



FACULTAD CIENCIAS Y TECNOLOGÍAS COMPUTACIONALES

Título: Registro de Pacientes Incluidos en Ensayos Clínicos

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias
Informáticas

Autor(es):

Rubén Álvarez Izquierdo
Michel Leandro Pérez Somoza

Tutor(es):

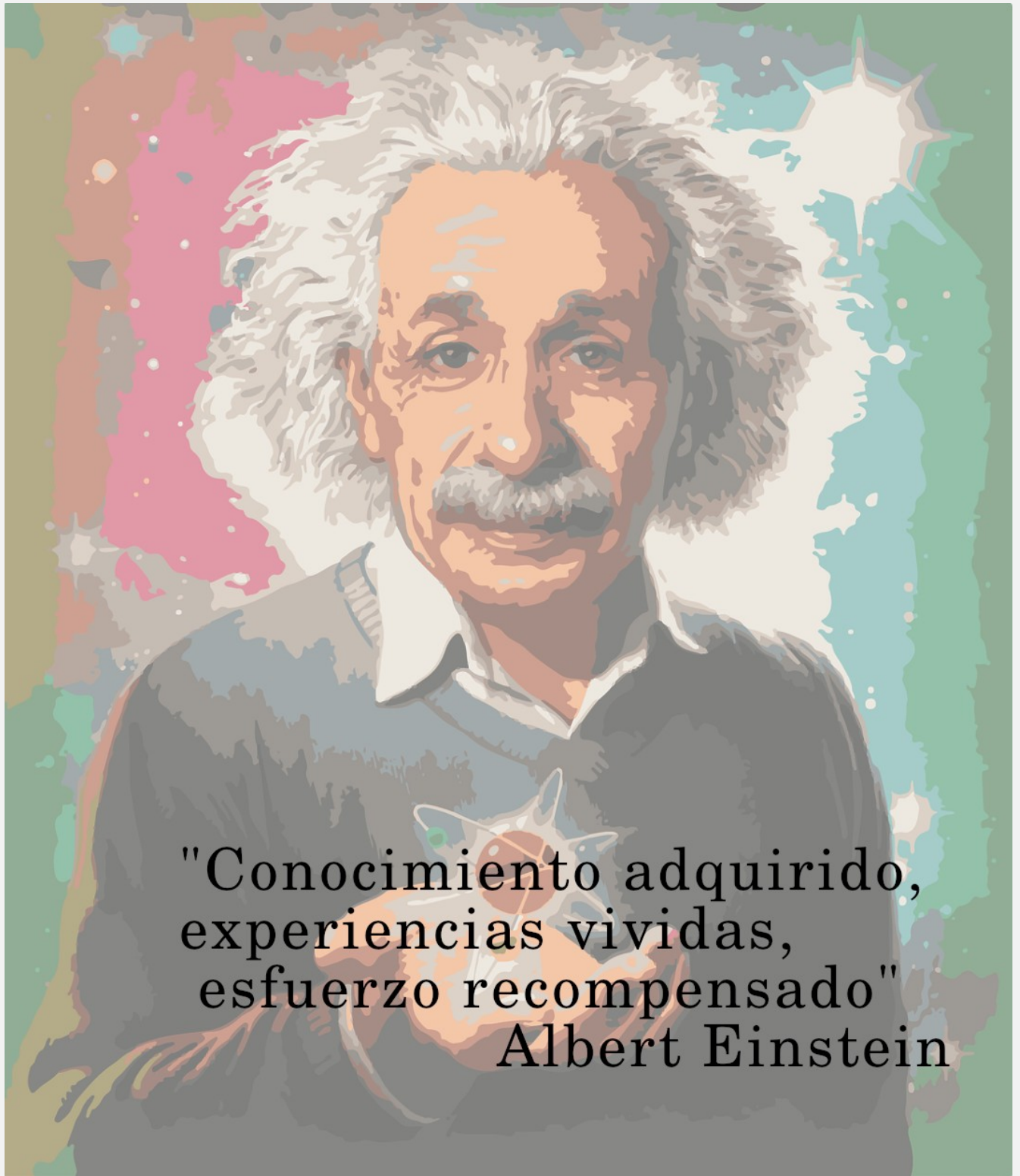
P.A Ing. Alejandro Martínez Valdivia

Consultante(s):

,

La Habana, junio de 2023

“Año 64 de la Revolución”



"Conocimiento adquirido,
experiencias vividas,
esfuerzo recompensado"
Albert Einstein

DECLARACIÓN DE AUTORÍA

DECLARACIÓN DE AUTORÍA

Los autores del trabajo de diploma con título "Registro de Pacientes Incluidos en Ensayos Clínicos" conceden a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar, se declaran como únicos autores de su contenido. Para que conste, firman la presente a los 6 días del mes de junio del año 2023.

Rubén Álvarez Izquierdo

Firma del Autor

Michel Leandro Pérez Somoza

Firma del Autor

P.A Ing. Alejandro Martínez Valdivia

Firma del Tutor

DATOS DE CONTACTO

DATOS DE CONTACTO

Tutor

- Nombre y apellidos: Alejandro Martínez Valdivia
- Graduado de nivel: Universitario
- Especialidad: Ingeniería en Ciencias Informáticas
- Especialista "B" en Ciencias Informáticas, Centro de Representación y Análisis de Datos.
- Área: Departamento de Matemática
- Correo: avaldivia@uci.cu

AGRADECIMIENTOS

Rubén Álvarez Izquierdo

Durante todos estos años de estudio para convertirme en ingeniero en Ciencias Informáticas, muchas personas me han ayudado a forjar la persona que soy. Por eso, quiero expresar mi más profundo agradecimiento a:

- Mi madre, mi padre, mi hermana y mi abuela por estar siempre presentes cuando más los necesité, brindándome su apoyo y aliento incondicional para cumplir mi sueño.
- A esos amigos que conocí en los últimos años, con quienes compartí momentos muy divertidos y especiales.
- A todas las personas que me brindaron su ayuda a lo largo de mi carrera, aportando su granito de arena para mi crecimiento y desarrollo.
- A cada uno de los profesores que tuve, quienes no solo me transmitieron conocimientos, sino también valores fundamentales.
- A la Universidad de las Ciencias Informáticas (UCI) y a la revolución cubana por brindarme la oportunidad de superarme de forma gratuita.
- A todos, ¡muchas gracias!

AGRADECIMIENTOS

Michel Leandro Perez Somoza

Quiero expresar mi mas profundo agradecimiento a mis padres y mi hermano por estar a mi lado en cada paso de mi vida. Su constante apoyo, amor incondicional y palabras de aliento han sido fundamentales para mi crecimiento y desarrollo como persona. Siempre han estado dispuestos a ayudarme en momentos de dificultad y han sido mi mayor impulso para seguir adelante. No tengo palabras suficientes para describir todo lo que significan para mi. Gracias por creer en mi y por ser mi apoyo incondicional.

Tambien quiero agradecer a mis abuelas y mi bisabuela, quienes siempre han pensado en mi como un futuro profesional exitoso. Su vision y confianza en mis capacidades han sido un gran estimulo para perseguir mis sueños y alcanzar mis metas academicas. Siempre han estado presentes en momentos importantes de mi vida, animandome a seguir adelante y recordandome la importancia de la educacion. Estoy eternamente agradecido por su amor y aliento constante.

Por ultimo, quiero expresar mi gratitud a mis amistades dentro y fuera de la universidad. Nuestra amistad y apoyo han sido un pilar fundamental durante mi tiempo en la universidad. Nuestra energia positiva, vuestras risas y nuestro animo incondicional han hecho que este camino sea mas llevadero. Gracias por estar siempre dispuestos a escucharme, aconsejarme y acompañarme en cada paso que he dado. Nuestra amistad ha enriquecido mi experiencia universitaria y la vida en general. No puedo expresar con palabras lo agradecido que estoy por teneros en mi vida. Gracias por ser parte de mi historia.

RESUMEN:

RESUMEN:

Cuba se encuentra inmersa en un proceso de informatización de la sociedad desde hace varios años, siendo uno de los pilares fundamentales de la gestión gubernamental. Aunque aún se necesitan más acciones para lograr un desarrollo completo del país, se han logrado avances significativos. El uso de las Tecnologías de la Información y las Comunicaciones (TIC) en el sector de la salud es clave dentro del Programa Estratégico de Informatización del MINSAP en Cuba. Como parte de este programa, se ha desarrollado el Sistema de Información para la Salud (SISalud), proyecto que busca mejorar la gestión de la información del Sistema Nacional de Salud cubano.

SISalud es una solución integral que abarca diferentes áreas de la prestación de servicios de salud, incluyendo la Atención Primaria, Secundaria y Terciaria, el Sistema Integrado de Urgencia Médica, la Vigilancia en Salud, la Docencia Médica, la Telemedicina, los Medicamentos y Fármacos, la Epidemiología, entre otros. Para llevar a cabo el diseño de esta solución, se cuenta con la colaboración de la Empresa SOFTEL, la Universidad de Ciencias Informáticas y el Centro de Informática Médica (CESIM).

El objetivo de esta investigación es desarrollar un subsistema para SISalud que permita calcular, gestionar y manejar la información de los pacientes incluidos en los ensayos clínicos que son manejados por el sistema Xavia SIDEC, desarrollado por CESIM. Como resultado de este desarrollo, se ha creado una aplicación web que se basa en la metodología ágil AUP en su versión UCI. La base de datos se ha implementado utilizando PostgreSQL, mientras que los lenguajes de programación utilizados son JavaScript en conjunto con TypeScript, aprovechando los frameworks AngularJS y NestJS.

El beneficio fundamental que ofrece este subsistema radica en el perfeccionamiento y la organización de la información relacionada con los pacientes incluidos en los ensayos clínicos dentro de SISalud. Esto permite una gestión más rápida, eficiente y precisa de dicha información, asegurando su accesibilidad.

Palabras clave: SISalud, Xavia SIDEC, Gestión de la información de pacientes, Ensayos Clínicos, Informatización, AUP versión UCI, PostgreSQL, JavaScript, TypeScript, Angular y NestJS.

ABSTRACT

ABSTRACT

Cuba has been immersed in a process of computerization of society for several years, one of the three axes that support government management. The actions carried out, although they are still not sufficient for the development of the country, have led to indisputable progress. The use of new Information and Communications Technologies (ICT) in the health sector is one of the keys to the MINSAP Strategic Informatization Program in Cuba. As part of said Strategic Informatization Program, the Health Information System (SISalud) is being developed, a project that will improve the management of information related to the Cuban National Health System.

SISalud as a comprehensive solution is the articulation of a pattern in the provision of health services, which includes: Primary, Secondary and Tertiary Care, the Integrated Medical Emergency System, Health Surveillance, Medical Teaching, Telemedicine, Medications and Drugs, Epidemiology, among others. The SOFTEL Company, in charge of carrying out the design of this solution, organizes and directs the production process together with the University of Computer Sciences and the Center for Medical Informatics (CESIM).

The objective of this research is to develop a subsystem for SISalud that allows calculating, managing and handling the registration information of patients included in clinical trials managed by the Xavia SIDEC system, developed by CESIM.

With the development of this subsystem, a Web application was obtained, which is based for its creation on an agile methodology, AUP in its UCI version. The database was created in PostgreSQL, JavaScript was used as programming languages in conjunction with TypeScript, leveraged in two frameworks such as AngularJS and NestJS.

As a fundamental benefit granted by the subsystem, it focuses on the improvement and organization of information related to patients included in clinical trials within SISalud, in such a way that the ease of management of said information is faster, more efficient and at the same time accessible and accurate.

KEYWORDS:

SISalud, Xavia SIDEC, Patient information management, Clinical Trials, Computerization, AUP UCI version, PostgreSQL, JavaScript, TypeScript, Angular and NestJS.

ÍNDICE

DECLARACIÓN DE AUTORÍA.....	I
DATOS DE CONTACTO.....	II
AGRADECIMIENTOS.....	III
RESUMEN:.....	V
ABSTRACT.....	VI
ÍNDICE DE TABLAS.....	X
ÍNDICE DE FIGURAS.....	XII
AVAL DEL CLIENTE.....	XIV
INTRODUCCIÓN.....	1
CAPÍTULO 1. FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LOS PROCESOS DE REGISTRO DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS..	6
1.1. INTRODUCCIÓN DEL CAPÍTULO.....	6
1.2. CONCEPTOS RELACIONADOS CON EL PROCESO DE DESARROLLO DEL SISTEMA DE GESTIÓN DE PACIENTES INCLUIDOS EN ENSAYOS Clínicos.....	6
1.2.1. ENSAYOS CLÍNICOS.....	6
1.2.2. GESTIÓN DE ENSAYOS CLÍNICOS.....	7
1.2.3. SISALUD.....	7
1.2.4. XAVIA SIDEC.....	9
1.2.5. SISTEMA ÚNICO DE IDENTIDAD NACIONAL (SUIN).....	10
1.2.6. GESTIÓN DE PACIENTES:.....	10
1.3. ANÁLISIS DE SOLUCIONES SIMILARES.....	10
1.3.1. OPENCLÍNICA.....	11
1.3.2. ALASCLÍNICAS.....	12
1.3.3. MEDIDATA RAVE:.....	14

ÍNDICE

1.3.4. RESULTADOS DEL ANALISIS DE SOLUCIONES SIMILARES.....	15
1.4. HERRAMIENTAS Y TECNOLOGÍAS.....	16
1.4.1. METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	24
1.4.2 CARACTERÍSTICAS DE AUP-UCI, ESCENARIO 4.....	25
CAPÍTULO 2: DESARROLLO DEL SISTEMA DE PACIENTES.....	27
INTRODUCCIÓN DEL CAPÍTULO.....	27
2.1 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN.....	27
2.1.1 MODELO CONCEPTUAL.....	28
2.1.2 DEFINICIÓN DE CLASES.....	29
2.2 DEFINICIÓN DE REQUISITOS.....	30
2.2.1 REQUISITOS FUNCIONALES.....	30
2.2.2 REQUISITOS NO FUNCIONALES.....	34
2.3 DESCRIPCIÓN DE LOS REQUISITOS.....	36
2.2.4 DIAGRAMA DE CLASES DEL DISEÑO.....	40
2.3 DISEÑO DE LA ARQUITECTURA DE SISTEMA.....	41
2.3.1 PATRONES DE DISEÑO.....	42
2.3.2 PATRONES GOF.....	45
2.4 DIAGRAMA DE BASE DE DATOS.....	46
CONCLUSIONES DEL CAPÍTULO.....	46
CAPÍTULO 3. VALIDACIÓN DEL SISTEMA DE GESTIÓN DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.....	48
3.1. INTRODUCCIÓN DEL CAPÍTULO.....	48
3.2. IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN.....	48

ÍNDICE

3.2.1. ESTÁNDARES DE CODIFICACIÓN.....	48
3.2.2. MODELO DE COMPONENTES.....	49
3.2.3. MODELO DE DESPLIEGUE.....	50
3.3. VALIDACIÓN DE REQUISITOS.....	51
3.3.1. MÉTRICAS APLICADAS A LOS REQUISITOS.....	52
3.4. PRUEBAS DE SOFTWARE.....	53
3.4.2 ESTRATEGIA DE PRUEBA.....	54
3.4.3 PRUEBA DE UNIDAD, CAMINO BÁSICO.....	54
3.4.4 PRUEBA DE INTEGRACIÓN.....	56
3.4.4.1 DETALLES DE LA PRUEBA.....	57
3.4.4.2 EJEMPLO DE PRUEBA DE INTEGRACIÓN.....	58
3.4.5 PRUEBA DE SISTEMA, PRUEBA DE RENDIMIENTO.....	58
3.5. CONCLUSIONES DEL CAPÍTULO.....	61
CONCLUSIONES GENERALES.....	62
REFERENCIAS BIBLIOGRÁFICAS.....	63
ANEXOS.....	70

ÍNDICE DE TABLAS

ÍNDICE DE TABLAS

Tabla 1: Tabla resumen.....	16
Tabla 2: Requisitos funcionales.....	34
Tabla 3: Requisitos no funcionales.....	36
Tabla 4: HU Registrar Paciente Incluido.....	38
Tabla 5: HU Buscar Paciente Incluido.....	40
Tabla 6: Pruebas aplicadas a la aplicación.....	54
Tabla 7: Prueba unitaria.....	56
Tabla 8: caso de prueba de integración del componente“HealthPromoterListComponent”	58
Tabla 9: HU Buscar cantidad de pacientes incluidos por año de inclusión.....	72
Tabla 10: HU Buscar cantidad de pacientes incluidos por ensayos clínicos.....	74
Tabla 11: HU Buscar cantidad de pacientes incluidos por promotores.....	76
Tabla 12: HU Buscar cantidad de pacientes incluidos por provincias.....	78
Tabla 13: HU Buscar cantidad de pacientes incluidos por rango de edades.....	80
Tabla 14: HU Buscar cantidad de pacientes incluidos por sitios de investigación.....	82
Tabla 15: HU Registrar Codificador de Ensayos.....	84
Tabla 16: HU Registrar Codificador de Promotores.....	86

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE FIGURAS

Figura 1: Página principal OpenClínica.....	11
Figura 2: AlasClínicas.....	12
Figura 3: Medidata RAVE.....	14
Figura 4: Modelo conceptual.....	28
Figura 5: Interfaz gráfica insertar paciente incluido.....	38
Figura 6: Interfaz gráfica buscar paciente incluido.....	40
Figura 7: Diagrama de clase Gestionar Pacientes Incluidos.....	41
Figura 8: Patrón arquitectónico.....	42
Figura 9: Estructura de app.module.ts.....	43
Figura 10: Ejemplo del routing.....	44
Figura 11: Patrón decorador.....	45
Figura 12: Diagrama de base de datos.....	46
Figura 13: Modelo de componentes.....	50
Figura 14: modelo de despliegue.....	51
Figura 15: Método loadPagePatients(number).....	55
Figura 16: Grafo representativo.....	56
Figura 17: Resultado de jmeter.....	61
Figura 18: Buscar cantidad de pacientes incluidos por año de inclusión.....	72
Figura 19: Buscar cantidad de pacientes incluidos por ensayos clínicos.....	74

ÍNDICE DE FIGURAS

Figura 20: Buscar cantidad de pacientes incluidos por promotores.....	76
Figura 21: Buscar cantidad de pacientes incluidos por provincias.....	78
Figura 22: Buscar cantidad de pacientes incluidos por rango de edades.....	80
Figura 23: Buscar cantidad de pacientes incluidos por sitios de investigación.....	82
Figura 24: Registrar Codificador de Ensayos.....	84
Figura 25: Registrar Codificador de Promotores.....	86

AVAL DEL CLIENTE

AVAL DEL CLIENTE

<Contenido del aval del cliente sobre la solución desarrollada>

INTRODUCCIÓN

INTRODUCCIÓN

A lo largo de la historia, siempre ha habido una necesidad de innovar y mejorar los mecanismos de gestión, con el objetivo de hacerlos más efectivos y eficientes. Con el transcurso del tiempo, se han desarrollado las Tecnologías de la Información y las Comunicaciones (TIC), que se definen como un conjunto de técnicas, avances y dispositivos que integran capacidades de almacenamiento, procesamiento y transmisión de datos. En la actualidad, las TIC generan cambios y transformaciones significativas en aspectos sociales, culturales y económicos, con un impacto directo en los sectores principales de la sociedad.

A pesar de las sanciones económicas, comerciales y financieras impuestas por Estados Unidos a Cuba durante más de medio siglo, la isla ha logrado avanzar en su proceso de informatización. Este progreso se ha basado en gran medida en los recursos humanos, materiales y financieros disponibles, y ha puesto un fuerte énfasis en el uso colectivo y social de las TIC. En este contexto, Cuba puede ser considerada como una sociedad en desarrollo en el ámbito de la información, ya que ha implementado estas tecnologías en diversos sectores, incluyendo la educación, la cultura y la salud.

En uno de estos sectores, en el campo de la salud, los notables avances tecnológicos han proporcionado servicios de mayor calidad que anteriormente eran impensables o inaccesibles. La incorporación de las TIC en este ámbito se ha convertido en un tema de gran interés, no solo para los profesionales de la salud y las instituciones, sino también para la sociedad en general. Se han logrado avances significativos en la aplicación de las TIC con el propósito de proporcionar al Sistema Nacional de Salud (SNS) información precisa, coherente y puntual para respaldar la toma de decisiones y mejorar los procesos médicos y asistenciales. Esto ha contribuido al aumento de la calidad y seguridad en la atención médica brindada a la población.

Diversas instituciones en Cuba han trabajado en el desarrollo de sistemas para informatizar el sector de la salud. Sin embargo, estas soluciones carecían de integración, lo que dificultaba su implementación generalizada. Por esta razón, a partir de 2003, el Ministerio de Salud Pública (MINSAP) inició el desarrollo de un sistema que pudiera integrar la información proveniente de las distintas aplicaciones de salud. Con el objetivo de lograr esta integración, se estableció la necesidad de que todas estas aplicaciones formaran parte de un programa general de informatización del Sistema Nacional de Salud (SNS). Como resultado, surgió la necesidad de desarrollar una plataforma

INTRODUCCIÓN

que garantizara de manera sencilla y eficiente el cumplimiento de estos objetivos, dando inicio así al desarrollo del Sistema de Información para la Salud (SISalud).

SISalud es una plataforma de aplicaciones con una interfaz de programación abierta que facilita la incorporación de nuevos módulos, registros y subsistemas compatibles entre sí, lo que posibilita la integración de sus componentes para la prestación de servicios de salud, que contempla: la Atención Primaria, Secundaria y Terciaria, el Sistema Integrado de Urgencia Médica, la Vigilancia en Salud, Docencia Médica, Telemedicina, Medicamentos y Fármacos, Epidemiología, entre otros.

La Empresa SOFTEL, encargada de llevar a cabo la creación de esta plataforma, organiza y dirige el proceso productivo en conjunto con la Universidad de Ciencias Informáticas y el Centro de Informática Médica (CESIM) para la creación de varios módulos y subsistemas que permitan optimizar y resolver determinados problemas en el ámbito de salud.

Situación problemática.

El sistema Xavia SIDEC, elaborado por CESIM, facilita la planificación y diseño de ensayos clínicos mediante la administración de estudios y los lugares donde se llevan a cabo, la gestión de formularios de recopilación de datos, la administración de grupos de participantes, la creación de un cronograma general y la definición de etapas y momentos de seguimiento del estudio. Toda esa información, se maneja en el entorno al ensayo clínico y no se enfoca directamente en el paciente vinculado a dicho ensayo. Por tal motivo, se requiere de un sistema que sea apto de operar con la información de los pacientes asociados a los ensayos clínicos y que sea capaz de contribuir al perfeccionar, calcular y almacenar esta información de forma estadística de los pacientes y aportar un significativo avance al proceso de informatización de nuestra sociedad.

SISalud, cuenta con un conjunto de registros en los que se trabaja con mucha información relacionada al SNS, y también posibilita la integración de otros sistemas de gestión de información con dichos registros, a razón de lo anterior, se le asigna a la empresa SOFTEL la realización de un sistema que gestione la información estadística de pacientes que se encuentran conectados con los ensayos clínicos de Xavia SIDEC y se anexe como subsistema a SISalud. Los datos de los pacientes deben de estar vinculados a varios registros con los que cuenta la plataforma en su módulo de Registros Básicos y Codificadores., de los que se identifican:

- Registro de Unidades de Salud.
- Registro de Fallecidos.

Estos registros dependen en gran medida, de la interacción que tiene SISalud con otros sistemas como el Sistema. Único de Identificación Nacional (SUIN) en los que se maneja la

INTRODUCCIÓN

información de ciudadanos y los relaciona con entidades de salud o Centros de salud que existen en toda la isla.

Problema a resolver.

¿Cómo gestionar la información concerniente a los pacientes participantes en los ensayos clínicos gestionados por Xavia SIDEC?

Objeto de estudio.

Proceso de gestión de la información en proyectos de gestión web.

Campo de acción.

La gestión de la información concerniente a los pacientes participantes en los ensayos clínicos.

Objetivo general.

Consiste en desarrollar un sistema que permita la gestión de la información de los participantes en los ensayos clínicos gestionados en la plataforma Xavia SIDEC.

Tareas de la investigación.

1. Elaboración del marco teórico de la investigación relacionado con el manejo y distribución de proyectos a todos los niveles de análisis y los conceptos asociados.
2. Análisis de las dificultades existentes en la gestión de la información concerniente a los pacientes participantes en los ensayos clínicos.
3. Análisis de sistemas de gestión de proyectos existentes.
4. Desarrollo del módulo de desarrollo del sistema de pacientes incluidos en ensayos clínicos.
5. Implementar el sistema de pacientes incluidos en ensayos clínicos.
6. Aplicación de la estrategia de validación a partir de métodos científico-técnicos, que demuestre la validez del resultado obtenido.
7. Realizar la Integración con los componentes del Sistema de Información para la Salud. (SISalud)

Métodos científicos seleccionados.

Comprender los pormenores del problema en cuestión y los objetivos de la solución a veces puede ser complicado. Por esta razón, es esencial al inicio de la investigación establecer los métodos científicos fundamentales que se aplicarán más adelante. Estos métodos posibilitan el estudio de características que no son directamente observables y de aspectos fenomenológicos.

Estos métodos representan una fase de la investigación en la que se toma información de la experiencia y se somete a un proceso de análisis racional. Existen diversos métodos que pueden ser

INTRODUCCIÓN

aplicados en investigaciones para obtener resultados de manera más eficaz y confiable. En el caso de esta investigación, se han definido los siguientes métodos:

Métodos teóricos:

- **Histórico – Lógico:** facilita la investigación de los fenómenos y eventos en una secuencia cronológica, lo que permite comprender la evolución y desarrollo de la esencia de la investigación en curso.

- **Analítico-Sintético:** será útil para condensar, expresar y detallar los requisitos funcionales de la solución propuesta. Esto se debe a que este enfoque nos permite investigar a fondo la esencia del problema en cuestión, sus características distintivas y sus rasgos distintivos. Además, facilita la identificación y extracción de los elementos más significativos relacionados con el objeto de estudio

- **Modelación:** se utilizó para representar el proceso de gestión del entorno general y de los grupos de participantes a través de diagramas y esquemas. Esto contribuye a una comprensión más profunda y clara de la operación del negocio.

Métodos empíricos:

Entrevista: empleado en los encuentros entre personal de SOFTEL y de CESIM para obtener la información necesaria que permita determinar las características, cualidades y requisitos con los que debe contar la propuesta de solución.

Observación: se utilizó para identificar buenas prácticas y vulnerabilidades de los portales similares para la visibilidad de servicios de SISalud.

Análisis documental: En el proceso de revisión bibliográfica, la exploración de fuentes de información, el análisis de documentos clasificados y la asimilación de elementos teóricos que respaldan la idea de la creación de un portal web son esenciales.

Estructura del Documento.

Este documento consta de tres capítulos que abordan todos los aspectos relacionados con la investigación. A continuación, se presenta una descripción de cada uno de los capítulos:

- **Capítulo 1.** Fundamentos y referentes teórico-metodológicos sobre los procesos de registro de pacientes incluidos en ensayos clínicos: En este capítulo, se lleva a cabo un análisis exhaustivo de la base teórica que sustenta el desarrollo de la solución propuesta. Se proporciona una descripción detallada de los principales conceptos relacionados con el entorno y se estudian sistemas homólogos. Se exploran las tendencias, técnicas, herramientas, metodologías y tecnologías utilizadas en la propuesta.

INTRODUCCIÓN

- **Capítulo 2.** Desarrollo del Sistema de Pacientes incluidos en ensayos clínicos: En este capítulo, se identifican y explican los conceptos relacionados con el dominio del problema y los procesos vinculados con el negocio, teniendo en cuenta la metodología seleccionada en el capítulo anterior. Se definen los requisitos funcionales y no funcionales, así como se presenta el modelo de datos.

- **Capítulo 3.** Validación del sistema de gestión de pacientes incluidos en ensayos clínicos: En este capítulo, se establece un estándar de codificación que servirá como guía para la implementación de la solución propuesta. También se describe la estrategia de pruebas que se aplicará para garantizar el correcto funcionamiento del sistema.

Además de los tres capítulos principales, el trabajo incluye Conclusiones, Recomendaciones, Referencias Bibliográficas y Anexos.

CAPÍTULO 1. FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LOS PROCESOS DE REGISTRO DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

CAPÍTULO 1. FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LOS PROCESOS DE REGISTRO DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

1.1. INTRODUCCIÓN DEL CAPÍTULO.

La informatización de la sociedad ha impulsado el desarrollo de diversas herramientas informáticas para optimizar los servicios de salud, por lo que es fundamental analizar distintos sistemas a nivel nacional e internacional que faciliten el diseño de un ensayo clínico y el dominio de los conceptos clave relacionados con este tema, para comprender mejor cómo se lleva a cabo este proceso en la actualidad. En el siguiente capítulo se realiza un estudio de temas y conceptos relacionados con la gestión de pacientes vinculados a ensayos clínicos, así como un breve análisis de diferentes sistemas homólogos que tienen un alto impacto dentro y fuera de nuestro país. Se describen, además, un análisis a los modelos de bases de datos, tecnologías, herramientas y metodologías utilizadas para dar solución al problema en cuestión de esta investigación.

1.2. CONCEPTOS RELACIONADOS CON EL PROCESO DE DESARROLLO DEL SISTEMA DE GESTIÓN DE PACIENTES INCLUIDOS EN ENSAYOS Clínicos

A continuación, se relacionan los principales conceptos analizados y estudiados en el presente trabajo.

1.2.1. ENSAYOS CLÍNICOS.

El ensayo clínico, experimento diseñado en cuidadosa observancia de la ética, en el que los sujetos se asignan a las diferentes intervenciones de manera aleatoria y simultánea (en un mismo período de tiempo), son supervisados también simultáneamente. Este tipo de estudio, que permite la evaluación de la eficacia y seguridad de un fármaco o producto en sujetos humanos, se fundamenta en: comparaciones con el patrón terapéutico o diagnóstico; la existencia de al menos dos grupos concurrentes de pacientes a comparar entre sí. (López et al. 2011)

CAPÍTULO 1. FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LOS PROCESOS DE REGISTRO DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

1.2.2. GESTIÓN DE ENSAYOS CLÍNICOS.

La gestión de ensayos clínicos es el proceso de planificar, ejecutar, monitorear y reportar los estudios que evalúan la seguridad y eficacia de nuevos tratamientos o intervenciones médicas. La gestión de ensayos clínicos implica el uso de sistemas de software, protocolos, diseños y estándares de calidad para asegurar que los ensayos se realicen de acuerdo con las buenas prácticas clínicas y los requisitos regulatorios. Algunas de las funciones y actividades de la gestión de ensayos clínicos son:

- Seleccionar y capacitar al personal del estudio, incluyendo a los investigadores, coordinadores, monitores y auditores.
- Identificar y reclutar a los participantes del estudio, asegurando que cumplan con los criterios de inclusión y exclusión.
- Supervisar la administración de los tratamientos o intervenciones, así como la recolección y el manejo de los datos clínicos.
- Verificar la calidad y la integridad de los datos clínicos, resolviendo las discrepancias y realizando el análisis estadístico.
- Preparar y presentar los informes finales del estudio, así como las solicitudes de autorización o registro de los productos o dispositivos evaluados.

La gestión de ensayos clínicos es una disciplina compleja y multidisciplinaria que requiere de conocimientos científicos, médicos, éticos, legales y administrativos. La gestión de ensayos clínicos tiene como objetivo garantizar que los ensayos se realicen con rigor científico, protegiendo los derechos y el bienestar de los participantes y generando evidencia válida y confiable para apoyar las decisiones clínicas y regulatorias. (*Ensayos clínicos - Cómo funcionan los ensayos clínicos | NHLBI, NIH sin fecha*)

1.2.3. SISALUD.

El Sistema de Información para la Salud (SISalud), forma parte de la Informatización del Sistema Nacional de Salud. Este sistema propone que todas las aplicaciones informáticas para la salud estén integradas; abarca: Administración del Sistema, Registros Básicos y Codificadores, Sistemas de Atención Médica para los niveles de Atención Primaria, Secundaria y Terciaria, Ayuda a

CAPÍTULO 1. FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LOS PROCESOS DE REGISTRO DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

la decisión y Otros Sistemas, mostrados en el Anexo 2. Como Registros Básicos y Codificadores se incluye el Registro Informatizado de Salud (RIS) que contiene los nomencladores nacionales que brindarán información para el funcionamiento del resto de las aplicaciones. Se puede citar para gestionar los recursos: Registro de Unidades de Salud (RUS), Registro Personal de la Salud (RPS), Registro del Ciudadano (RC), Registro de Equipos Médicos y No Médicos (REQ) y en los que gestionan los servicios: Registro de Servicios Médicos (RSM). Con estos registros desarrollados e implementados se garantiza por primera vez la gestión en tiempo real y con alcance nacional de la información referida a los principales recursos y servicios de la salud, así como la obtención de un grupo de estadísticas para ser analizadas a cualquier instancia de dirección del Servicio nacional de salud (SNS) para la toma de decisiones. Son capaces de interactuar entre ellos y de esta forma reutilizar la información gestionada por cada uno, y por el resto de las aplicaciones que conforman SiSalud

Ventajas que reporta la implementación de SiSalud:

- Es portable en diferentes sistemas operativos, tanto en los servidores como en los clientes y replicable en otros entornos.
- Es una plataforma en constante desarrollo, que crece en la medida en que se implementan nuevos módulos.
- Comparada con otras soluciones, es menos costosa, ya que la cantidad de servidores involucrados se reduce considerablemente.
- Es una solución fácil de mantener, actualizar y soportar técnicamente, ya que la aplicación es centralizada.
- Permite la integración con todos los procesos relacionados con la informatización de la sociedad cubana.
- Facilita a todos los niveles de salud la información estandarizada y homogénea.
- Sirve de guía, consulta y aprendizaje para la actividad pre-profesional y post-profesional.
- Permite contar con los nomencladores nacionales para conocer las definiciones centrales del MINSAP en cuanto a las disposiciones que rigen el funcionamiento del resto de las aplicaciones.
- Acelera y mejora las posibilidades de estudios de tipo comparativo, investigativos y estadísticos. (*Softel | Soluciones Informáticas sin fecha*)

CAPÍTULO 1. FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LOS PROCESOS DE REGISTRO DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

1.2.4. XAVIA SIDEC.

El Sistema para el manejo de datos de ensayos clínicos Xavia SIDEC fue diseñado con el objetivo de contribuir a mejorar el diseño y conducción de los ensayos clínicos que se realizan en el Centro de Inmunología Molecular (CIM) de Cuba y sitios clínicos asociados.

En Xavia SIDEC se diseñan el 100% de los ensayos clínicos de este centro. Esto es posible a partir de la gestión de los estudios y centros donde se realizan los ensayos, gestión de hojas del Cuaderno de Recogida de datos (CRD), gestión de grupos de sujetos, cronograma general, etapas y momentos de seguimiento del estudio. Permite además definir reglas para la validación de los campos de las hojas CRD.

El sistema ha sido desplegado en todas las provincias del país, alcanzando no solo el nivel especializado de hospitales, sino que está siendo utilizado en unidades de atención primaria de salud. El sistema XAVIA SIDEC propicia a los especialistas del CIM, coordinadores e investigadores, realizar la captura electrónica de datos, eliminando la doble entrada de los mismos durante la conducción de los ensayos clínicos y que a su vez permita la transmisión remota de datos entre la red nacional de hospitales y el CIM.

El desarrollo de XAVIA SIDEC para Cuba propicia:

- Generar ahorros significativos, posibilitando la adquisición de un software desarrollado en el país, garantizando a su vez soporte o mantenimiento sin incurrir en grandes gastos por concepto de licencias o servicios.
- Cumplir los estándares internacionales de calidad, seguridad y buenas prácticas clínicas según GCP (Good Clinical Practices), HIPAA (Health Insurance Portability and Accountability Act).
- Aumentar la eficiencia en la conducción de los ensayos clínicos en el CIM y demás centros del Polo Científico, como empresas biotecnológicas, con el consecuente impacto en la aceleración del desarrollo de varios productos y por ende el aumento de la competitividad de los mismos en el mercado internacional.
- Facilitar y agilizar el proceso de diseño y aprobación de los Cuadernos de Recogida de Datos.
- Entre otros.

Xavia SIDEC 3.0 ya se está implantando en el Instituto Finlay de Vacunas y el Centro de Ingeniería Genética y Biotecnología, con los ensayos clínicos de mayor impacto e importancia. (

CAPÍTULO 1. FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LOS PROCESOS DE REGISTRO DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

SIDEC(SIDEC 3.0 | Universidad de las Ciencias Informáticas sin fecha)

1.2.5. SISTEMA ÚNICO DE IDENTIDAD NACIONAL (SUIN).

El Sistema Único de Identificación Nacional de la República de Cuba fue desarrollado con un enfoque a procesos. Integra tecnologías para la verificación biométrica, aspecto clave para la identificación de las personas. Está diseñado bajo el precepto de la creación de una base de datos nacional, poblada a partir de la información vigente y de la información resultante de los distintos trámites que se realizan en las oficinas de identificación y registros. Cuenta con un enfoque de mejora y depuración continua, que garantizará el manejo de una identidad única para cada persona natural o extranjera residente en el país. Este sistema permite la gestión de los diferentes procesos requeridos para la ejecución segura y eficiente de trámites de identificación. Incluye la captura de los datos alfanuméricos, documentales y biométricos, la validación de la identidad por comparación biométrica, y la obtención de reportes para este nivel. Se responsabiliza de la interacción con AFIS para la validación a partir de rasgos biométricos, así como de la comunicación con sistemas o servicios externos que se requieran para el correcto funcionamiento de los procesos. Este sistema está desplegado desde el 2009 en la República de Cuba. El mismo no incluye el proceso de tramitación de pasaportes corrientes de los ciudadanos cubanos que residen en el exterior. (Cortiñas Martínez 2017)

1.2.6. GESTIÓN DE PACIENTES:

La gestión de pacientes se refiere a la estrategia que permite sistematizar y ordenar los procesos de atención sanitaria de forma adecuada y eficiente, sustentados en la mejor evidencia científica del momento y con la participación de profesionales en la gestión para la toma de decisiones en torno al paciente. *(La gestión clínica: ¿qué es y para qué sirve? | Conexión ESAN sin fecha)*

Esta gestión se realiza mediante un software hospitalario específico, que agiliza todos los trámites de ingresos, altas y asignación de camas a los pacientes.

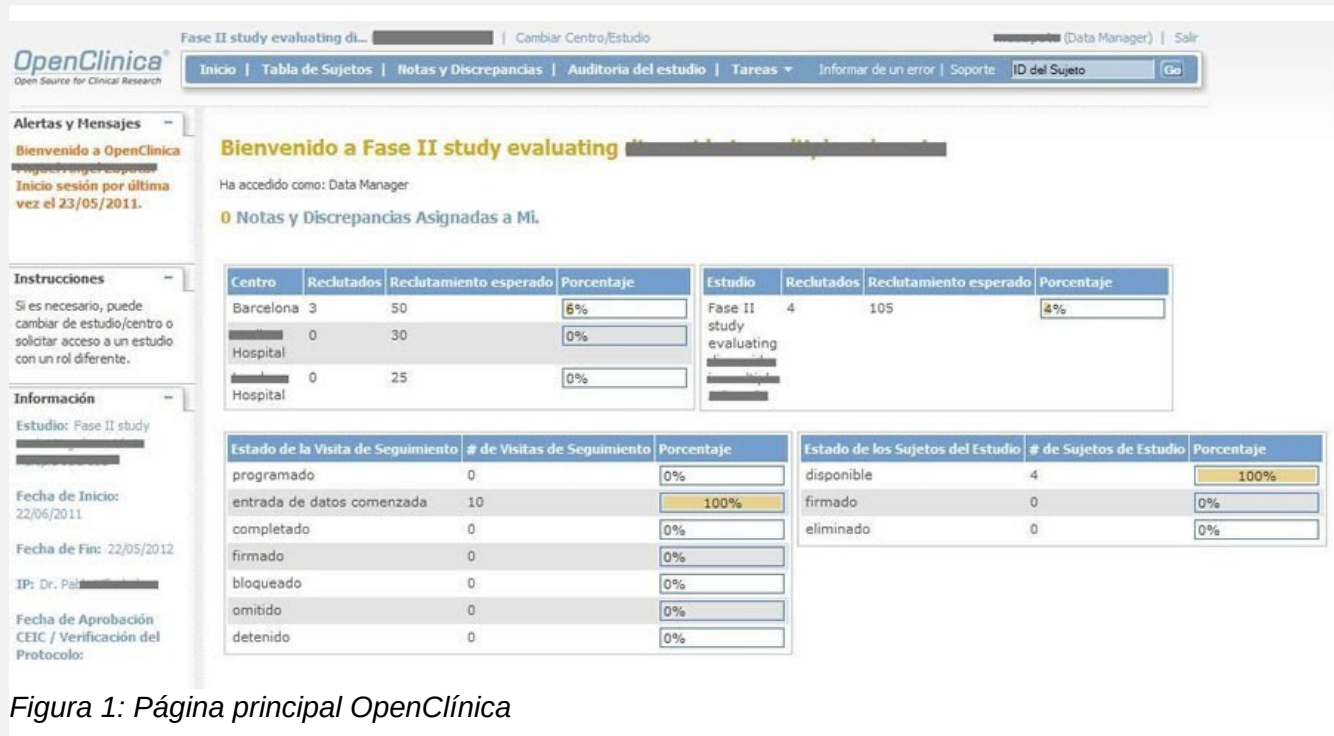
La gestión de pacientes es fundamental para el funcionamiento y la logística de un centro sanitario u hospital. Implica un cambio de actitud no solo del personal administrativo, sino de todo el personal asistencial, que deberá asumir el liderazgo de las tareas administrativas, garantizando el logro de los objetivos de la institución. *(La gestión de pacientes: el servicio de admisión y documentación sin fecha)*

1.3. ANÁLISIS DE SOLUCIONES SIMILARES.

A continuación, se presenta un análisis de los sistemas de salud más utilizados en el mundo con el objetivo de conocer las tendencias actuales y tecnologías empleadas en el desarrollo de estos.

CAPÍTULO 1. FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LOS PROCESOS DE REGISTRO DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

1.3.1. OPENCLÍNICA.



OpenClinica es un sistema diseñado bajo la filosofía de software libre para el manejo de estudios clínicos. Puede utilizarse en hospitales o universidades para investigaciones médicas. Es diseñada sobre una aplicación web para la gestión de protocolos de investigaciones clínicas (GesPIC), diseños de cuadernos de recogidas de datos (CRD's) electrónicos, captura, almacenamiento y gestión electrónica de datos. Además, es escalable, modular y basado en estándares, incluye diferentes niveles de acceso a lectura, creación y modificación de los datos. (Openclinica from Akaza Research. sin fecha)

Algunas de sus características son:

- Licencia de código abierto LGPL (sin costes por licencia).
- Sistema Operativo: Linux, Windows.
- Basado en estándares abiertos e independientes (CDISC ODM, HL7).
- Robusto y modular diseñada con Java/J2EE.
- Base de Datos Independiente (Postgres (por defecto), Oracle, MySQL, SQL Server, etc.)

CAPÍTULO 1. FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LOS PROCESOS DE REGISTRO DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

- Arquitectura Orientada a Servicios (SOA).

Soportes que brinda:

- Consultas por Email.
- Gestión de BBDD personalizada.
- Gestión de configuración.
- Procedimientos de copia de seguridad.
- Integración de sistemas.
- Soporte online con base a servicios.
- Asistencia en conocimiento y requerimientos.
- Implementación y adaptación.

1.3.2. ALASCLÍNICAS



Figura 2: AlasClínicas.

Es un sistema desarrollado en la Universidad de las Ciencias Informáticas en Cuba, abarca las funcionalidades de OpenClínica y además brinda otras como: la realización de un Cronograma de

CAPÍTULO 1. FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LOS PROCESOS DE REGISTRO DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

ejecución general para los ensayos y la generación del cronograma específico para cada paciente, la validación de las variables de los ensayos clínicos, permitiendo la disminución y detección de errores en los datos, el reporte de las trazas de las acciones realizadas en el sistema; así como la restricciones de las direcciones IP desde las cuales se puede acceder al mismo, entre otras. (García et al. 2012)

Aportes del sistema:

- Las principales agencias reguladoras internacionales ya han adecuado sus normativas al modo de conducción de Ensayos Clínicos en forma electrónica. Gran parte de estos procesos en el mundo ya están informatizados, lo cual, a cualquier centro que lo utilice, le garantiza:
- Reducir los costos en cuanto a transportación y material de oficina, ya que se ocupará menos espacio físico al eliminar los CRD impresos.
- Respuesta rápida a problemas que en el modo tradicional tardan semanas y meses en conocerse y por consecuencia más tiempo en solucionarse.
- Facilitar la comunicación entre todos los centros que participen en la conducción de los ensayos.
- Chequear y optimizar el proceso de monitoreo que se realiza en las diferentes provincias y que no resulta factible de chequear físicamente.
- Estandarizar la información que se maneja, restringiendo a los especialistas al uso de los formatos establecidos por los entes responsables. (Rodríguez García et al. 2012)

Sistemas de este tipo son sumamente costosos y realizados por empresas bajo licencias no libres, imposibilitando a países del tercer mundo su adquisición. Como muestra de ello, se estima que el monto inicial de un software de este tipo puede llegar a tener un valor de 800 000 dólares, adicionándole a esta suma los pagos de las licencias anuales con un costo de 100 000 dólares cada una, los servicios de mantenimiento, soporte y capacitación del personal también deben ser pagados, como resultado final, el país que lo solicita tiene que pagar cifras que superan los 800 000 dólares.

La información manejada, en la mayoría de los casos, se encuentra en los servidores de las empresas que ofertan el servicio, es decir, las empresas solicitantes no tienen el control de los datos manejados, lo que no garantiza la confiabilidad de los mismos.

CAPÍTULO 1. FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LOS PROCESOS DE REGISTRO DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

Cuba, por ser un país bloqueado, posee aún más restricciones. Además, muchos de estos sistemas no están en correspondencia a los requisitos identificados en los diferentes centros de salud cubano y no ocupan todas las etapas de los ensayos clínicos.

1.3.3. MEDIDATA RAVE:

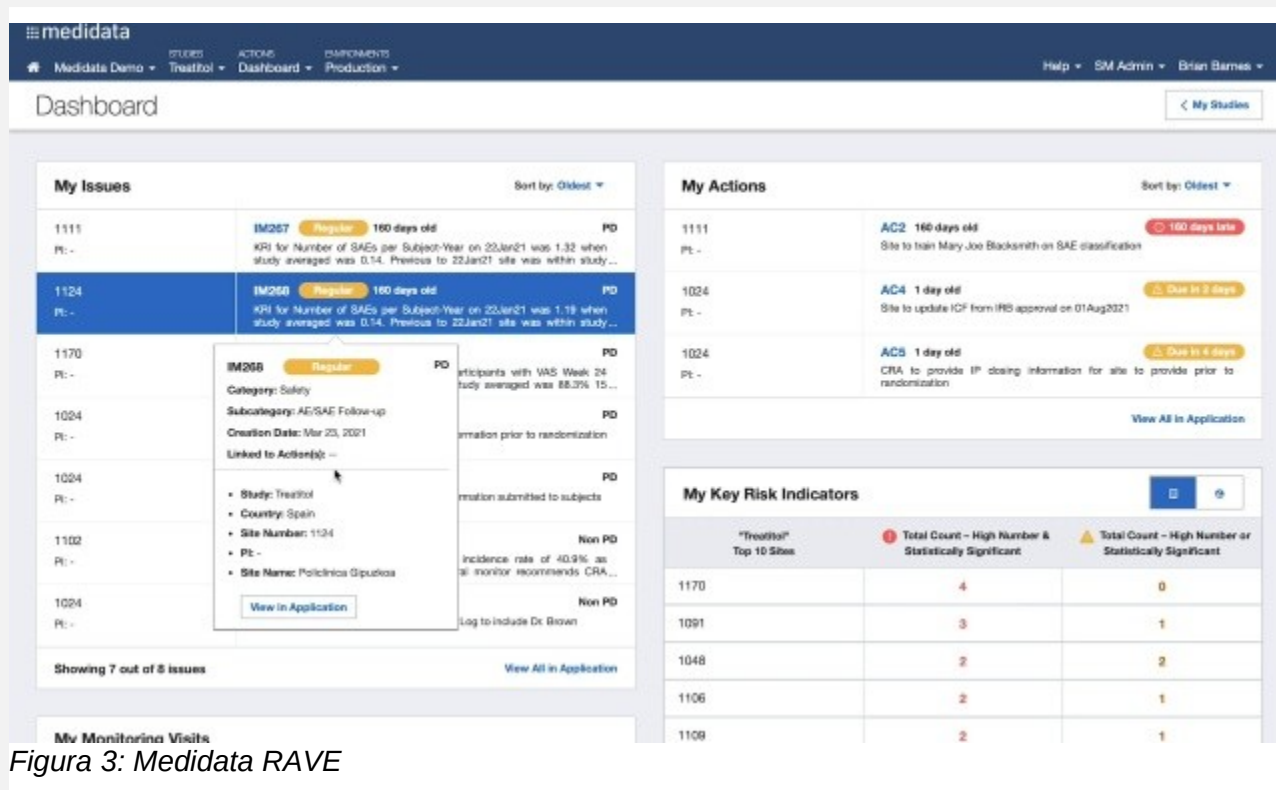


Figura 3: Medidata RAVE

Este sistema es una solución de la Compañía Medidata Solutions (proveedor mundial de soluciones de gestión de datos clínicos en formato electrónico con clientes en 40 países), destinada a mejorar la eficacia de los ensayos, racionalizando el proceso de recogida, verificación, consolidación y análisis de datos de investigaciones clínicas. El sistema ha sido evaluado por IBM Global Services y actualmente es usado por Bayer Healthcare AG. Esta última es una filial de Bayer AG, empresa alemana, líder mundial en el ámbito de la salud y la fabricación de medicamentos y productos médicos, para el diagnóstico, la prevención y el tratamiento de enfermedades.

Desventajas:

CAPÍTULO 1. FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LOS PROCESOS DE REGISTRO DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

Sistemas como Medidata RAVE además de otros como Hipócrates y Oracle ClinTrial, son sistemas propietarios a través de los cuales las compañías que los promueven solo ofrecen sus servicios online. Esto se refiere a que el cliente deberá pagar un alto precio por el servicio de conducir un ensayo en el sistema y en ocasiones el precio podría variar en dependencia de la cantidad de pacientes que serán incluidos. Así mismo, la capacitación del personal también sería otro de los gastos en los que habría que incurrir. Además del gran costo monetario que significa el uso de los servicios, la información generada en los estudios, que por su significado deberá ser de gran fiabilidad e integridad, se encontrará en servidores externos a la institución que gestiona los ensayos; lo cual no es conveniente teniendo en cuenta que los datos que se manejan podrían ser vulnerados, y estos constituyen la base de los EC y de la producción de fármacos en el país. (Medidata | Unified Life Science Platform | Medidata Solutions sin fecha)

1.3.4. RESULTADOS DEL ANALISIS DE SOLUCIONES SIMILARES.

A continuación, se muestra un resumen en forma de tabla teniendo en cuenta los siguientes parámetros:

- **Procedencia:** referente a la procedencia del producto. Puede tomar valores Nacional o Extranjero.
- **Gestión de paciente:** referente al si el tipo de gestión que realiza el sistema analizado se relaciona a información de los pacientes incluidos en los ensayos clínicos.
- **Libre de costo:** referente a si el sistema es gratis o no.
- **Código abierto:** referente al si el sistema se puede editar libremente para añadirle nuevas funcionalidades.
- **Aplicación Web:** referente a si el sistema cuenta con una aplicación web.

CAPÍTULO 1. FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LOS PROCESOS DE REGISTRO DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

Sistema	Procedencia	Gestión de paciente	Libre de costo	Código abierto	Aplicación web
OpenClínica	Extranjera	No	No	Si	Si
AlasClínicas	Nacional	No	No	Si	Si
MediData Rave	Extranjera	SI	No	No	Si

Tabla 1: Tabla resumen.

Al realizar un estudio en el ámbito nacional e internacional de sistemas que manejan datos vinculados a pacientes incluidos en ensayos clínicos se concluye que: los sistemas a nivel internacional integran las tecnologías más avanzadas y utilizan disímiles funcionalidades que dan respuesta a complejas exigencias, pero no cumplen con los requerimientos necesarios para gestionar la información de los pacientes.

1.4. HERRAMIENTAS Y TECNOLOGÍAS.

Para construir la presentación online de la empresa que se adapte a las necesidades tanto de la organización como de los clientes es necesario dedicar grandes cantidades de tiempo y esfuerzo. Se trabaja con herramientas y tecnologías que sean de gran ayuda al desarrollador. Para el desarrollo de este portal web se ha decidido optar por las siguientes tecnologías.

Microsoft Visual Studio Code:

Es un editor de texto desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para depuración, control de Git integrado, resaltado de sintaxis, finalización de código inteligente, fragmentos de código y refactorización de código. También es personalizable, de modo que los usuarios pueden cambiar el tema del editor, los métodos abreviados de teclado y las preferencias. Es gratuito y de código abierto. (*Visual Studio Code - Code Editing. Redefined* sin fecha)

Ventajas

CAPÍTULO 1. FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LOS PROCESOS DE REGISTRO DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

- Visual Studio Code es una herramienta que tiene soporte nativo para gran variedad de lenguajes, entre ellos podemos destacar los principales del desarrollo Web: HTML, CSS, y JavaScript, entre otros.
- Posibilidad de configurar la interfaz a nuestro gusto. De esta forma, podremos tener más de un código visible al mismo tiempo, las carpetas de nuestro proyecto y también acceso a la terminal o un detalle de problemas, entre otras posibilidades.
- Existencia de una amplia gama de temas o estilos visuales para Visual Studio Code, que hacen el trabajo con el software más agradable a la vista.
- Goza de un soporte técnico formidable, pues debido a su frecuente uso por la comunidad de desarrolladores, se puede encontrar fácilmente documentación y ayuda en foros y sitios relacionados.

Herramienta para el modelado de la solución.

Una herramienta de ingeniería de software asistida por computadora es una herramienta individual para ayudar al desarrollador de *software* o administrador de proyecto durante una o más fases del desarrollo de software o mantenimiento. En el modelado de la propuesta de solución se empleó Visual Paradigm v15.1.2 para UML y Axure RP para el diseño visual de la aplicación.

Axure RP

Axure RP pone el poder del prototipado en manos de gerentes de productos, analistas de negocios y profesionales de la experiencia de usuario. Antes de escribir una sola línea de código, las soluciones pueden ser prototipadas y validadas por las personas que mejor entienden su negocio, productos y clientes. La principal utilidad es el diseño de *sitemaps*, *wireframes* y prototipos básicos o avanzados para proyectos de diseño de experiencia de usuario (UX por sus siglas en inglés).

Entre las principales ventajas de Axure RP destaca, que se trata de una herramienta muy completa y a la vez, fácil de utilizar. En función de los conocimientos y de la propia experiencia del diseñador, éste podrá ir incorporando más y más funcionalidades que simularán todas las interacciones propuestas en su proyecto.

Esto lleva directamente a otra de sus enormes ventajas, la interactividad, pues al tratarse de una herramienta de prototipado, va a simular el comportamiento de las interfaces de usuario. Y como

CAPÍTULO 1. FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LOS PROCESOS DE REGISTRO DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

Axure RP es HTML, los prototipos que se diseñan con ella funcionan en cualquier navegador web (Safari, Mozilla Firefox, Chrome, Opera...). (*Axure, poderosa herramienta en el Diseño UX* sin fecha)

Visual Paradigm

Visual Paradigm es una herramienta CASE: Ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.

Visual Paradigm ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta privada disponible en varias ediciones, cada una destinada a satisfacer diferentes necesidades: *Enterprise, Professional, Community, Standard, Modeler* y *Personal*. Existe una alternativa libre y gratuita de este *software*. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos.

Esta herramienta permite aumentar la calidad del software, a través de la mejora en el desarrollo y mantenimiento del mismo, de igual forma potencia la reutilización del software y estandarización de la documentación, además del uso de las distintas metodologías propias de la Ingeniería del Software. (*Herramienta ideal de modelado y diagramación para una colaboración ágil en equipo* sin fecha)

Lenguaje para el modelado UML.

Por sus siglas en inglés UML, *Unified Modeling Language*: es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (*Object Management Group*). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y

CAPÍTULO 1. FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LOS PROCESOS DE REGISTRO DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

componentes de software reutilizables. UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas. (*¿Qué es UML* sin fecha)

Lenguaje de programación.

En informática, cualquier lenguaje artificial que puede utilizarse para definir una secuencia de instrucciones para su procesamiento por un ordenador o computadora. Es complicado definir qué es y qué no es un lenguaje de programación. Se asume generalmente que la traducción de las instrucciones a un código que comprende la computadora debe ser completamente sistemática. Normalmente, es la computadora la que realiza la traducción.

JavaScript.

Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS).

Características:

- Es simple, no hace falta tener conocimientos avanzados de programación para aprender a manejar JavaScript y es recomendado por muchos expertos a la hora de encontrar un lenguaje para comenzar a programar.
- Maneja objetos dentro de la página Web y sobre ese objeto se puede definir diferentes eventos. Dichos objetos facilitan la programación de páginas interactivas, a la vez que se evita la posibilidad de ejecutar comandos que puedan ser peligrosos para la máquina del usuario, tales como formateo de unidades, modificar archivos, etc.
- Es dinámico, responde a eventos en tiempo real. Eventos como presionar un botón, pasar el puntero del *mouse* sobre un determinado texto o el simple hecho de cargar la página o caducar un tiempo. Con esto podemos cambiar totalmente el aspecto de nuestra página al gusto del usuario, evitándonos tener en el servidor una página para cada gusto, hacer cálculos en base a variables cuyo valor es determinado por el usuario, etc.

CAPÍTULO 1. FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LOS PROCESOS DE REGISTRO DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

- Existen tecnologías utilizadas en varios campos basadas en JavaScript, algunos ejemplos son Node, Vue o React, este último creado por Facebook. (*JavaScript* | MDN 2023)

HTML.

Lenguaje de Marcado de Hipertexto o HyperText Markup Language (HTML por sus siglas en inglés) es el elemento de construcción más básico de una página web y se utiliza para crear y representar visualmente elementos en la web. Determina el contenido de la página web, pero no su funcionamiento. HTML5 se refiere a la quinta versión del lenguaje y propone nuevos elementos y comportamientos. Contiene un conjunto más amplio de tecnologías que permiten a los sitios web y a las aplicaciones ser más diversas y de gran alcance. Esta versión permite describir con mayor precisión cuál es el contenido y establece conexiones al servidor de manera novedosa. Su uso proporciona una mayor optimización de la velocidad y un mejor uso del hardware. (MDN web docs, 2019).(*HTML: Lenguaje de etiquetas de hipertexto* | MDN 2023)

CSS 3.

Lenguaje de Hojas de Estilos en Cascada o Cascading Style Sheets (CSS) es un lenguaje de hojas de estilo utilizado para describir la presentación de un documento escrito en un lenguaje de marcado. Aunque se utiliza con mayor frecuencia para establecer el estilo visual de las páginas web y las interfaces de usuario escritas en HTML y XHTML. Es aplicable a la representación en voz o en otros medios de comunicación. Junto con HTML y JavaScript, CSS es una tecnología fundamental utilizada por la mayoría de los sitios web para crear páginas web visualmente atractivas, interfaces de usuario para aplicaciones web e interfaces de usuario para muchas aplicaciones móviles. CSS está diseñado principalmente para permitir la separación del contenido del documento de la presentación del documento, incluidos aspectos como el diseño, los colores y las fuentes. Esta separación puede mejorar la accesibilidad al contenido, proporcionar más flexibilidad y control en la especificación de las características de presentación, permitir que múltiples páginas HTML compartan el formato al especificar el CSS relevante en un archivo .CSS separado y reducir la complejidad y la repetición en el contenido estructural. (CSS | MDN 2023)

TypeScript:

CAPÍTULO 1. FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LOS PROCESOS DE REGISTRO DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

TypeScript es un lenguaje para JavaScript a escala de aplicación. TypeScript agrega tipos opcionales a JavaScript que admiten herramientas para aplicaciones JavaScript a gran escala para cualquier navegador, cualquier host y cualquier sistema operativo. TypeScript se compila en JavaScript legible y basado en estándares. (*Mecanografiado 2023*)

Docker:

Docker implementa una API de alto nivel para proporcionar contenedores livianos que ejecutan procesos de manera aislada.

Construido sobre las facilidades proporcionadas por el kernel Linux (principalmente *cgroups* y *namespaces*), un contenedor Docker, a diferencia de una máquina virtual, no requiere incluir un sistema operativo independiente. En su lugar, se basa en las funcionalidades del kernel y utiliza el aislamiento de recursos (CPU, la memoria, el bloque E/S, red, etc.) y *namespaces* separados para aislar la vista de una aplicación del sistema operativo. Docker accede a la virtualización del kernel Linux, ya sea directamente a través de la biblioteca *libcontainer* (disponible desde Docker 0.9), o indirectamente a través de *libvirt*, *LXC* o *systemd-nspawn*.

Mediante el uso de contenedores, los recursos pueden ser aislados, los servicios restringidos, y se otorga a los procesos la capacidad de tener una visión casi completamente privada del sistema operativo con su propio identificador de espacio de proceso, la estructura del sistema de archivos, y las interfaces de red. Contenedores múltiples comparten el mismo núcleo, pero cada contenedor puede ser restringido a utilizar solo una cantidad definida de recursos como CPU, memoria y E / S.

Usar Docker para crear y gestionar contenedores puede simplificar la creación de sistemas altamente distribuidos, permitiendo que múltiples aplicaciones, las tareas de los trabajadores y otros procesos funcionen de forma autónoma en una única máquina física o en varias máquinas virtuales. Esto permite que el despliegue de nodos se realice a medida que se dispone de recursos o cuando se necesiten más nodos, lo que permite una plataforma como servicio (PaaS - Platform as a Service) de estilo de despliegue y ampliación de los sistemas como Apache Cassandra, MongoDB o Riak. Docker también simplifica la creación y el funcionamiento de las tareas de carga de trabajo o las colas y otros sistemas distribuidos. (*Docker: desarrollo acelerado de aplicaciones de contenedores 2022*)

Marco de trabajo. (Framework)

CAPÍTULO 1. FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LOS PROCESOS DE REGISTRO DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

Un marco de trabajo no es más que una estructura de soporte definida mediante la cual otro proyecto de software puede ser desarrollado y organizado. Puede incluir soporte de programas, biblioteca y un lenguaje interpretado entre otros softwares para ayudar desarrollar y unir los diferentes componentes de un proyecto.

Nestjs

Nest es un marco para crear aplicaciones del lado del servidor Node.js escalables y eficientes . Utiliza JavaScript moderno, está construido con TypeScript (conserva la compatibilidad con JavaScript puro) y combina elementos de OOP (Programación orientada a objetos), FP (Programación funcional) y FRP (Programación reactiva funcional).

En el fondo, Nest utiliza Express , pero también proporciona compatibilidad con una amplia gama de otras bibliotecas, como por ejemplo Fastify , lo que permite un uso sencillo de los innumerables complementos de terceros que están disponibles. (*GitHub - nestjs/nest: un marco progresivo de Node.js para crear aplicaciones de servidor eficientes, escalables y de nivel empresarial con TypeScript/JavaScript*  sin fecha)

Node.JS.

Node.js es un entorno de desarrollo en la capa del servidor basado en el estándar ECMAScript y en el motor V8 de Google Chrome, con comunicación de datos en una arquitectura orientada a eventos. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como, por ejemplo, servidores web.

Incorpora varios módulos compilados en el propio binario, como por ejemplo el módulo de red, que proporciona una capa para programación de red asíncrona, y otros módulos fundamentales, como por ejemplo *Path*, *FileSystem*, *Buffer*, *Timers* y el de propósito más general *Stream*.

Funciona con un modelo de evaluación de un único hilo de ejecución, usando entradas y salidas asíncronas las cuales pueden ejecutarse concurrente en un número de hasta cientos de miles sin incurrir en costos asociados al cambio de contexto. Este diseño de compartir un único hilo de ejecución entre todas las solicitudes atiende a necesidades de aplicaciones altamente concurrentes.

CAPÍTULO 1. FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LOS PROCESOS DE REGISTRO DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

Node.js puede ser combinado con una base de datos documental, como MongoDB o MariaDB y JSON lo que permite desarrollar en un entorno de desarrollo JavaScript unificado. Con la adaptación de los patrones para desarrollo del lado del servidor tales como MVC (Modelo vista controlador) y sus variantes MVP (Modelo vista plantilla), etc. Node.js facilita la reutilización de código del mismo modelo de interfaz entre el lado del cliente y el lado del servidor. Suele ser muy utilizado en aplicaciones de red, por ejemplo, servidores web, streams, sockets, etc; debido a su alta concurrencia. (Acerca sin fecha)

Angular JS

Angular JS es un marco de JavaScript MVC desarrollado por Google, el cual permite construir aplicaciones front-end bien estructuradas y fáciles de comprobar y mantener.

El marco consiste en un conjunto de herramientas bien integradas que ayudarán a construir aplicaciones del lado del cliente, bien estructuradas en un sistema modular, con menos código y más flexibilidad.

Angular JS extiende HTML, proporcionando directrices que añaden funcionalidad al margen de beneficio y permite crear plantillas dinámicas poderosas. También puede crear directrices personalizadas, elaborando componentes reutilizables que completan las necesidades y abstrayendo toda la lógica de manipulación del DOM.

También implementa vinculación de datos de dos vías, conectando el HTML (vistas) a los objetos de JavaScript (modelos) sin problemas. En términos simples, esto significa que cualquier actualización del modelo se reflejará inmediatamente en la vista, sin necesidad de ningún tipo de manipulación del modelo de objetos del documento (por sus siglas en ingles DOM) o el control de eventos (por ejemplo, con jQuery).

Angular presta servicios en la parte superior de XHR (O XMLHTTP *Extensible Markup Lenguaje/Hypertext Transfer Protocol* es una interfaz para realizar peticiones web) que simplifican considerablemente el código y permite abstraer llamadas API en servicios reutilizables. Con esto, puedes mover tu modelo y lógica de negocio para el front-end y construir aplicaciones web back-end independientes (agnósticas).

CAPÍTULO 1. FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LOS PROCESOS DE REGISTRO DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

Por último, Angular permite mucha flexibilidad en cuanto a la comunicación con el servidor. Como la mayoría de los marcos de JavaScript MVC, Angular permite trabajar con cualquier tecnología de servidor, siempre que puede servir a la aplicación a través de una API REST Web. (*AngularJS: marco MVW de JavaScript superheroico* sin fecha)

PostgreSQL

PostgreSQL viene con muchas funciones destinadas a ayudar a los desarrolladores a crear aplicaciones, a los administradores a proteger la integridad de los datos y crear entornos tolerantes a fallas, y ayudarlo a administrar sus datos sin importar cuán grande o pequeño sea el conjunto de datos. Además de ser gratuito y de código abierto, PostgreSQL es altamente extensible. Por ejemplo, puede definir sus propios tipos de datos, crear funciones personalizadas e incluso escribir código desde diferentes lenguajes de programación sin tener que volver a compilar su base de datos.

PostgreSQL intenta cumplir con el estándar SQL donde dicha conformidad no contradice las características tradicionales o podría conducir a malas decisiones arquitectónicas. Muchas de las características requeridas por el estándar SQL son compatibles, aunque a veces con una sintaxis o función ligeramente diferente. Se pueden esperar nuevos avances hacia la conformidad con el tiempo. A partir del lanzamiento de la versión 16 en septiembre de 2023, PostgreSQL cumple con al menos 170 de las 179 características obligatorias para la conformidad con SQL:2023 Core.

Además, PostgreSQL es altamente extensible: muchas funciones, como los índices, tienen API definidas para que pueda desarrollarlas con PostgreSQL para resolver sus desafíos.

Se ha demostrado que PostgreSQL es altamente escalable tanto en la gran cantidad de datos que puede administrar como en la cantidad de usuarios simultáneos que puede acomodar. Hay clústeres de PostgreSQL activos en entornos de producción que administran muchos terabytes de datos y sistemas especializados que administran petabytes. (*PostgreSQL: Acerca de* sin fecha)

1.4.1. METODOLOGÍA DE DESARROLLO DE SOFTWARE.

Crear un sistema informático que funcione bien y que satisfaga las necesidades planteadas es un trabajo muy duro y complejo. Las metodologías para el desarrollo del software establecen un proceso ordenado para crear software con el fin de hacerlo más fiable y eficaz. Una metodología de desarrollo de software tiene como principal finalidad mejorar la calidad del software que se genera en

CAPÍTULO 1. FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LOS PROCESOS DE REGISTRO DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

todas y cada una de sus etapas de desarrollo. No hay una metodología de software que sirva para todo, ya que cada metodología debe ajustarse a las particularidades de cada proyecto (equipo de desarrollo, recursos, etc.) requiriéndose así que el proceso sea flexible. Las metodologías de desarrollo se pueden clasificar en dos grupos según sus características y los propósitos que buscan: ágiles y robustas. (*AngularJS: marco MVW de JavaScript superheróico* sin fecha)

AUP-UCI.

El Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Este explica de una manera sencilla y fácil de comprender la forma de crear aplicaciones de software de negocio con el uso de técnicas ágiles y conceptos que aún se conservan válidos en RUP. Divide el proceso de desarrollo en ciclos, en cada uno, se obtiene un producto final y se divide en cuatro fases: Inicio, Elaboración, Construcción y Transición, que termina con un hito bien definido. Para el ciclo de vida de los proyectos de la Universidad de las Ciencias Informáticas se decide mantener la fase de Inicio, pero cambiando el objetivo de la misma, se unen las restantes tres fases en la fase de Ejecución y se añade la fase de Cierre.

AUP plantea siete disciplinas (Modelo, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyecto y Entorno). Se decide para el ciclo de vida de los proyectos de la UCI conservar las mismas disciplinas, pero a un nivel más detallado que el definido en AUP. Los flujos de trabajo: Modelado de negocio, Requisitos, Análisis y Diseño en AUP están unidos en la disciplina Modelo, en la variación para la UCI se consideran a cada uno de ellos disciplinas. Se conserva la disciplina Implementación, en el caso de Prueba se dividen en 3 disciplinas: Pruebas Internas, de Liberación y Aceptación. Las restantes 3 disciplinas de AUP asociadas a la parte de gestión para la variación UCI se cubren con las áreas de procesos que define CMMI-DEV v1.3 para que pueda usarse en los proyectos productivos de la Universidad de las Ciencias Informáticas (UCI) para el nivel 2, serían CM (Gestión de la configuración), PP (Planificación de proyecto) y PMC (Seguimiento y control de proyecto). (*Metodología de Desarrollo de Software Variación de AUP para la UCI - EcuRed* sin fecha)

1.4.2 CARACTERÍSTICAS DE AUP-UCI, ESCENARIO 4.

Escenario No 4.

CAPÍTULO 1. FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE LOS PROCESOS DE REGISTRO DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos, ya que una HU no debe poseer demasiada información.

Todas las disciplinas antes definidas (desde Modelado de negocio hasta Pruebas de Aceptación) se desarrollan en la Fase de Ejecución, de ahí que en la misma se realicen Iteraciones y se obtengan resultados incrementales.

CONCLUSIONES DEL CAPÍTULO.

La solución propuesta debe incluir mecanismos que faciliten la integración entre las diversas aplicaciones de SIsalud y para lograr esto, se hará uso de una aplicación web. En el presente capítulo se definieron los principales conceptos relacionados con el sistema y se caracterizaron los lenguajes, tecnologías y metodologías a utilizar para el desarrollo del Sistema de Gestión de pacientes incluidos en los ensayos clínicos. Una vez concluida la investigación sobre los sistemas existentes para el registro de pacientes se contempló la necesidad de desarrollar una aplicación web para integrar al sistema de la plataforma SIsalud, ya que actualmente no se cuenta con un registro como el anterior descrito y el análisis de los sistemas homólogos realizado se definen que tampoco presentan un registro con la información detallada de cada paciente e implementarlos en esos sistemas hay que pagar un elevado costo. Asimismo, se definieron las herramientas a utilizar para la documentación e implementación de la sugerencia como solución adoptando como ambiente de desarrollo según las pautas establecidas por SOFTEL, la metodología AUP-UCI en su escenario 4, apoyado con el lenguaje de modelado UML, como IDE de desarrollo el Visual Studio Code, Visual Paradigm y para la base de datos MySQL ; empleando los frameworks NestJS para trabajar de lado al servidor y AngularJS para trabajar de lado al cliente , ambos con soporte de TypeScript y JavaScript como lenguajes de programación utilizados.

CAPÍTULO 2: DESARROLLO DEL SISTEMA DE PACIENTES.

INTRODUCCIÓN DEL CAPÍTULO.

El siguiente capítulo se plantearán aspectos relacionados con el diseño del sistema, incluyendo la especificación de requisitos funcionales y no funcionales, historias de usuarios y sus correspondientes prototipos, representación y características de la arquitectura elegida para el proyecto, así como los diagramas a elaborar de acuerdo con la metodología y el escenario para la realización del sistema seleccionado mostrando así, una comprensión clara de las funciones que debe realizar el software una vez en ejecución.

2.1 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN.

Después de haber realizado un estudio del arte en el capítulo anterior y reflejar que no existe un mecanismo que sea encargado de administrar la información y estadística de los pacientes incluidos en ensayos clínicos, como resultado, es muy necesario la creación de una aplicación que se encargue de gestionar, calcular y comunicar los datos de cada paciente agregado en los ensayos clínicos del sistema Xavia Sidec.

Se propone una aplicación web que permita el control de toda la información referente al paciente y su ensayo clínico asociado, permitiendo la gestión (modificar, insertar, eliminar) y la búsqueda de dicha información ya sea saber la cantidad de pacientes dentro de un mismo ensayo clínico, el rango de edades de paciente entre todos los ensayos y la información de cada paciente según su sitio de investigación. Se dispondrá una interfaz para mostrar la información listada de los pacientes y los ensayos en los que se incluyen, donde se podrán realizar las principales funciones como buscar, insertar, eliminar, etc. Dicha interfaz se mantendrá en completa comunicación con los sistemas, de gestión de ensayos clínicos Xavia Sidec y el sistema único de Identificación Nacional SUIN. Para el acceso, el usuario debe pasar por el sistema de autenticación de sistema SISalud, para luego tener acceso a un Registro de Pacientes Incluidos, redireccionando a la aplicación. Para facilitar el trabajo estadístico, el sistema deberá proveer los siguientes informes:

- Cantidad de pacientes incluidos por ensayos clínicos.
- Cantidad de pacientes incluidos por provincias.
- Cantidad de pacientes incluidos por sitios de investigación.

CAPÍTULO 2: DESARROLLO DEL SISTEMA DE PACIENTES.

- Cantidad de pacientes incluidos por promotores.
- Cantidad de pacientes incluidos por año de inclusión.
- Cantidad de pacientes incluidos por rangos de edades, incluyendo edad pediátrica.

2.1.1 MODELO CONCEPTUAL.

Los mapas conceptuales son organizadores gráficos que permiten representar el conocimiento como una serie de conceptos que se conectan con palabras vinculadas para formar una proposición, dan una idea clara de conceptos complejos y facilitan la enseñanza-aprendizaje (GARCÍA FRANCO, Vilma, GARCÍA NÚÑEZ, Rubén Darío, LORENZO GONZÁLEZ, Marisela, HERNÁNDEZ CABEZAS, Marilys, GARCÍA FRANCO, Vilma, GARCÍA NÚÑEZ, Rubén Darío, LORENZO GONZÁLEZ, Marisela y HERNÁNDEZ CABEZAS, Marilys, 2020).

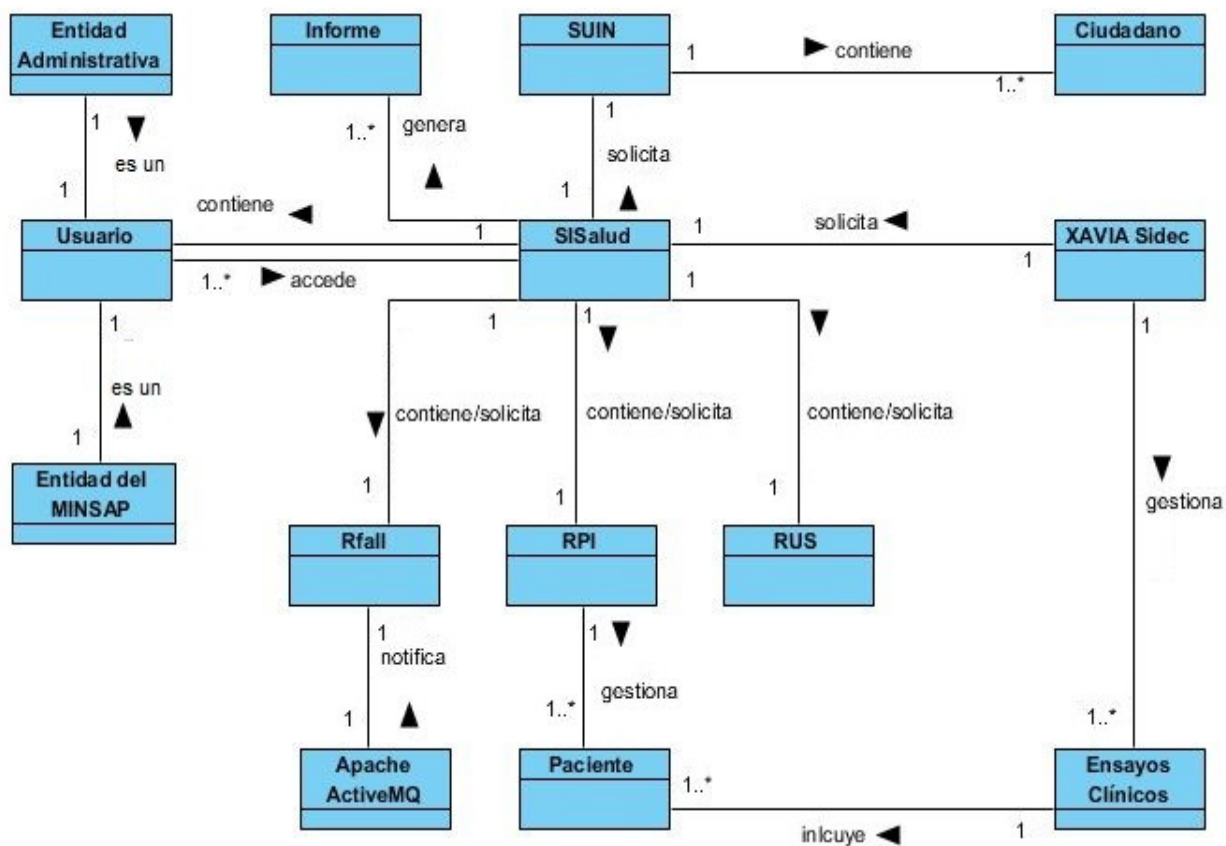


Figura 4: Modelo conceptual.

CAPÍTULO 2: DESARROLLO DEL SISTEMA DE PACIENTES.

2.1.2 DEFINICIÓN DE CLASES.

Entidad Administrativa: Es una persona con un perfil que se encarga de la gestión y mantenimiento de aplicaciones web. Este perfil es muy versátil, ya que independientemente de que no desarrolle la aplicación debe saber gestionarla y mantenerla, así como generar lo que se precise para su mejora.

Entidad del MINSAP: Es un usuario de una aplicación web de salud es una persona perteneciente al MINSAP que accede a un sitio web o un programa informático que ofrece servicios o información relacionada con la salud. Por ejemplo, un usuario puede consultar algún historial médico, revisar una cita, recibir consejos de prevención, etc. Los usuarios de una aplicación web de salud pueden tener diferentes roles, como doctores, médicos, enfermeros u otro personal de salud. dependiendo de sus necesidades y permisos.

Usuario: Un usuario en una aplicación web es una persona que utiliza un sitio web o un programa informático que ofrece servicios o información. En informática, un usuario es una persona que utiliza una computadora o un servicio de red. En el contexto de una aplicación web, un usuario puede tener diferentes roles, como pacientes, médicos, administrativos, etc., dependiendo de sus necesidades y permisos.

Informe: Un informe que genera una aplicación web es un documento que se crea automáticamente a partir de los datos almacenados en la aplicación. Los informes pueden ser personalizados y diseñados para satisfacer las necesidades específicas de los usuarios. Por ejemplo, un informe puede mostrar estadísticas de uso, resultados de encuestas, resúmenes de ventas. Los informes pueden ser generados en diferentes formatos, como PDF, Excel, HTML.

SUIN: es el sistema que el Ministerio del Interior (MININT) de Cuba provee para la identificación de ciudadanos que radican en el país.

Ciudadano: es una persona que ha creado una cuenta en la aplicación web y ha proporcionado sus datos de información personal, como su nombre, dirección, fecha de nacimiento y más.

CAPÍTULO 2: DESARROLLO DEL SISTEMA DE PACIENTES.

XAVIA Sidec: El Sistema para el manejo de datos de Ensayos Clínicos XAVIA SIDEDEC es un software desarrollado por el Centro de Inmunología Molecular (CIM) de Cuba para la gestión y conducción de ensayos clínicos.

Ensayo clínico: Un ensayo clínico es una investigación científica que se realiza con personas voluntarias para evaluar la seguridad y eficacia de nuevos tratamientos, vacunas, pruebas diagnósticas o cambios de hábitos relacionados con la salud.

SISalud: El sistema SISALUD es un sistema de información para la gestión de la salud en Cuba. Es un conjunto de aplicaciones informáticas que permiten el registro, procesamiento, análisis y difusión de datos e indicadores relacionados con la salud de la población, los recursos humanos, los servicios, los programas y las investigaciones.

RFall: Registro de fallecidos.

Apache ActiveMQ Es un servicio de intermediación de mensajes que permite la comunicación entre diferentes aplicaciones o sistemas mediante el intercambio de mensajes

RPI: Registro de pacientes incluidos en ensayos clínicos.

Paciente: es un ciudadano registrado en este caso en una aplicación web de salud, un usuario de una aplicación web de salud, o un participante en un ensayo clínico.

RUS: Registro de Unidades de Salud.

2.2 DEFINICIÓN DE REQUISITOS.

Para un sistema los requisitos son descripciones detalladas de las funciones, servicios y las restricciones operacionales de software. El documento de requisitos del sistema define con exactitud lo que se implementará. A menudo, los requisitos de software se clasifican como requisitos funcionales o requisitos no funcionales.(I. Sommerville 2016).

2.2.1 REQUISITOS FUNCIONALES.

Son tareas, capacidades o condiciones que el sistema debe cumplir. Dichas tareas dirigen un entendimiento de cuál será el impacto del software sobre el negocio, qué quieren los

CAPÍTULO 2: DESARROLLO DEL SISTEMA DE PACIENTES.

clientes y cómo los usuarios finales interactuarán con dicho software. Deben ser lo más completos, claros y precisos posibles (S. Pressman y B. R. Maxim 2015).

ID	NOMBRE	DESCRIPCIONES	PRIORIDAD	COMPLEJIDAD
RF-1	Registrar paciente incluido	El sistema debe permitirle al usuario registrar un paciente incluido en los ensayos clínicos mediante los datos: Nombre Apellido Edad Sexo Estado (Vivo o Muerto)	alta	media
RF-2	Buscar paciente incluido	El sistema debe permitirle al usuario realizar una búsqueda filtrando por los datos de los pacientes incluidos en los ensayos clínicos.	alta	media
RF-3	Registrar codificador de ensayo	El sistema debe permitirle al usuario registrar un codificador de ensayo mediante los datos: Código ensayo Descripción Nombre corto implícitamente se registrará la unidad que lo realiza.	alta	alta
RF-4	Registrar codificador de promotores	El sistema debe permitirle al usuario registrar un codificador de promotores mediante los datos: Nombre entidad Siglas Correo del responsable	alta	alta

CAPÍTULO 2: DESARROLLO DEL SISTEMA DE PACIENTES.

RF-5	Editar codificador de ensayo	El sistema debe permitirle al usuario editar un codificador de ensayo mediante los datos: Código ensayo Descripción Nombre corto implícitamente se registrará la unidad que lo realiza.	alta	alta
RF-6	Editar codificador de promotores	El sistema debe permitirle al usuario registrar un codificador de promotores mediante los datos: Nombre entidad Siglas Correo del responsable	alta	alta
RF-7	Eliminar paciente incluido	El sistema debe permitirle al usuario realizar una búsqueda filtrando por los datos de los pacientes incluidos en los ensayos clínicos.	alta	media
RF-8	Editar paciente incluido	El sistema debe permitirle al usuario modificar un paciente incluido mediante los datos: Nombre Apellido Edad Sexo Estado (Vivo o Muerto)	alta	alta
RF-9	Listar paciente incluido	El sistema debe permitirle al usuario listar todos los pacientes incluidos en los ensayos clínicos.	alta	baja
RF-10	Listar codificador de	El sistema debe permitirle al usuario listar todos los	alta	baja

CAPÍTULO 2: DESARROLLO DEL SISTEMA DE PACIENTES.

	ensayos	codificadores de ensayos.		
RF-11	Listar codificador de promotores	El sistema debe permitirle al usuario listar todos los codificadores de promotores.	alta	baja
RF-12	Buscar cantidad de pacientes incluidos por ensayos clínicos	El sistema debe permitirle al usuario realizar una búsqueda filtrando por los datos de los pacientes incluidos en los ensayos clínicos.	alta	media
RF-13	Buscar cantidad de pacientes incluidos por provincias	El sistema debe permitirle al usuario realizar una búsqueda de los pacientes por provincia	media	media
RF-14	Buscar cantidad de pacientes incluidos por sitios de investigación	El sistema debe permitirle al usuario realizar una búsqueda por los sitios de investigación.	media	media
RF-15	Buscar cantidad de pacientes incluidos por promotores	El sistema debe permitirle al usuario realizar una búsqueda por los datos de los promotores guardados en el sistema.	media	media

CAPÍTULO 2: DESARROLLO DEL SISTEMA DE PACIENTES.

RF-16	Buscar cantidad de pacientes incluidos por año de inclusión	El sistema debe permitirle al usuario realizar una búsqueda por el año de inclusión.	media	media
RF-17	Buscar cantidad de pacientes incluidos por rangos de edades, incluyendo edad pediátrica para menores de 18 años	El sistema debe permitirle al usuario realizar una búsqueda por rangos de edad.	media	media

Tabla 2: Requisitos funcionales.

2.2.2 REQUISITOS NO FUNCIONALES.

Definen cuáles son las propiedades con las que el sistema debe contar, así como sus restricciones fundamentales. Este tipo de requisitos garantizan que el producto sea fácil de usar, seguro y atractivo. Hacen referencia a las propiedades emergentes de un sistema como la fiabilidad, el tiempo de respuesta, el rendimiento y la seguridad (S. Pressman y B. R. Maxim 2015).

CAPÍTULO 2: DESARROLLO DEL SISTEMA DE PACIENTES.

ID	NOMBRE	DESCRIPCIONES	PRIORIDAD	COMPLEJIDAD
RNF-1	Integración Xavia Sidec con RPI	Luego de pasar por el sistema de autenticación de SiSalud, el usuario tendrá acceso al RPI, mostrándosele la pantalla de Registro de Paciente Incluido (RF 1), con los datos del ensayo ya cargados y deshabilitados. Para ello, Xavia Sidec deberá proporcionar los datos del ensayo clínico con los que registrará al paciente o los pacientes incluidos en el mismo.	alta	alta
RNF-2	Integración de SUIN con RPI	Una vez introducido el CI en el formulario de registro de paciente incluido (RF1), se cargan los datos de Xavia Sidec, en caso en caso de no existir la persona en la BD, El sistema SUIN deberá proporcionar los datos relevantes.	media	alta
RNF-3	Usabilidad	Debe tener una opción de ayuda sobre las principales funcionalidades que brinda el sistema reconocer su adecuación y sus iconos respectivos, para un mejor entendimiento.	baja	media
RNF-4	Seguridad	Capacidad de protección contra el acceso a datos e información no autorizados, ya sea Autenticidad accidental o deliberadamente.	alta	alta

CAPÍTULO 2: DESARROLLO DEL SISTEMA DE PACIENTES.

RNF-5	Portabilidad	Capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes navegadores web, principalmente a Google Chrome y Firefox.	Alta	Baja
-------	--------------	--	------	------

Tabla 3: Requisitos no funcionales.

2.3 DESCRIPCIÓN DE LOS REQUISITOS.

Historia de Usuario	
Número: 1	Rol: Usuario
Título: Registrar Paciente Incluido	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Tiempo estimado: 4 Horas	Iteración Asignada:1
Programador Responsable: Rubén Álvarez Izquierdo.	
Descripción: El sistema insertará el paciente incluido en Ensayos Clínicos para darle seguimiento y manipular la información estadística con el que se relaciona.	
No	Criterio de Aceptación
1	El usuario debe tener asignado los permisos para la funcionalidad registrar paciente incluido. Debe haberse autenticado en el sistema y haber accedido a la funcionalidad Registrar Paciente Incluido.
2	Al insertar el carnet de identidad el usuario debe seleccionar el botón, en caso de que el paciente ya se encuentra registrado en la Base de Datos Ciudadano, el sistema rellena los

CAPÍTULO 2: DESARROLLO DEL SISTEMA DE PACIENTES.

	siguientes campos automáticamente: Nombre, Apellido, Edad, Sexo, Estado (Vivo o Muerto). Si el paciente no se encuentra registrado en la Base de Datos Ciudadano, el sistema mantiene la página sin cambios y el usuario debe presionar el botón y registrar los datos del nuevo ciudadano en la BD Ciudadano.
3	El código del paciente, es un campo obligatorio, y único, autogenerado por el usuario y responde al siguiente patrón: [Siglas entidad promotora]_[Nombre corto del ensayo clínico]_[Siglas de la institución donde se incluyó el sujeto]_[Iniciales del nombre del sujeto]_[Número de inclusión].
4	El código del ensayo clínico es obligatorio, y el usuario debe seleccionar el ensayo relacionado con el paciente incluido. Los ensayos se listaran en el Codificador de Ensayo.
5	El sitio de investigación es obligatorio y el usuario tendrá la posibilidad de teclear el Código o el Nombre del sitio si los conoce, de lo contrario, el usuario selecciona el sitio a través de un listar Unidades de Salud del RUS.
6	El promotor es obligatorio y el usuario selecciona el promotor de salud relacionado con el paciente incluido previamente captado en el Codificador de Promotores.
7	La fecha de inclusión es obligatoria, se debe guardar en el formato año/mes/día. Esta fecha hace referencia a la instancia en que al paciente le realizan el primer ensayo clínico.
8	El número de inclusión es único y obligatorio, y permite al usuario una entrada de texto libre.
9	La fecha de interrupción debe ser mayor que la fecha .de inclusión y hace referencia a la fecha en que se interrumpe el ensayo clínico del paciente incluido.
Prototipo elemental de interfaz gráfica de usuario:	

CAPÍTULO 2: DESARROLLO DEL SISTEMA DE PACIENTES.

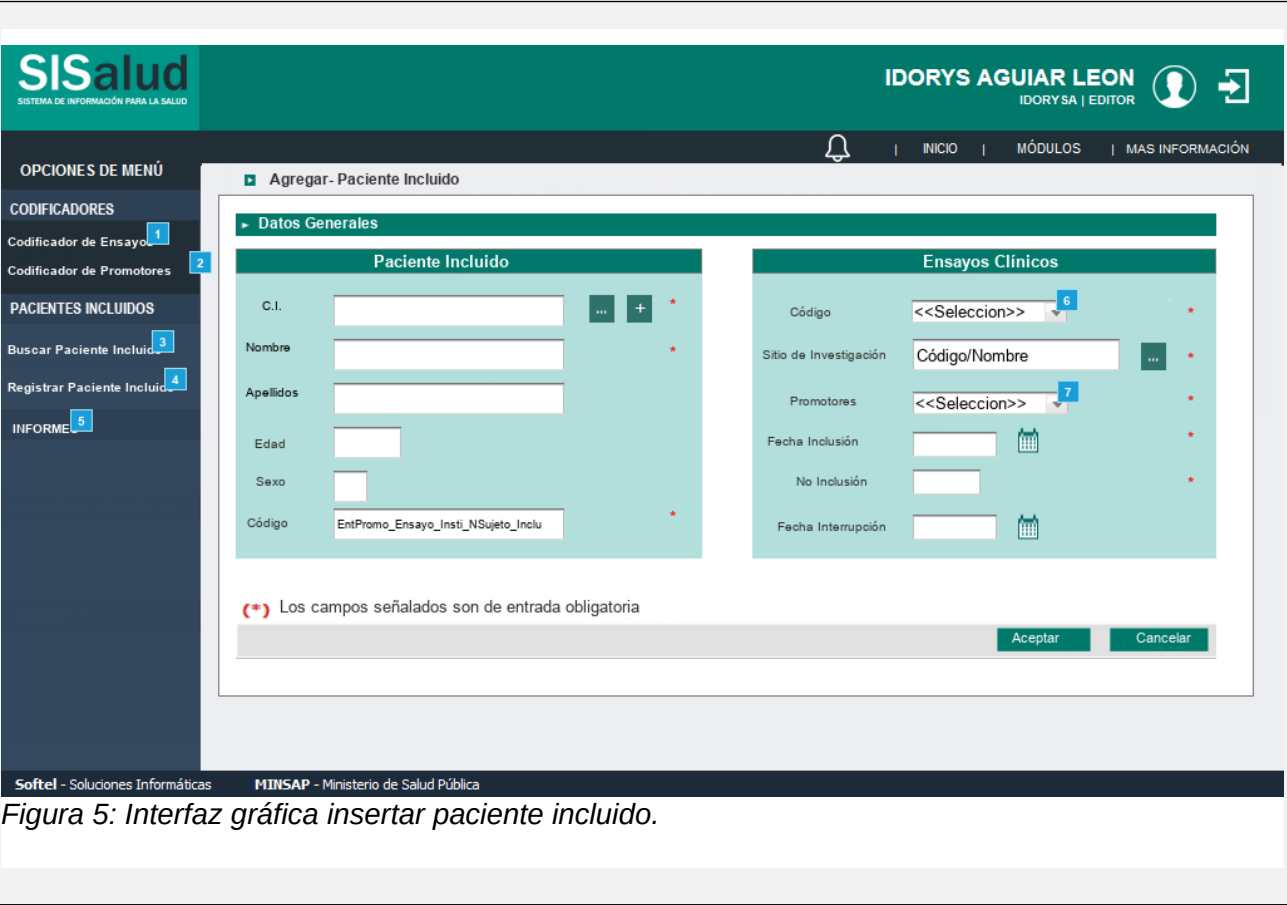


Figura 5: Interfaz gráfica insertar paciente incluido.

Tabla 4: HU Registrar Paciente Incluido

Historia de Usuario	
Número: 2	Rol: Usuario
Título: Buscar Paciente Incluido	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Tiempo estimado: 4 Horas	Iteración Asignada:1
Programador Responsable: Michel Leandro Pérez Somoza.	

CAPÍTULO 2: DESARROLLO DEL SISTEMA DE PACIENTES.

Descripción: El sistema permitirá al usuario buscar y eliminar un Paciente Incluido en Ensayos Clínicos.	
No	Criterio de Aceptación
1	El usuario debe tener asignado los permisos para la funcionalidad buscar paciente incluido. Debe haberse autenticado en el sistema y haber accedido a la funcionalidad Buscar Paciente Incluido. Para eliminar un paciente, el usuario debe tener asignados los permisos correspondientes, acceder a la funcionalidad Buscar Paciente Incluido>Eliminar.
2	Dentro de la Sección de Búsqueda del Paciente, el usuario podrá filtrar por uno de los siguientes parámetros: carnet de identidad (único, asociado al paciente incluido que el usuario desea buscar), Nombre del Paciente Incluido, Apellidos del Paciente Incluido, código del Ensayo Clínico, el sitio de investigación asociado con el paciente incluido que desea filtrar la búsqueda, número de inclusión(único) y fecha de inclusión por rango de fechas.
3	<i>Al presionar el botón: Eliminar, el paciente debe seleccionar si desea eliminar el Paciente Incluido ->Aceptar si desea confirmar la operación de eliminar el paciente incluido, en caso contrario, el usuario debe presionar el botón cancelar y el sistema debe redireccionar a la vista Buscar Paciente incluido.</i>
Prototipo elemental de interfaz gráfica de usuario:	

CAPÍTULO 2: DESARROLLO DEL SISTEMA DE PACIENTES.

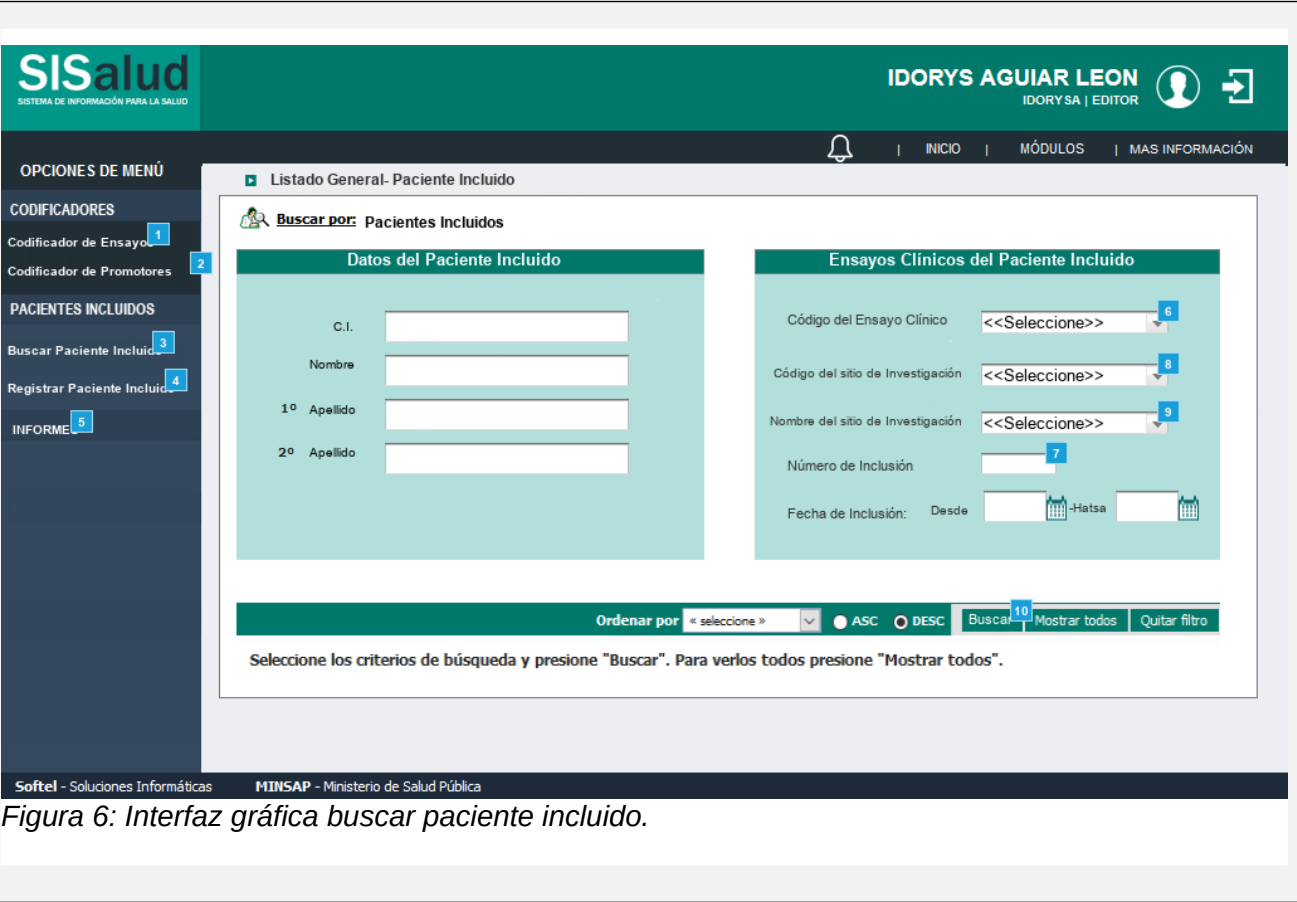


Figura 6: Interfaz gráfica buscar paciente incluido.

Tabla 5: HU Buscar Paciente Incluido

2.2.4 DIAGRAMA DE CLASES DEL DISEÑO.

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación, es un conjunto de definiciones de entidades software más que de conceptos del mundo real. Las clases del diagrama pueden tener relaciones entre ellas tales como relaciones de dependencia, asociativas y herencia (Craig Larman sin fecha). A continuación, en la Figura se muestra el DCD del requisito funcional Gestionar Modificación de los pacientes incluidos.

CAPÍTULO 2: DESARROLLO DEL SISTEMA DE PACIENTES.

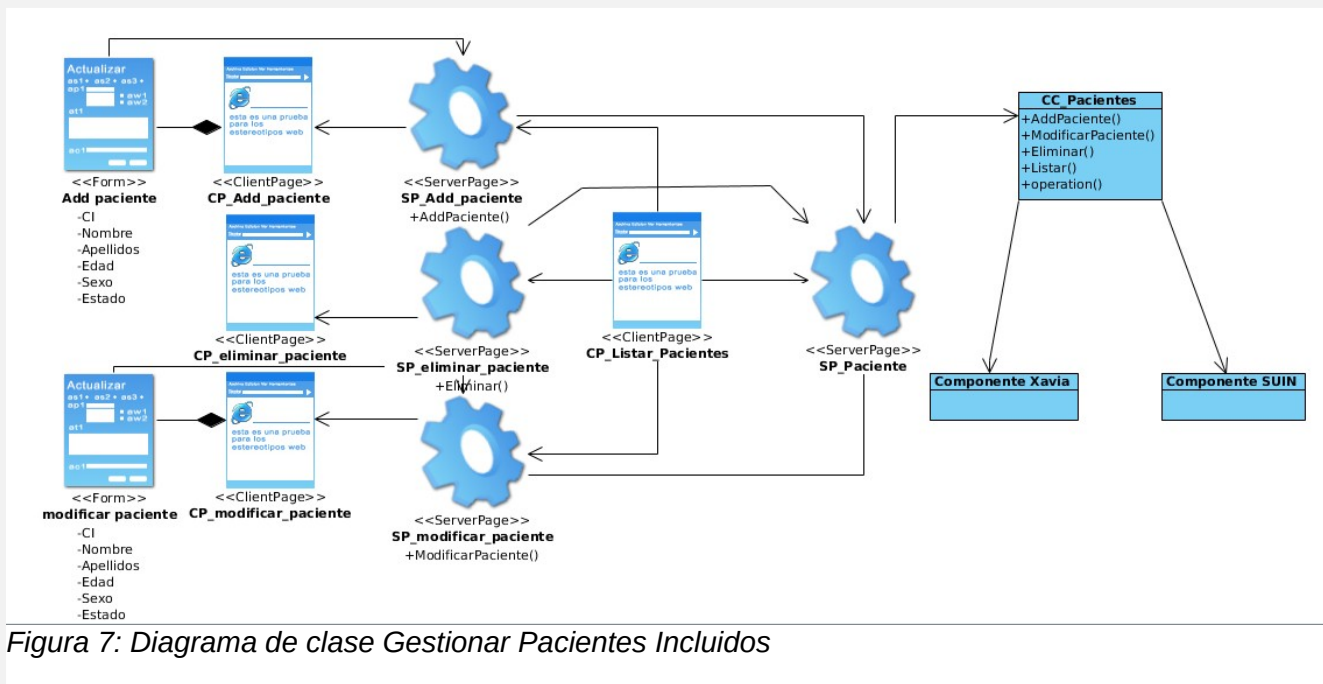


Figura 7: Diagrama de clase Gestionar Pacientes Incluidos

2.3 DISEÑO DE LA ARQUITECTURA DE SISTEMA.

La arquitectura de software conforma el esqueleto de cualquier sistema, y es la principal responsable de los atributos de calidad del sistema (parte de los requerimientos no funcionales del mismo) la misma identifica los elementos más importantes de un sistema, así como sus relaciones, es decir, da una visión global del sistema.

En el caso del componente desarrollado se utiliza el patrón arquitectónico MVC (modelo-vista-controlador) utilizado por el marco de trabajo Angular, este se basa en separar las funcionalidades del sistema en tres componentes: modelo, vista, y controlador. A continuación, se muestra el patrón arquitectónico MVC

- **Modelo:** es una representación de los datos o el estado de la aplicación, contiene (o provee una interfaz) para la lógica de la aplicación.
- **Vista:** es un componente de interfaz de usuario que produce una representación del modelo para el usuario y permite a este la entrada de datos.

CAPÍTULO 2: DESARROLLO DEL SISTEMA DE PACIENTES.

- **Controlador:** maneja la interacción entre el modelo y la vista, convirtiendo las acciones del usuario en cambios en el modelo o la vista.

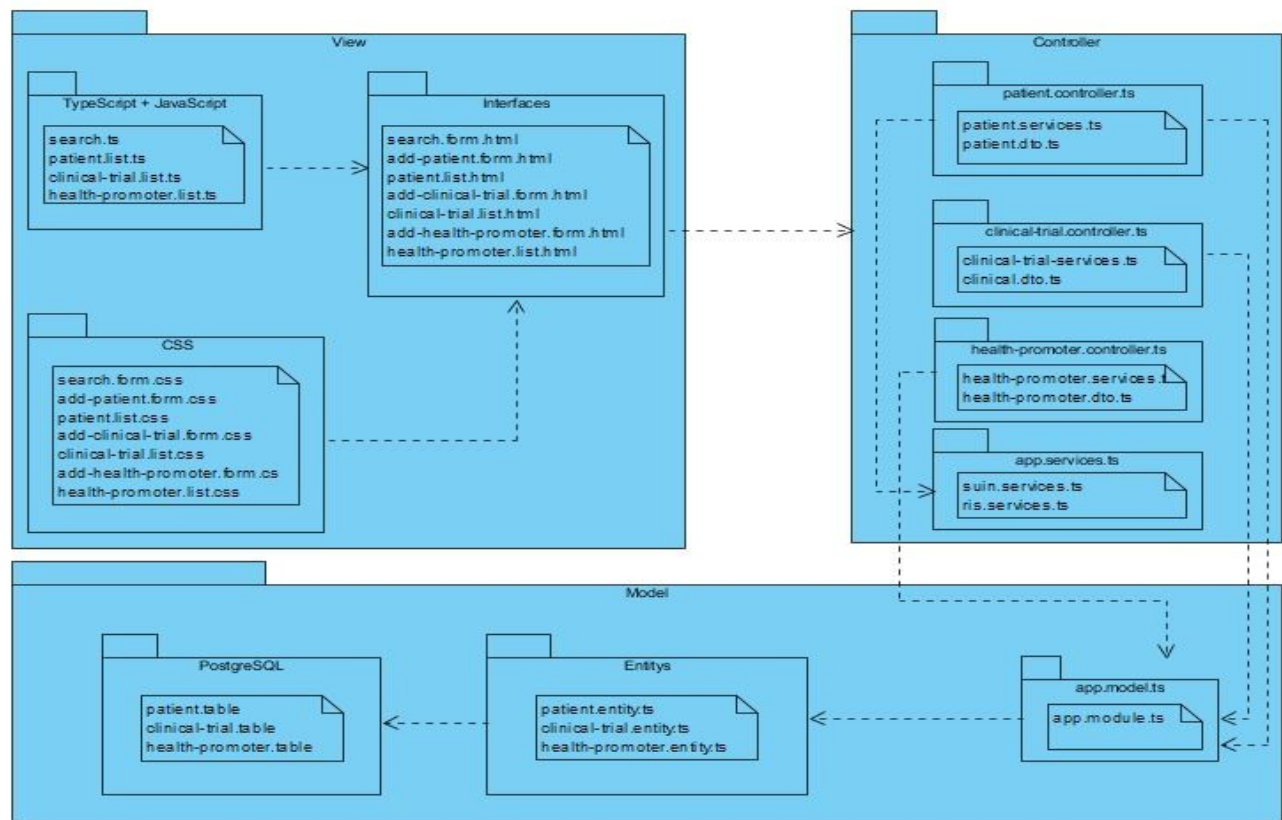


Figura 8: Patrón arquitectónico.

2.3.1 PATRONES DE DISEÑO.

Un patrón de diseño provee un esquema para refinar los subsistemas o componentes de un sistema de software, o las relaciones entre ellos. Describe la estructura comúnmente recurrente de los componentes en comunicación, que resuelve un problema general de diseño en un contexto particular. Tienden a ser independientes de los lenguajes y paradigmas de programación y su aplicación no afecta necesariamente al sistema completo, pero sí a un subsistema o parte del mismo (Pressman sin fecha).

CAPÍTULO 2: DESARROLLO DEL SISTEMA DE PACIENTES.

A continuación, se muestran los patrones de diseño empleados en la implementación de la propuesta:

Experto: Se encarga de asignar la responsabilidad al experto en la información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. Brinda la posibilidad de conservar el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide, lo que provee un bajo nivel de acoplamiento. Promueve clases sencillas y cohesivas que son más fáciles de mantener y comprender (Pressman sin fecha). Se pone en práctica en las clases **Layouts** que es la que posee toda la información necesaria para mostrar y almacenar los datos adquiridos.

Creador: Consiste en asignar a un Objeto la responsabilidad de crear otro Objeto. Un objeto es responsable de crear una nueva instancia de alguna clase si: agrega o contiene objetos de ella, registra las instancias de sus objetos o tiene los datos de inicialización que serán enviados a ella cuando el objeto sea creado (Pressman sin fecha). Se refleja en la clase **app.module.ts**, donde se cargan todos los módulos presentes y se ejecutan cada uno de ellos.

```
21 PatientListComponent,  
22 ...AppComponent,  
23 ...],  
24 ...imports:[  
25 ...CommonModule,  
26 ...NgSelectModule,  
27 ...FormsModule,  
28 ...BrowserModule,  
29 ...AppRoutingModule,  
30 ...NgbModule,  
31 ...HttpClientModule,  
32 ...ReactiveFormsModule,  
33 ...],  
34 ...providers:[ClinicalTrialService, HealthPromoterService],
```

Figura 9: Estructura de app.module.ts

CAPÍTULO 2: DESARROLLO DEL SISTEMA DE PACIENTES.

Bajo acoplamiento: El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo (o débil) acoplamiento no depende de muchas otras. El Bajo Acoplamiento es un principio que debemos recordar durante las decisiones de diseño: es la meta principal que es preciso tener presente siempre. Es un patrón evaluativo que el diseñador aplica al juzgar sus decisiones de diseño (*Patrones Grasp (Craig Larman) Parte I | El Mundo Informático* sin fecha). En la propuesta de solución se utilizó este patrón en las clases presentes en la carpeta **src/app/services/**, la cual se relacionan con la menor cantidad de clases posibles para el cumplimiento de sus funciones.

Controlador: El patrón ofrece una guía para tomar decisiones sobre los eventos de entrada, asignando la responsabilidad del manejo de mensajes de los eventos del sistema a una clase controladora, ya que los elementos de interfaz y sus controladores de eventos, no deben ser

```
1 import { NgModule } from '@angular/core';
2 import { RouterModule, Routes } from '@angular/router';
3 import { ClinicalTrialListComponent } from '../clinical-trial-list/clinical-trial-list.component';
4 import { HealthPromoterListComponent } from '../health-promoter-list/health-promoter-list.component';
5 import { PatientListComponent } from '../patient-list/patient-list.component';
6
7 const routes: Routes = [
8   {
9     path: 'patient',
10    component: PatientListComponent,
11  },
12  {
13    path: 'clinical-trial',
14    component: ClinicalTrialListComponent,
15  },
16  {
17    path: 'health-promoter',
18    component: HealthPromoterListComponent,
19  },
20 ];
21 @NgModule({
22   imports: [RouterModule.forRoot(routes)],
23   exports: [RouterModule],
24 })
25 export class AppRoutingModule {}
```

Figura 10: Ejemplo del routing

CAPÍTULO 2: DESARROLLO DEL SISTEMA DE PACIENTES.

responsables de controlar los eventos del sistema (Pressman sin fecha). En la propuesta de solución se utilizó este método en las clases que realizan solicitudes del usuario, la cual va a controlar las funciones sobre la interfaz principal.

Alta Cohesión: Nos dice que la información que almacena una clase debe de ser coherente y debe estar (en la medida de lo posible) relacionada con la clase. La clase **Routes** está altamente cohesionada al ser responsable del envío en tiempo real de los datos obtenidos de las inspecciones.

2.3.2 PATRONES GOF.

Los patrones GOF son aquellos que describen soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos.

Patrón observador: Este patrón está presente en las clases de Angular las cuales están pendientes de validar la información que introduce el usuario y en caso de que esta no está correcta se notificaran en los formularios.

```
1 <nav class="navbar navbar-expand-lg navbar-light bg-light">
2   <div class="collapse navbar-collapse">
3     <ul class="navbar-nav mr-auto">
4       <li class="nav-item">
5         <a class="nav-link" routerLink="/patient" routerLinkActive="active">Pacientes</a>
6       </li>
7       <li class="nav-item">
8         <a class="nav-link" routerLink="/clinical-trial" routerLinkActive="active">Ensayos Clínicos</a>
9       </li>
10      <li class="nav-item">
11        <a class="nav-link" routerLink="/health-promoter" routerLinkActive="active">Promotores de Salud</a>
12      </li>
13    </ul>
14  </div>
15 </nav>
16
17 <!-- Router Outlet -->
18 <router-outlet></router-outlet>
19
```

Figura 11: Patrón decorador.

CAPÍTULO 2: DESARROLLO DEL SISTEMA DE PACIENTES.

Patrón decorador: Se evidencia en las vistas de Angular, las cuales contienen estructuras y diseños comunes para todo el sistema, heredando componentes de manera dinámica y decorándola con contenido propio del mismo.

2.4 DIAGRAMA DE BASE DE DATOS.

Para la gestión de la base de datos se utilizará mysql. La base de datos constará de la siguiente estructura.

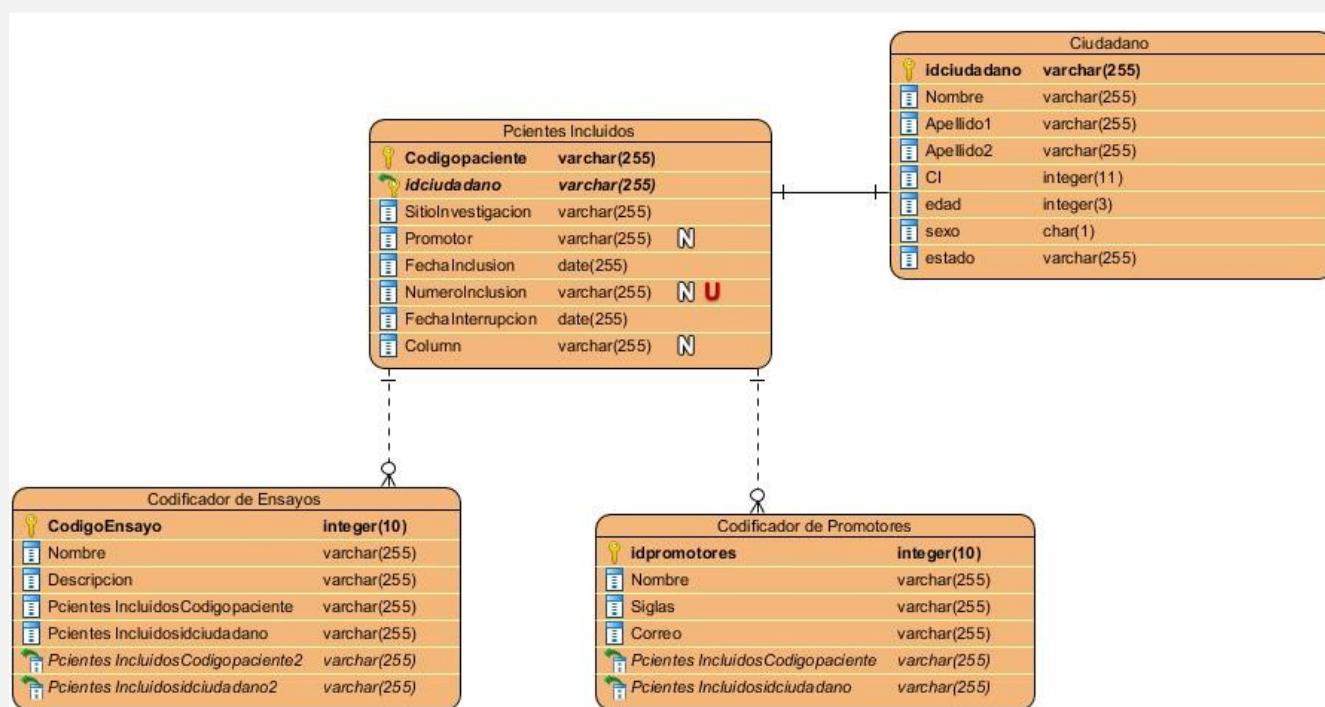


Figura 12: Diagrama de base de datos

CONCLUSIONES DEL CAPÍTULO.

En el capítulo se describe los principales conceptos del dominio, conociéndose el procedimiento para el registro de pacientes. Se identificaron los actores del negocio, así como las clases a tener en cuenta durante todo el proceso. Se definieron los requisitos funcionales y se obtuvo un listado de requerimientos no funcionales a tener en cuenta para el desarrollo de la aplicación, mostrándose las descripciones de cada uno de los requisitos funcionales en historias de usuario. Igualmente se describen la las clases del diseño, la arquitectura del sistema en conjunto con los

CAPÍTULO 2: DESARROLLO DEL SISTEMA DE PACIENTES.

patrones de diseño y GOF empleados, el diagrama de base de datos. Toda la información va dando una visión global de cómo está prevista la implementación y cómo va a ser el funcionamiento de la propuesta de solución.

CAPÍTULO 3. VALIDACIÓN DEL SISTEMA DE GESTIÓN DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

3.1. INTRODUCCIÓN DEL CAPÍTULO.

En este capítulo se describe la implementación de la propuesta de solución, donde se realiza el producto final y se cumplen los requisitos especificados en el capítulo anterior, realizando el prototipado y diseñando interfaces funcionales y otros diagramas. Se definen reglas o estándares de codificación para uniformidad y mayor comprensión del código fuente. Del código generado y su puesta en marcha se diseñan y ejecutan las pruebas de software, técnicas y métodos de pruebas para la solución desarrollada. Para la validación de los requisitos se realizan revisiones, se generan casos de pruebas y prototipos y para la validación de la solución se define una estrategia de prueba aplicando varios niveles de pruebas con sus respectivos métodos y técnicas.

3.2. IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN.

El modelo de implementación describe cómo los elementos del diseño se implementan en componentes. También describe la organización de los componentes según los mecanismos de estructuración disponibles en el entorno de desarrollo, el lenguaje de programación utilizado, y la dependencia entre componentes. A partir de los resultados del análisis y el diseño se realiza la implementación del sistema, generando el modelo de implementación como artefacto de esta disciplina. Este modelo está conformado por el modelo de componentes y el modelo de despliegue.

3.2.1. ESTÁNDARES DE CODIFICACIÓN.

TypeScript proporciona dos maneras de organizar el código: espacios de nombres y módulos. Se explica cómo organizar el código de TypeScript mediante espacios de nombres. Aunque tanto los espacios de nombres como los módulos proporcionan una forma de categorizar el código relacionado de maneras significativas y permitirle controlar qué código se expone al espacio de nombres global de las aplicaciones, hay algunas diferencias entre ellos.

Los espacios de nombres (denominados "módulos internos" en versiones anteriores de TypeScript) son una forma específica de TypeScript de organizar y categorizar el código, lo que le

CAPÍTULO 3. VALIDACIÓN DEL SISTEMA DE GESTIÓN DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

permite agrupar el código relacionado. Los espacios de nombres permiten agrupar variables, funciones, interfaces o clases relacionadas con las reglas de negocio en un espacio de nombres y la seguridad en otro.

El código dentro de un espacio de nombres se extrae del ámbito global y entra en el ámbito del espacio de nombres. Esta colocación puede ayudarle a evitar conflictos de nomenclatura entre los componentes del espacio de nombres global y puede ser beneficioso al trabajar con equipos de desarrollo distribuidos que pueden utilizar nombres de componentes similares. (GeekTrainer sin fecha)

Puede utilizar espacios de nombres o módulos para la organización de código y ambos pueden contener código y declaraciones.

Y, a partir de ECMAScript 2015, los módulos son parte nativa del lenguaje y todas las implementaciones del motor compatibles los deben admitir. Por lo tanto, para los proyectos nuevos, se recomiendan los módulos para la organización del código. (GeekTrainer sin fecha)

3.2.2. MODELO DE COMPONENTES.

Los diagramas de componentes muestran las interacciones y relaciones de los componentes de un modelo. Entendiéndose como componente a una clase de uso específico, que puede ser implementada desde un entorno de desarrollo, ya sea de código fuente, binario o ejecutable, dichos componentes poseen tipo, y están unidos mediante relaciones de dependencia (Pressman sin fecha). A continuación, se presenta el diagrama de componentes del sistema.

CAPÍTULO 3. VALIDACIÓN DEL SISTEMA DE GESTIÓN DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

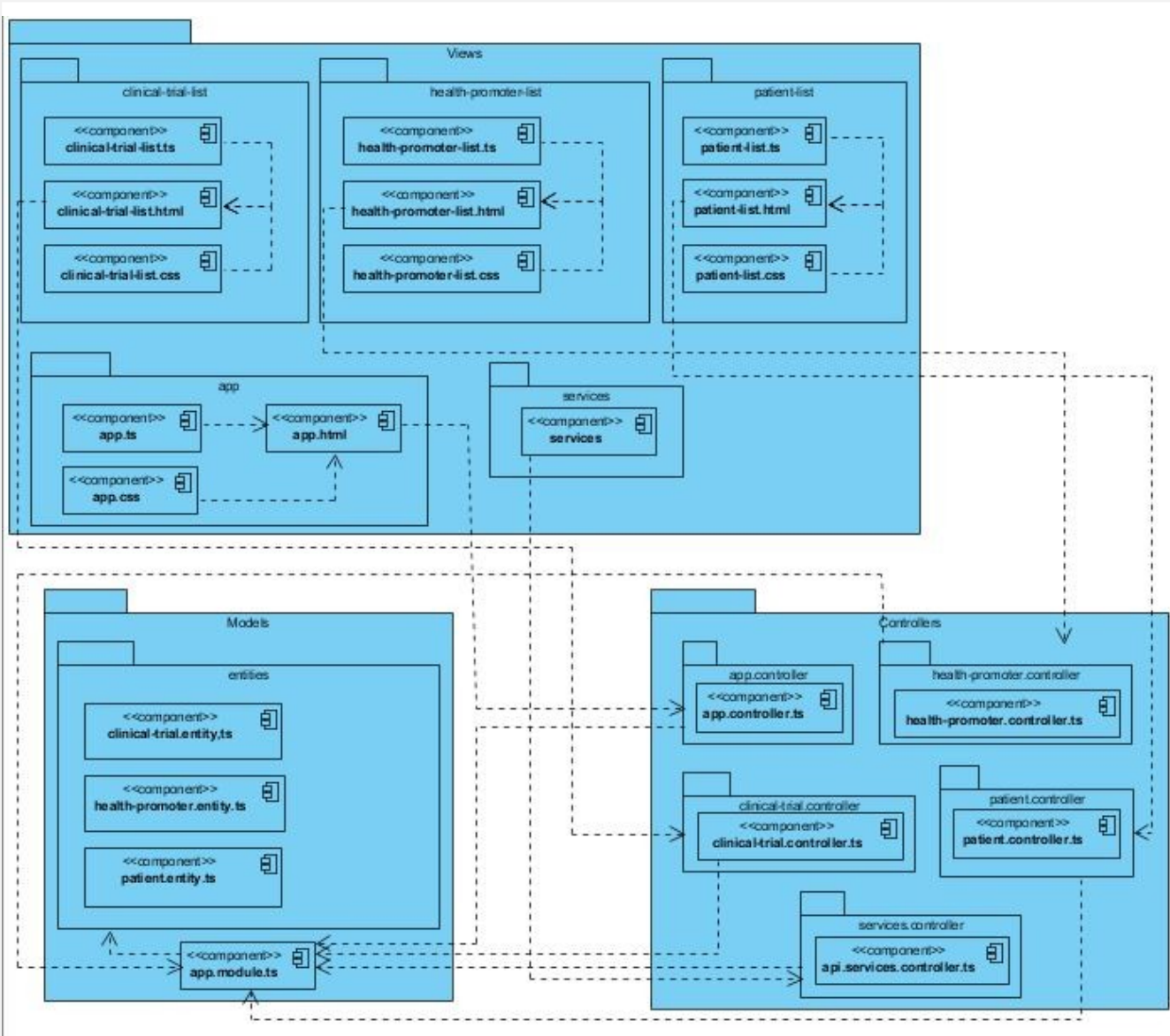


Figura 13: Modelo de componentes.

3.2.3. MODELO DE DESPLIEGUE.

El modelo de despliegue se realiza como parte de la implementación para describir la distribución física del sistema. Establece la correspondencia entre la arquitectura lógica, los procesos y los nodos. Cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar. Los nodos poseen relaciones que

CAPÍTULO 3. VALIDACIÓN DEL SISTEMA DE GESTIÓN DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

representan medios de comunicación entre ellos (Pressman sin fecha). Se reflejan en este artefacto los protocolos de comunicación mediante los cuales se comunican los nodos respectivos.

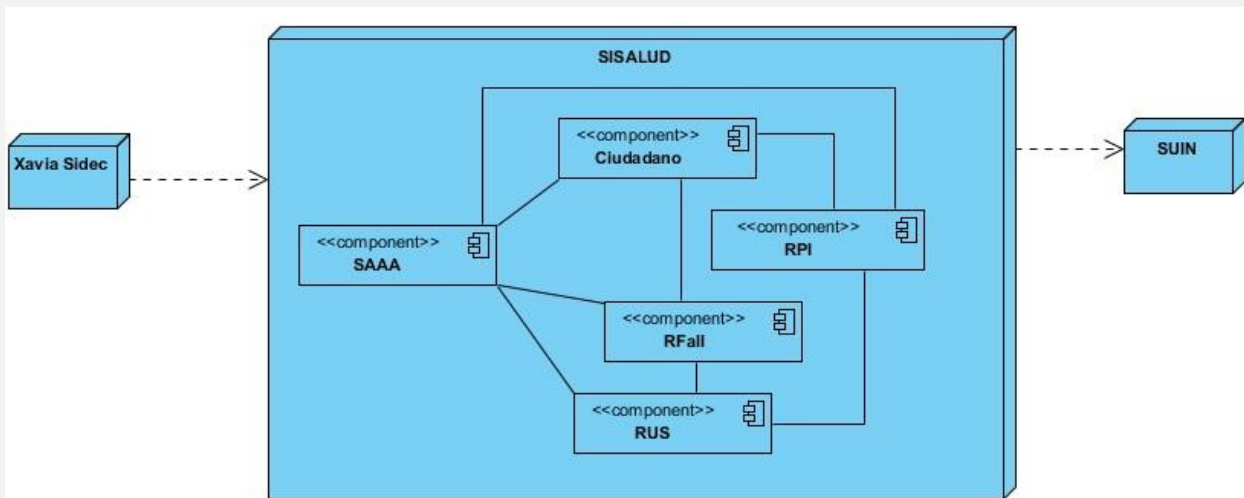


Figura 14: modelo de despliegue.

3.3. VALIDACIÓN DE REQUISITOS.

Con el objetivo de comprobar que los requisitos del software obtenido se ajusten al componente que se necesita, se realizó un proceso de validación de los mismos, para el cual se emplearon las siguientes técnicas:

Revisiones de los requisitos: Se realizaron revisiones a cada uno de los requisitos por parte del equipo de desarrollo.

Generación de casos de prueba: Como parte del proceso de validación de los requisitos funcionales se diseñaron casos de pruebas, se define uno por cada requisito.

Diseño de prototipos web: Se muestra al cliente un modelo ejecutable que permita tener una visión preliminar de cómo se vería el componente y a través de la interacción con estos se comprueba satisfacción del cliente hacia los prototipos diseñados.

CAPÍTULO 3. VALIDACIÓN DEL SISTEMA DE GESTIÓN DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

3.3.1. MÉTRICAS APLICADAS A LOS REQUISITOS.

Con el fin de medir la calidad de la especificación de los requisitos, se aplicó la métrica **Calidad de la especificación (CE)**, una de las métricas propuestas para la metodología AUP (UCI).

A continuación, se muestra el cálculo del total de requisitos de la especificación con el objetivo de obtener qué entendibles y precisos son los mismos:

- **Nr:** total de requisitos de especificación.
- **Nf:** cantidad de requisitos funcionales.
- **Nnf:** cantidad de requisitos no funcionales.

Teniendo en cuenta que:

$$\mathbf{Nr = Nf + Nnf}$$

$$\mathbf{Nr = 17 + 5}$$

$$\mathbf{Nr = 22}$$

Como resultado de la sustitución de los valores para el componente se obtiene que: el total de requisitos de especificación tiene un valor de 19, quiere decir que, existe un total de 19 requisitos para ser analizados y tras este se procede a determinar cuán entendibles y precisos son.

Especificidad de los Requisitos (ER)

Nui: número de requisitos para los cuales todos los revisores tuvieron interpretaciones idénticas. Mientras más cerca de 1 esté el valor de ER, menor será la ambigüedad. Para determinar, la Especificidad de los Requisitos (ER) o ausencia de ambigüedad en los mismos se realiza la siguiente operación:

Teniendo en cuenta que:

$$\mathbf{ER = Nui / Nr}$$

Para el caso de los requisitos obtenidos, cuatro produjeron contradicción en las interpretaciones.

CAPÍTULO 3. VALIDACIÓN DEL SISTEMA DE GESTIÓN DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

Sustituyendo las variables se obtiene:

$$ER = 18/ 22$$

$$ER = 0.81$$

Obteniéndose así, un resultado final satisfactorio, con un grado de ambigüedad de los requisitos bajo (19%) ya que el 81% son entendibles. Los requisitos ambiguos fueron modificados y validados para garantizar una correcta interpretación.

3.4. PRUEBAS DE SOFTWARE.

Método de caja negra.

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra no son una alternativa para las técnicas de caja blanca. En vez de ello, es un enfoque complementario que es probable que descubra una clase de errores diferentes que los métodos de caja blanca (S. Pressman y B. R. Maxim 2015).

Las pruebas de caja negra intentan encontrar errores en las categorías siguientes:

- Funcionalidades incorrectas o faltantes.
- Errores de interfaz.
- Errores en las estructuras de datos o en el acceso a bases de datos externas.
- Errores de comportamiento o rendimiento
- Errores de inicialización y terminación.

Método de caja blanca:

La prueba de caja blanca, en ocasiones llamada prueba de caja de vidrio, es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de prueba.(S. Pressman y B. R. Maxim 2015)

CAPÍTULO 3. VALIDACIÓN DEL SISTEMA DE GESTIÓN DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

Al usar los métodos de prueba de caja blanca, se pueden derivar casos de prueba que:

- Garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez.
- Revisen todas las decisiones lógicas en sus lados verdadero y falso.
- Ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas.
- Revisen estructuras de datos internas para garantizar su validez.

3.4.2 ESTRATEGIA DE PRUEBA.

Para evaluar la calidad del componente que se está desarrollando y verificar el cumplimiento de los objetivos trazados, se diseña la siguiente estrategia de prueba para aplicar a la solución desarrollada:

Prueba	Método	Técnica
Prueba de unidad	Caja blanca	Camino básico
Prueba de integración	Caja negra	Incremental
Prueba de sistema: Prueba de rendimiento	Caja negra	Semiautomático

Tabla 6: Pruebas aplicadas a la aplicación.

3.4.3 PRUEBA DE UNIDAD, CAMINO BÁSICO.

La prueba de unidad enfoca los esfuerzos de verificación en la unidad más pequeña del diseño de software: el componente o módulo de software. Al usar la descripción del diseño de componente como guía, las rutas de control importantes se prueban para descubrir errores dentro de la frontera del módulo. La relativa complejidad de las pruebas y los errores que descubren están limitados por el ámbito restringido que se establece para la prueba de unidad. Las pruebas de unidad se enfocan en la lógica de procesamiento interno y de las estructuras de datos dentro de las fronteras de un componente (S. Pressman y B. R. Maxim 2015).

Para la realización de la prueba se utiliza la técnica de camino básico, la cual permite obtener una medida de la complejidad lógica del diseño procedimental y usar esa medida como guía para la

CAPÍTULO 3. VALIDACIÓN DEL SISTEMA DE GESTIÓN DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

definición de un conjunto básico de caminos de ejecución, garantizando con estos que durante la prueba se ejecute por lo menos una vez cada sentencia del programa.

A continuación, se muestra el caso de prueba del método `get(id)` perteneciente a la clase **health-promoter.controller.ts**, como resultado la funcionalidad se espera obtener a partir de la id proporcionada, un promotor con todos los datos correspondientes existentes en el sistema.

```
113  ..loadPagePatients(page: number)·{  
115  ...this._patientService  
116  ...  .getPage(page, this.filterPatientForm.value)  
117  ...  .subscribe((resp: any) => { 1  
118  ...    this.patients = resp.items.map((item: any) => { 2  
119  ...      item.inclusionDate = (item.inclusionDate as string).split('T')[0];  
120  ...      item.interruptionDate = (item.interruptionDate as string).split(  
121  ...        'T',  
122  ...      )[0];  
123  ...      item.state = item.state === 'alive' ? 'Si' : 'No'; 3  
124  ...      switch (item.sex as string) { 4  
125  ...        case 'male': 5  
126  ...          item.sex = 'M'; 6  
127  ...          break;  
128  ...        case 'female': 7  
129  ...          item.sex = 'F'; 8  
130  ...          break;  
131  ...        default:  
132  ...          item.sex = 'Other';  
133  ...      }  
134  ...      return item;  
135  ...    });  
136  ...    this.page = resp.currentPage;  
137  ...    this.totalItems = resp.total;  
138  ...  });  
139  }
```

Figura 15: Método `loadPagePatients(number)`

CAPÍTULO 3. VALIDACIÓN DEL SISTEMA DE GESTIÓN DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

Prueba estructural de caja blanca	Código de caso de prueba
Método	get(id)
Probador	Rubén Álvarez Izquierdo
Complejidad ciclomática: $V(G)=E-N+2$ $V(G)=9-8+2$ $V(G) = 3$	
Caminos independientes	1,2,3,4,5,8 1,2,3,4,7,8 1,2,3,4,6,8

Tabla 7: Prueba unitaria.

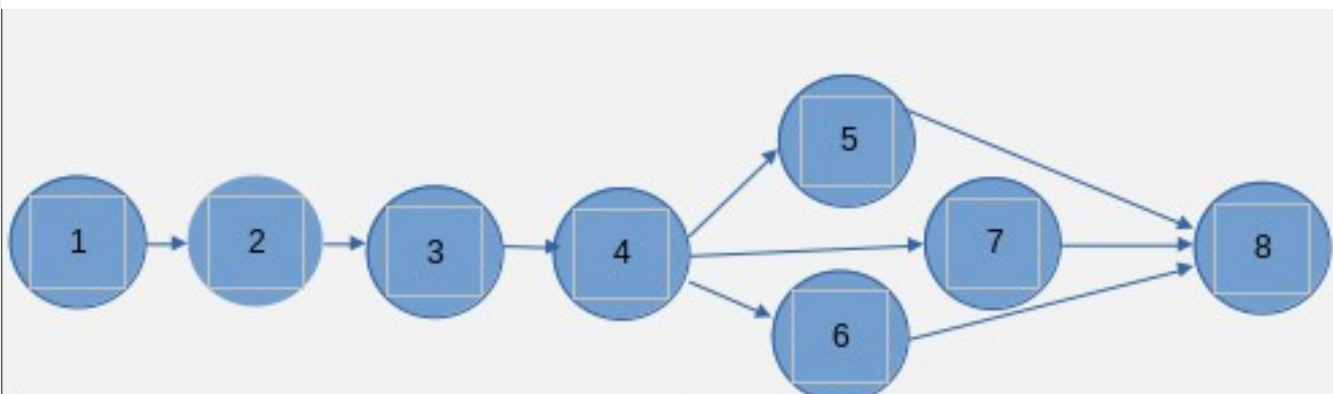


Figura 16: Grafo representativo.

3.4.4 PRUEBA DE INTEGRACIÓN.

Las pruebas de integración son una técnica sistemática para construir la arquitectura del software y, al mismo tiempo, realizar pruebas para descubrir errores asociados con la interconexión.

CAPÍTULO 3. VALIDACIÓN DEL SISTEMA DE GESTIÓN DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

El objetivo es tomar componentes probados por unidades y construir la estructura del programa que se diseñó (S. Pressman y B. R. Maxim 2015).

En el desarrollo de esta prueba se utiliza la técnica de integración incremental. El programa se construye y se prueba en pequeños incrementos, donde los errores son más fáciles de aislar y corregir. Es más probable que las interfaces se prueben completamente y se puede aplicar un enfoque de prueba sistemático (S. Pressman y B. R. Maxim 2015). Se emplea el método de integración incremental por las facilidades que brinda al desarrollador de realizar pruebas al programa menos complejas y obteniendo mejores resultados.

Dentro de la técnica de integración incremental se utilizó la estrategia de integración top-down (descendente). La prueba de integración descendente es un enfoque incremental para la construcción de la arquitectura del software. Los módulos se integran moviéndose hacia abajo a través de la jerarquía de control, comenzando con el módulo de control principal (programa principal). Los módulos subordinados al módulo de control principal se incorporan a la estructura ya sea en profundidad o primero en amplitud. El proceso de integración se realiza en una serie de cinco pasos (S. Pressman y B. R. Maxim 2015):

1. El módulo de control principal se utiliza como controlador de prueba y los apéndices se sustituyen por todos los componentes directamente subordinados al módulo de control principal.
2. Dependiendo del enfoque de integración seleccionado (es decir, primero en profundidad o primero en amplitud), los apéndices subordinados se reemplazan uno a la vez con componentes reales.
3. Las pruebas se realizan a medida que se integra cada componente.
4. Al completar cada conjunto de pruebas, se reemplaza otro apéndice con el componente real.
5. Se pueden realizar pruebas de regresión para garantizar que no se hayan introducido nuevos errores.

El proceso continúa desde el paso 2 hasta que se construye toda la estructura del programa.

3.4.4.1 DETALLES DE LA PRUEBA.

La prueba se encuentra dividida en tres fases.

CAPÍTULO 3. VALIDACIÓN DEL SISTEMA DE GESTIÓN DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

1- **Interacción de usuario:** en esta fase se revisan los datos de entrada y salida, el formato de presentación de los datos, el procesamiento de errores y la presentación de los mismos.

2- **Comunicación:** en esta fase se revisa la capacidad del componente para comunicarse con la Plataforma “Xavia Sidec”.

3- **Procesamiento de eventos:** en esta fase se revisan las acciones del componente en el procesamiento de los eventos”.

3.4.4.2 EJEMPLO DE PRUEBA DE INTEGRACIÓN

Se aplica la prueba de integración al componente *“HealthPromoterListComponent”*, obteniéndose los resultados que a continuación se muestran.

Sistema/componente que se integra	al Xavia Sidec
Condiciones de ejecución	Hay un usuario autenticado en Xavia Sidec
Descripción de la prueba	Comprobar que el componente <i>“HealthPromoterListComponent”</i> proporcione la información referente a los promotores registrados en la base de datos de Xavia Sitec.
Pasos de ejecución	El probador interactúa con el componente. El probador comprueba que el componente muestre todos los Promotores existentes en la base de datos, con su correspondiente información. Verificar que el usuario que se utiliza para la realización de la prueba aparezca en la cabecera de la ventana como el usuario conectado
Resultados esperados	El componente <i>“Mostrar lista de usuarios”</i> brinda toda la información solicitada.
Evaluación	

Tabla 8: caso de prueba de integración del componente *“HealthPromoterListComponent”*.

3.4.5 PRUEBA DE SISTEMA, PRUEBA DE RENDIMIENTO.

La prueba del rendimiento ocurre a lo largo de todos los pasos del proceso de prueba. Incluso en el nivel de unidad, puede accederse al rendimiento de un módulo individual conforme se realizan

CAPÍTULO 3. VALIDACIÓN DEL SISTEMA DE GESTIÓN DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

las pruebas. Sin embargo, no es sino hasta que todos los elementos del sistema están plenamente integrados cuando puede determinarse el verdadero rendimiento de un sistema (Pressman sin fecha).

Las pruebas de rendimiento por lo general requieren instrumentación de hardware y de software, es decir, con frecuencia es necesario medir la utilización de los recursos (por ejemplo, ciclos del procesador) en forma meticulosa. La instrumentación externa puede monitorear intervalos de ejecución y eventos de registro (por ejemplo, interrupciones) conforme ocurren, y los muestreos del estado de la máquina de manera regular. Con la instrumentación de un sistema, la persona que realiza la prueba puede descubrir situaciones que conduzcan a degradación y posibles fallas del sistema (Pressman sin fecha).

Pruebas de carga

Mediante la ejecución de las pruebas de Carga es posible identificar la capacidad de recuperación de un sistema cuando es sometido a cargas variables tanto de usuarios como de procesos. Al realizar las pruebas de carga se puede determinar el tiempo de respuesta de todas las transacciones críticas del sistema y encontrar cuellos de botella de la misma (Pressman sin fecha).

Pruebas de estrés

Mediante las pruebas de estrés es posible identificar la capacidad de respuesta de un sistema bajo condiciones de carga extrema, representadas por una alta concurrencia de Usuarios y/o procesos, una vez realizadas las pruebas de estrés se podrá conocer el punto de quiebre del aplicativo en términos de capacidad de respuesta, con lo cual será posible establecer acciones de optimización en diferentes niveles para asegurar una mejor capacidad de concurrencia de usuarios y/o procesos que se verá reflejada en una óptima operación de negocio (Pressman sin fecha).

Resultados de las pruebas de rendimiento

Para la realización de las pruebas de carga y estrés al sistema Diagnóstico se utilizó el software Apache JMeter en su versión 2.13.19.

Hardware de prueba (PC cliente):

- Tipo de procesador: Intel(R) Celeron(R) 2 Duo CPU @ 1.6GHz.

CAPÍTULO 3. VALIDACIÓN DEL SISTEMA DE GESTIÓN DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

- RAM: 2 GB DDR2.
- Tipo de Red: Ethernet 10/100Mbps.

Hardware de prueba (PC servidor):

- Tipo de procesador: Intel(R) Celeron(R) 2 Duo CPU @ 1.6GHz.
- RAM: 2 GB DDR2.
- Tipo de Red: Ethernet 10/100Mbps.

Software instalado en ambas PC:

- Tipo de servidor web: Node 20.5.0 y docker 24.0.6.
- Plataforma: Linux Lite 6.6 (PC cliente y servidor)
- Servidor de BD: MySQL v5.1.38.

La prueba define 250 hilos de concurrencia, los cuales simulan 250 usuarios accediendo concurrentemente. Se simularon un total de mil peticiones a cinco direcciones del módulo en el servidor. En la Tabla se puede observar los resultados obtenidos por el sistema.

Descripción de los parámetros evaluados:

Muestras: cantidad de hilos utilizados para la URL.

Media: tiempo promedio en milisegundos para un conjunto de resultados.

Min: tiempo mínimo que demora un hilo en acceder a una página.

Max: tiempo máximo que demora un hilo en acceder a una página.

%Error: porcentaje de error de las respuestas de las peticiones.

Rendimiento: rendimiento medido en los requerimientos por segundo / minuto / hora.

Kb/s Recibidos: rendimiento medido en Kbyte por segundo

CAPÍTULO 3. VALIDACIÓN DEL SISTEMA DE GESTIÓN DE PACIENTES INCLUIDOS EN ENSAYOS CLÍNICOS.

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Media de Bytes
inicio	250	91	3	895	170,90	0,00%	13,2/sec	11,35	881,0
Paciente	250	100	2	891	164,95	0,00%	13,8/sec	11,87	881,0
ensayos clini...	250	68	2	710	99,70	0,00%	14,0/sec	12,05	881,0
promotor	250	63	3	698	92,33	0,00%	13,8/sec	11,90	881,0
Total	1000	81	2	895	137,69	0,00%	49,9/sec	42,90	881,0

Figura 17: Resultado de jmeter.

3.5. CONCLUSIONES DEL CAPÍTULO.

En este capítulo, se ha llevado a cabo la implementación del Subsistema de registro de pacientes incluidos en ensayos clínicos dentro del Sistema Informatizado de Salud, logrando obtener un producto que cumple con todas las funcionalidades solicitadas. Además, se ha destacado la relación existente entre todos los nodos que componen el sistema, así como su conexión con la base de datos mediante el uso del Diagrama de Componentes. Este enfoque permite dividir el sistema en partes más comprensibles para los programadores. Como resultado, se proporciona a los usuarios del sistema una solución funcional que permite una fácil interacción para la visualización de la información.

CONCLUSIONES GENERALES

Una vez culminado el presente trabajo de diploma se puede concluir que se le dio cumplimiento al objetivo general de la investigación, resaltando que:

- La elaboración del marco teórico conceptual de la investigación permitió sentar las bases de la misma.
- La especificación de los requisitos, así como el análisis y diseño del sistema para la gestión de Ensayos clínicos, permitió definir los elementos necesarios de la implementación del mismo.
- El empleo de las herramientas y tecnologías seleccionadas para la implementación de la solución, propició la correspondencia entre los resultados obtenidos y los esperados.
- lo cual aseguró el nivel de precisión en el análisis y diseño de la aplicación.
- El empleo de la metodología AUP-UCI en su escenario 4, permitió que se trazara una buena planificación de todo el proyecto, ahorrando tiempo y recursos.
- El uso de patrones GRASP y los patrones de diseño GOF identificados se cohesionan al estilo arquitectónico modelo-vista-controlador agregando a la solución un alto grado de flexibilidad ante posibles modificaciones.
- La realización de pruebas de software permitió comprobar el correcto funcionamiento del sistema. Estas pruebas arrojaron resultados satisfactorios en relación al código y el conjunto de interfaces implementadas.

REFERENCIAS BIBLIOGRÁFICAS

Acerca (no date) Node.js. Available at: <https://nodejs.org/es/about> (Accessed: 17 September 2023).

AngularJS: marco MVW de JavaScript superherico (no date). Available at: <https://angularjs.org/> (Accessed: 28 September 2023).

Cortiñas Martínez, O. (2017) Informatización del proceso de tramitación de pasaportes corrientes biométricos a partir de solicitudes consulares. bachelorThesis. Universidad de las Ciencias Informáticas. Facultad 1. Available at: <https://repositorio.uci.cu/jspui/handle/123456789/8055> (Accessed: 17 September 2023).

CSS | MDN (2023). Available at: <https://developer.mozilla.org/es/docs/Web/CSS> (Accessed: 17 September 2023).

Express: marco de aplicación web Node.js (no date). Available at: <https://expressjs.com/> (Accessed: 17 September 2023).

García, L.R. et al. (2012) “ALASCLÍNICAS”: Sistema de gestión de ensayos clínicos’, Revista Cubana de Informática Médica, 4(1). Available at: <https://revinformatica.sld.cu/index.php/rcim/article/view/185> (Accessed: 17 September 2023).

Herramienta ideal de modelado y diagramación para una colaboración ágil en equipo (no date). Available at: <https://www.visual-paradigm.com/> (Accessed: 17 September 2023).

HTML: Lenguaje de etiquetas de hipertexto | MDN (2023). Available at: <https://developer.mozilla.org/es/docs/Web/HTML> (Accessed: 17 September 2023).

JavaScript | MDN (2023). Available at: <https://developer.mozilla.org/es/docs/Web/JavaScript> (Accessed: 17 September 2023).

La gestión clínica: ¿qué es y para qué sirve? | Conexión ESAN (no date). Available at: <https://www.esan.edu.pe/conexion-esan/la-gestion-clinica-que-es-y-para-que-sirve> (Accessed: 28 September 2023).

La gestión de pacientes: el servicio de admisión y documentación (no date). Available at: <https://www.ilerna.es/blog/fp-online/gestion-de-pacientes-admision-y-documentacion/> (Accessed: 28 September 2023).

López, M.A.P. et al. (2011) ‘Surgimiento, evolución y principales resultados del Centro Nacional Coordinador de Ensayos Clínicos’, Revista Cubana de Farmacia, 45(1), pp. 4–18.

Medidata | Unified Life Science Platform | Medidata Solutions (no date). Available at: <https://www.medidata.com/> (Accessed: 17 September 2023).

Metodología de Desarrollo de Software Variación de AUP para la UCI - EcuRed (no date). Available at: <https://www.ecured.cu/Metodolog>

REFERENCIAS BIBLIOGRÁFICAS

%C3%ADa_de_Desarrollo_de_Software_Variaci%C3%B3n_de_AUP_para_la_UCI (Accessed: 17 September 2023).

Openclinica from Akaza Research. (no date) ResearchGate. Available at: https://www.researchgate.net/figure/Openclinica-from-Akaza-Research_fig2_51200568 (Accessed: 17 September 2023).

¿Qué es UML (no date). Available at: <http://profesores.fi-b.unam.mx/carlos/aydoo/uml.html> (Accessed: 17 September 2023).

SIDEC 3.0 | Universidad de las Ciencias Informáticas (no date). Available at: <https://www.uci.cu/investigacion-y-desarrollo/productos/xavia/sidec-30> (Accessed: 17 September 2023).

Softel | Soluciones Informáticas (no date). Available at: <https://www.softel.cu/solucionesInformaticasPortal/show/?id=40289082529dc80501529dd9ff420011> (Accessed: 17 September 2023).

Acerca, sin fecha Node.js [en línea]. Recuperado a partir de: <https://nodejs.org/es/about> [accedido 17 septiembre 2023].

Angular JS: marco MVW de JavaScript superheróico, sin fecha [en línea]. Recuperado a partir de: <https://angularjs.org/> [accedido 28 septiembre 2023].

CORTIÑAS MARTÍNEZ, Osniel, 2017. Informatización del proceso de tramitación de pasaportes corrientes biométricos a partir de solicitudes consulares [en línea]. bachelorThesis . Universidad de las Ciencias Informáticas. Facultad 1. Recuperado a partir de : <https://repositorio.uci.cu/jspui/handle/123456789/8055> [accedido 17 septiembre 2023]. Accepted: 2019-11-20T15:48:36Z

CSS | MDN, 2023 [en línea]. Recuperado a partir de: <https://developer.mozilla.org/es/docs/Web/CSS> [accedido 17 septiembre 2023].

Express: marco de aplicación web Node.js, sin fecha [en línea]. Recuperado a partir de : <https://expressjs.com/> [accedido 17 septiembre 2023].

GARCÍA, Lucía Rodríguez et al., 2012. «ALASCLÍNICAS»: Sistema de gestión de ensayos clínicos. Revista Cubana de Informática Médica [en línea]. Vol. 4, número 1. Recuperado a partir de : <https://revinformatica.sld.cu/index.php/rcim/article/view/185> [accedido 17 septiembre 2023].

Herramienta ideal de modelado y diagramación para una colaboración ágil en equipo, sin fecha [en línea]. Recuperado a partir de : <https://www.visual-paradigm.com/> [accedido 17 septiembre 2023].

HTML: Lenguaje de etiquetas de hipertexto | MDN, 2023 [en línea]. Recuperado a partir de: <https://developer.mozilla.org/es/docs/Web/HTML> [accedido 17 septiembre 2023].

JavaScript | MDN, 2023 [en línea]. Recuperado a partir de: <https://developer.mozilla.org/es/docs/Web/JavaScript> [accedido 17 septiembre 2023].

REFERENCIAS BIBLIOGRÁFICAS

La gestión clínica: ¿qué es y para qué sirve? | Conexión ESAN, sin fecha [en línea]. Recuperado a partir de : <https://www.esan.edu.pe/conexion-esan/la-gestion-clinica-que-es-y-para-que-sirve> [accedido 28 septiembre 2023].

La gestión de pacientes: el servicio de admisión y documentación, sin fecha [en línea]. Recuperado a partir de: <https://www.ilerna.es/blog/fp-online/gestion-de-pacientes-admision-y-documentacion/> [accedido 28 septiembre 2023].

LÓPEZ, María Amparo Pascual et al., 2011. Surgimiento, evolución y principales resultados del Centro Nacional Coordinador de Ensayos Clínicos. Revista Cubana de Farmacia. Vol. 45, número 1, pp. 4-18.

Medidata | Unified Life Science Platform | Medidata Solutions, sin fecha [en línea]. Recuperado a partir de: <https://www.medidata.com/> [accedido 17 septiembre 2023].

Metodología de Desarrollo de Software Variación de AUP para la UCI - EcuRed, sin fecha [en línea]. Recuperado a partir de : <https://www.ecured.cu/Metodologia%20de%20Desarrollo%20de%20Software%20Variacion%20de%20AUP%20para%20la%20UCI> [accedido 17 septiembre 2023].

Openclinica from Akaza Research., sin fechaResearchGate [en línea]. Recuperado a partir de: https://www.researchgate.net/figure/Openclinica-from-Akaza-Research_fig2_51200568 [accedido 17 septiembre 2023].

Patrones Grasp (Craig Larman) Parte I | El Mundo Informático, sin fecha [en línea]. Recuperado a partir de : <https://jorgesaavedra.wordpress.com/2006/08/17/patrones-grasp-craig-larman/> [accedido 3 octubre 2023].

PRESSMAN, Roger S, sin fecha. Ingeniería del Software. Un Enfoque Practico. .

¿Qué es UML, sin fecha [en línea]. Recuperado a partir de: <http://profesores.fi-b.unam.mx/carlos/aydoo/uml.html> [accedido 17 septiembre 2023].

SIDEC 3.0 | Universidad de las Ciencias Informáticas, sin fecha [en línea]. Recuperado a partir de : <https://www.uci.cu/investigacion-y-desarrollo/productos/xavia/sidec-30> [accedido 17 septiembre 2023].

Softel | Soluciones Informáticas, sin fecha [en línea]. Recuperado a partir de : <https://www.softel.cu/solucionesInformaticasPortal/show/?id=40289082529dc80501529dd9ff420011> [accedido 17 septiembre 2023].

Acerca, sin fechaNode.js [en línea]. Recuperado a partir de: <https://nodejs.org/es/about> [accedido 17 septiembre 2023].

Angular JS: marco MVW de JavaScript superheróico, sin fecha [en línea]. Recuperado a partir de: <https://angularjs.org/> [accedido 28 septiembre 2023].

CORTIÑAS MARTÍNEZ, Osniel, 2017. Informatización del proceso de tramitación de pasaportes corrientes biométricos a partir de solicitudes consulares [en línea]. bachelorThesis . Universidad de las Ciencias Informáticas. Facultad 1. Recuperado a partir de : <https://repositorio.uci.cu/jspui/handle/123456789/8055> [accedido 17 septiembre 2023]. Accepted: 2019-11-20T15:48:36Z

REFERENCIAS BIBLIOGRÁFICAS

CRAIG LARMAN, sin fecha. Uml Y Patrones [en línea]. Recuperado a partir de : <http://archive.org/details/larmanumlypatrones2daedicion> [accedido 4 octubre 2023].

CSS | MDN, 2023 [en línea]. Recuperado a partir de : <https://developer.mozilla.org/es/docs/Web/CSS> [accedido 17 septiembre 2023].

Express: marco de aplicación web Node.js, sin fecha [en línea]. Recuperado a partir de : <https://expressjs.com/> [accedido 17 septiembre 2023].

GARCÍA FRANCO, VILMA, GARCÍA NÚÑEZ, RUBÉN DARÍO, LORENZO GONZÁLEZ, MARISELA, HERNÁNDEZ CABEZAS, MARILYS, GARCÍA FRANCO, VILMA, GARCÍA NÚÑEZ, RUBÉN DARÍO, LORENZO GONZÁLEZ, MARISELA Y HERNÁNDEZ CABEZAS, MARILYS, 2020. Los mapas conceptuales como instrumentos útiles en el proceso enseñanza-aprendizaje. MediSur. en línea. Vol. 18, número 6, pp. 1154-1162.

GARCÍA, Lucía Rodríguez et al., 2012. «ALASCLÍNICAS»: Sistema de gestión de ensayos clínicos. Revista Cubana de Informática Médica [en línea]. Vol. 4, número 1. Recuperado a partir de : <https://revinformatica.sld.cu/index.php/rcim/article/view/185> [accedido 17 septiembre 2023].

Herramienta ideal de modelado y diagramación para una colaboración ágil en equipo, sin fecha [en línea]. Recuperado a partir de: <https://www.visual-paradigm.com/> [accedido 17 septiembre 2023].

HTML: Lenguaje de etiquetas de hipertexto | MDN, 2023 [en línea]. Recuperado a partir de: <https://developer.mozilla.org/es/docs/Web/HTML> [accedido 17 septiembre 2023].

I. SOMMERVILLE, 2016. Software engineering. Boston: Pearson.

JavaScript | MDN, 2023 [en línea]. Recuperado a partir de: <https://developer.mozilla.org/es/docs/Web/JavaScript> [accedido 17 septiembre 2023].

La gestión clínica: ¿qué es y para qué sirve? | Conexión ESAN, sin fecha [en línea]. Recuperado a partir de : <https://www.esan.edu.pe/conexion-esan/la-gestion-clinica-que-es-y-para-que-sirve> [accedido 28 septiembre 2023].

La gestión de pacientes: el servicio de admisión y documentación, sin fecha [en línea]. Recuperado a partir de : <https://www.ilerna.es/blog/fp-online/gestion-de-pacientes-admision-y-documentacion/> [accedido 28 septiembre 2023].

LÓPEZ, María Amparo Pascual et al., 2011. Surgimiento, evolución y principales resultados del Centro Nacional Coordinador de Ensayos Clínicos. Revista Cubana de Farmacia. Vol. 45, número 1, pp. 4-18.

Medidata | Unified Life Science Platform | Medidata Solutions, sin fecha [en línea]. Recuperado a partir de: <https://www.medidata.com/> [accedido 17 septiembre 2023].

Metodología de Desarrollo de Software Variación de AUP para la UCI - EcuRed, sin fecha [en línea]. Recuperado a partir de : https://www.ecured.cu/Metodolog%C3%ADa_de_Desarrollo_de_Software_Variaci%C3%B3n_de_AUP_para_la_UCI [accedido 17 septiembre 2023].

REFERENCIAS BIBLIOGRÁFICAS

Openclinica from Akaza Research. sin fecha ResearchGate [en línea]. Recuperado a partir de: https://www.researchgate.net/figure/Openclinica-from-Akaza-Research_fig2_51200568 [accedido 17 septiembre 2023].

Patrones Grasp (Craig Larman) Parte I | El Mundo Informático, sin fecha [en línea]. Recuperado a partir de: <https://jorgesaavedra.wordpress.com/2006/08/17/patrones-grasp-craig-larman/> [accedido 3 octubre 2023].

PRESSMAN, Roger S, sin fecha. Ingeniería del Software. Un Enfoque Practico. .

¿Qué es UML, sin fecha [en línea]?. Recuperado a partir de: <http://profesores.fi-b.unam.mx/carlos/aydoo/uml.html> [accedido 17 septiembre 2023].

S. PRESSMAN Y B. R. MAXIM, 2015. Software Engineering: A Practitioner's Approach. McGraw Hill Education.

SIDEC 3.0 | Universidad de las Ciencias Informáticas, sin fecha [en línea]. Recuperado a partir de: <https://www.uci.cu/investigacion-y-desarrollo/productos/xavia/sidec-30> [accedido 17 septiembre 2023].

Softel | Soluciones Informáticas, sin fecha [en línea]. Recuperado a partir de: <https://www.softel.cu/solucionesInformaticasPortal/show/?id=40289082529dc80501529dd9ff420011> [accedido 17 septiembre 2023].

Acerca, sin fecha Node.js [en línea]. Recuperado a partir de : <https://nodejs.org/es/about> [accedido 17 septiembre 2023].

AngularJS: marco MVW de JavaScript superherico, sin fecha [en línea]. Recuperado a partir de : <https://angularjs.org/> [accedido 28 septiembre 2023].

Axure, poderosa herramienta en el Diseño UX, sin fecha [en línea]. Recuperado a partir de : <https://esdima.com/axure-poderosa-herramienta-en-el-diseno-ux/> [accedido 14 octubre 2023].

CORTIÑAS MARTÍNEZ, Osniel, 2017. Informatización del proceso de tramitación de pasaportes corrientes biométricos a partir de solicitudes consulares [en línea]. bachelorThesis . Universidad de las Ciencias Informáticas. Facultad 1. Recuperado a partir de : <https://repositorio.uci.cu/jspui/handle/123456789/8055> [accedido 17 septiembre 2023]. Accepted: 2019-11-20T15:48:36Z

CRAIG LARMAN, sin fecha. Uml Y Patrones [en línea]. Recuperado a partir de : <http://archive.org/details/larmanumlypatrones2daedicion> [accedido 4 octubre 2023].

CSS | MDN, 2023 [en línea]. Recuperado a partir de : <https://developer.mozilla.org/es/docs/Web/CSS> [accedido 17 septiembre 2023].

Docker: desarrollo acelerado de aplicaciones de contenedores, 2022 [en línea]. Recuperado a partir de : <https://www.docker.com/> [accedido 13 octubre 2023].

Ensayos clínicos - Cómo funcionan los ensayos clínicos | NHLBI, NIH, sin fecha [en línea]. Recuperado a partir de : <https://www.nhlbi.nih.gov/es/investigacion/ensayos-clinicos/como-funcionan> [accedido 5 noviembre 2023].

REFERENCIAS BIBLIOGRÁFICAS

GARCÍA FRANCO, VILMA, GARCÍA NÚÑEZ, RUBÉN DARÍO, LORENZO GONZÁLEZ, MARISELA, HERNÁNDEZ CABEZAS, MARILYS, GARCÍA FRANCO, VILMA, GARCÍA NÚÑEZ, RUBÉN DARÍO, LORENZO GONZÁLEZ, MARISELA Y HERNÁNDEZ CABEZAS, MARILYS, 2020. Los mapas conceptuales como instrumentos útiles en el proceso enseñanza-aprendizaje. *MediSur*. en línea. Vol. 18, número 6, pp. 1154-1162.

GARCÍA, Lucía Rodríguez et al., 2012. «ALASCLÍNICAS»: Sistema de gestión de ensayos clínicos. *Revista Cubana de Informática Médica* [en línea]. Vol. 4, número 1. Recuperado a partir de : <https://revinformatica.sld.cu/index.php/rcim/article/view/185> [accedido 17 septiembre 2023].

GEEKTRAINER, sin fecha. Introducción - Training. [en línea]. Recuperado a partir de : <https://learn.microsoft.com/es-es/training/modules/typescript-namespaces-organize-code/1-introduction> [accedido 13 octubre 2023].

GitHub - nestjs/nest: un marco progresivo de Node.js para crear aplicaciones de servidor eficientes, escalables y de nivel empresarial con TypeScript/JavaScript , sin fecha [en línea]. Recuperado a partir de : <https://github.com/nestjs/nest> [accedido 13 octubre 2023].

Herramienta ideal de modelado y diagramación para una colaboración ágil en equipo, sin fecha [en línea]. Recuperado a partir de : <https://www.visual-paradigm.com/> [accedido 17 septiembre 2023].

HTML: Lenguaje de etiquetas de hipertexto | MDN, 2023 [en línea]. Recuperado a partir de : <https://developer.mozilla.org/es/docs/Web/HTML> [accedido 17 septiembre 2023].

I. SOMMERVILLE, 2016. *Software engineering*. Boston: Pearson.

JavaScript | MDN, 2023 [en línea]. Recuperado a partir de : <https://developer.mozilla.org/es/docs/Web/JavaScript> [accedido 17 septiembre 2023].

La gestión clínica: ¿qué es y para qué sirve? | Conexión ESAN, sin fecha [en línea]. Recuperado a partir de : <https://www.esan.edu.pe/conexion-esan/la-gestion-clinica-que-es-y-para-que-sirve> [accedido 28 septiembre 2023].

La gestión de pacientes: el servicio de admisión y documentación, sin fecha [en línea]. Recuperado a partir de : <https://www.ilerna.es/blog/fp-online/gestion-de-pacientes-admision-y-documentacion/> [accedido 28 septiembre 2023].

LÓPEZ, María Amparo Pascual et al., 2011. Surgimiento, evolución y principales resultados del Centro Nacional Coordinador de Ensayos Clínicos. *Revista Cubana de Farmacia*. Vol. 45, número 1, pp. 4-18.

Mecanografiado [logiciel] [en línea]. 14 octubre 2023. Microsoft. [accedido 14 octubre 2023]. Recuperado a partir de : <https://github.com/microsoft/TypeScript> [accedido 14 octubre 2023].

Medidata | Unified Life Science Platform | Medidata Solutions, sin fecha [en línea]. Recuperado a partir de : <https://www.medidata.com/> [accedido 17 septiembre 2023].

REFERENCIAS BIBLIOGRÁFICAS

Metodología de Desarrollo de Software Variación de AUP para la UCI - EcuRed, sin fecha [en línea]. Recuperado a partir de : https://www.ecured.cu/Metodolog%C3%ADa_de_Desarrollo_de_Software_Variaci%C3%B3n_de_AUP_para_la_UCI [accedido 17 septiembre 2023].

Openclinica from Akaza Research., sin fecha *ResearchGate* [en línea]. Recuperado a partir de : https://www.researchgate.net/figure/Openclinica-from-Akaza-Research_fig2_51200568 [accedido 17 septiembre 2023].

Patrones Grasp (Craig Larman) Parte I | El Mundo Informático, sin fecha [en línea]. Recuperado a partir de : <https://jorgesaavedra.wordpress.com/2006/08/17/patrones-grasp-craig-larman/> [accedido 3 octubre 2023].

PostgreSQL: Acerca de, sin fecha [en línea]. Recuperado a partir de : <https://www.postgresql.org/about/> [accedido 5 noviembre 2023].

PRESSMAN, Roger S, sin fecha. *Ingeniería del Software. Un Enfoque Práctico*. .

¿Qué es UML, sin fecha [en línea]. Recuperado a partir de : <http://profesores.fi-b.unam.mx/carlos/aydoo/uml.html> [accedido 17 septiembre 2023].

RODRÍGUEZ GARCÍA, Lucía et al., 2012. *ALASCLÍNICAS: Sistema de gestión de ensayos clínicos*. *Revista Cubana de Informática Médica*. Vol. 4, número 1, pp. 5-16.

S. PRESSMAN Y B. R. MAXIM, 2015. *Software Engineering: A Practitioner's Approach*. McGraw Hill Education.

SIDEC 3.0 | Universidad de las Ciencias Informáticas, sin fecha [en línea]. Recuperado a partir de : <https://www.uci.cu/investigacion-y-desarrollo/productos/xavia/sidec-30> [accedido 17 septiembre 2023].

Softel | Soluciones Informáticas, sin fecha [en línea]. Recuperado a partir de : <https://www.softel.cu/solucionesInformaticasPortal/show?id=40289082529dc80501529dd9ff420011> [accedido 17 septiembre 2023].

Visual Studio Code - Code Editing. Redefined, sin fecha [en línea]. Recuperado a partir de : <https://code.visualstudio.com/> [accedido 17 septiembre 2023].

ANEXOS

Historia de Usuario	
Número: 1	Rol: Usuario
Título: Buscar cantidad de pacientes incluidos por año de inclusión	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Tiempo estimado: 4 Horas	Iteración Asignada: 1
Programador Responsable: Rubén Álvarez Izquierdo.	
Descripción: El sistema permitirá al usuario buscar la cantidad de pacientes por año de inclusión en uno de los informes.	
No	Criterio de Aceptación
1	El usuario debe tener asignado los permisos para trabajar con la pestaña informes. Debe haberse autenticado en el sistema y haber accedido a la funcionalidad Informes Cantidad de pacientes incluidos por años de inclusión.
2	EL usuario debe ubicar el año, por el cuál desea filtrar la búsqueda y presionar el botón Imprimir.
3	El sistema debe ser capaz de devolver la cantidad de pacientes incluidos por año de inclusión, a través de la suma de todos los identificadores de pacientes incluidos en el año de inclusión por el cual se filtró la búsqueda, en el componente Paciente Incluido en Ensayos Clínicos.
Prototipo elemental de interfaz gráfica de usuario:	

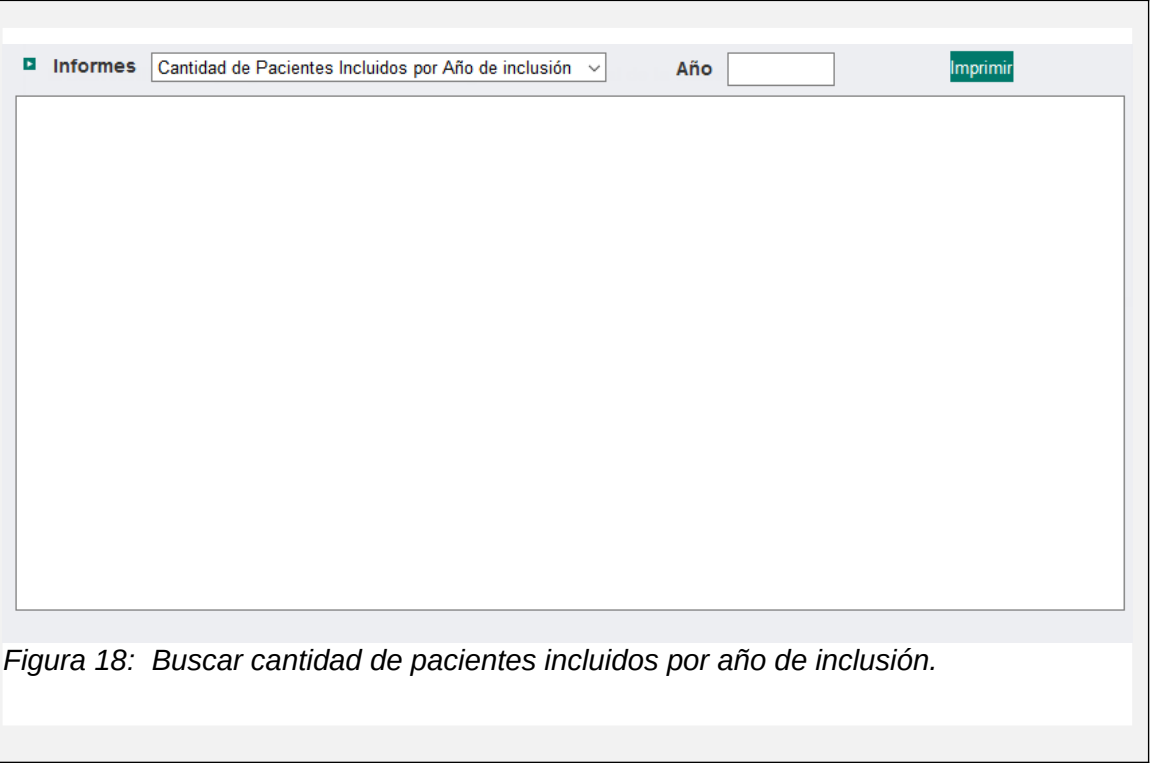


Tabla 9: HU Buscar cantidad de pacientes incluidos por año de inclusión.

Historia de Usuario	
Número: 1	Rol: Usuario
Título: Buscar cantidad de pacientes incluidos por ensayos clínicos	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Tiempo estimado: 4 Horas	Iteración Asignada: 1
Programador Responsable: Rubén Álvarez Izquierdo y Michel Leandro Pérez Somoza.	
Descripción: El sistema permitirá al usuario buscar la cantidad de pacientes incluidos por ensayos clínicos a partir de uno de los informes.	
No	Criterio de Aceptación
1	El usuario debe tener asignado los permisos para trabajar con la pestaña informes. Debe haberse autenticado en el sistema y haber accedido a la funcionalidad Cantidad de pacientes incluidos por ensayos clínicos.
2	EL usuario debe ubicar el año, por el cuál desea filtrar la búsqueda y presionar el botón Imprimir
3	El sistema debe ser capaz de devolver la cantidad de pacientes incluidos por ensayos clínicos y agruparlo por ensayo clínico, a través de la suma de todos los identificadores de pacientes incluidos en un ensayo clínico en el componente Paciente Incluido en Ensayos Clínicos. Este procedimiento se realiza para todos los pacientes Incluidos en dichos componentes registrados con anterioridad y se devuelve en una misma página en forma de fila, Ejemplo (10, Ensayo1); el 10 hace referencia a la cantidad de PI y Ensayo1, al ensayo correspondiente con dicho paciente.
Prototipo elemental de interfaz gráfica de usuario:	

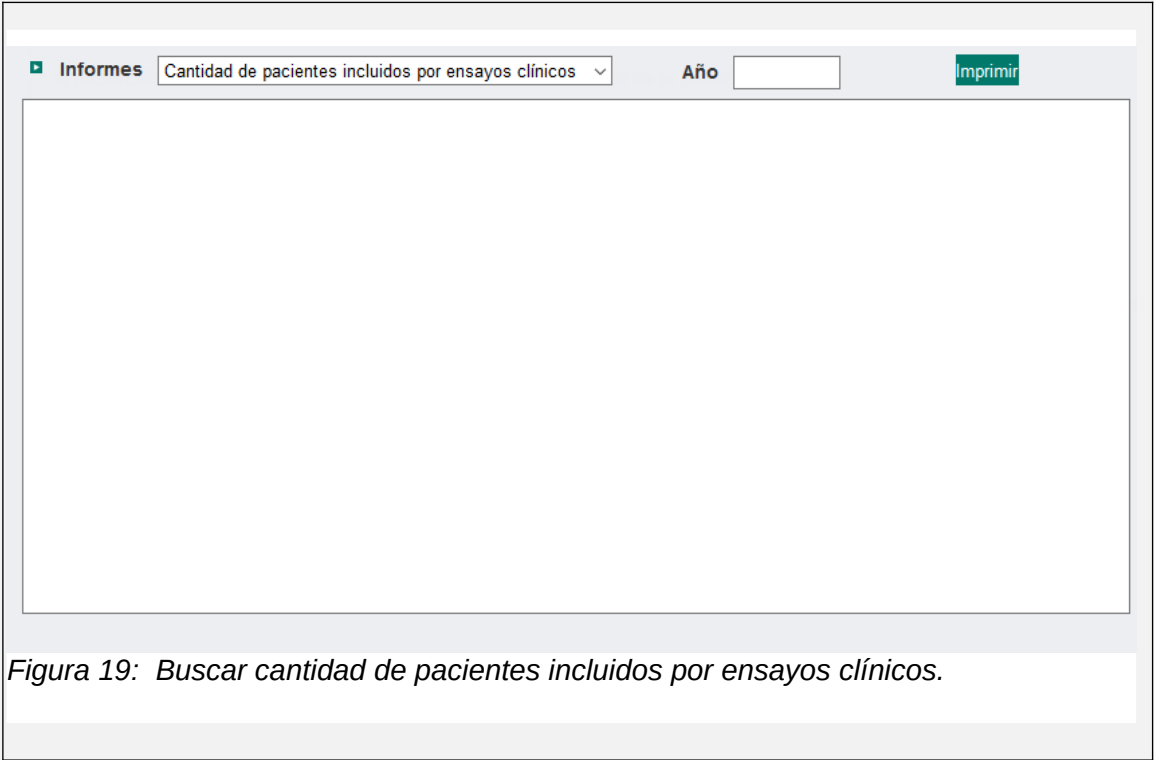


Tabla 10: HU Buscar cantidad de pacientes incluidos por ensayos clínicos.

Historia de Usuario	
Número: 1	Rol: Usuario
Título: Buscar cantidad de pacientes incluidos por promotores.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Tiempo estimado: 4 Horas	Iteración Asignada: 1
Programador Responsable: Rubén Álvarez Izquierdo y Michel Leandro Pérez Somoza.	
Descripción: El sistema permitirá al usuario buscar la cantidad de pacientes incluidos por promotores a partir de uno de los informes.	
No	Criterio de Aceptación
1	El usuario debe tener asignado los permisos para trabajar con la pestaña informes. Debe haberse autenticado en el sistema y haber accedido a la funcionalidad Cantidad de pacientes incluidos por promotores.
2	EL usuario debe ubicar el año, por el cuál desea filtrar la búsqueda y presionar el botón Imprimir.
3	El sistema debe ser capaz de devolver la cantidad de pacientes incluidos por promotores, y agruparlo por promotor de salud, a través de la suma de todos los identificadores de pacientes incluidos que sean promotor de salud, en el componente Paciente Incluido en Ensayos Clínicos. Este procedimiento se realiza para todos los pacientes Incluidos en dicho componente registrado con anterioridad y se devuelve en una misma página en forma de fila.
Prototipo elemental de interfaz gráfica de usuario:	

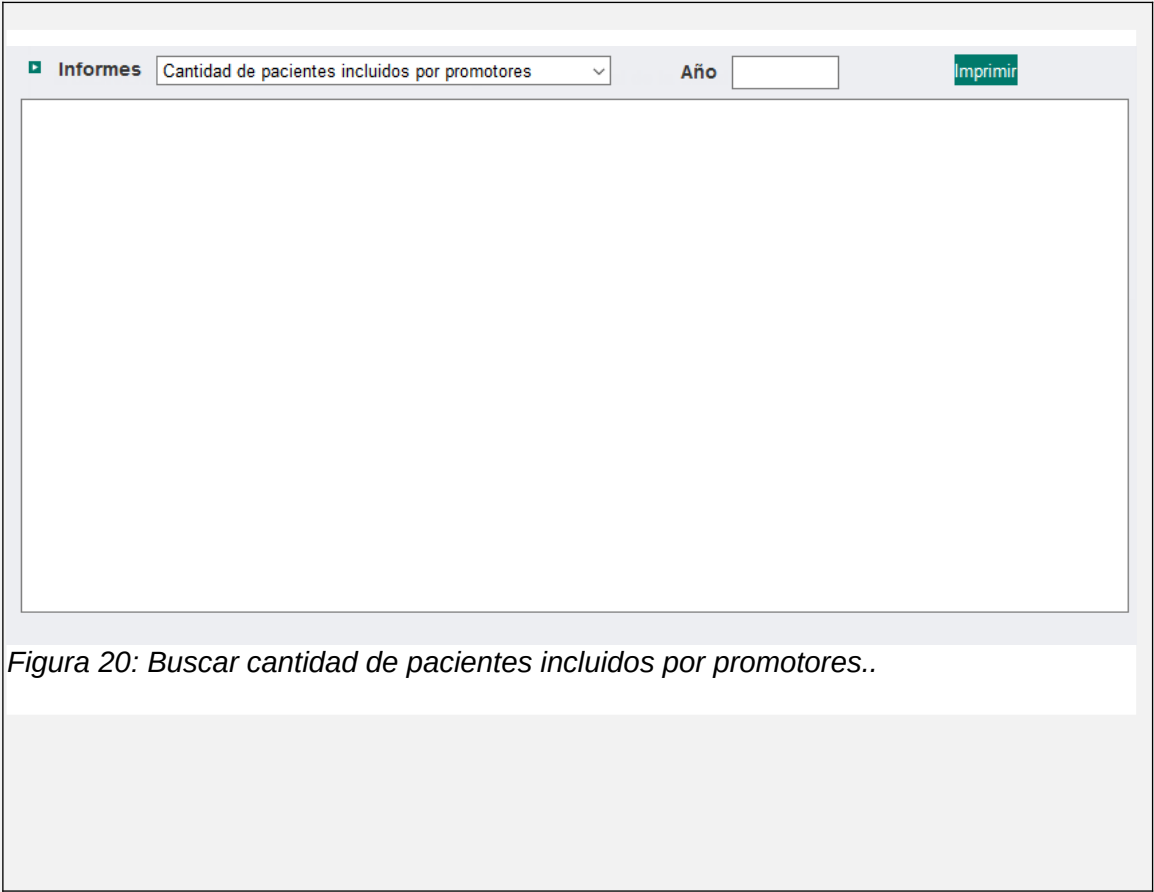


Tabla 11: HU Buscar cantidad de pacientes incluidos por promotores.

Historia de Usuario	
Número: 1	Rol: Usuario
Título: Buscar cantidad de pacientes incluidos por provincias.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Tiempo estimado: 4 Horas	Iteración Asignada: 1
Programador Responsable: Rubén Álvarez Izquierdo y Michel Leandro Pérez Somoza.	
Descripción: El sistema permitirá al usuario buscar la cantidad de pacientes incluidos por provincias a partir de uno de los informes.	
No	Criterio de Aceptación
1	El usuario debe tener asignado los permisos para trabajar con la pestaña informes. Debe haberse autenticado en el sistema y haber accedido a la funcionalidad Cantidad de pacientes incluidos por provincias.
2	EL usuario debe ubicar el año, por el cuál desea filtrar la búsqueda y presionar el botón Imprimir.
3	El sistema debe ser capaz de devolver la cantidad de pacientes incluidos por provincias, y agruparlo por provincia, a través de la suma de todos los identificadores de pacientes incluidos en una provincia en el componente Paciente Incluido en Ensayos Clínicos. Este procedimiento se realiza para todos los pacientes Incluidos en dicho componente registrado con anterioridad y se devuelve en una misma página en forma de fila Ejemplo (15, Cienfuegos), el 15 hace referencia a la cantidad de PI y Cienfuegos, a la provincia correspondiente con dicho paciente.
Prototipo elemental de interfaz gráfica de usuario:	

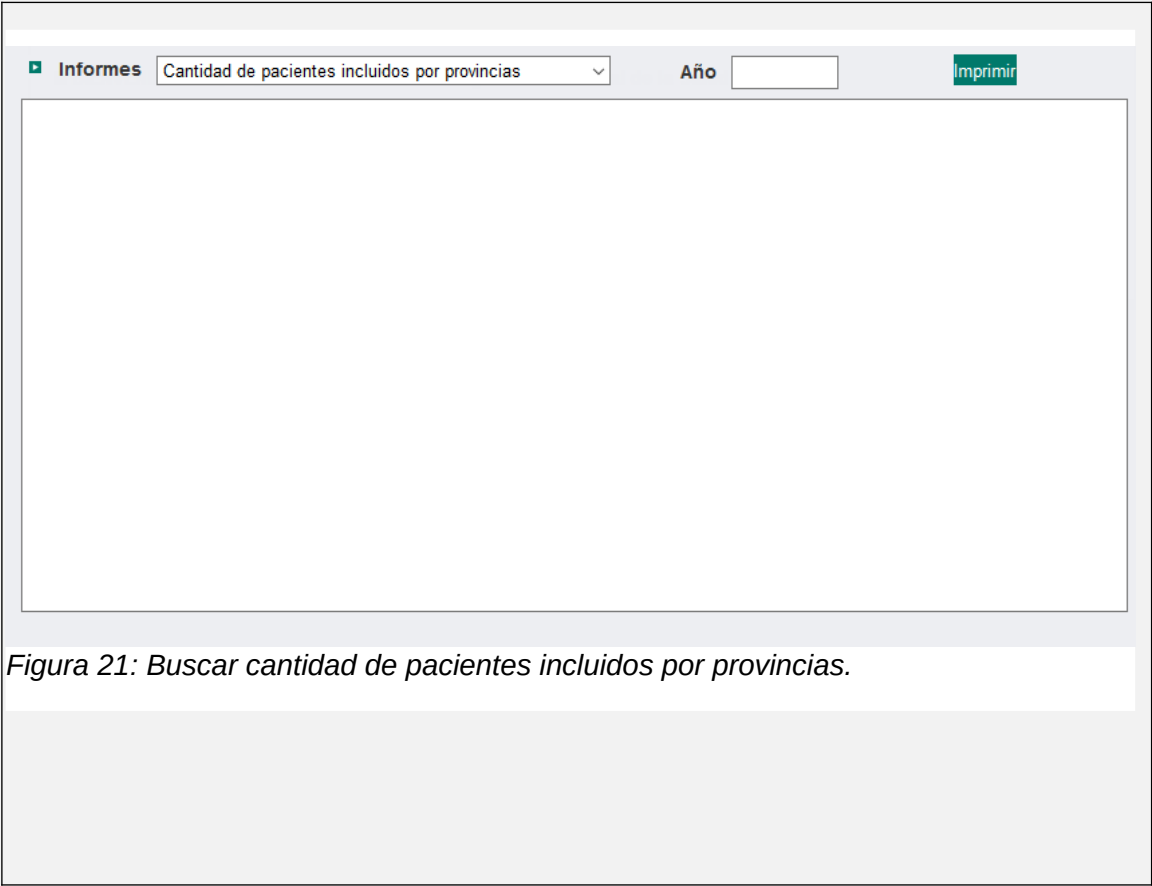


Tabla 12: HU Buscar cantidad de pacientes incluidos por provincias.

Historia de Usuario	
Número: 1	Rol: Usuario
Título: Buscar cantidad de pacientes incluidos por rango de edades.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Tiempo estimado: 4 Horas	Iteración Asignada: 1
Programador Responsable: Rubén Álvarez Izquierdo y Michel Leandro Pérez Somoza.	
Descripción: El sistema permitirá al usuario buscar la cantidad de pacientes incluidos por rangos de edades incluyendo edad pediátrica para menores de 18 años.	
No	Criterio de Aceptación
1	El usuario debe tener asignado los permisos para trabajar con la pestaña informes. Debe haberse autenticado en el sistema y haber accedido a la funcionalidad Cantidad de pacientes incluidos por rango de edades.
2	EL usuario debe ubicar el año, por el cuál desea filtrar la búsqueda y presionar el botón Imprimir.
3	El sistema debe ser capaz de devolver la cantidad de pacientes incluidos por rango de edades, y agruparlos por edades de cinco años de diferencia, y un rango independiente que son los menores de 18 años, incluyendo los 18, a través de la suma de todos los identificadores de pacientes incluidos que se encuentren en el rango de edad, en el componente Paciente Incluido en Ensayos Clínicos. Este procedimiento se realiza para todos los pacientes Incluidos en dicho componente registrado con anterioridad y se devuelve en una misma página en forma de fila Eje (20, [19,24]), 20 hace referencia a la cantidad de pacientes incluidos que se encuentran en el rango de edad de 19 a 24 años.
Prototipo elemental de interfaz gráfica de usuario:	

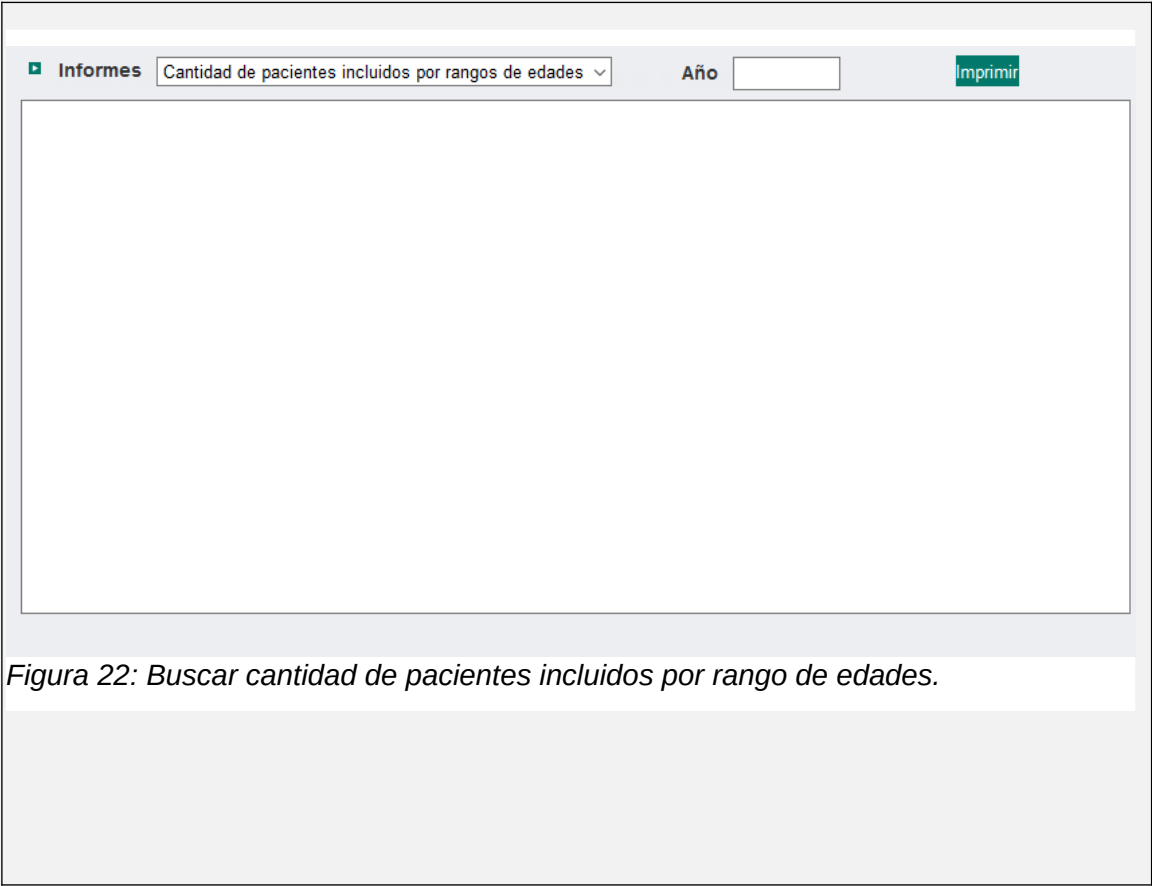


Tabla 13: HU Buscar cantidad de pacientes incluidos por rango de edades.

Historia de Usuario	
Número: 1	Rol: Usuario
Título: Buscar cantidad de pacientes incluidos por sitios de investigación	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Tiempo estimado: 4 Horas	Iteración Asignada: 1
Programador Responsable: Rubén Álvarez Izquierdo y Michel Leandro Pérez Somoza.	
Descripción: El sistema permitirá al usuario buscar la cantidad de pacientes incluidos por sitios de investigación a partir de uno de los informes.	
No	Criterio de Aceptación
1	El usuario debe tener asignado los permisos para trabajar con la pestaña informes. Debe haberse autenticado en el sistema y haber accedido a la funcionalidad Cantidad de pacientes incluidos por provincias.
2	<i>EL usuario debe ubicar el año, por el cuál desea filtrar la búsqueda y presionar el botón Imprimir.</i>
3	El sistema debe ser capaz de devolver la cantidad de pacientes incluidos por sitios de investigación, y agruparlo por sitio de investigación, a través de la suma de todos los identificadores de pacientes incluidos en un sitio de investigación en el componente Paciente Incluido en Ensayos Clínicos. Este procedimiento se realiza para todos los pacientes Incluidos en dicho componente registrado con anterioridad y se devuelve en una misma página en forma de fila.
Prototipo elemental de interfaz gráfica de usuario:	

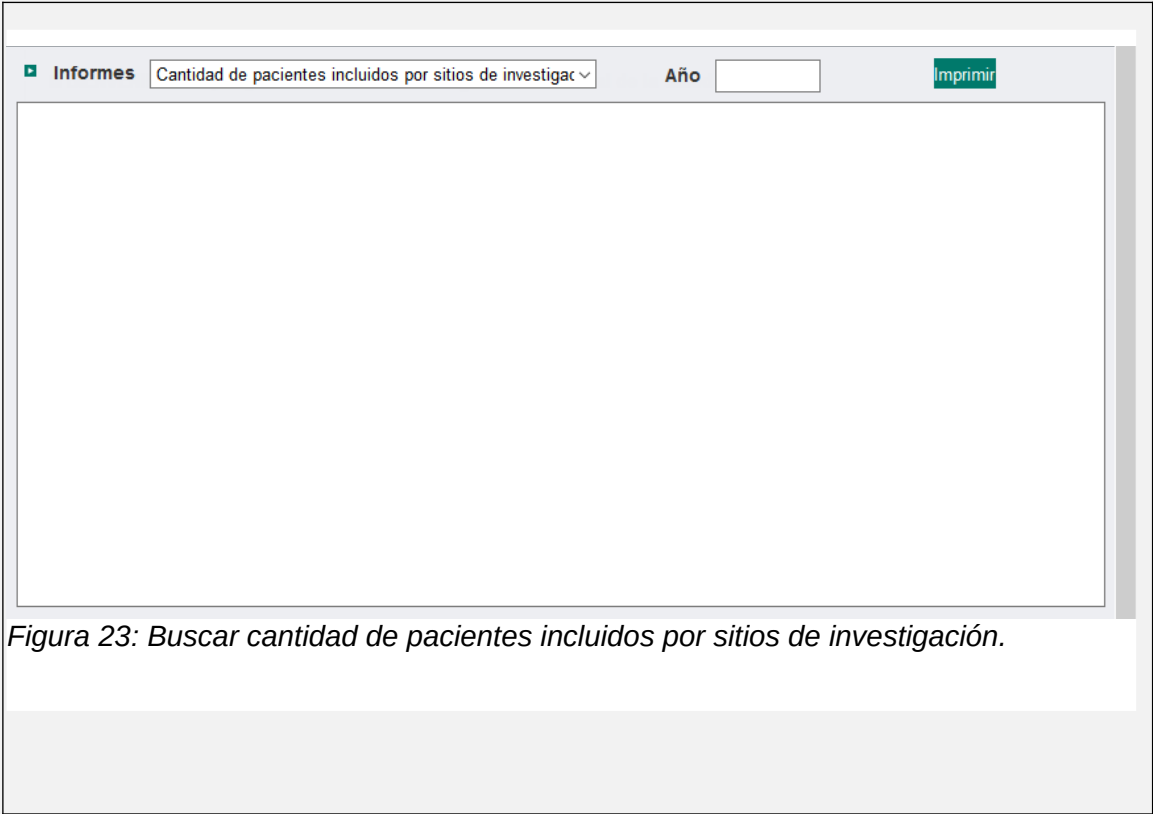


Figura 23: Buscar cantidad de pacientes incluidos por sitios de investigación.

Tabla 14: HU Buscar cantidad de pacientes incluidos por sitios de investigación.

ANEXOS

Historia de Usuario	
Número: 1	Rol: Usuario
Título: Registrar Codificador de Ensayos.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Tiempo estimado: 4 Horas	Iteración Asignada: 1
Programador Responsable: Rubén Álvarez Izquierdo y Michel Leandro Pérez Somoza.	
Descripción: El sistema registrara los ensayos según la unidad de Salud correspondiente.	
No	Criterio de Aceptación
1	El usuario debe tener asignado los permisos para registrar el codificador de ensayo. Debe haberse autenticado en el sistema y haber accedido a la funcionalidad Registrar Codificador de Ensayos.
2	Para registrar el codificador de ensayo, el usuario debe llenar los siguientes campos Código ensayo, Descripción, Nombre corto e implícitamente se registrará la unidad que lo realiza.
Prototipo elemental de interfaz gráfica de usuario:	

■ Codificador de Ensayos

Codificador de Ensayos

Codificador

Descripción

Nombre

Aceptar

Cancelar

Tabla 15: HU Registrar Codificador de Ensayos.

Historia de Usuario	
Número: 1	Rol: Usuario
Título: Registrar Codificador de Promotores.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Tiempo estimado: 4 Horas	Iteración Asignada: 1
Programador Responsable: Rubén Álvarez Izquierdo y Michel Leandro Pérez Somoza.	
Descripción: El sistema registrara los promotores de salud para poder utilizarlos como entidades cliente	
No	Criterio de Aceptación
1	El usuario debe tener asignado los permisos para registrar el codificador de ensayo. Debe haberse autenticado en el sistema y haber accedido a la funcionalidad Registrar Codificador de Ensayos.
2	Para registrar el codificador de promotores, el usuario debe llenar los siguientes campos: Nombre entidad, Siglas, Correo del responsable
Prototipo elemental de interfaz gráfica de usuario:	

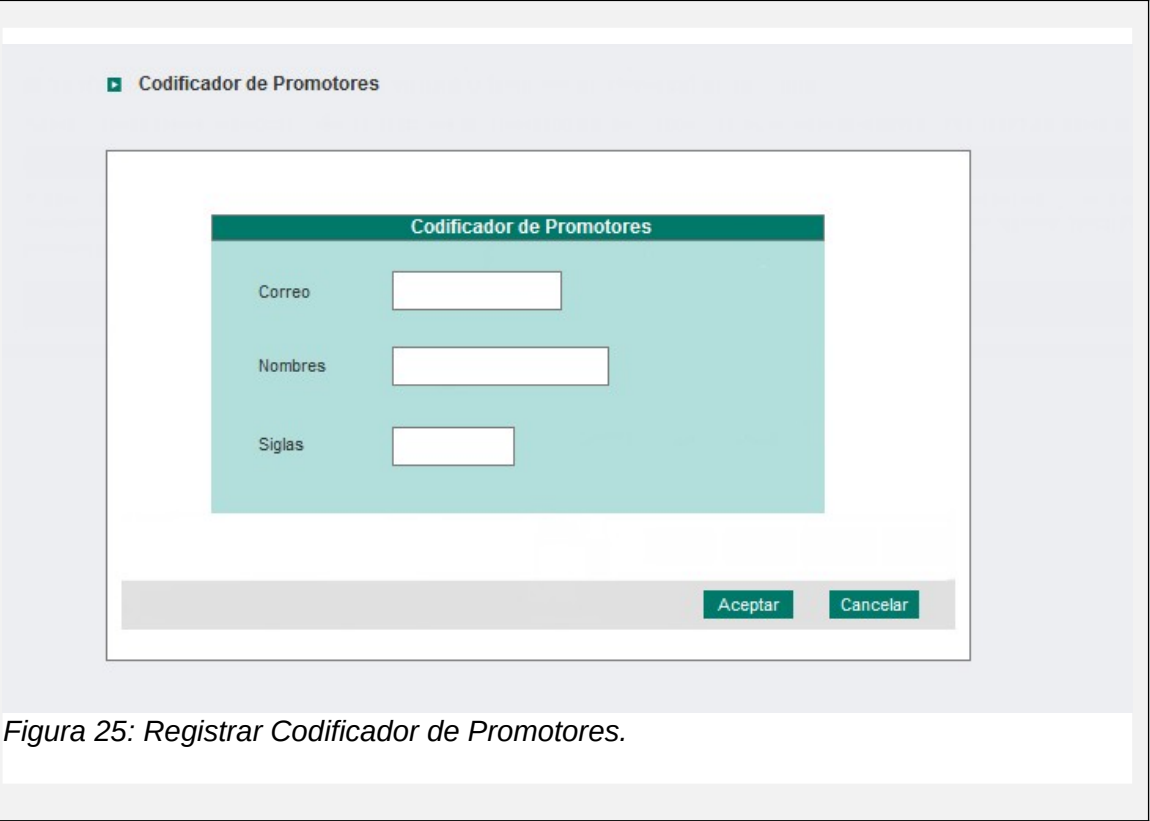


Figura 25: Registrar Codificador de Promotores.

Tabla 16: HU Registrar Codificador de Promotores.