO CUATRIMESTRE**

**GRUPO N°5**

<b>Apellido y nombre</b>	<b>Documento</b>	<b>E-Mail</b>
Perez, Luis Rubén	33.309.392	luisrperez2006@gmail.com
Di Ludovico, Maximiliano	26.122.379	mdlprofesional@yahoo.com.ar
Tillar, Romina Paola	28.238.933	rominatillar@gmail.com
Benítez, Sabrina	29.696.633	sbprofesional@hotmail.com
Cuozzo, Leandro	33.745.336	leandrocuzzo@gmail.com

# **Sistema de Seguimiento de Pacientes Crónicos (SEPAC)**

## **Introducción**

Las Tecnologías de la Información y la Comunicación (Tics) son hoy una realidad que marca oportunidades y desafíos. Vemos día tras día el impacto significativo de la aplicación de las mismas al área de la salud.

En este sentido, en el PLAN DE ACCION 2008 -2013 PARA LA ESTRATEGIA MUNDIAL DE LUCHA CONTRA LAS ENFERMEDADES NO TRANSMISIBLES (ENT), la Organización Mundial de la Salud (OMS) señala que estas enfermedades, principalmente las cardiovasculares, las afecciones respiratorias crónicas, la diabetes y el cáncer, son las principales causas de mortalidad en el mundo.

Las tecnologías de la información y de las comunicaciones son instrumentos esenciales para facilitar la implementación de un nuevo modelo de gestión de las enfermedades crónicas, que favorezca la participación del paciente en el cuidado de su salud y la colaboración entre el paciente y un equipo de salud proactivo.

## **El problema**

El aumento de la incidencia y de la mortalidad por enfermedades no transmisibles, agravado por la escasa adherencia a los tratamientos; el gasto creciente de los sistemas de salud para atender estas enfermedades; y el impacto negativo en la rentabilidad y en la sostenibilidad de los sistemas, son algunos de las dificultades actuales en el ámbito de la salud.

También se verifican los siguientes problemas:

- La adherencia al tratamiento a largo plazo de las enfermedades crónicas en los países desarrollados promedia 50%; en los países en desarrollo, las tasas son aún menores.
- Aumentar la efectividad de las intervenciones sobre adherencia terapéutica puede tener una repercusión mucho mayor sobre la salud de la población que cualquier mejora de los tratamientos médicos específicos”.
- Los estudios que se realizan descubren sistemáticamente ahorros de costos y aumentos significativos de la efectividad de las intervenciones de salud que son atribuibles a las intervenciones de bajo costo para mejorar la adherencia terapéutica.

## **Sistema de Seguimiento de Pacientes Crónicos (SEPAC)**

Se requiere un sistema Web que posibilite la Admisión y Seguimiento de Pacientes Crónicos según protocolos totalmente configurables y actualizables. Que permite informatizarlos a través de la definición y configuración de controles y comunicaciones que se ejecutarán a lo largo del tiempo. También reemplazarlos o modificarlos de manera ágil; es decir, la inclusión y el manejo de un sistema protocolizado por patología y/o medicamentos u otras prestaciones, que indica los criterios de seguimiento según el estadio de la enfermedad, y las prescripciones e indicaciones médicas a ser cumplidas.

Que permite trascender, de manera eficiente, la “medicina episódica” e ir hacia un seguimiento de la evolución de los pacientes afectados por enfermedades crónicas.

El sistema deberá estar compuesto mínimamente por dos módulos que pueden funcionar de forma independiente.

### ***Módulo de Admisión***

Utiliza un motor de reglas que permite la asignación automática de un protocolo a un paciente.

Admite e ingresa al paciente, según criterios médicos preestablecidos, y selecciona un protocolo de seguimiento que determina los controles y comunicaciones que se ejecutarán a través del tiempo, en un orden y con periodicidades preestablecidas. Interroga y registra co-morbilidades.

### ***Módulo de Seguimiento***

Verifica, a lo largo del tiempo, el cumplimiento del protocolo asignado.

El Sistema debe estar diseñado para que los operadores de un call center reciban de manera automática, en cada momento, los datos de los pacientes y los controles que deben efectuar.

Que permita relevar valores, evaluarlos y enviar alertas automáticas, verificar la adquisición de medicamentos, el cumplimiento de visitas al médico, el Índice de Masa Corporal, reacciones adversas y la realización de exámenes establecidos por el protocolo correspondiente, entre otros.

Deberá de manera automática y a través de diferentes canales de comunicación (e-mail, sms, etc.) brindar a los pacientes información sobre los correctos hábitos alimentarios y de cuidados específicos para los enfermos crónicos, promueve la realización de actividad física sistemática, la constitución de grupos y foros de pacientes, y otras comunicaciones que fomentan la adherencia al tratamiento. También emite recordatorios sobre los exámenes y controles que debe realizar.

Debe contar con Interfaces para dispositivos móviles permiten recibir y visualizar información de los controles y su grado de cumplimiento, y conocer el estado de evolución del tratamiento.

Se deberán registrar, lo que permite su análisis cualitativo y cuantitativo, tanto de los pacientes como de los profesionales, consumos de medicamentos y cumplimiento de tratamientos. La utilización de algoritmos de análisis de datos especializados facilita la detección de patrones de comportamiento y la correlación entre las distintas variables que intervienen a lo largo del proceso.

Ambos módulos trabajan con mecanismos especiales de encriptación y resguardo de la información. Cada operador habilitado sólo accede a una pequeña parte de la base de datos de pacientes.

### ***Supervisión***

Ante el repetido incumplimiento del consumo de medicamentos y de prácticas fijadas en el protocolo, la imposibilidad de comunicarse con el beneficiario o problemas de salud detectados, el sistema envía alertas a Supervisión.

### ***Informes***

Periódicamente se deberán emitir Informes que señalan la evolución de un conjunto de indicadores referidos a beneficiarios, profesionales, cumplimiento de programas, entre otros.

## Teniendo de base Proceso Unificado

1. Definir objetivos, límites y alcances del sistema dado para caso de estudio.
2. Metodologías Ágiles
  - a. Realice una investigación sobre distintas herramientas de software libre que permitan gestionar proyectos basados en metodologías ágiles, comparando las ventajas y desventajas de cada una de ellas.
  - b. Suponiendo que para el proyecto del caso de estudio se decidiera aplicar la metodología Scrum, se pide utilizar una herramienta para definir:
    - i. Roles
    - ii. Product Backlog
    - iii. Sprints con sus entregables.
3. Realizar el diseño de la solución con UML, deberá realizar, para el caso de estudio, los siguientes diagramas:
  - a. Diagrama de Casos de Uso
  - b. Diagrama de Actividad
  - c. Diagrama de Estado, para aquellos objetos que considere necesario.
  - d. Diagrama de Comunicación (para los casos de Uso que los docentes le indiquen)
  - e. Diagrama de Secuencia (para los casos de Uso que los docentes le indiquen)
  - f. Diagrama de Clases de Diseño
  - g. Diagrama de Componentes
  - h. Diagrama de Despliegue
4. Indique que patrones de Diseño podría haber utilizado, justificando su elección, rediseñe el Diagrama de Clases aplicando dichos patrones.
5. Seleccione una Metodología y Realice el Documento de Arquitectura del sistema utilizando la documentación correspondiente a la metodología seleccionada.
6. Diseño de la Interfaz de Usuario
  - a. Realice una investigación sobre distintas herramientas de software libre, que permitan realizar prototipos de interfaces gráficas para aplicaciones Web.
  - b. Realice el diagrama de navegación para la aplicación Web del caso de estudio.
7. Diseñe el prototipo Web de la interfaz gráfica para el caso de estudio.

### **Nota:**

- Cada una de estas actividades tendrá una fecha límite de entrega indicada por los docentes.
- Al finalizar el cuatrimestre deberá presentar para la evaluación final una carpeta con todas las actividades aprobadas, y además un CD rotulado.

## Índice

<b>1- Objetivos de la Arquitectura y Restricciones.....</b>	<b>7</b>
<b>2- Metodologías Ágiles.....</b>	<b>9</b>
Herramientas para Metodologías Ágiles.....	9
Metodología SCRUM ( <i>Herramienta Sprintometer</i> ).....	14
<b>3- Diseño UML.....</b>	<b>19</b>
<b>4- Patrones de diseño.....</b>	<b>27</b>
<b>5- Documento de Arquitectura de Software.....</b>	<b>29</b>
<b>6- Diseño de la Interfaz de Usuario.....</b>	<b>35</b>
<b>7- Prototipo Web (Interfaz operador/ modulo de seguimiento).....</b>	<b>39</b>
<b>Anexo.....</b>	<b>41</b>

# **1- Objetivos de la Arquitectura y Restricciones**

## **Objetivos**

Proveer un sistema vía Web para la admisión, control y seguimiento de pacientes con enfermedades crónicas que realice el monitoreo del tratamiento y el pronóstico diario las 24 horas. El sistema proporcionará un canal de acceso para la información de cuidados generales y comunicación con el paciente pudiendo contar con apoyo de un equipo de médicos en tiempo real y que además brinde un sistema de alarmas como medio de prevención.

## **Límite**

Los límites están comprendidos desde el momento que se admite al paciente hasta que se termina el tratamiento o se produce la baja del mismo (o inhabilitación en caso de interrumpirse debido a problemas administrativos).

## **Alcance**

1. El sistema permitirá el registro de un nuevo paciente en cualquier momento de las 24 hs del día, los siete días de la semana así como la baja en caso de ser solicitado o ameritado.
2. El sistema debe permitir ingresar todos los datos del paciente que serán ingresados por un operador.
3. El sistema permitirá elegir el protocolo de control y seguimiento del paciente según su enfermedad.
4. El sistema podrá actualizar los protocolos siempre que estén disponibles.
5. El mismo proporcionará un conjunto de herramientas para asignar un centro de gestión y tratamiento al paciente que se registre en el sistema así como la creación automática de una historia clínica.
6. El sistema contará con un menú de asignación de enfermedades y tratamientos que proporcionarán los médicos y exámenes asociados a la enfermedad como también así las patologías relacionadas con las enfermedades.
7. El sistema asignará de una lista un diagnóstico según evaluación y análisis médico.
8. El sistema ofrecerá una lista de registro de comorbilidades para elegir según el diagnóstico del paciente.
9. El sistema contará con métodos de gestión de medicamentos.
10. El sistema contará con un registro del índice de masa corporal (IMC) de cada paciente.
11. El sistema mostrará un registro de cada paciente a los operadores del programa de seguimiento los controles (datos, diagnóstico seleccionado, realización de exámenes, controles, citas médicas, adquisición e ingesta de medicamentos, reacciones adversas) para su gestión.
12. El sistema proveerá la realización de actividad física sistemática.
13. El sistema emitirá una alarma al equipo médico en caso de detectar un problema de salud o deterioro producido por la enfermedad.
14. El sistema emitirá recordatorios de exámenes y controles al paciente por medios de comunicación en forma de mail y sms.
15. El sistema enviará al paciente información general de hábitos alimentarios y cuidados específicos relacionados con su enfermedad.

16. El sistema verificará si el paciente esta inscripto en grupos de ayuda.
17. El sistema elaborará informes que serán enviados a especialistas encargados de realizar el control y seguimiento de cada paciente.
18. El sistema contará con un conjunto de alarmas para notificar que el paciente no esta realizando el tratamiento correctamente como se le asigno.
19. El sistema emitirá una alerta al equipo medico en caso de detectar la imposibilidad de comunicación o ausencia de respuesta del paciente.

### **Requisitos no Funcionales**

20. El sistema debe tener un rendimiento aceptable (tiempo de acceso menor a 5 seg.).
21. El sistema debe verificar en todo momento la disponibilidad de stock de los productos.
22. El acceso al sistema debe ser seguro (verificación de registros de ingreso).
23. El sistema debe ser fácil de mantener.
24. El sistema debe poder instalarse en múltiples plataformas (portabilidad).

## 2- Metodologías Ágiles

### Herramientas para Metodologías Ágiles

#### i. Cardmeeting

Una sencilla herramienta de colaboración simultánea en la que los usuarios pueden ir añadiendo sus cartas con ideas, conceptos, tareas, etc.

##### Ventajas

- Puede resultar útil para celebrar una tormenta de ideas o brainstorming.
- Realizar una planificación ágil, o simplemente para fijar y compartir las ideas con otros usuarios.
- Las cartas pueden clasificarse luego en diferentes categorías temáticas (curioso, sorprendente, divertido, frustrante) y colores, de forma que podemos asignar un orden a las ideas formuladas de manera dispersa.

##### Desventajas

- Todavía está en versión Alfa.

#### ii. Rally

En Rally afirman ser los auténticos y genuinos números 1 en el software de gestión ágil. Su aplicación cubre todo el ciclo del proyecto, y permite crear un flujo entre la labor de los gestores, desarrolladores y responsables de testing.

##### Ventajas

- Sus productos están centrados en las siguientes funciones: organización del proyecto, el reporte y control, la planificación y seguimiento, gestión de calidad, colaboración de equipos, customización e integración, y administración de permisos.
- Entre los clientes de Rally se encuentran varias grandes compañías.

#### iii. Scrumdesk

Es una herramienta de gestión diseñada específicamente para trabajar con SCRUM, la metodología de desarrollo ágil más conocida.

##### Ventajas

- Permite automatizar los procedimientos básicos de SCRUM, como son la elaboración del product Backlog (lista de requisitos de alto nivel ordenada según la prioridad), el sprint Backlog (lista de tareas que deben ser completadas en un plazo de entre 1 y 4 semanas) o las historias de usuario.
- Asignarle prioridad a las tareas, trabajar en diferentes escritorios de prueba.

#### iv. Sistemas Multiagentes

El bloque fundamental de un sistema multiagente son los agentes. Una de las formas en las que se consideran a los agentes es, por lo general, una entidad inteligente, que en términos computacionales sería como un proceso de un sistema con la capacidad de comunicarse a través de un mecanismo de comunicación inter-procesos, en cierto contexto y ambiente. Los sistemas multiagente pueden tener 2



enfoques dentro de la Inteligencia artificial por los que pueden ser construidos:

1. **Formal:** consiste en dotar a los agentes de la mayor inteligencia posible, utilizando descripciones formales del problema a resolver y hacer reposar el funcionamiento del sistema en tales capacidades cognitivas.
2. **Constructivo:** persigue la idea de brindarle inteligencia al conjunto de agentes del sistema para que a través de mecanismos de interacción, el sistema genere comportamiento inteligente que no necesariamente estaba planeado desde un principio o definido dentro de los agentes.

## v. AgilShell

Es una herramienta para el modelado de procesos. Esta herramienta para el diseño e implementación de sistemas multi-agente empleando una metodología ágil, cualquiera que esta sea.

Su visión se basa en 3 pasos:

1. Expertos del dominio construyen un modelo es los procesos de trabajo existentes.
2. Estos procesos existentes se analizan para detectar escenarios de aplicación de agentes.
3. Estos procesos se van incrementando para introducir agentes en ellos.

**Process Graph:** La vista principal de AGIL-Shell es un gráfico en el cual los ingenieros y expertos pueden definir procesos ordenando actividades y datos como nodos, y representando el flujo de datos y la secuencia de actividades mediante flechas. Para cada actividad, su nombre y su rol son visualizados. Para cada agente mensaje, el comando y la ontología son mostradas.

**Process Explorer:** Modelos de proceso complejos se representan y se visualizan empleando un orden jerárquico de procesos y sub-procesos en una vista de árbol.

**Customizer:** Las propiedades de la actividad seleccionada actualmente, de los procesos, mensajes, agentes u ontologías pueden ser editadas.

**Community Viewer:** Este gráfico muestra los caminos de ejecución entre los roles (humanos, actores,...) en el sistema. Los agentes son mostrados en nodos, los cuales muestra el tipo de rol. Esta vista permite detectar cuellos de botella. Los nodos con muchas flechas entrantes y salientes representan roles que operan en una gran cúmulo de datos. Estos puntos de información puede ser que no lleguen a beneficiarse de los servicios de información basados en agentes. Además de estas vistas, existen otras vistas de procesos que pueden derivarse del metamodelo. Por ejemplo, la secuencia de actividades permite extraer una vista de ciclo de vida en la cual solo las actividades que son llevadas a cabo por un agente o un humano son mostradas. Estas vistas permiten realizar el diseño con los procesos humanos participantes, los cuales (desde su punto de vista) pueden evaluar si su rutina diaria está suficientemente cubierta por el conjunto de todos los procesos.

## vi. Agent Factory

Para cumplir los principios del manifiesto ágil una de las fases más importantes es la de producción de código, que en este tipo de sistemas se basa en la herramienta Agent Factory para la generación automática de las estructuras del agente.

Esta herramienta:

- Analiza los diagramas de identificación de agentes y de la Ontología y genera un primer esqueleto de las clases del agente necesarias.

- Permite introducir patrones en el proyecto para mejorar la funcionalidad.
- Genera código para el sistema multiagente. Este código consiste en un esqueleto del agente y clases con sus tareas, que más tarde se completa con los patrones reutilizados. Algunos experimentos han demostrado un porcentaje de reutilización de código que es de aproximadamente 50-60%.
- Otra parte importante son las pruebas ya que estas comprueban el cumplimiento de los requisitos. Estas pruebas deberían prepararse antes de la fase de codificación, de acuerdo con las especificaciones. El Agent Factory facilita el testeado de los agentes.

## **vii. X-Man (Extreme Manager)**

X-Man (Extreme Manager) herramienta poderosa y de uso fácil, la cual está diseñada para dar soporte a la gestión y seguimiento de los proyectos desarrollados con XP. Fue diseñada para ser utilizada por todos los interesados (stakeholders) envueltos en el desarrollo del proyecto. Los managers del proyecto pueden usar la herramienta para darle seguimiento al estado diario del mismo, mientras que los desarrolladores y probadores la usan para chequear cuales de las historias de usuarios están abiertas o cerradas.

El cliente también puede utilizarla para mantener un claro entendimiento del estado del proyecto en cualquier momento, de manera que puede constatar cuales historias de usuarios están listas para ofrecer retroalimentación.

X-Man extrapola el día de completamiento de las iteraciones basándose en el alcance del segundo día y estima una desviación del alcance original por la fecha de fin planificada. Otra característica de interés es que es compatible con Office 2007 y la mayoría de sus datos pueden ser exportados como hojas de cálculo de Excel.

## **viii. Extreme Planner**

Solución Web de gestión ágil de proyectos para el desarrollo de software en equipos que trabajan en diferentes localizaciones. Las herramientas tradicionales de gestión de proyectos se basan principalmente en una lista de tareas, ExtremePlanner se enfoca en la planificación y seguimiento de los requisitos o historias de usuario que tienen valor real para los clientes.

Está diseñada específicamente para dar soporte a las metodologías Ágiles, así como a líderes, desarrolladores, soporte técnico, en fin todas las partes interesadas en el desarrollo de un proyecto ágil.

Entre las características que posee que ofrece una vista de las entregas planificadas, quienes están trabajando sobre que tarea, las iteraciones en progreso y detalles relevantes del proyecto. Además permite a partir de las historias de usuario, hacer una planificación de las entregas, de manera drag-and-drop, o sea, para la una entrega determinada tantas historias de usuario como se consideren; de esta manera se agilizan los planes de entrega e iteraciones, incluso hace advertencias cuando el plan de entregas está por debajo del esfuerzo estimado del equipo. Otra de las potentes funcionalidades es el manejo de las historias de usuario las cuales desde el momento en que son capturadas son planificadas junto con las tareas, casos de prueba asociados todo en una misma vista. Además permite importar las tareas asociadas a las historias de usuario desde un Excel, exportar reportes al mismo formato, a Word o a XML.

El Manager tiene la posibilidad de que si se están desarrollando varios proyectos, limitar el acceso de los usuarios del sistema solamente a los proyectos en los que están involucrados, además le brinda

información sobre los cambios sufridos en algún proyecto con un completo registro auditable.

Por último es completamente integrable a los IDEs Eclipse y Visual Studio en cuanto a la actualización de la información de las tareas, de esta manera los desarrolladores pueden ver esta información.

## ix. Pivotal Tracker

Aplicación Web similar a Extreme Planning. Propone el siguiente flujo de trabajo (workflow):

1. Historias de Usuarios.
2. Estimar complejidad.
3. Organizar las historias por etiqueta.
4. Establecer prioridad de las historias.
5. Convertir las historias en software funcionando.
6. Colaboración de Equipo.
7. Aceptación del Cliente.
8. Seguimiento de velocidad y fechas de entrega.

Esta herramienta da soporte para todo este flujo de trabajo, proporcionando al equipo una interfaz amigable, fácil de usar y sobre todo eficiente desde el punto de vista organizacional, ya que utiliza las nuevas tecnologías como e-mail y redes sociales para la colaboración del equipo en el desarrollo de historias de usuario que necesiten de comunicación a través de notificaciones. Además soporta control de versiones (Subversión, GitHub y Git) para la integración de las tareas de historias de usuario. Existen herramientas 3rd-party que traen dentro su integración como son:

- Lighthouse
- Zendesk
- JIRA
- Bugzilla
- Ofrece un framework abierto para desarrollar su integración con cualquier otra herramienta a través de un API de desarrollo.

## x. IceScrum

Aplicación Web diseñada para el desarrollo de proyectos utilizando Scrum, cubre con las necesidades de equipos ágiles geográficamente distribuidos. Bajo licencia GNU GPL V3 (opensource), posee una comunidad muy amplia y facilita a los equipos su transición hacia la agilidad implementando las prácticas ágiles claves: visión, requisitos, historias de usuario, estimación de esfuerzos, pizarra de tareas, etc.

Entre las características que ofrece se encuentran:

- **Backlog del producto (Product Backlog):** contiene las historias, las cuales son seleccionadas, extendidas o movidas con facilidad por el propietario del producto. De esta manera el equipo cuenta con un Backlog siempre listo para usar.
- **Plan de Entregas (Release Plan):** las historias de usuario son relacionadas con sprints, de manera casi automática teniendo en cuenta la capacidad del equipo.
- **Plan de Sprint (Sprint Plan):** posee las tareas para la implementación de las historias de

usuarios. Cada desarrollador define, escoge, comienza y termina las tareas del Sprint. Trae incorporada prácticas de Trabajo en Progreso (Work In Process WIP) para ello.

- **Sandbox:** espacio donde todas las partes interesadas hacen comentarios sobre las historias con el objetivo de ayudar a decidir al líder de proyecto que hacer con las historias, si darle o no mayor prioridad, etc. Es una extensión del Backlog del producto.
- **Línea de Tiempo (Timeline):** vista gráfica del ciclo de vida de vida del proyecto así como su estado.
- **Actores (Actors):** contiene los usuarios del producto y sus características. Los actores son usados las historias del Backlog, los actores son asociados a las mismas y permite filtrar las historias según las características definidas del actor.
- **Requisitos (Features):** ofrece la posibilidad de manera sencilla de asociar requisitos a historias, así como las prioridades de estos de acuerdo a su valor para el cliente.
- **Equipo (Team):** gestiona los roles y permisos con gran flexibilidad. Cuando se une un miembro al equipo es posible cambiar de rol e incluso poseer un avatar.
- **Indicadores (Indicators):** produce automáticamente indicadores del Sprint, entregas y nivel del producto, teniendo en cuenta indicadores emblemáticos de Scrum.

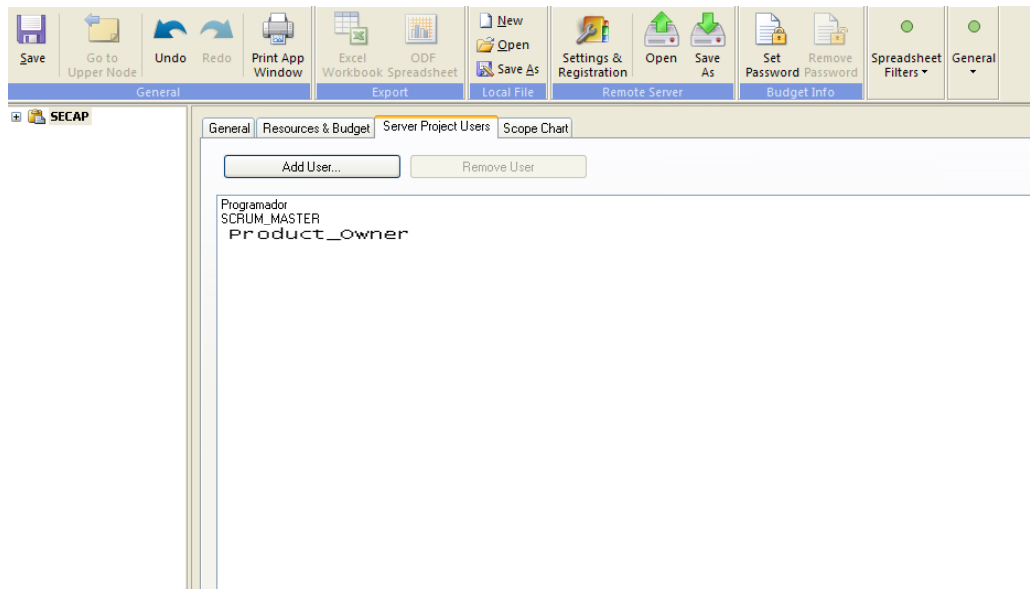
## xi. Version One

Básicamente aplicación Web que cubre todo el ciclo de vida de la gestión ágil de proyectos, cualquiera que sea la metodología utilizada para desarrollar el mismo. El ciclo de vida propuesto es el siguiente:

- **Planificación del producto:** llevada a cabo por todas las partes interesadas. Permite la planificación tanto a alto nivel como detallada.
- **Planificación de entregas:** llevada a cabo por los propietarios del producto. Ofrece la realización de predicciones y realizar planes de contingencia.
- **Planificación de Sprint:** llevada a cabo por los managers, donde se dividen historias en sus respectivas tareas y pruebas. Además se puede llevar a cabo una planificación de las capacidades del equipo en cada sprint.
- **Seguimiento de producto:** llevado a cabo por el equipo de desarrollo, permitiendo que cada miembro personalice sus reportes de seguimiento. Además permite dado un cambio en las prioridades cambiar de manera muy fácil y eficiente realizar cambios en la planificación
- **Gestión de Programas:** llevada a cabo por los ejecutivos en la cual estos elaboran una jerarquía de proyectos según la complejidad, determinando características comunes entre ellos, de manera tal que puedan llevar a cabo varios proyectos simultáneamente.
- **Reportes y estadísticas:** llevada a cabo por líderes y managers. VersionOne ofrece reportes utilizando más de 50 métricas del desarrollo ágil. Además ofrece la posibilidad a clientes de indagar en estos reportes a través de un Data Mart al cual puede acceder desde herramientas como Bussines Objects, Crystal Reports, Cognos y LogiXML.

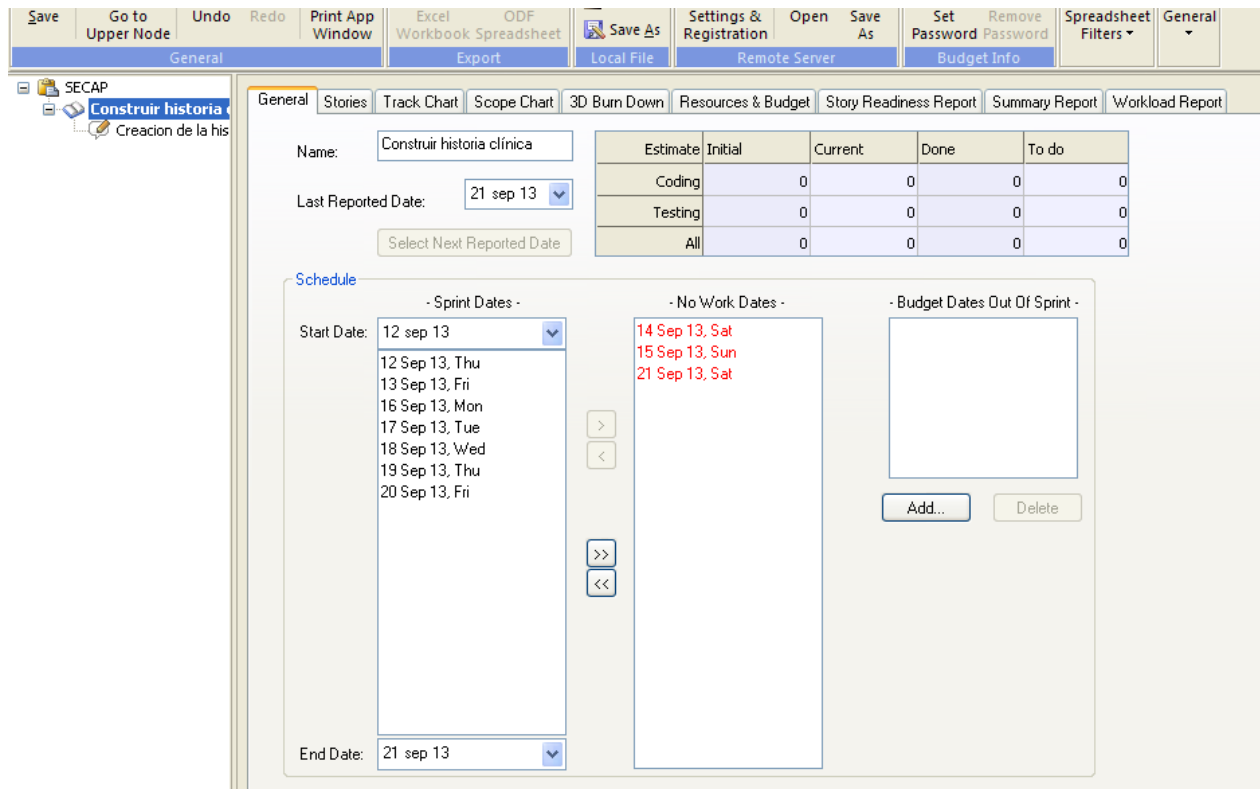
## Metodología SCRUM (Herramienta Sprintometer)

### Roles



### Product Backlog:

#### Primer sprint



SECAP

Construir historia clínica

Selección de la historia clínica

1: Selección de la historia clínica

2

General

Task No: 1

Name: Selección de la historia clínica

Description: El medico selecciona la historia clínica del paciente

Initial Estimation (days): 15

Current Estimation (days): 15

Work Type

Coding

Change Request

Issued: 12/11/2013

Date:	Sep 12	Sep 13	Sep 16	Sep 17	Sep 18	Sep 19	Sep 20
Work day:	1	2	3	4	5	6	7
Done %:	-	-	-	-	-	-	-
Done today/to do:	0/15	0/15	0/15	0/15	0/15	0/15	0/15
Done today/to do:	0/15	0/15	0/15	0/15	0/15	0/15	0/15

Construir historia clínica

historia clínica

1: Creación de la clase historia clínica

General

Stories

Track Chart

Scope Chart

3D Burn Down

Resources & Budget

Story Readiness Report

Summary Report

Workload Report

Construir historia clínica: Sprint track upon work left/complete

Work Day	Track upon work left (Days)	Track upon work complete (Days)
Sep 12	-2	0
Sep 13	-4	0
Sep 16	-6	0
Sep 17	-8	0
Sep 18	-10	0
Sep 19	-12	0
Sep 20	-16	0

Work Type: ---ALL---

Print Chart

14

## Segundo Sprint

SECAP

Construir historia clínica

Selección de la historia clínica

1: Selección de la historia clínica

Elegir protocolo

Selección de protocolo

1: Selección un protocolo

General

Stories

Track Chart

Scope Chart

3D Burn Down

Resources & Budget

Story Readiness Report

Summary Report

Work

Name:

Elegir protocolo

Last Reported Date:

24 sep 13

Select Next Reported Date

Estimate	Initial	Current	Done	To do
Coding	20	20	0	20
Testing	0	0	0	0
All	20	20	0	20

Schedule

- Sprint Dates -

Start Date:

24 sep 13

24 Sep 13, Tue

25 Sep 13, Wed

26 Sep 13, Thu

27 Sep 13, Fri

30 Sep 13, Mon

01 Oct 13, Tue

End Date:

01 oct 13

- No Work Dates -

28 Sep 13, Sat

29 Sep 13, Sun

- Budget Dates Out Of Sprint -

Add...

Delete

SECAP

Construir historia clínica

Selección de la historia clínica

1: Selección de la historia clínica

Elegir protocolo

Selección de protocolo

1: Selección un protocolo

General

Task No:

1

Name:

Selección un protocolo

Description:

El medico selecciona el protocolo a seguir

Initial Estimation (days):

20

Current Estimation (days):

20

Work Type

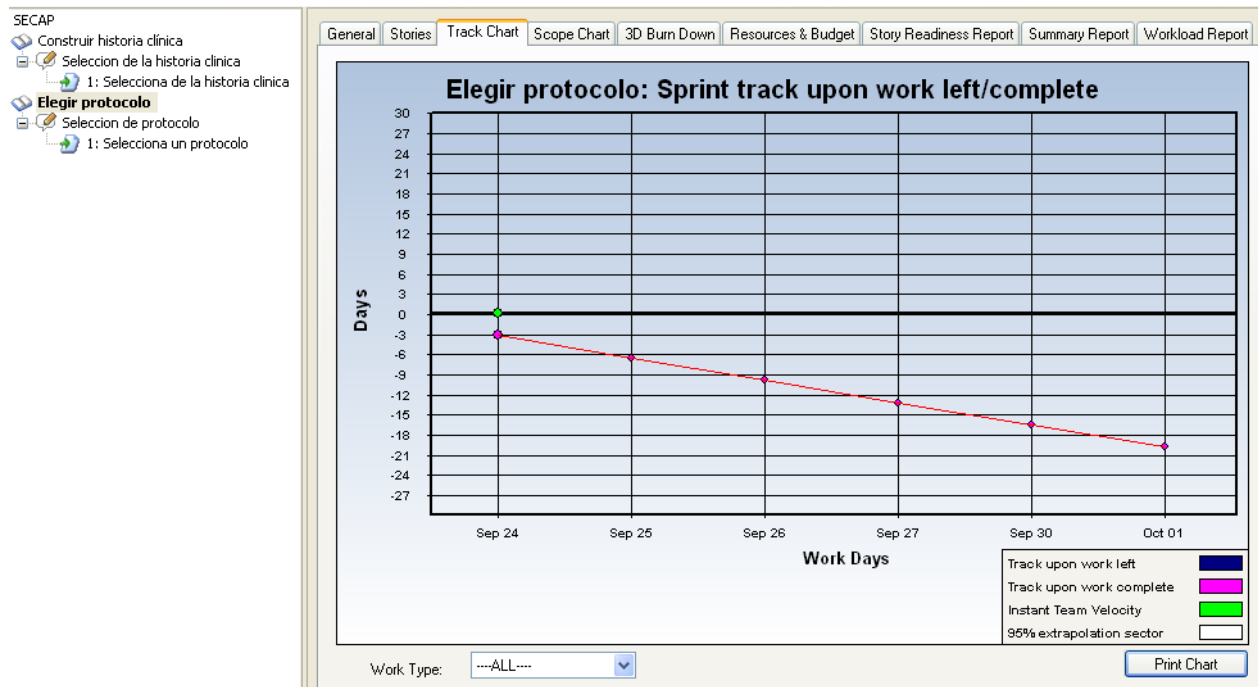
Coding

Change Request

Issued:

12/11/2013

Date:	Sep 24
Work day:	1
Done %:	-
Done today/to do:	0/20
Done today/to do:	0/20



## Tercer Sprint

General | Export | Local File | Remote Server | Budget Info

General | Stories | **Track Chart** | Scope Chart | 3D Burn Down | Resources & Budget | Story Readiness Report | Summary Report | Wc

Name:

Last Reported Date:

Select Next Reported Date

Estimate	Initial	Current	Done	To do
Coding	0	0	0	0
Testing	10	10	0	10
All	10	10	0	10

Schedule

- Sprint Dates -

Start Date:

21 Oct 13, Mon  
22 Oct 13, Tue  
23 Oct 13, Wed  
24 Oct 13, Thu  
25 Oct 13, Fri  
28 Oct 13, Mon  
29 Oct 13, Tue  
30 Oct 13, Wed  
31 Oct 13, Thu  
01 Nov 13, Fri

- No Work Dates -

19 Oct 13, Sat  
20 Oct 13, Sun  
26 Oct 13, Sat  
27 Oct 13, Sun

- Budget Dates Out Of Sprint -

Add... Delete

End Date:



SECAP

- Construir historia clinica
  - Selección de la historia clinica
    - 1: Selección de la historia clinica
- Elegir protocolo
  - Selección de protocolo
    - 1: Selección de un protocolo
- Testing de alarma
  - Pruebas de alarma
    - 1: Prueba de alarmas del sistema

**General**

Task No:

Name:  Assigned to:

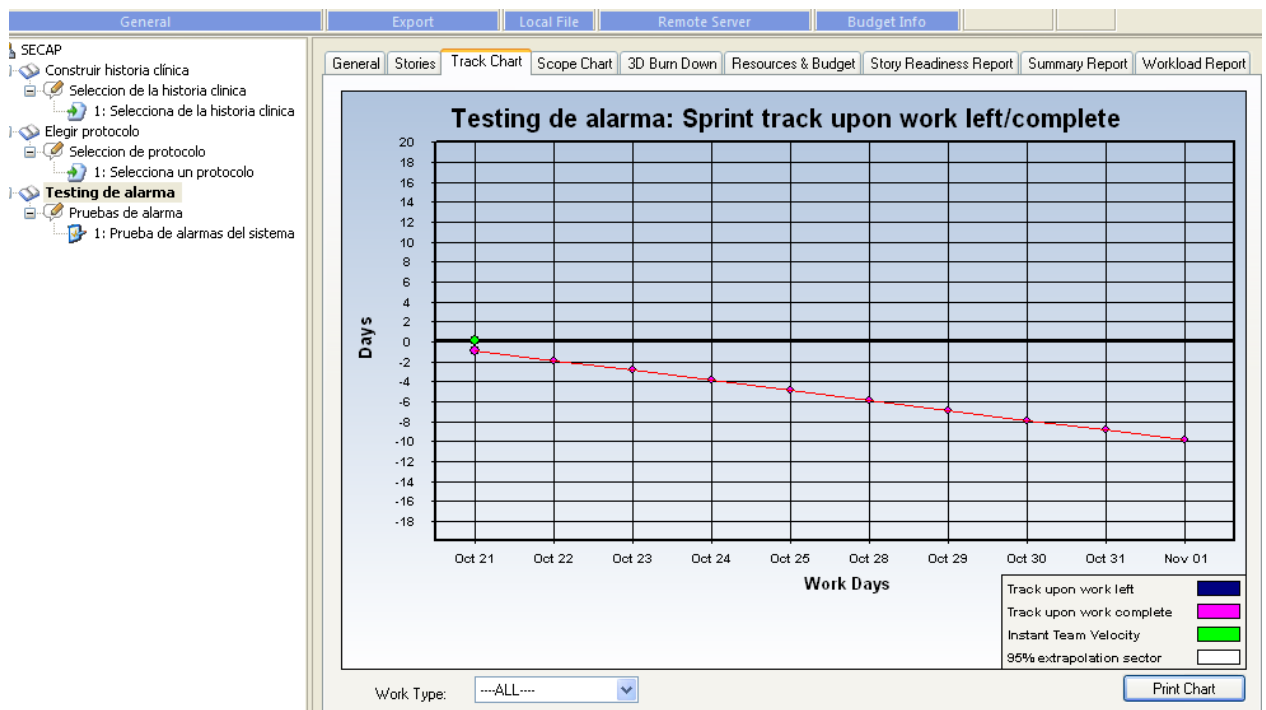
Description:

Initial Estimation (days):  Current Estimation (days):

Work Type:  ☐ Change Request Issued:

Date:	Oct 21
Work day:	1
Done %:	-
Done today/to do:	0/10
Done today/to do:	0/10

Undo Unc



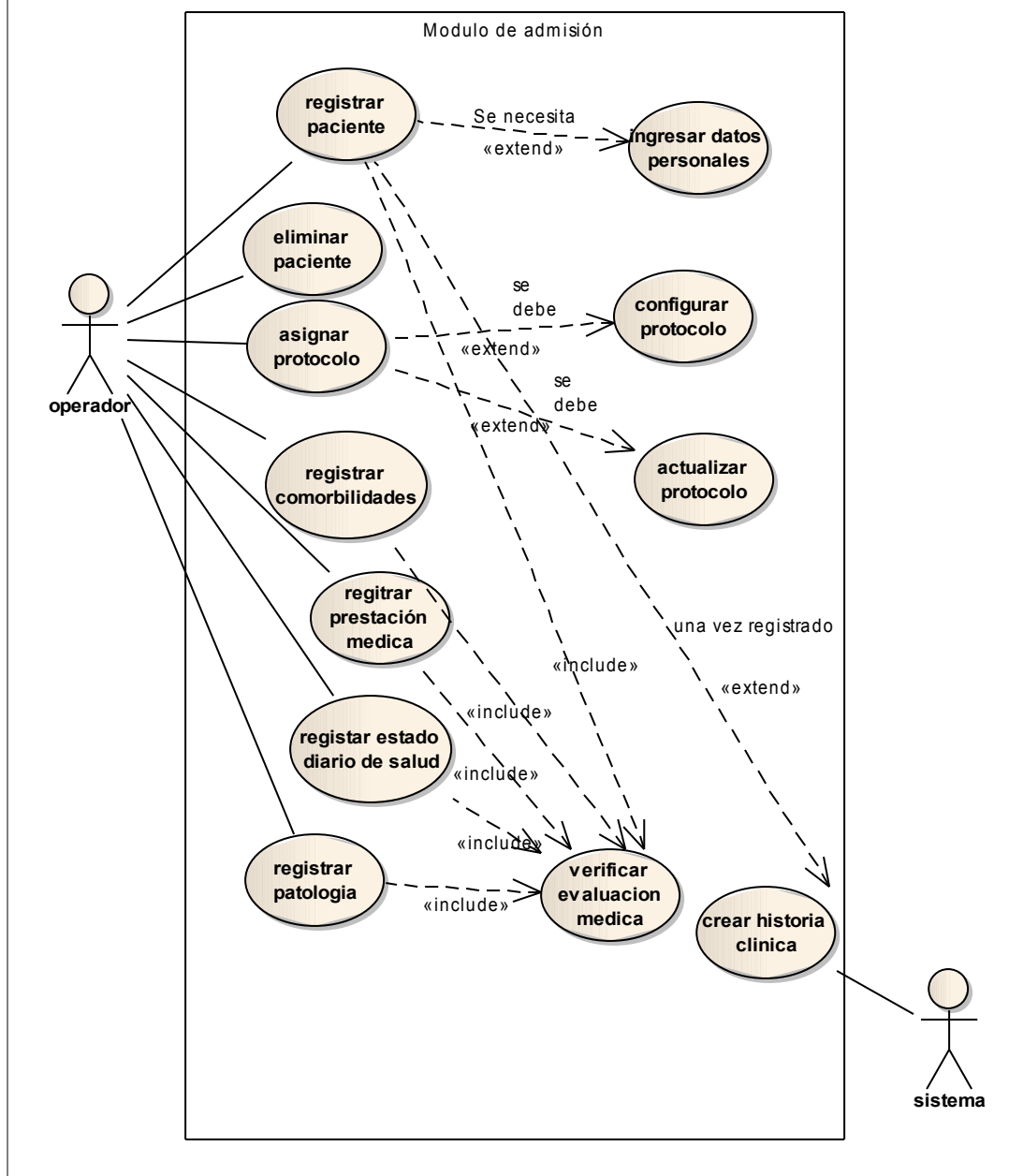
### 3- Diseño UML

#### Diagrama de Casos de Uso

A continuación se ilustran los casos de uso correspondientes a cuatro tipos de actores diferentes:

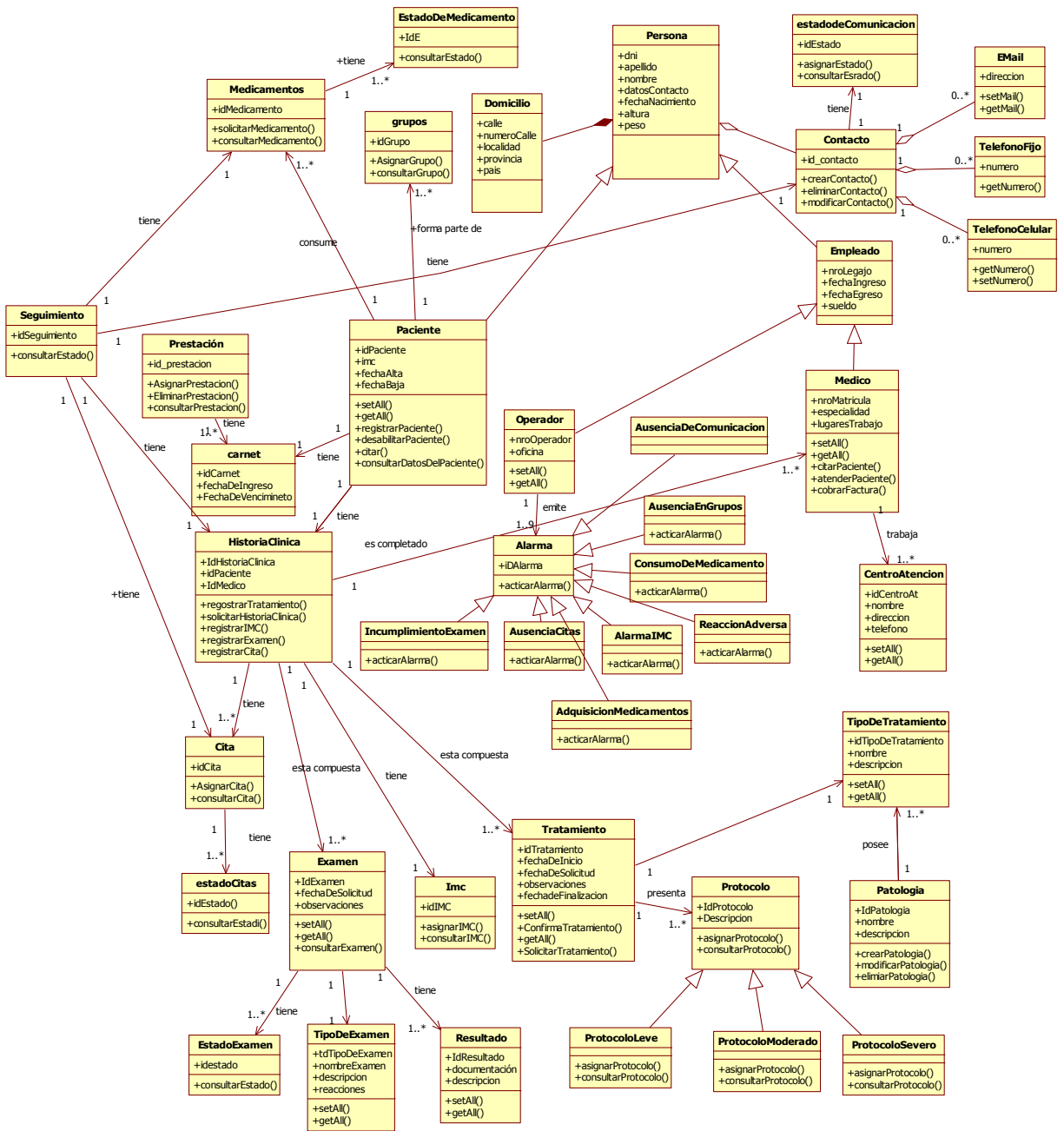
**Operador:** Admite e ingresa al paciente en el sistema, además realiza, controla y califica el seguimiento de los pacientes, envía información sobre cuidados, nutrición y recordatorios de próximos turnos, elabora registros de seguimiento y da avisos de alerta en caso de no estar cumpliendo con el protocolo asignado.

**Sistema:** es el encargado de monitorear los controles médicos, controlar la salud y la comunicación con el paciente y en caso de fallar uno de ellos se emite una alerta.

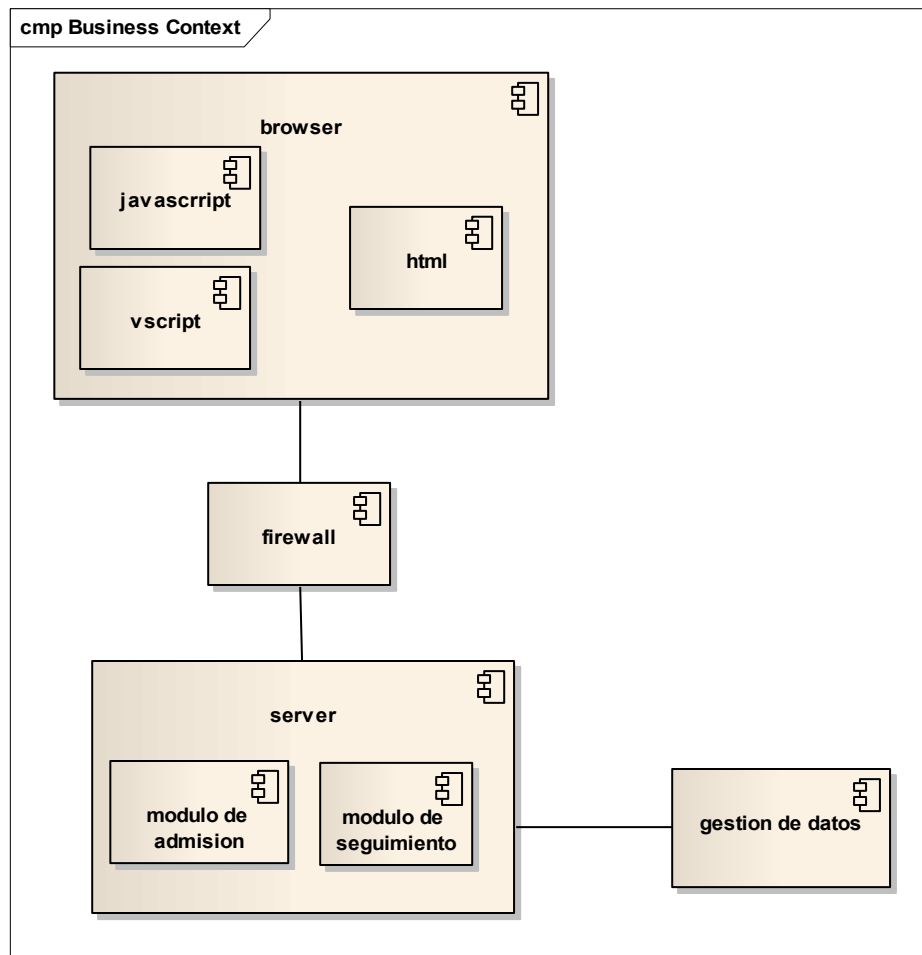




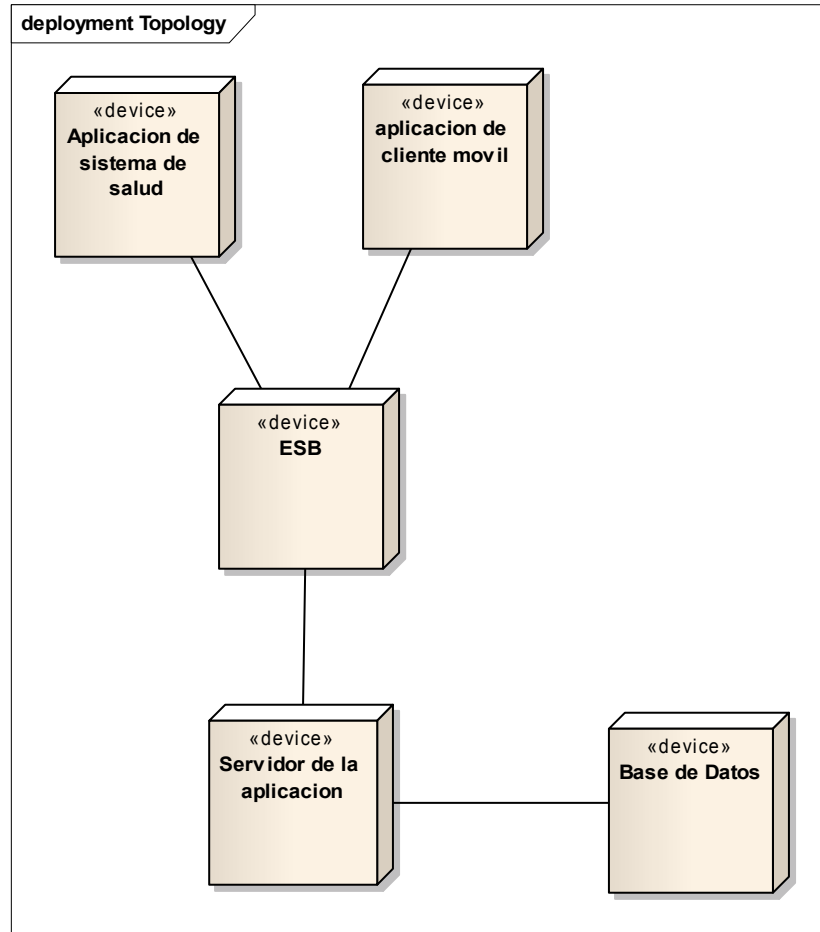
## Diagrama de clases



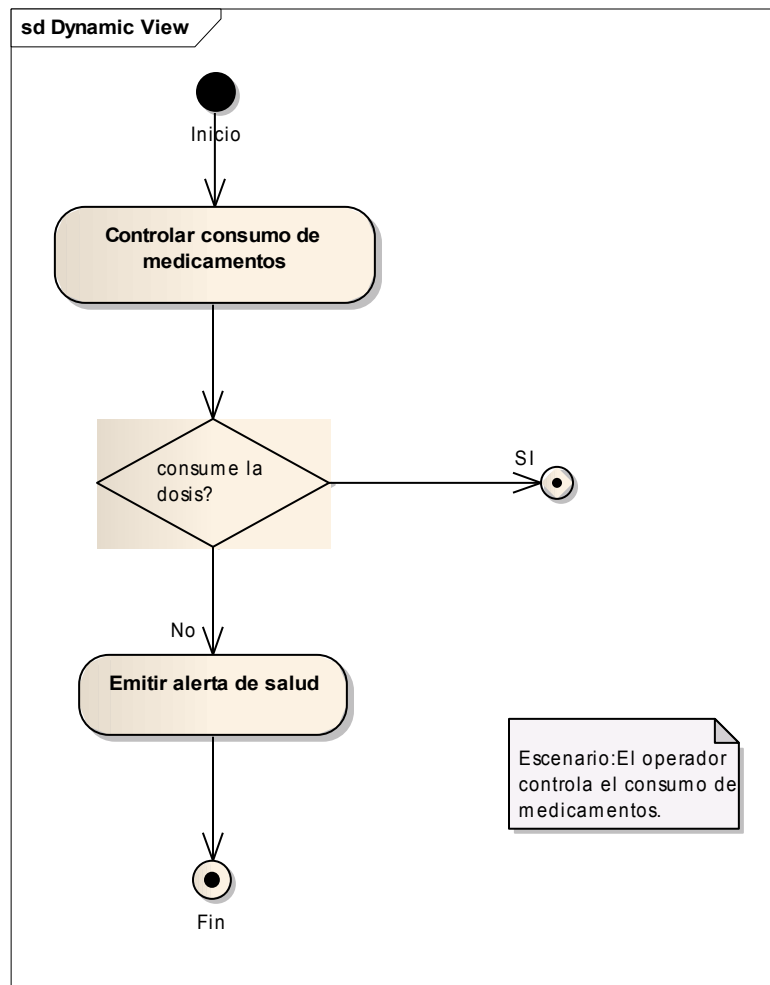
## Diagrama de Componentes



## Diagrama de Despliegue

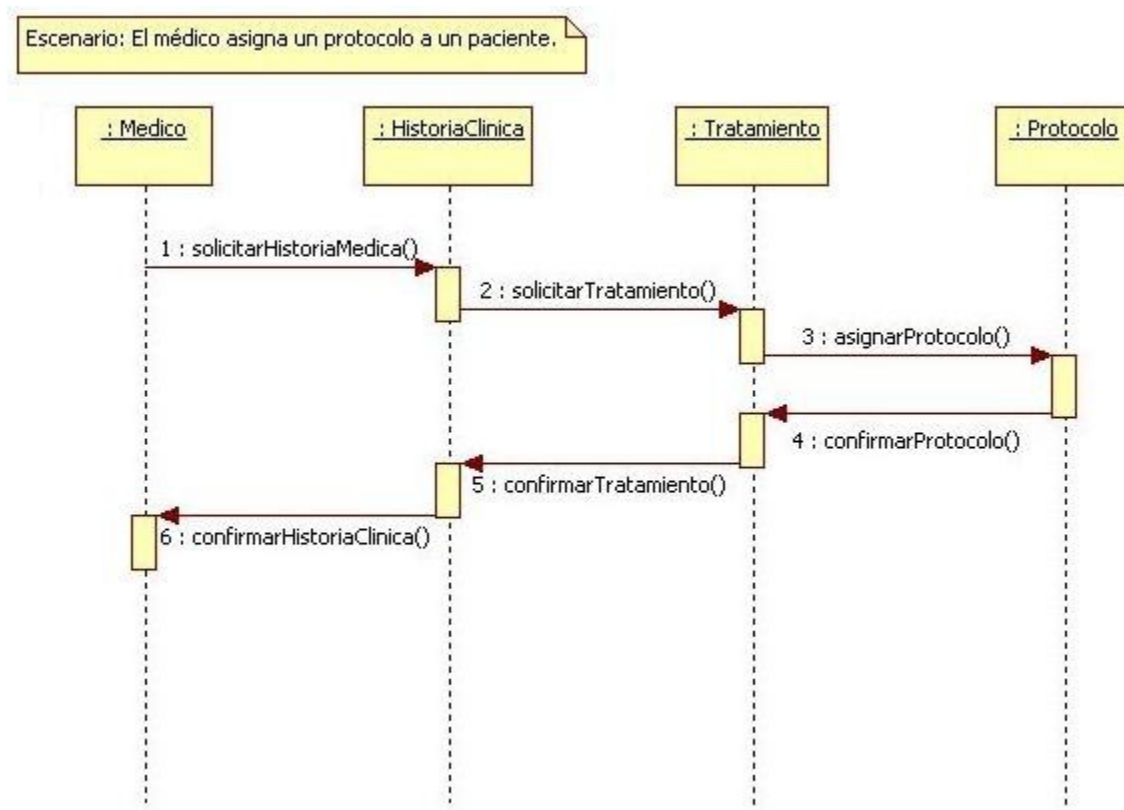


## Diagrama de Actividad

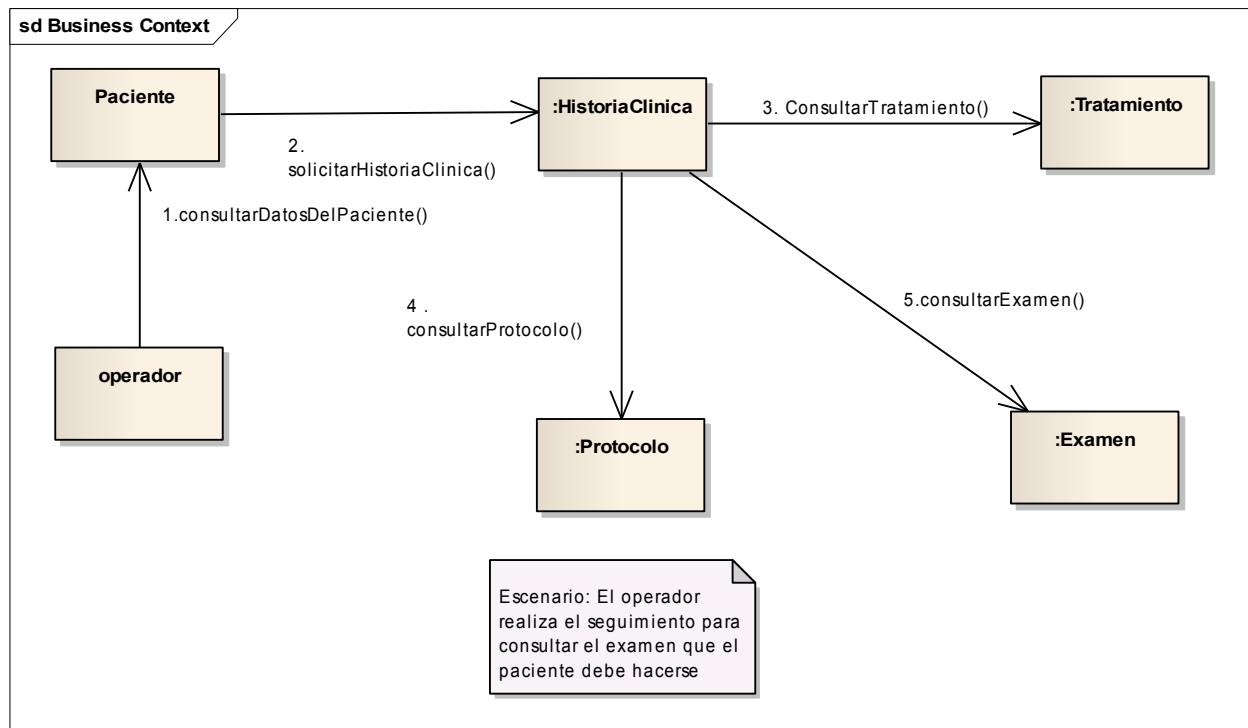




## Diagrama de Secuencia

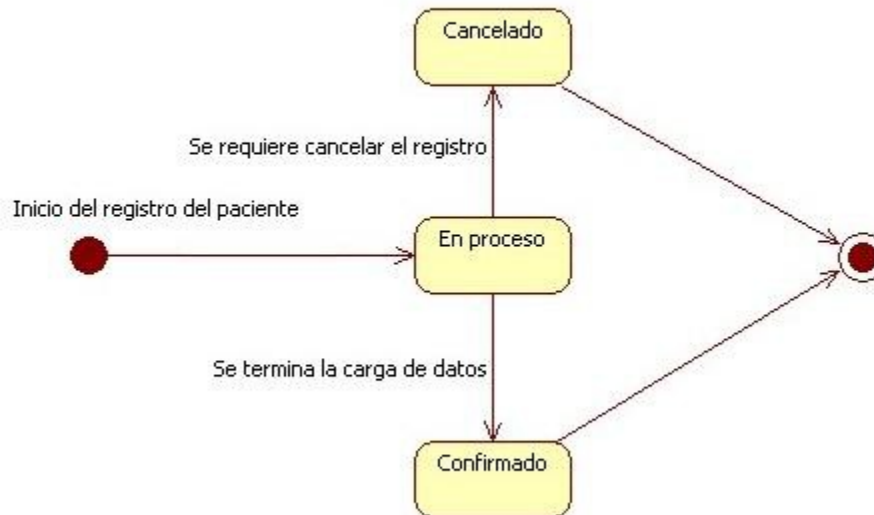


## Diagrama de Comunicación



## Diagrama de Estado

Escenario: Registro de un paciente por parte del operador de admisión



## 4- Patrones de diseño

Se aplicaron los siguientes patrones al diagrama de clases.

### **Singleton:**

Clase Persona, Historia clínica.

Se utiliza para no permitir que existan múltiples instancias de una clase, sino solamente una sola.

### **Strategy:**

Clase Alarma:

Declara una interfaz común para todos los tipos de tratamiento.

Concrete Strategy de 1 a 9: AdquisicionMedicamentos, AusenciaDeComunicacion , AusenciaDeGrupos, ConsumoDeMedicamentos, ReaccionAdversa, IMC, AusenciaDeCitas, Control, IncumplimientoExamen.

Clase Contacto:

Concrete Strategy A: Email.

Concrete Strategy B: TelefonoCelular.

Concrete Strategy C: TelefonoFijo.

Clase Protocolo:

Concrete Strategy A: ProtocoloLeve.

Concrete Strategy B: ProtocoloModerado.

Concrete Strategy C: ProtocoloSevero.

Concrete Strategy: Porque encapsula la complejidad de los distintos protocolos y puede acceder a sus métodos desde otras clases vinculadas a esta.

### **Observer:**

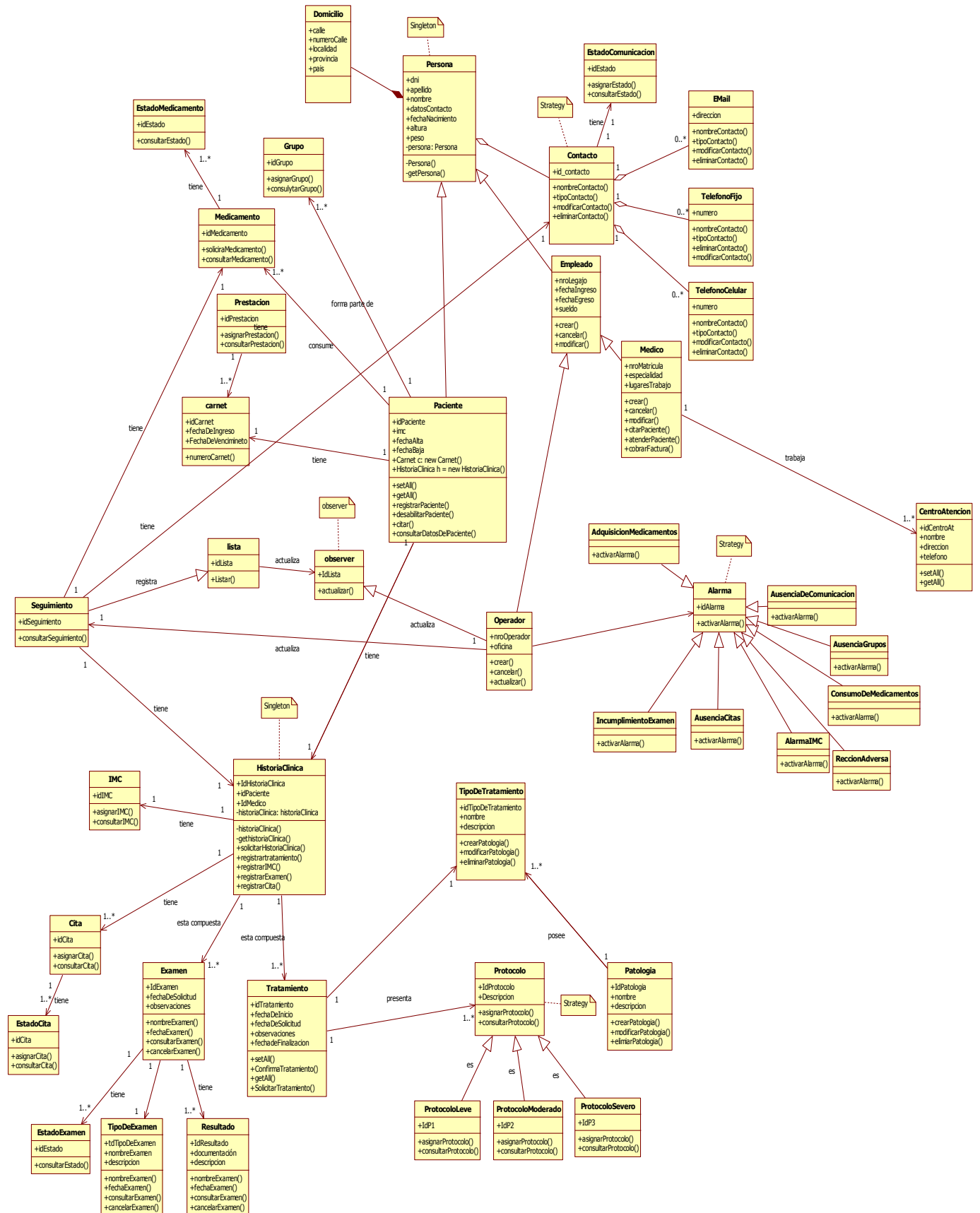
Define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él. Podemos decir entonces que dicho patrón se compone de un objeto observable u Observado (**SujetoConcreto**) y objetos observadores (**ObservadorConcreto**).

Clase Observer: Observer.

Clase concrete Observer: Operador.

Clase sujeto: Lista

Clase sujeto concreto: Seguimiento.



## 5- Diseño de Software – Construcción de Sistemas II

### Introducción

Este documento proporciona una visión general de la arquitectura del sistema, proporcionando diferentes vistas para describir aspectos del mismo a través de diagramas que muestran tanto los componentes que forman el sistema como su interacción. Para dar a conocer las decisiones de diseño que tomará el equipo en cuanto al diseño y desarrollo del proyecto.

### Representación Arquitectónica

La arquitectura de software representa el diseño de más alto nivel de la estructura del sistema.

La misma fue seleccionada y diseñada con base en los objetivos y restricciones planteados inicialmente. Se tienen en cuenta no solo requerimientos de tipo funcional, también otros como la mantenibilidad, flexibilidad, confiabilidad, disponibilidad, etc. que son requerimientos no funcionales del sistema. Se definen como restricciones aquellas limitaciones derivadas de las tecnologías disponibles para implementar el sistema.

El sistema se representa tanto con visiones estáticas como dinámicas.

**Visiones estáticas:** diagrama de clases.

**Visiones dinámicas:** diagrama de estados, diagrama de comunicación, diagrama de actividades, diagrama de secuencia

### Arquitectura

La arquitectura elegida será **Cliente-Servidor**, permitiendo tener clientes distribuidos en los distintos puntos geográficos, accediendo a los servidores que correrán la aplicación y proveerán los servicios necesarios para el acceso y el normal desenvolvimiento del sistema.

El programa cliente, que accederá mediante una aplicación Web, permitirá que el usuario registrado (paciente) entre al sistema, solicitando los datos guardados y ciertos servicios al servidor para continuar con el seguimiento de su enfermedad. A su vez, la aplicación permitirá a médicos y operadores continuar con la supervisión del paciente.

Este programa actuará como interfaz, mostrando los diferentes aspectos del seguimiento así como también las diferentes opciones que tenga el paciente dentro del mismo.

El programa servidor se encargará de realizar la mayor parte del procesamiento, devolviendo los resultados para que sean interpretados por el programa cliente, y así graficados en la interfaz.

La elección de esta arquitectura responde a criterios que se tuvieron en cuenta al momento del diseño, y que se observan en los diferentes juegos similares que se pueden encontrar en línea, a saber:

**Centralización:** el programa servidor contará con la mayoría del control y procesamiento del seguimiento, lo que se hará en una infraestructura que pueda soportar tanto la cantidad de clientes como sus solicitudes de servicios. Al ser esta infraestructura superior a la que cualquier usuario podría tener, se permitirá una velocidad de procesamiento mayor, una restricción de acceso que quedará en manos de los dueños del programa, así como también permitirá una mayor fluidez en el desenvolvimiento normal del mismo.

**Escalabilidad:** La cantidad de clientes no estará restringida, así como también el acceso desde diferentes dispositivos. Además, siempre será posible agregar nuevos servidores para cubrir futuras demandas, siendo transparente para los clientes.

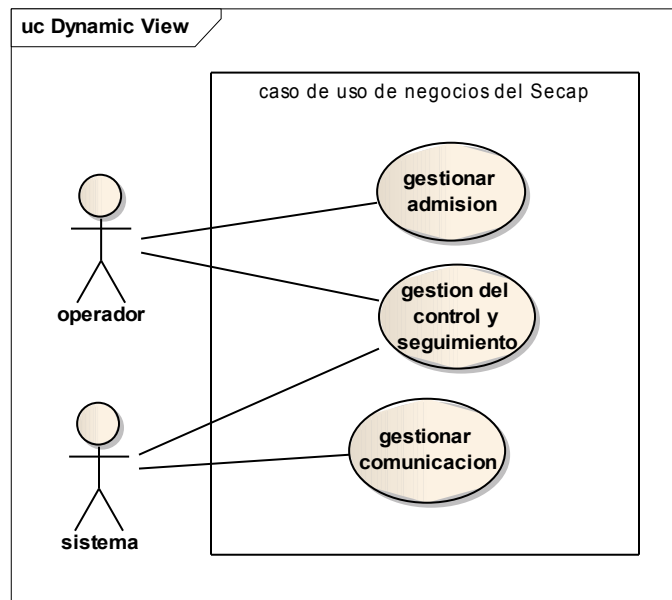
**Fácil mantenimiento:** dado que los programas servidores estarán del lado del propietario de SECAP, la adición de nuevas funcionalidades y elementos en el juego será casi transparente para los programas clientes, quienes podrán o no necesitar una actualización. Asimismo, no se interrumpirá el acceso al seguimiento en ningún momento, de ser necesario realizar cambios.

## Objetivos, límite y alcance del sistema

Los límites, objetivos y alcance del sistema se encuentran detallados en el punto 1

## Vista Funcional

El objetivo de este punto del documento es evidenciar las funcionalidades esenciales del sistema que se está diseñando. Esto puede ser observado mediante el diagrama de casos de uso de negocios, donde se puede observar las funciones clave que permite realizar el sistema, y que además son las que interactúan directamente con el usuario. La explicación de cada una de ellas permite dar un aspecto mas abarcativo de lo que realmente hace la funcionalidad, aquellos aspectos que se le presentan al usuario al momento de interactuar con la interfaz.



A continuación, la descripción de cada uno de ellos:

### **Gestionar admisión:**

Admite e ingresa al paciente, según criterios médicos preestablecidos, y selecciona un protocolo de seguimiento que determina los controles y comunicaciones que se ejecutarán a través del tiempo, en un orden y con periodicidades preestablecidas. Interroga y registra co-morbididades.

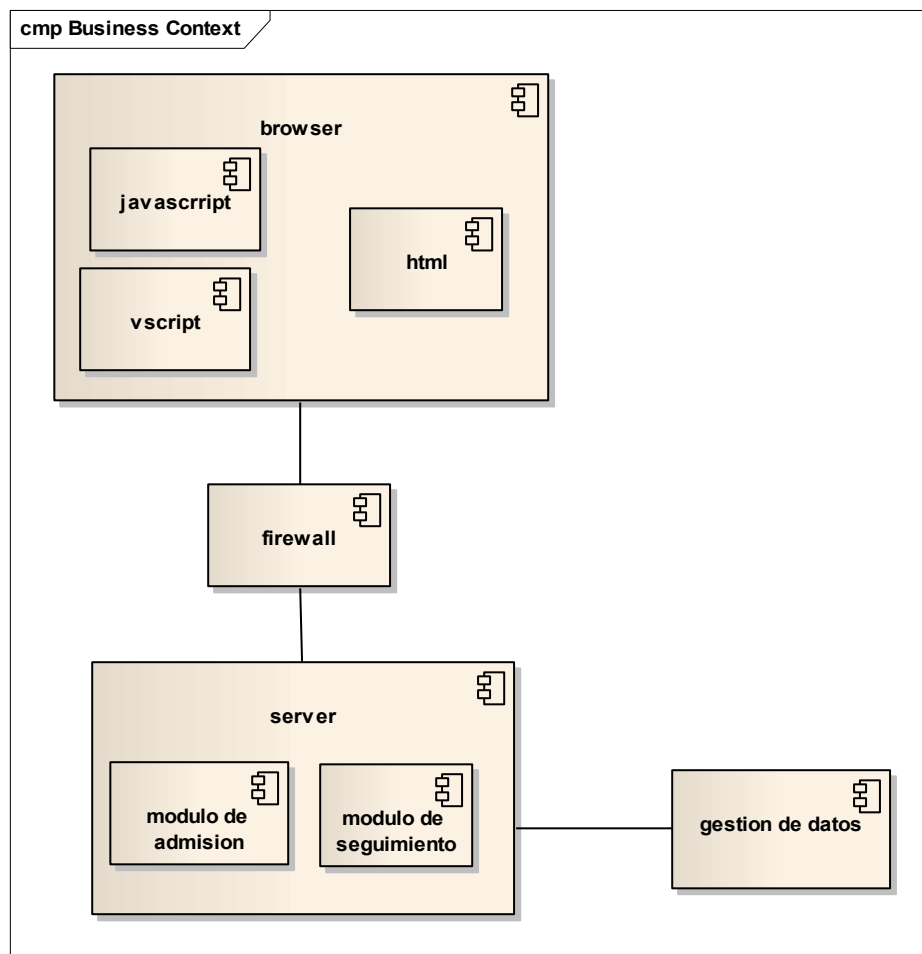
### **Gestión del control y seguimiento:**

Verifica, a lo largo del tiempo, el cumplimiento del protocolo asignado. Los operadores reciben de manera automática, en cada momento, los datos de los pacientes y los controles que deben efectuar. Permite

relevar valores, evaluarlos y enviar alertas automáticas, verificar la adquisición de medicamentos, el cumplimiento de visitas al médico, el Índice de Masa Corporal, reacciones adversas y la realización de exámenes establecidos por el protocolo correspondiente, entre otros. Ante el repetido incumplimiento del consumo de medicamentos y de prácticas fijadas en el protocolo, la imposibilidad de comunicarse con el beneficiario o problemas de salud detectados, el sistema o el operador envía alertas. A su vez, periódicamente se deberán emitir Informes que señalan la evolución de un conjunto de indicadores referidos a beneficiarios, profesionales, cumplimiento de programas, entre otros.

## Vista Lógica

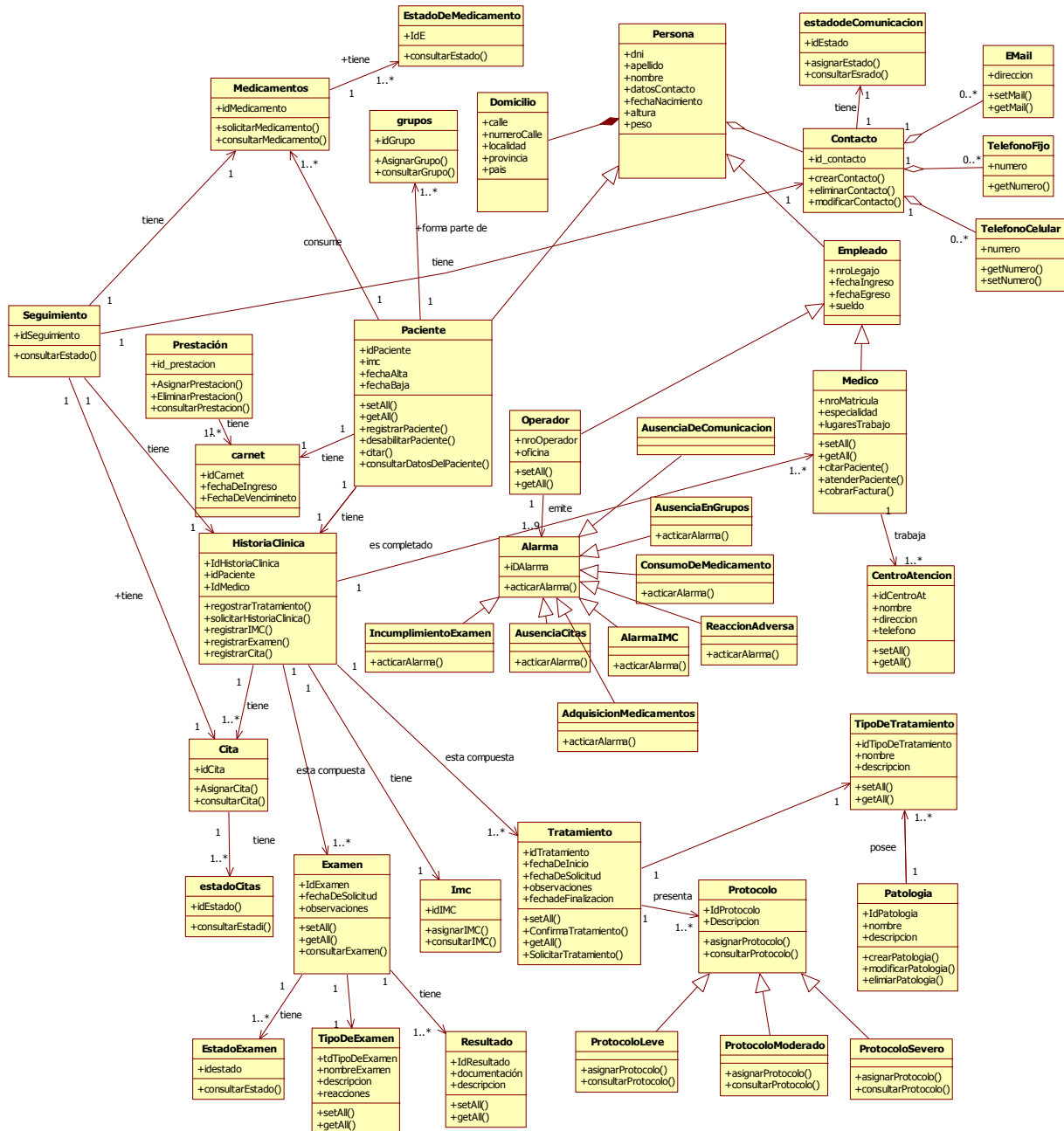
Esta vista nos permite dilucidar la estructura del sistema mediante sus componentes y sus interacciones. Así, se presenta los componentes esenciales que componen el sistema de software, donde se encuentra cada componente en la arquitectura elegida, así como también las interacciones que hay entre los componentes para la arquitectura elegida. A continuación, se ilustra lo expuesto en este punto:



## Vista de Proceso

Esta vista permite observar las clases de análisis tenidas en cuenta para el diseño del software que se realiza, además de presentarse las clases del sistema con sus relaciones estructurales y de herencia. Muchas de las clases que se han incluido en el diagrama presentado a continuación surgieron del análisis del software y su funcionamiento, y se tomaron en cuenta los casos de uso principales propuestos anteriormente. Muchas de las clases especificadas a continuación son vitales para el desarrollo de los

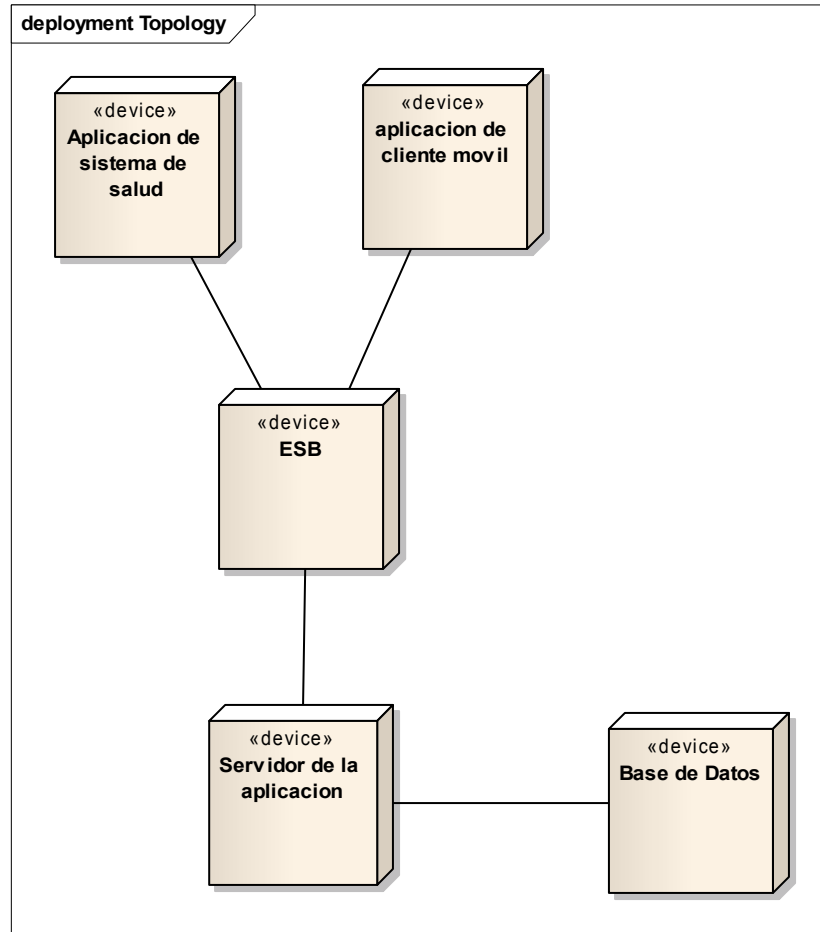
misimos. A continuación se presenta el modelo de clases de análisis elaborado para el sistema analizado



## Vista de Despliegue

El siguiente diagrama de despliegue muestra de manera fácil y grafica los elementos que conforman el sistema, así como la interacción entre los distintos elementos, para que el software pueda realizar lo especificado en los requerimientos.





## Calidad

Para realizar la evaluación correspondiente, se tuvieron en cuenta los siguientes atributos de calidad:

### **Usabilidad:**

El sistema contara con una interfaz intuitiva y fácil de entender, permitiendo que personas de todas las edades puedan hacer uso de la aplicación con poco conocimiento de la misma.

### **Eficiencia:**

El sistema se desenvolverá de manera ágil y fluida en todo momento, sea que corra por medio de una aplicación Web, o una aplicación local, sin importar el dispositivo en el cual se ejecute.

### **Mantenibilidad**

El sistema está pensado para poder ser sujeto a mantenimiento y que la aplicación no se vea afectada en ningún momento. Tanto la arquitectura como la infraestructura soportaran este aspecto de la calidad.

### **Seguridad:**

El sistema no permitirá el ingreso indebido de usuarios no registrados, así como también utilizara los medios necesarios para comprobar la autenticidad de los datos ingresados. Esto deberá realizarse mediante diferentes medidas de seguridad, y preguntas al usuario cuando cambie de dispositivo o zona de conexión habitual.

***Disponibilidad:***

La arquitectura elegida, así como también la infraestructura donde se monte, deberá considerar el máximo tiempo de disponibilidad de la aplicación.

## 6- Diseño de la Interfaz de Usuario

### ¿Qué es un prototipo Web?

Un prototipo Web es un bosquejo o modelo inicial a partir del cual se empieza a concebir y desarrollar la idea original del diseño de un sitio Web.

Es más sencillo y económico hacer un prototipo, y ajustarlo gradualmente hasta lograr el mejor modelo posible, que hacer un diseño Web real y luego modificarlo hasta alcanzar lo que se busca.

El prototipado de las páginas Web resuelven básicamente los siguientes aspectos:

- qué elementos deben conformar la interface gráfica (de cada página),
- cuántos elementos deben conformar la interface gráfica para que haya suficiencia en la información/interacción pero evitando la saturación de elementos (de cada página),
- cómo debe organizarse la jerarquía visual (en qué orden y disposición van los elementos)
- qué extensión (superficie visual o tamaño) adecuado deben tener aprovechando eficientemente el espacio bidimensional disponible

### ¿Cómo modularizar la interface gráfica?

Podemos separar la interface gráfica en dos áreas:

1. **Layout genéricos** (partes comunes que se reutilizan en varias páginas Web, ej: header, footer).
2. **Wireframes** (elementos individuales que tienen usos particulares: grilla de datos, buscador, etc.).

## Herramientas de diseño de prototipos

### i. Hot Gloo

Es una aplicación en línea desarrollada en Flash para crear Wireframes, cuenta con funciones como arrastrar y soltar elementos, capas y más, además se puede recibir una URL para compartir el trabajo con colegas.

#### **Ventajas:**

- i. Fácil de usar.
- ii. Independiente.
- iii. Colaboración.
- iv. Interactivo.

### ii. Mockingbird

Mockingbird es una herramienta popular para la creación de maquetas y wireframes. Te permite fácilmente crear, previsualizar, relacionar entre sí, y compartir todos tus wireframes con quien quieras.

#### **Ventajas:**

- i. Ahorro de tiempo y dinero.
- ii. Satisfacción del cliente.
- iii. Foco en el equipo.
- iv. Construcción de mejores productos.

### iii. Pencil Project

Esta es una extensión para Firefox que nos permite hacer diagramas detallados e interfaces de usuario increíbles, viene con una variedad de características tales como documentos de varias páginas, en la edición de texto en pantalla, soporte de texto enriquecido, páginas de fondo, la exportación de HTML, PNG y Openoffice.

**Ventajas:**

- i. Extensibilidad.
- ii. Herramientas de dibujo vectorial.
- iii. Plantillas, prototipos.

#### **iv. ProtoShare**

ProtoShare proporciona una herramienta de colaboración fácil de utilizar que permite la creación de proyectos dinámicos que requieren de prototipos y wireframes.

**Ventajas:**

- i. Colaboración fluida.
- ii. Gestión automática y mantenimiento.
- iii. Acceso único a la información.

#### **v. Balsamiq**

Balsamiq es una herramienta interactiva de wireframing que se siente como si estuvieras dibujando con un lápiz y papel, sólo una en la Web. La diferencia de esta herramienta se encuentra en el control total de los elementos en cuanto su movilidad y posicionamiento.

**Ventajas:**

- i. Crear mejor software.
- ii. Mejora la comunicación con el equipo.
- iii. Centrarse en lo que importa.
- iv. Utilización con el cliente.

#### **vi. Flair Builder**

FlairBuilder permite un rápido despliegue de wireframes mediante el uso de una plataforma cruzada que alberga las diferentes funciones y características útiles dentro de una sola paleta.

**Ventajas:**

- i. Alta fidelidad.
- ii. Inclusión total de todos en el proceso.
- iii. Impresiona a clientes.
- iv. Prototipar, refinar y revisar.

#### **vii. Cacao**

Cacao es una gran herramienta de trabajo para dibujo en línea que te permite desarrollar varios diagramas como los gráficos de una red alámbrica, o los mapas de un sitio, además puedes permitir que varios usuarios accedan a tus prototipos para hacer cambios o simplemente ver tu progreso, por lo que lo convierte en una herramienta de colaboración.

**Ventajas:**

- i. Crear, compartir diagramas.
- ii. Colaboración.

- iii. Gestión de esquemas.
- iv. Idiomas y zonas horarias.
- v. Seguridad.
- vi. API.

### **viii. Serena: Prototype Composer**

Prototype Composer permite ver como se será una aplicación antes de empezar a escribir código.

#### **Ventajas:**

- i. Ingeniería de valor.
- ii. On demand.
- iii. Visión de conjunto.

### **ix. MockFlow**

MockFlow viene con una amplia biblioteca en línea de plantillas que te permiten usarlas libremente y mejorar la calidad visual de tus wireframes.

#### **Ventajas:**

- i. Editor.
- ii. Biblioteca.
- iii. Publicación.
- iv. Colaboración.
- v. Negocios.
- vi. Herramientas de pizarra.

### **x. fluidIA**

fluidIA es una herramienta ágil para el diseño de prototipos de interfaces de usuario ricas. La gran idea detrás de este experimento es si nosotros, como diseñadores podemos crear nuestra propia herramienta de creación de prototipos de una manera abierta. fluidIA también proporciona velocidad, agilidad, colaboración increíble, y una rica interacción entre el diseñador y la interfaz de usuario.

#### **Ventajas:**

- i. Velocidad y agilidad.
- ii. Colaboración.
- iii. Interacción Rich.
- iv. Alternancia.

### **xi. GUI Studio Design**

GUI Design Studio es una herramienta de diseño de software especializado para cualquiera involucrado en el diseño de interfaz de usuario de aplicaciones, incluyendo diseñadores de experiencia de usuario, analistas, desarrolladores, administradores de proyectos y consultores.

#### **Ventajas:**

- i. Ahorrar tiempo y dinero.
- ii. Disminuir riesgos.
- iii. Mejora de usabilidad y comprensión por parte del equipo y el cliente.

### **xii. iPlotz**

iPlotz permite crear wireframes en los que se puede hacer clic y navegar a través de ellos, esencialmente se trata de crear o recrear la experiencia de un sitio Web real.

**Ventajas:**

- i. Compatibilidad.
- ii. Colaboración.
- iii. Proyecto y gestión de tareas.
- iv. Wireframing.

### **xiii. OmniGraffle**

OmniGraffle puede ayudarte a hacer alucinantes documentos gráficos rápidamente por gracias a la técnica de líneas conectadas como diagramas, proporcionando herramientas de diseño de gran alcance, además te permite importar y exportar archivos de Microsoft Visio.

**Ventajas:**

- i. Diversidad de estilos.
- ii. Grafos, tablas, interacciones.
- iii. Sincronización
- iv. Herramientas gráficas sofisticadas.

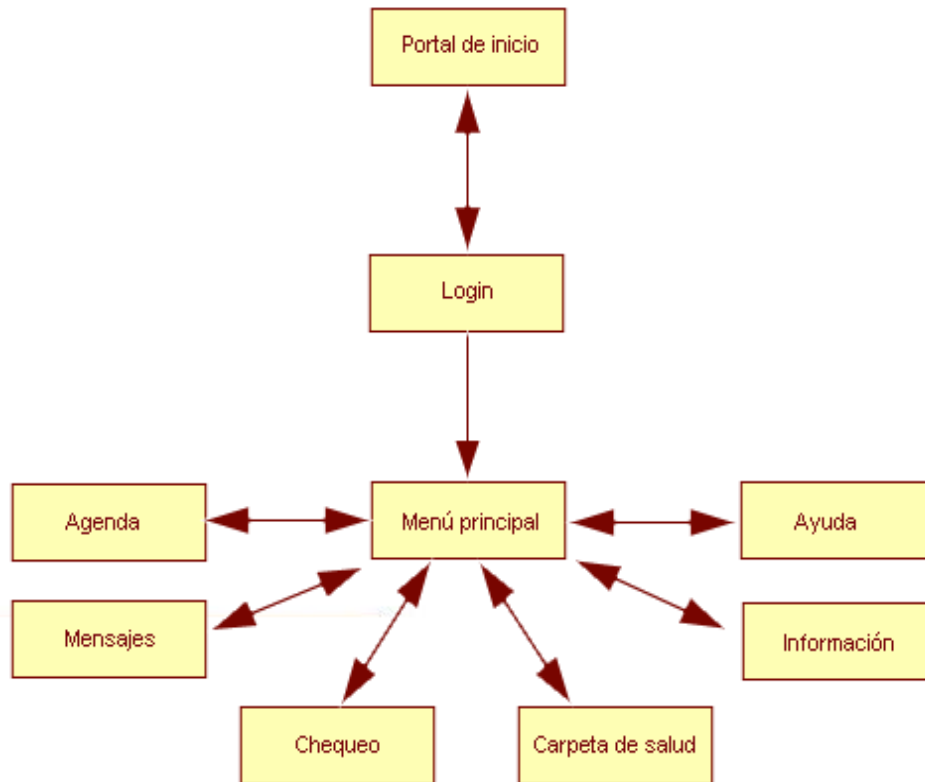
### **xiv. MockupScreens**

MockupScreens es una herramienta interactiva que te permite dibujar maquetas en pantalla, crear un sistema de retroalimentación, y obtener los mejores resultados posibles. Así podrás ver cómo interactúan todos antes de empezar con el código.

**Ventajas:**

- i. Aclarar requisitos.
- ii. Manejar expectativas de usuario.
- iii. Acelerar diseño y desarrollo.
- iv. Creación de maquetas.

## Diagrama de navegación



## 7- Prototipo Web (Interfaz operador/ modulo de seguimiento)

Seguimiento de Pacientes Crónicos

Operador/a: Ariela Gimenez

Salir

Paciente: Oscar Martinez

sepac

Buscar paciente

Ingrese nombre y/o apellido

Realizar nueva busqueda

Datos del paciente

Nombre: Oscar

Apellido: Martinez

ID: 3388-77

Plan: DB1

Enfermedad: diabetes tipo A

Protocolo: TBD1 asignado

Información paciente



Biomedidas tomadas



Controles pendientes



Exámenes realizados



Verificar cita médica

Seguimiento



Reacciones adversas



Consumo de medicamentos



Adquisición de medicamentos



Ejercicios

Tareas del paciente

12:00 Toma de biomedidas

13:00 Toma de biomedidas

14:00 Teleconferencia

15:00 Cita médica

Ver toda la agenda

Últimos mensajes

No tiene mensajes nuevos

Ver mensajes anteriores



## Anexo

Código de las clases más importantes:

```
public class Paciente extends Persona
{
    /** */
    public Object idPaciente;
    /** */
    public Object imc;
    /** */
    public Object fechaAlta;
    /** */
    public Object fechaBaja;
    /** */
    public Citar Unnamed1;
    /** */
    public Factura Unnamed2;
    /** */
    public HistoriaClinica Unnamed3;
    /** */
    public Factura Unnamed4;
    /** */
    public HistoriaClinica Unnamed5;
    /** */
    public carnet Unnamed6;
    /** */
    public HistoriaClinica Unnamed7;
```

```

/** */
public carnet Unnamed8;

/** */
public void setAll()
{
}

/** */
public void getAll()
{
}

/** */
public void registrarPaciente()
{
}

/** */
public void desabilitarPaciente()
{
}

/** */
public void citar()
{
}

/** */
public void consultarDatosDelPaciente()
{
}
}

```

```

public class Tratamiento
{
    /** */
    public Object idTratamiento;
    /** */
    public Object fechaDelInicio;
    /** */
    public Object fechaDeSolicitud;
    /** */
    public Object observaciones;
    /** */
    public Object fechadeFinalizacion;
    /** */
    public HistoriaClinica Unnamed1;
    /** */
    public TipoDeTratamiento Unnamed2;
    /** */
    public Patologia Unnamed3;
    /** */
    public CentroAtencion Unnamed4;
    /** */
    public Protocolo Unnamed5;
    /** */
    public Protocolo Unnamed6;
    /** */
    public TipoDeTratamiento Unnamed8;
    /** */

```

```

    public void setAll()
    {
    }

    /** */

    public void ConfirmaTratamiento()
    {
    }

    /** */

    public void getAll()
    {
    }

    /** */

    public void SolicitarTratamiento()
    {
    }
}

public class HistoriaClinica
{
    /** */

    public Object IdHistoriaClinica;

    /** */

    public Object idPaciente;

    /** */

    public Object IdMedico;

    /** */

    public Paciente Unnamed1;

    /** */

```

```

    public Examen Unnamed2;

    /** */

    public Tratamiento Unnamed3;

    /** */

    public Medico Unnamed5;

    /** */

    public void setAll()

    {

    }

    /** */

    public void getAll()

    {

    }

    /** */

    public void solicitarHistoriaClinica()

    {

    }

}

public class Patologia

{

    /** */

    public Object IdPatologia;

    /** */

    public Object nombre;

    /** */

    public Object descripcion;

    /** */

```

```
public Tratamiento Unnamed1;

/** */

public TipoDeTratamiento Unnamed2;

/** */

public TipoDeTratamiento Unnamed3;

/** */

public void crearPatologia()

{

}

/** */

public void modificarPatologia()

{

}

/** */

public void elimiarPatologia()

{

}

}
```