

UNIVERSIDAD NACIONAL DE INGENIERÍA

Facultad de Electrotecnia y Computación



**DESARROLLO DE UN SISTEMA DE GESTIÓN DE CORRESPONDENCIA
PARA EL LABORATORIO DE INVESTIGACIONES CLINICAS
DIVINO NIÑO – NINDIRÍ, MASAYA**

Trabajo monográfico
Para optar al título de ingeniero en computación

Realizado por:

Br. Kerolin Alemán Rivera

Tutor:

Ing. José Leónidas Díaz Chow.

Febrero 2022

Managua, Nicaragua

DEDICATORIA

A la memoria de aquella señora que decía que buscara aquello a lo que ella llamaba el cartón y quien me decía incesantemente que un lápiz pesa menos que una pala, a mi abuela doña Paula López Arias, a mi madre y a mi compañerita de batallas, mi hija Amy N. Domínguez, quien me ha enseñado otra faceta de mi vida que nunca me imaginé vivir.

A ellas, las mujeres de mi vida.

AGRADECIMIENTO

Agradezco a Dios Padre todo poderoso por permitirme llegar a esta etapa de mi vida.

A mis padres, mi hija y familiares por todo el apoyo brindado durante todo este proceso de formación.

A mis profesores por todo el apoyo y paciencia.

A mi tutor, el Ing. José Díaz Chow, por todo el apoyo brindado en la elaboración de este documento monográfico y por su oportuna instrucción.

A mis colegas y amigos por siempre animarme a superarme cada día.

RESUMEN

El presente trabajo monográfico “Desarrollo de un Sistema de Gestión de Correspondencia para el Laboratorio Divino Niño – Nindirí, Masaya”, consistió en la creación de un sistema web para proveer soporte informático al registro de la correspondencia física del laboratorio, facilitando el control de su gestión y mejorando la eficiencia de la administración de las comunicaciones formales en esta empresa de servicios clínicos.

El sistema permite registrar la información relevante de toda correspondencia física entrante o recibida (ya sean cartas, facturas, proformas o cualquier otro documento físico dirigido a una persona dentro de la organización) y saliente o enviada (ya sea por mensajería física o correo postal). Envía notificaciones por correo electrónico a los destinatarios de la correspondencia recibida y permite registrar la evidencia de entrega o recibo de la correspondencia enviada. Facilita realizar seguimiento a la gestión de la correspondencia a través de consultas dinámicas y reportes. Asimismo, conforme lo requerido por las autoridades del Laboratorio, el sistema permite llevar registro de la entrega de los resultados de los estudios clínicos contratados por los clientes, a fin de tener control centralizado del universo de toda documentación física saliente de la organización.

Conforme a los requerimientos técnicos definidos, el sistema se desarrolló sobre tecnología Microsoft ASP.Net ©, con arquitectura basada en el patrón estructural Modelo-Vista-Controlador (MVC), empleando Microsoft SQL Server © 2014 como gestor de base de datos y C# como lenguaje de desarrollo del lado del servidor. Del lado del cliente se empleó HTML 5.0, CSS y javascript (framework Bootstrap), que son las tecnologías que integra la plantilla de Microsoft Visual Studio © para un proyecto de aplicación Web con arquitectura MVC.

Para dirigir el proceso de desarrollo se empleó la metodología de ingeniería de software Proceso Racional Unificado (RUP por sus siglas en inglés) de IBM y el Lenguaje de Modelado Unificado (UML por sus siglas en inglés) para la creación de los modelos para el análisis y diseño del sistema. El diseño de interfaz de usuario y demás decisiones arquitecturales se ajustaron a los estándares y políticas definidas por El Laboratorio.

TABLA DE CONTENIDO

DEDICATORIA	i
AGRADECIMIENTO	ii
RESUMEN	iii
1 INTRODUCCIÓN	1
1.1 Antecedentes	2
2 OBJETIVOS	6
2.1 Objetivo General	6
2.2 Objetivos Específicos	6
3 JUSTIFICACIÓN	7
4 MARCO TEÓRICO	8
4.1 Gestión de Correspondencia.....	8
4.1.1 Normas técnicas y estándares de gestión de Correspondencia	8
4.2 Tecnologías para Desarrollo de Software.	9
4.2.1 Herramienta de desarrollo: Microsoft Visual Studio	9
4.2.2 Lenguaje de desarrollo C#.....	11
4.2.3 <i>Marco de trabajo ORM: Entity Framework</i>	15
4.2.4 Marco de trabajo de capa de presentación Bootstrap	16
4.2.5 Marco de trabajo ASP.NET MVC V5	16
4.2.6 Gestor de base de datos relacionales SQL Server.....	17
4.3 Metodología de Desarrollo de Software.....	18
4.3.1 Proceso Racional Unificado (RUP).....	18
4.3.2 Lenguaje de Modelado Unificado (UML)	22
5 DESARROLLO DEL SISTEMA	25
5.1 Fase de Inicio.....	25

5.1.1	Requerimientos del Sistema	25
5.1.2	Factibilidad del Sistema	30
5.2	Fase de Elaboración	37
5.2.1	Modelo de Casos de Uso.....	37
5.2.2	Modelo de Dominio del Problema.....	56
5.2.3	Modelo de Clases	57
5.2.4	Diseño de la Base de Datos	62
5.2.5	Modelo de Interacción.....	67
5.3	Fase de Construcción	74
5.3.1	Arquitectura del Sistema	74
5.3.2	Desarrollo del Sistema	75
5.4	Fase de Implantación	79
5.4.1	Instalación de ambiente de capacitación a usuarios	79
5.4.2	Capacitación y pruebas de usuario	79
5.4.3	Evaluación del uso del sistema	80
5.4.4	Instalación final del sistema en servidores de producción y puesta en servicio ...	85
6	CONCLUSIONES Y RECOMENDACIONES.....	86
6.1	Conclusiones	86
6.2	Recomendaciones.....	87
7	BIBLIOGRAFÍA.....	88
8	ANEXOS.....	- 1 -

Índice de Tablas

Tabla 1: Requerimientos Funcionales	26
Tabla 2: Requerimientos No Funcionales	29
Tabla 3: Requerimientos mínimos de hardware del servidor.....	30
Tabla 4: Requerimientos de hardware de la computadora de desarrollo.....	31
Tabla 5: Requerimientos de hardware de computadora de usuario.....	32
Tabla 6: Cálculo del esfuerzo del proyecto.	34
Tabla 7: Resumen de costos del SGC	35
Tabla 8: CU1: Autenticarse en el sistema	40
Tabla 9: CU2: Administrar Sistema	42
Tabla 10: CU3: Gestionar Catálogos.....	43
Tabla 11: CU4: Registrar Correspondencia	45
Tabla 12: CU4.1: Registrar Correspondencia Entrante	47
Tabla 13: CU4.2: Registrar Correspondencia Saliente	49
Tabla 14: CU4.3: Actualizar Correspondencia.....	50
Tabla 15: CU5: Entregar Resultado de Estudio Clínico.....	51
Tabla 16: CU6: Dar Seguimiento a Correspondencia	53
Tabla 17: CU7: Generar Reportes	54
Tabla 18: Diccionario de Datos de Dominio de Aplicación	65
Tabla 19: Diccionario de datos de Seguridad de Usuarios.....	66
Tabla 20: Medida de las variables de fiabilidad.....	82
Tabla 21: Medida de variables de desempeño.	83
Tabla 22: Resultados de evaluación de fiabilidad.	83
Tabla 23: Resultados evaluación de desempeño.....	83

Índice de Figuras

Figura 4-1. Fases del RUP	19
Figura 4-2: Ciclo de vida del software de RUP	20
Figura 5-1 Diagrama de Caso de Uso del Sistema.....	38
Figura 5-2: Diagrama de Dominio	57
Figura 5-3: Diagrama de Clases.....	59
Figura 5-4: Diagrama de Tablas de Dominio de Aplicación	63
Figura 5-5: Diagrama de Tablas de Seguridad de Usuario	64
Figura 5-6: Diagrama de Secuencia para la operación Crear	68
Figura 5-7: Diagrama de Secuencia para la operación Editar	70
Figura 5-8: Diagrama de Secuencia para la operación Borrar	72
Figura 5-9 Arquitectura del SGC.....	74
Figura 5-10 Desarrollo del SGC	75
Figura 5-11: Diagrama de modelo de datos del Entity Framework	77

1 INTRODUCCIÓN

La comunicación es un aspecto vital para la gestión de toda organización, principalmente la comunicación formal. La comunicación en forma escrita tiene más relevancia en la gestión por cuanto tiene un carácter oficial y puede tener connotación e implicaciones administrativas y legales (Manene, 2013).

Son aspectos muy importantes de la información que se comunica: la oportunidad y la confidencialidad, por lo cual, las organizaciones tienen sumo cuidado en gestionar las comunicaciones confidencial y diligentemente (Gómez, 2007). Los avances tecnológicos han permitido dar pasos significativos en la automatización de los procesos y servicios claves, como es la gestión de la comunicación, a través de herramientas informáticas, mejorando así el desempeño de las empresas (García & Martín, 2017). Desafortunadamente, en muchas organizaciones, el control de la correspondencia (comunicación escrita) se sigue gestionando mediante controles manuales que son vulnerables a errores humanos.

El laboratorio de Investigaciones Clínicas Divino Niño, del municipio de Nindirí, en el departamento de Masaya, ha experimentado dificultades para gestionar adecuadamente su correspondencia, debido a que actualmente su control es manual y la cantidad de correspondencia se ha incrementado considerablemente al incorporarse la entrega de resultados de exámenes de laboratorio como parte de la misma. El débil control manual ha generado una serie de problemas como demoras e incumplimientos en plazos de entrega de comunicaciones a proveedores y pacientes, falta de respuesta oportuna en las comunicaciones y pérdida de eficiencia por fallas de comunicación.

Debido a lo sensible de la información que se maneja y al volumen de la correspondencia que generan las comunicaciones del laboratorio, es requerido sistematizar la gestión de la correspondencia entrante y saliente de la organización.

Con el presente trabajo monográfico se provee una solución al problema planteado, a través del desarrollo de un sistema de gestión de correspondencia, que permite registrar la información relevante de toda la correspondencia física de la organización, facilitando así el control de su gestión y mejorando su eficiencia.

1.1 Antecedentes

El Laboratorio de Investigaciones Clínicas Divino Niño (en adelante referido también como “El Laboratorio”) está ubicado en la ciudad de Nindirí del departamento de Masaya. Éste inició sus operaciones el 20 de Julio del 2008. Actualmente se encuentra bajo la administración del Dr. Hellman Castillo Quintanilla y la Lic. Francisca Bravo Toruño.

El control de la correspondencia en el laboratorio de Investigaciones Clínicas Divino Niño se realiza actualmente de forma manual. La recepcionista del área administrativa anota en un cuaderno de actas cada comunicación que recibe con la fecha y hora de recepción. Posteriormente procede a enviarla al destinatario de la misma a lo interno de la estructura organizativa del Laboratorio.

En el caso de envío de correspondencia saliente, actualmente no se registra información. Queda a responsabilidad del remitente llevar su control. La recepcionista del área administrativa se limita a gestionar el envío de la correspondencia por mensajería. Si la comunicación requiere acuso de recibo, se encarga de regresar al remitente la copia de recibido.

Este método de trabajo ha generado inconvenientes de diversa índole derivado de errores humanos y falta de oportunidad del registro. Por mencionar algunos de estos:

- a. Se ha olvidado entregar correspondencia recibida a tiempo.
- b. Se ha traspapelado correspondencia.
- c. No se ha enviado oportunamente información urgente y no se tiene control de dónde se atrasa el proceso.

- d. No se lleva control centralizado de la correspondencia enviada, por lo que se dificulta saber cuáles y cuántas comunicaciones se han enviado a un destinatario específico.

Estos inconvenientes han afectado la gestión del área administrativa y del Laboratorio en general, mermando la calidad del servicio que se presta a la población. Adicionalmente, ha impactado negativamente en la evaluación de los indicadores de desempeño de los trabajadores y de la oficina administrativa en su conjunto.

El gerente general, Dr. Castillo¹, planteó a la autora de este trabajo monográfico, la necesidad de sistematizar la gestión de la correspondencia del laboratorio para superar la problemática descrita anteriormente. En la indagación realizada en las dependencias de informática a lo interno del Laboratorio no se encontró información de trabajos previos asociados a atender esta necesidad específica. Por ende, no existe antecedente alguno correspondiente a la sistematización de la gestión documental en el Laboratorio Divino Niño de la ciudad de Nindirí, Masaya.

En revisión bibliográfica de trabajos similares realizados, se encontraron varios casos elaborados a nivel nacional e internacional, como los presentados en (Guarachi & Rodrigo, 2014), (Rodríguez & Alfonso, 2015), (Carmona, Sáenz, & Morales, 2013), y (Rodríguez, 2013). Sin embargo, todos los trabajos citados son soluciones a la medida para diversas instituciones, como universidades y cooperativas, que por ende no estarían a disposición del Laboratorio y no necesariamente se ajustan a las necesidades del mismo.

¹ Dr. Hellman Castillo, médico cirujano, dueño y Gerente General del laboratorio de Investigaciones Clínicas Divino Niño, manifestado en entrevista realizada con la autora de este documento.

Así también se indagó sobre gestores de documentos de libre disponibilidad que pudieran adaptarse como solución a la problemática planteada, encontrándose los siguientes:

- **OpenKM:** Es un sistema de gestión de contenidos de código fuente libre; desarrollado en Java, concebido para colaborar y gestionar documentación e información a escala empresarial. Compatible con Windows, Linux y otros sistemas (S.L., 2019).

OpenKM se ofrece en 3 versiones distintas:

- OpenKM Community: modalidad de licenciamiento GNU General Public Licence version 2. El código fuente de OpenKM está disponible para toda la comunidad, la cual es libre de utilizarlo, modificarlo y redistribuirlo bajo las premisas de dicha licencia.
 - OpenKM Profesional: licencia de código abierto y estándares abiertos, con la posibilidad de soporte comercial y propietario a escala empresarial.
 - OpenKM Cloud: es la modalidad SaaS o Software como servicio de OpenKM.
- **OpenProDoc:** Es un gestor documental que puede instalarse en múltiples plataformas. Fue desarrollado en Java y publicado con licencia de código abierto (Hierro, 2019).

Además de las funciones de gestor documental, OpenProdoc incluye de forma integrada funciones de gestor de tesauros y gestión de referencias bibliográficas.

Ya que las opciones de software previamente mencionadas abordan de manera bastante completa el proceso de gestión de correspondencia, se presentó y revisó en conjunto con el Gerente General del Laboratorio las características, ventajas y desventajas de los gestores de documentos listados, concluyendo el Gerente General que prefiere una solución de desarrollo a medida sobre la adopción de uno de los sistemas de software libre, por las siguientes razones:

- Las soluciones de software libre revisadas son muy especializadas e incluyen muchas más funciones de las requeridas, obligando al personal que la opere a tener una curva de aprendizaje bastante grande.
- El software a medida será de pertenencia del interesado, por tanto, su mantenimiento, evolución y soporte técnico está bajo control total del cliente.
- El tener control del desarrollo del software a medida le confiere mayor seguridad y confianza.
- El control de versiones y copias de seguridad están a total disposición del interesado.

Por consiguiente, el trabajo monográfico se orientó al desarrollo de un sistema de gestión de información basado en computadora y sobre plataforma Web a fin de satisfacer las necesidades y preferencias del Gerente General del Laboratorio Divino Niño, al cual denominamos: “Sistema de Gestión de Correspondencia”, abreviadamente “SGC”.

2 OBJETIVOS

2.1 Objetivo General

Desarrollar un Sistema de Gestión de Correspondencia para el Laboratorio de Investigaciones Clínicas Divino Niño de la ciudad de Nindirí, en el Departamento de Masaya.

2.2 Objetivos Específicos

- a) Identificar los requerimientos que debe cumplir el SGC para superar la problemática planteada.
- b) Realizar el diseño de la aplicación siguiendo la metodología de desarrollo de Software RUP y UML como lenguaje de modelado.
- c) Codificar el SGC sobre la base del diseño realizado.
- d) Implantar el SGC, desarrollado en el laboratorio de Investigaciones Clínicas Divino Niño.
- e) Evaluar el resultado del uso del SGC, verificando su efectividad en la solución del problema.

3 JUSTIFICACIÓN

Por cuanto la búsqueda de soluciones para resolver la problemática en la gestión de correspondencia del Laboratorio Clínico Divino Niño, demostró que:

- a) Los trabajos académicos similares que se encontraron en la investigación documental no son aplicables.
- b) Las soluciones de software libres existentes en el mercado tienen mayor alcance que el requerido, pero no cubren de manera exacta las necesidades y requerimientos en el proceso de gestión de correspondencia con las particularidades específicas, por lo que el Gerente de El Laboratorio desestimó su uso.

Por tanto, se consideró pertinente proveer una solución al problema a través del desarrollo de un sistema informático a medida que automatice el proceso de gestión de correspondencia de El Laboratorio.

El sistema SGC desarrollado permite registrar la información pertinente de cada correspondencia recibida y enviada, generando a los trabajadores del Laboratorio los siguientes beneficios:

- 1. Información completa y precisa de cada correspondencia enviada y recibida en un lugar centralizado.
- 2. Capacidad de llevar control y seguimiento de cada una de las comunicaciones que se gestionan.
- 3. Generación de notificaciones por correo electrónico para informar oportunamente de las recepciones y asegurar una gestión a tiempo de las respuestas de correspondencia.
- 4. Capacidad de consultar la información de la correspondencia gestionada por la organización.
- 5. Capacidad de generación de estadísticas de las comunicaciones entrantes y salientes del Laboratorio.

4 MARCO TEÓRICO

A continuación, se presentan los conceptos relacionados al negocio y la tecnología a emplear para el desarrollo del sistema de gestión de correspondencia física SGC para El Laboratorio.

4.1 Gestión de Correspondencia

La gestión de la correspondencia es una de áreas más relevantes de la gestión documental de cualquier organización. La correspondencia es un documento esencial desde siempre, ya no sólo como parte de actividades institucionales y empresariales, sino también privadas y particulares (gestion.org, 2018).

4.1.1 Normas técnicas y estándares de gestión de Correspondencia

Si bien en la investigación documental realizada no se encontraron normas técnicas específicas o estándares para la gestión de correspondencia, si se encontraron normas y estándares referidos a la gestión documental. De éstas las más relevantes son:

- **Norma ISO 15489 “Información y Documentación”** (UNE, 2005): Esta norma se orienta a estandarizar las políticas y los procedimientos de la gestión de documentos de archivo, asegura la adecuada atención y protección de los mismos, y permite que la evidencia y la información que contienen puedan ser recuperadas de un modo más eficiente y más eficaz usando prácticas y procedimientos normalizados. Si bien su ámbito es más general, las recomendaciones referidas al diseño de un sistema de gestión de documentos y a gestión de calidad, establecen un marco de referencia para el presente trabajo.

- **DoD 5015: Design Criteria Standard for Electronic Records Management Software Application** (Bustelo, 2015): Corresponde a especificaciones funcionales para los sistemas de software para la gestión documental del departamento de defensa de los Estados Unidos de América. Es un estándar de alto perfil para la gestión de registros electrónicos con gran volumen de información confidencial. Se enfoca en tácticas centrales como planes de archivo, declaración de registros, preservación, registros físicos, registros vitales, marcas suplementarias y transferencias de registros al almacenamiento fuera de línea u otros repositorios certificados. Si bien es cierto su alcance es mucho mayor que el requerido en este trabajo monográfico, se emplearán las recomendaciones y guías generales para la definición de los requerimientos del SGC a desarrollar.
- **Manual de archivo de correspondencia** (Trigos, Claro, Claro, & Barbosa, 2019): Constituye una recopilación de elementos básicos según la norma nacional colombiana para la clasificación, creación y clasificación de documentos de archivo de correspondencia. Provee un marco conceptual y de clasificación de los documentos de correspondencia a emplear en el presente trabajo.

4.2 Tecnologías para Desarrollo de Software.

4.2.1 Herramienta de desarrollo: Microsoft Visual Studio

Es un entorno de desarrollo de software para sistemas operativos Windows. Este conjunto de herramientas se utiliza para crear sitios y aplicaciones web, así como generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio aplicaciones móviles.

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para Windows y macOS. Es compatible con múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web, como ASP.NET MVC, Django, etc., a lo cual hay que sumarle las nuevas capacidades en línea bajo Windows Azure en forma del editor Mónico.

Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno compatible con la plataforma .NET (a partir de la versión .NET 2002). Así, se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos y videoconsolas, entre otros.

Marco de Trabajo (framework) .NET

El framework .NET ofrece un compilador C# que se puede utilizar directamente por línea de comandos después de haber introducido el código del programa en un editor de texto, como NotePad. El procedimiento es posible pero no muy productivo. Evidentemente, el desarrollador busca utilizar un entorno de desarrollo totalmente integrado, que le ayude durante la escritura del código, le permita realizar la interfaz gráfica, la puesta a punto y el despliegue de su aplicación. Este programa es un **IDE** (Integrated Development Environment). Integra al menos un editor de código fuente, herramientas que automatizan las compilaciones y un unlocker.

Visual Studio es la herramienta de desarrollo propuesta por Microsoft. Potente, estable y con un entorno amigable, Visual Studio se presenta en varias versiones: Community, Professional y Enterprise. Es importante mencionar que las versiones Professional y Enterprise son soluciones de pago mientras Visual Studio 2019 Community es un IDE gratuito ideal para estudiantes, colaboradores de código abierto y usuarios particulares.

El framework .NET representa el conjunto de interfaces de programación que forman el núcleo de la plataforma .NET y que permiten maximizar el rendimiento, la fiabilidad y la seguridad de los servicios Web XML.

Todos los lenguajes .NET emplean el Common Language Runtime (CLR) y comparten un gran conjunto de recursos como son:

- Un modelo de programación orientado a objetos (herencia, polimorfismo, manejo de excepciones y colección de basura).
- Modelo de seguridad.
- Sistema de tipos.
- Base Class Library (BCL) (Biblioteca de Clases Base).
- Desarrollo, depuramiento y herramientas.
- Administración de ejecución y código.
- Traductores y optimizadores de MSIL (MicroSoft Intermediate Language) a código nativo.

4.2.2 Lenguaje de desarrollo C#

C# (pronunciado "si sharp" en inglés) es un lenguaje de programación moderno, basado en objetos y con seguridad de tipos, que permite a los desarrolladores crear muchos tipos de aplicaciones seguras y sólidas que se ejecutan en el ecosistema de .NET. C# tiene sus raíces en la familia de lenguajes C, y a los programadores de C, C++, Java y JavaScript les resultará familiar inmediatamente.

C# es un lenguaje de programación **orientado a componentes**, orientado a objetos. C# proporciona construcciones de lenguaje para admitir directamente estos conceptos, por lo que se trata de un lenguaje natural en el que crear y usar componentes de software. Desde su origen, C# ha agregado características para admitir nuevas cargas de trabajo y prácticas de diseño de software emergentes. (Microsoft, Microsoft Docs, 2021)

C# es el único que ha sido diseñado específicamente para ser utilizado en .Net, por lo que programar usando C# es más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes. Por esta razón, Microsoft suele referirse a C# como el

lenguaje nativo de .NET, y, de hecho, gran parte de la librería de clases base de .NET ha sido escrito en este lenguaje.

C# es un lenguaje orientado a objetos sencillo, moderno, amigable y fácilmente legible que recoge las mejores características de muchos otros lenguajes, fundamentalmente Visual Basic, Java y C++, para combinarlas en uno sólo en el que se une la alta productividad y facilidad de aprendizaje de Visual Basic con la potencia de C++.

Características de C#

- Visual C# distingue entre minúsculas y mayúsculas.
- Dispone de todas las características propias de cualquier lenguaje orientado a objetos: encapsulación, herencia y polimorfismo.
- Tiene a su disposición un recolector de basura que libera al programador de la tarea de tener que eliminar las referencias a objetos que no son útiles, evitando que se agote la memoria en áreas ya liberadas y reasignadas.
- Incluye soporte nativo para EVENTOS y DELEGADOS. Los delegados son similares a los punteros a funciones y los eventos son mecanismos mediante los cuales los objetos pueden notificar de la ocurrencia de sucesos.
- Incorpora propiedades, que permiten el acceso controlado a miembros de una clase. Gracias a ellos se evita la pérdida de legibilidad que en otros lenguajes causa la utilización de métodos Set () y Get () y se mantienen todas las ventajas de un acceso controlado.
- Admite atributos que no son miembros de las clases sino información sobre éstas que podemos incluir en su declaración.
- Es un lenguaje que controla que todas las conversiones entre tipos se realicen de forma compatible.

- Asegura que nunca se acceda fuera del espacio de memoria ocupado por un objeto evitando frecuentes errores de programación y consigue que los programas no puedan poner en peligro la integridad de otras aplicaciones.
- Incluye la recolección de elementos no utilizados y la seguridad en el tratamiento de tipos.
- Seguridad implementada por medio de mecanismos de confianza intrínsecos del código.
- Compatible con componentes XML basados en Web y metadatos extensibles.
- Plena interoperabilidad por medio de los servicios de COM+ y .NET Framework con un acceso basado en bibliotecas permitiendo la integración de código existente.
- Proporciona interoperabilidad con otros lenguajes, entre plataformas y con datos heredados.
- Admite el control de versiones para facilitar la administración y la implementación.

Nacimiento de C#

Andrés Hejlsberg decidió formar un equipo de trabajo en 1999 para crear un nuevo lenguaje de programación, que hoy conocemos como C#. En cambio, en sus inicios el nombre inicial que se barajó fue **Cool** (C Object Oriented Language), lo que en español traduciríamos con un lenguaje de programación orientado a objetivos.

C# tiene sus orígenes en la familia de lenguajes C y su primera versión, tal y como explica Microsoft, se parecía mucho a Java. De hecho, se creó con el fin de ser una alternativa viable en Windows. Muchas de sus características fueron evolucionando y mejorando hasta llegar a la versión actual.

Por ejemplo, ahora el lenguaje de C# admite los conceptos de encapsulación, herencia y polimorfismo y facilita el desarrollo de componentes de software mediante varias construcciones de lenguaje innovadoras. (School, 2018)

Uso en páginas web

C# es en términos generales, multiplataforma, aunque al tratarse de un proyecto de Microsoft ha sido específicamente diseñado para .NET Framework. Es muy frecuente encontrarlo bajo el nombre “Visual C#”, sobre todo como implementación. Conceptualmente, se trata de una evolución de Java y C++ que amplía el modelo orientado a objetos gracias a los llamados atributos, que almacenan información sobre clases, objetos y métodos, y a los delegados, que representan referencias a métodos determinados. Principalmente, esto nos permite conseguir una descripción de error más eficiente durante la compilación de código, algo que ahorra tiempo a los desarrolladores.

Para muchos, C# es, junto a Java, el lenguaje de programación más importante y que todo desarrollador web debería aprender. Como lenguaje de programación orientado a objetos, C# ofrece la mejor combinación entre funcionalidad y potencia. Sus críticos advierten del problema derivado de vincular el uso de C# a .NET Framework de Microsoft. No obstante, con esta sintaxis, los programadores cubren un gran sector del mercado: muchos confían actualmente en C# a la hora de programar para sistemas Windows o videojuegos para Xbox y PC. Como lenguaje de programación web, C# se utiliza principalmente en las API web y en varias aplicaciones web.

4.2.3 Marco de trabajo ORM: Entity Framework

Entity Framework es marco de trabajo que facilita el mapeo objeto – relacional (ORM) para sistemas de gestión de información desarrollados en paradigma de programación orientado a objetos (OOP) pero que persisten los datos en un sistema gestor de base de datos relacional (RDBMS). Se compone de un conjunto de tecnologías en ADO.NET (Es parte de la biblioteca de clases base que están incluidas en el Microsoft .NET Framework) que respaldan el desarrollo de aplicaciones de software que acceden a bases de datos relacionales. Los arquitectos y desarrolladores de aplicaciones orientadas a datos han luchado con la necesidad de lograr dos objetivos muy diferentes. Deben modelar las entidades, las relaciones y la lógica de los problemas comerciales que están resolviendo, y también deben trabajar con los motores de datos utilizados para almacenar y recuperar los datos. Los datos pueden abarcar varios sistemas de almacenamiento, cada uno con sus propios protocolos; incluso las aplicaciones que funcionan con un único sistema de almacenamiento deben equilibrar los requisitos del sistema de almacenamiento con los requisitos de escribir código de aplicación eficiente y fácil de mantener.

Entity Framework permite a los desarrolladores trabajar con datos en forma de objetos y propiedades específicos del dominio, como clientes y direcciones de clientes, sin tener que preocuparse por las tablas y columnas de la base de datos subyacente donde se almacenan estos datos. Con Entity Framework, los desarrolladores pueden trabajar a un nivel más alto de abstracción cuando manejan datos y pueden crear y mantener aplicaciones orientadas a datos con menos código que en las aplicaciones tradicionales. Debido a que Entity Framework es un componente de .NET Framework, las aplicaciones de Entity Framework pueden ejecutarse en cualquier computadora en la que esté instalado .NET Framework a partir de la versión 3.5 SP1. (Microsoft, Microsoft Docs, 2018)

4.2.4 Marco de trabajo de capa de presentación Bootstrap

Bootstrap es un framework de capa de presentación basado en hojas de estilo en cascada (CSS) desarrollado por Twitter en 2010, para estandarizar las herramientas de la compañía.

Inicialmente, se llamó Twitter Blueprint y, un poco más tarde, en 2011, se transformó en código abierto y su nombre cambió para Bootstrap. Desde entonces fue actualizado varias veces y ya se encuentra en la versión 4.4.

El framework combina CSS y JavaScript para estilizar los elementos de una página HTML. Permite mucho más que, simplemente, cambiar el color de los botones y los enlaces.

Esta es una herramienta que proporciona interactividad en la página, por lo que ofrece una serie de componentes que facilitan la comunicación con el usuario, como menús de navegación, controles de página, barras de progreso y más.

Además de todas las características que ofrece el framework, su principal objetivo es permitir la construcción de sitios web responsive para dispositivos móviles. (Guest, 2020)

4.2.5 Marco de trabajo ASP.NET MVC V5

El ASP.NET MVC Framework es un framework de aplicaciones web que implementa el patrón modelo-vista-controlador (MVC).

El patrón de arquitectura MVC (model-view-controller) es un patrón de arquitectura que ayuda a crear una separación lógica entre el modelo (información y lógica de negocio), la vista (la lógica de presentación) y el controlador (intermediario entre la vista y el modelo).

Uno de los pilares básicos de ASP.NET MVC es el concepto de enrutamiento (routing), lo que permite a las aplicaciones aceptar peticiones a URL que no se corresponden con ficheros físicos en el servidor. (Microsoft, 2018).

4.2.6 Gestor de base de datos relacionales SQL Server

SQL Server es un sistema de gestión de bases de datos relacionales, o RDBMS (Relational Database Management System), desarrollado y comercializado por Microsoft.

Al igual que otro software RDBMS, SQL Server está construido sobre SQL, un lenguaje de programación estándar para interactuar con las bases de datos relacionales. El servidor SQL está vinculado a Transact-SQL, o T-SQL.

En cada instalación de SQL Server hay 4 bases de datos de sistema, y la capacidad de crear nuevas bases de datos por el usuario, en los cuales los datos están almacenados en tablas.

Estas bases de datos, creadas por parte de los usuarios, incluyen básicamente un archivo de datos (con el sufijo mdf) con las tablas y los distintos objetos a nivel de la base de datos; y un archivo de registro (con el sufijo ldf) con las transacciones abiertas, y transacciones cerradas, Sujeto al modelo de recuperación seleccionado (se puede acumular en el archivo de registro todos los cambios en la base de datos desde el último respaldo). Se puede crear un conjunto de archivos de datos además del principal (con el sufijo ndf) por consideraciones de eficiencia, partición de carga de trabajo entre los discos rígidos, etc. (Microsoft, Documentacion de Sql, 2017)

4.3 Metodología de Desarrollo de Software

4.3.1 Proceso Racional Unificado (RUP)

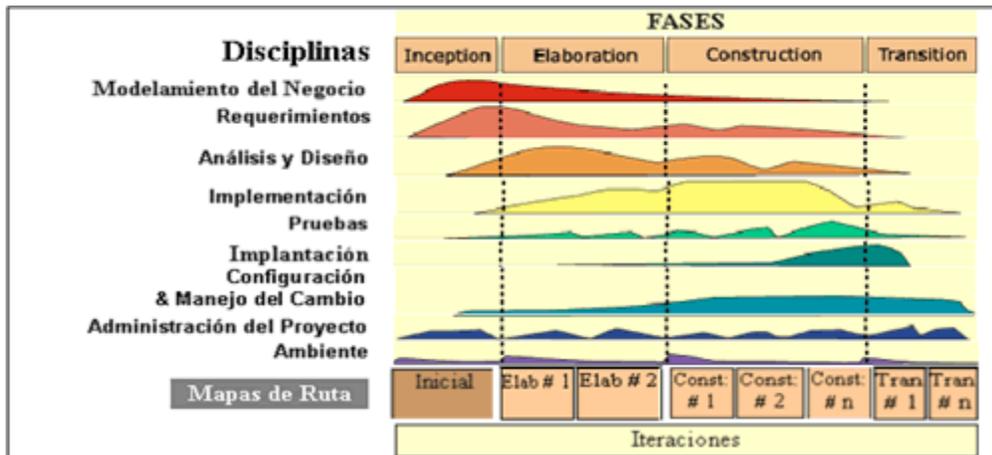
Es una metodología de ingeniería de software basada en programación orientada a objetos, denominada Rational Unified Process o Proceso Racional Unificado (RUP por sus siglas en inglés). Es de las metodologías estándar más completas y mejor estructuradas, que puede emplearse con gran efectividad tanto en proyectos de gran tamaño como en proyectos muy pequeños. Proporciona una manera de delegar tareas y responsabilidades dentro de una organización, asegurando la producción de un software de alta calidad ayudando a resolver las necesidades de los usuarios a través de una serie de pasos o iteraciones para cumplir con un presupuesto y tiempo delimitado. (RUP, 2019)

Organización del proceso RUP:

RUP es una metodología bidimensional que se estructura en fases y en cada fase se desarrollan actividades de diferentes disciplinas, las cuales se describen en la figura 4-1:

- En su eje horizontal se encuentra representado el tiempo y muestra los aspectos del ciclo de vida del proceso.
- En su eje vertical encontramos las diferentes disciplinas del proceso que se agrupan por actividades definidas de acuerdo a su naturaleza. En la imagen que se muestra a continuación podemos ver el enfoque de cada una de las disciplinas con respecto al tiempo y durante cada una de las fases.

Figura 4-1. Fases del RUP



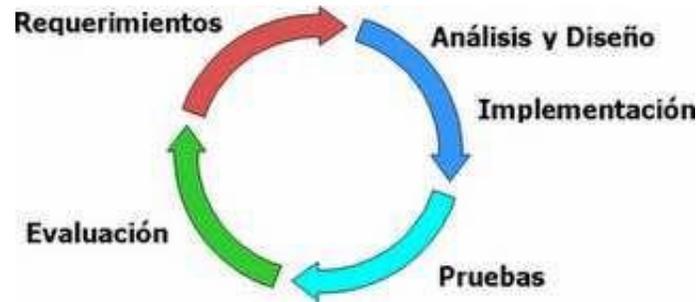
Fuente <https://metodoss.com/metodologia-rup/>. Fases de la Metodología RUP.

Características esenciales:

Dentro del proceso de desarrollo de RUP se menciona 3 características que son esenciales y están bien definidas:

- **Proceso dirigido por casos de uso:** Utiliza los Casos de Uso para el desarrollo y manejo de las especialidades con los artefactos, roles y actividades necesarias. El soporte principal en la implementación de las fases y especialidades del RUP son los Casos de Uso. Un Caso de Uso es una representación de una unidad discreta de trabajo realizada por un usuario (u otro sistema) usando el sistema en operación, está relacionado con los requerimientos y conlleva a su desarrollo e implementación.
- **Proceso iterativo e incremental:** Por medio de este modelo se propone la realización del proyecto a través de una serie de iteraciones, las cuales definen unos objetivos que se deben cumplir en cada una de las iteraciones y así llegar a completar todo el proyecto.

Figura 4-2: Ciclo de vida del software de RUP



Fuente <https://metodoss.com/metodologia-rup/>. Modelo RUP (Rational Unified Process)

Una de las ventajas de este modelo es que entrega pequeños avances al cliente, de esta manera se puede ir realizando pruebas mientras se avanza en otras etapas del proyecto, el proyecto crece de una manera más rápida hasta quedar completado en su totalidad.

- **Proceso centrado en la arquitectura:** Este proceso define los elementos que son de más importancia dentro del sistema. Está influenciado, entre otras, por plataformas de software, sistemas operativos, motores de bases de datos, protocolos, consideraciones de desarrollo como sistemas no heredados y requerimientos no funcionales. A medida que se avanza, RUP realiza una serie de mejoras sucesivas en una arquitectura ejecutable, creada como un prototipo evolutivo.

Fases de RUP

El ciclo de vida del software de RUP se divide en 4 fases, que se describen en la figura 4-2. Cada una de las fases tiene unos objetivos y puntos de control que ayudan a asegurar el cumplimiento con las expectativas y decidir el paso a la fase siguiente del proyecto. Las siguientes son las fases sobre las cuales se establece RUP:

Inicio: En esta fase se conoce el contexto del negocio, el alcance y la viabilidad del proyecto.

Los productos de la fase de inicio deben ser, principalmente los requerimientos del sistema.

Elaboración: El propósito de la fase de elaboración es analizar el dominio del problema, establecer los cimientos de la arquitectura, desarrollar el plan del proyecto y eliminar los mayores riesgos.

En esta fase se construye un prototipo de la arquitectura, que debe evolucionar en iteraciones sucesivas hasta convertirse en el sistema final.

Se desarrolla además el grueso de las disciplinas de análisis y diseño de la aplicación. Los principales productos de esta fase son: De la disciplina de análisis, modelado de casos de uso y modelo de dominio o conceptual. De la disciplina de diseño, diagrama de clases y diagrama de secuencia.

Construcción: La finalidad principal de esta fase es alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones.

Durante esta fase se realiza principalmente la codificación del sistema.

Los productos de la fase deben ser los diferentes componentes del sistema funcionales y probados.

Transición: La finalidad de la fase de transición es poner el producto en manos de los usuarios finales, por lo tanto, requiere entrenar al usuario en el manejo del producto, en general tareas relacionadas con el ajuste, configuración, instalación y puesta en servicio del producto.

Los productos de esta fase son: instalación del sistema, capacitación del usuario y puesta en servicio.

4.3.2 Lenguaje de Modelado Unificado (UML)

El lenguaje de modelado unificado (UML por sus siglas en inglés) se ha convertido en el lenguaje de modelado estándar para el paradigma de programación orientado a objetos. Construido a partir las investigaciones de James Rumbaugh (OMT), Grady Booch (Técnica Booch) e Ivar Jacobson (OOSE). UML surgió como una propuesta para:

- Estandarizar un lenguaje de modelado orientado a objetos sobre el proceso de desarrollo.
- Generar conceptos de modelado dentro de la comunidad orientada a objetos.

Ofrecer una semántica para el modelamiento de software en diferentes circunstancias. (que es el lenguaje unificado de modelado uml, 2019)

Componentes de UML

Los componentes de UML representan las diferentes perspectivas (vistas) del sistema, a partir de diagramas específicos utilizando elementos de modelado comunes.

Vista: Una vista es una abstracción construida a partir de una serie de diagramas. Cada vista muestra un aspecto particular del sistema y su conjunto la imagen completa del sistema. El sistema es descrito a partir de aspectos funcionales (estructura y comportamiento) y no funcionales (requerimientos de tiempo, con habilidad, despliegue, etc.).

Diagramas: Permiten describir el contenido de una vista. UML presente dos tipos de diagramas: estructural y comportamiento. Un diagrama puede ser parte de varias vistas, depende del contenido del diagrama. Los diagramas más comunes de UML son:

- **Diagrama de Casos de Uso:** Es la representación gráfica de todos los actores, casos de uso e interacciones, identificados para el sistema.
- **Diagrama de Dominio:** Representa las posibles entidades que conforman el dominio de la aplicación.
- **Diagrama de Clase:** Muestra una agrupación de elementos de modelado declarativo (estáticos), tales como clases, tipos y sus contenidos y relaciones.
- **Diagrama de Secuencia:** Representa la interacción entre objetos, poniendo el foco en la secuencia de los mensajes que se intercambian a medida pasa el tiempo durante la secuencia de eventos.
- **Diagrama de Estados:** Representa los posibles estados que pueda tener un objeto, a partir de los eventos que se generan.

- **Diagrama de Actividades:** Representa los procesos de negocios de alto nivel, incluidos el flujo de datos. Es utilizado para describir las actividades en una operación.
- **Diagrama de Componentes:** Representa la estructura física del código en términos de los componentes de código, tales como: los componentes, sus relaciones, interacciones y sus interfaces públicas.
- **Diagrama de Despliegue:** Muestra la arquitectura física del sistema (hardware y software). Las máquinas físicas y los procesadores se representan como nodos y la construcción interna puede ser representada por nodos o artefactos embebidos.
- **Diagrama de Comunicación:** Es un diagrama que enfoca la interacción entre líneas de vida, donde es central la arquitectura de la estructura interna y como ella se corresponde con el pasaje de mensajes. La secuencia de los mensajes se da a través de un esquema de numerado de la secuencia.

5 DESARROLLO DEL SISTEMA

El desarrollo del sistema de gestión de información para el control y seguimiento de las correspondencias físicas del laboratorio Divino Niño “SGC”, se realizó siguiendo la metodología de ingeniería de software Proceso Racional Unificado (RUP).

A continuación, se describen las actividades realizadas, así como los resultados obtenidos en cada una de las fases del proceso RUP, durante el desarrollo del sistema:

5.1 Fase de Inicio

En esta fase se delimitaron los requerimientos del sistema a partir de la visión del mismo planteada por el Gerente General del Laboratorio y se estudió su factibilidad. Los resultados (entregables) de la misma son:

- a) Requerimientos del sistema**
- b) Estudio de factibilidad.**

5.1.1 Requerimientos del Sistema

5.1.1.1 Requerimiento General

El Laboratorio Divino Niño requiere un sistema de gestión de información que sistematice el registro de la correspondencia física, facilitando el seguimiento de ésta a fin de asegurar el control de su gestión.

5.1.1.2 Requerimientos Funcionales

Son las funciones específicas del proceso de gestión de correspondencia que el sistema va a automatizar. El cuadro a continuación resume los requerimientos funcionales identificados para el desarrollo de la aplicación:

Tabla 1: Requerimientos Funcionales

Código	Requerimiento	Descripción
RF1	El sistema debe permitir registrar la correspondencia del Laboratorio.	<p>Para efectos del sistema, se denominará correspondencia a todo documento físico (cartas, facturas, contratos, etc.) que se recibe o se envía².</p> <p>La información mínima a registrar de cada correspondencia física es:</p> <ul style="list-style-type: none"> a) Remitente b) Destinatario c) Referencia (Identificador) d) Prioridad o urgencia e) Observación o comentario f) Fecha y hora de registro. g) Si se espera una respuesta a esta correspondencia, se debe registrar la fecha idónea de la respuesta. <p>Asimismo, si la correspondencia recibida es contestación o está relacionada a otra correspondencia, se debe registrar dicha relación.</p>
RF1.1	El sistema debe permitir registrar la correspondencia recibida o entrante.	Se considera correspondencia entrante, todo documento físico dirigido a una persona dentro de la organización, que se recibe en el mostrador de la recepción del Laboratorio.

² El Laboratorio requiere que el sistema lleve el control de las entregas de los resultados de estudios clínicos contratados por los clientes al Laboratorio, sin embargo, debido a la naturaleza de estos, no se considerarán correspondencia, sino que se tratarán de forma diferente (ver requerimiento RF4).

Código	Requerimiento	Descripción
		El sistema debe solicitar y registrar la información de la correspondencia, y una vez finalizado el registro, debe enviar notificación al destinatario a través de correo electrónico.
RF1.2	El sistema debe permitir registrar la correspondencia enviada o saliente.	<p>Se considera correspondencia saliente, todo documento físico que se envía a una persona (natural o jurídica) externa al Laboratorio.</p> <p>La correspondencia saliente se registra en la recepción del Laboratorio antes de ser enviada a entrega por mensajería (entrega física al destinatario por parte del mensajero del Laboratorio) o deposita en la oficina de correos.</p> <p>El Sistema debe permitir registrar:</p> <ul style="list-style-type: none"> a) La información básica de cada correspondencia saliente. b) La evidencia de entrega de la correspondencia saliente enviada por mensajería (a través de actualización del registro original).
RF3	El sistema debe permitir dar seguimiento a la correspondencia.	Debe permitir consultar la información de la correspondencia enviada y recibida de manera dinámica. La información a consultar debe poderse definir a través de diferentes parámetros de filtros a partir de los principales atributos de la correspondencia registrada y su estado de gestión.
RF4	El sistema debe permitir registrar la entrega de resultados de estudios clínicos a los clientes.	El sistema debe permitir a la recepcionista registrar la entrega de los resultados de estudios clínicos directamente al cliente o a quien este delegue, en la recepción del Laboratorio, una vez verifique la información correspondiente (cédula y factura).

Código	Requerimiento	Descripción
		<p>El Sistema debe permitir registrar:</p> <ul style="list-style-type: none"> a) Nombre de quien recibe el resultado. b) Fecha en que se entrega el resultado. c) Observación o comentario. d) <p>El sistema debe obtener la información de los resultados disponibles para entrega a través de interacción con el Sistema de Estudios Clínicos del Laboratorio.</p>
RF5	El sistema debe permitir generar reportes de las correspondencias.	Debe permitir obtener información consolidada o a detalle de la correspondencia en forma de reportes conforme la estructura establecida por el Laboratorio.
RF6	El sistema debe permitir registrar la información básica (catálogos) necesarios para su funcionamiento.	El sistema debe permitir a un usuario administrador registrar de forma centralizada la información de los catálogos del sistema.

5.1.1.3 Requerimientos No Funcionales

Consiste en otros requisitos técnicos que el sistema debe cumplir que no se refiere a funcionalidad (el qué) sino a características (el cómo) del sistema:

Tabla 2: Requerimientos No Funcionales

Código	Requerimiento	Descripción
RNF1	El sistema debe ser una aplicación web.	Debe ser compatible con los estilos web2, su interfaz debe ser autoadaptable (<i>responsive</i>) a diferentes factores de forma de diferentes dispositivos (PC, portátil, tableta y smarthphone).
RNF2	El sistema debe ser compatible con los demás sistemas del laboratorio.	<p>Su análisis y diseño, así como implementación debe considerar y ser compatible con los demás sistemas del laboratorio, tanto en estilo de interfaz de usuario como en la tecnología empleada para desarrollo, tanto a nivel de aplicación como de la base de datos.</p> <p>El proceso de desarrollo del sistema debe regirse conforme las políticas y estándares definidos en el Laboratorio para el desarrollo de software.</p>
RNF3	El sistema debe implementar seguridad de acceso.	Debe implementar restricciones de acceso, a través de roles de usuario para garantizar la separación de responsabilidades y el control interno.

5.1.2 Factibilidad del Sistema

Fue necesario conocer las condiciones actuales del Laboratorio de Investigaciones Clínicas Divino Niño antes de emprender el desarrollo e implementación del sistema, a fin de determinar la viabilidad de la implementación del sistema. Para ello se realizó un estudio mínimo de factibilidad que dimensiona técnica, económica y operativamente los requerimientos de recursos y esfuerzo para desarrollo y los contraste con las capacidades de la organización para determinar si la propuesta de implementación del SGC es viable.

5.1.2.1 *Factibilidad Técnica*

Esta dimensión del estudio analiza los recursos tecnológicos (hardware y software) necesarios para la implementación del proyecto, estableciendo los requerimientos mínimos de capacidad para ejecutar las distintas fases del mismo, tanto la de desarrollo y pruebas como la de producción per sé.

Asimismo, determina el perfil del personal requerido y el esfuerzo para el desarrollo del sistema.

➤ **Hardware**

➤ **Servidor**

A pesar de que el SGC va a operarse en la red local del Laboratorio de Investigaciones Clínicas Divino Niño, sin acceso externo al Internet, requiere para su correcto funcionamiento, además del sistema, las aplicaciones SQL Server e IIS, las cuales tienen los siguientes requerimientos:

Tabla 3: Requerimientos mínimos de hardware del servidor.

Característica	Mínimo	Recomendado
Procesamiento (Frecuencia x núcleo)	1.4 GHz x 1	2.0 GHz x 2
Memoria principal (Capacidad)	1 GB	4 GB
Almacenamiento (Capacidad)	6 GB	100 GB

En vista del tipo de aplicación y volumen de transacciones (en otros sistemas del Laboratorio) y que se pretende almacenar imágenes de documentos, se establecen las siguientes características técnica mínimas para el servidor:

- **Tipo:** Servidor de clase empresarial tipo torre con capacidad de fuente de poder y almacenamiento redundante.
- **Procesamiento:** Microprocesador Intel Xeon @ 2.0 GHz (1 CPU mínimo, 2 deseable).
- **Memoria Principal:** 16GB mínimo, 32GB deseable.
- **Almacenamiento:** 500 GB mínimo, 2 TB deseable.
- **Tarjeta de red:** Capacidad de 1 Gigabit.

➤ Computadora de Desarrollo

El equipo de desarrollo requiere soportar la plataforma de desarrollo Microsoft .Net Framework y el IDE de desarrollo Microsoft Visual Studio, la versión comunitaria del gestor de base de datos SQL Server Developer Edition, así como las aplicaciones ofimáticas Microsoft Office, las cuales tienen los siguientes requerimientos de hardware combinado:

Tabla 4: Requerimientos de hardware de la computadora de desarrollo.

Característica	Mínimo	Recomendado
Procesamiento (Frecuencia x núcleo)	1.4 GHz x 1	2.0 GHz x 2
Memoria principal (Capacidad)	4 GB	8 GB
Almacenamiento (Capacidad)	6 GB	60 GB

A fin de asegurar capacidad suficiente y movilidad, se establecen las siguientes características técnica mínimas para el equipo de desarrollo:

- **Tipo:** Computadora portátil de alta capacidad.
- **Procesamiento:** Microprocesador Intel Core I5 (1 CPU, 2 cores @ 2.0 GHz)
- **Memoria Principal:** 6 GB.
- **Almacenamiento:** 500 GB SSD.
- **Tarjeta de red:** Capacidad de 1 Gigabit.

➤ **Computadora de Usuario Final**

El equipo de usuario final solo requiere soportar los aplicativos de Microsoft Office y un navegador de Internet para operar la aplicación:

Tabla 5: Requerimientos de hardware de computadora de usuario.

Característica	Mínimo	Recomendado
Procesamiento (Frecuencia x núcleo)	1.4 GHz x 1	2.0 GHz x 2
Memoria principal (Capacidad)	2 GB	4 GB
Almacenamiento (Capacidad)	4 GB	10 GB

A fin de asegurar margen de crecimiento, se establecen las siguientes características técnica mínimas para el equipo de usuario final:

- **Tipo:** Computadora de escritorio.
- **Procesamiento:** Microprocesador Intel Pentium o superior (1 CPU x 2 cores @ 2.0 GHz)
- **Memoria Principal:** 6 GB.
- **Almacenamiento:** 500 GB.
- **Tarjeta de red:** Capacidad de 1 Gigabit.

➤ **Software**

➤ **Software en servidor de producción:**

Para la puesta en producción del SGC se requieren los siguientes:

- **Microsoft Windows Server** (versión 2012 o superior)
- **Microsoft SQL Server** (edición estándar o empresarial, licenciado por CPU)

➤ **Software de Desarrollo:**

Para el desarrollo del SGC se requieren los siguientes programas:

- **Microsoft Windows 10** (edición profesional)
- **Microsoft SQL Server Developer Edition** (versión 2014 o superior)

- **Microsoft Visual Studio** (versión 2015 o superior)
- **Microsoft Office** (versión 2013 o superior)
- **Software de usuario final:**

Para la operación del SGC, el usuario final requiere los siguientes programas:

- **Microsoft Windows 10** (edición profesional)
- **Microsoft Office** (versión 2013 o superior)
- **Navegador Google Chrome** (versión 98 o superior)

➤ **Personal para desarrollo del Sistema**

➤ **Perfil del Desarrollador de Software:**

Para el desarrollo del SGC se requiere un desarrollador de software con conocimiento y capacidad de ejecutar todo el ciclo de vida y roles del desarrollo del sistema, específicamente con dominio de la metodología de ingeniería de software seleccionada y las tecnologías de desarrollo definidas por el cliente:

- Dominio de la metodología de desarrollo RUP
 - Dominio del lenguaje de modelado UML
 - Análisis y diseño de sistemas en el paradigma de programación orientada a objetos con UML para RUP.
 - Programación de aplicaciones Web con Microsoft ASP.Net empleando arquitectura MVC.
 - Diseño, creación y gestión de bases de datos con Microsoft SQL Server
 - Realización de pruebas, implantación del sistema y capacitación al usuario final.
- **Esfuerzo:**

En función del alcance del sistema, debido a que un solo desarrollador asume todos los roles que establece la metodología RUP, se definen las horas de trabajo (esfuerzo) por cada una de las fases y el esfuerzo total del sistema:

Tabla 6: Cálculo del esfuerzo del proyecto.

Fase RUP	Horas
Inicio	70
Elaboración	100
Construcción	560
Implantación	80
Esfuerzo Total	810

5.1.2.2 Factibilidad Económica

El estudio de factibilidad económica permite dimensionar los costos asociados a la implementación del sistema y por tanto su viabilidad de implementación para la organización.

Para ello, se consolidan los costos de recursos tecnológicos y talento humano requerido para su desarrollo.

En el caso del SGC, el Laboratorio Divino Niño cuenta con el hardware requerido (servidor y equipos de usuario final) que cumplen con los requerimientos técnicos, así como la infraestructura tecnológica de redes de comunicaciones interna para la puesta en producción:

- ✓ **Servidor de Aplicaciones de Producción:**
 - Marca: HP, Modelo: ML310
 - Procesador: Intel Xeon E3-1240V3
 - Memoria Principal: 32 GB
 - Almacenamiento: 1 TB
 - Tarjeta de red: Gigabit
 - Sistema Operativo: Windows Server 2012
 - AppServer: Microsoft IIS.

- ✓ **Servidor de Base de Datos:**
 - Marca: HP, Modelo: ML310
 - Procesador: Intel Xeon E3-1240V3
 - Memoria Principal: 32GB
 - Almacenamiento: 1 TB
 - Tarjeta de red: Gigabit

- Sistema Operativo: Windows Server 2012
- RDBMS: SQL Server estándar 2014

✓ **Computadora del usuario final:**

- Marca: DELL, Modelo: OPTIPLEX 3090
- Procesador: Intel Core i3-1115G4
- Memoria Principal: 8 GB DDR4
- Disco Duro: SATA 500GB
- Tarjeta de red: Gigabit
- Sistema Operativo: Windows 10 Pro 64 bits

La autora de este documento empleó su equipo personal como el equipo de desarrollo:

✓ **Computadora de desarrollo:**

- Marca: ASUS, Modelo: A55V
- Procesador: Intel Core i7-3630QM
- Memoria Principal: 6 GB DDR3
- Disco Duro: SATA de 2.5" de 750GB
- Tarjeta de red: Gigabit
- Sistema Operativo: Windows 10 Pro 64 bits
- Plataforma de desarrollo: Visual Studio 2019 Community Edition
- RDBMS: SQL Server Developer Edition 2019

Para para construir el SGC se emplearon versiones gratuitas de las herramientas de desarrollo de Microsoft, lo cual no generó costos.

Asimismo, la autora proveyó la mano de obra para el desarrollo del proyecto. Por tanto, el mismo no generó costos adicionales para el Laboratorio. Sin embargo, a fines de dimensionar el costo del proyecto, a continuación, se calculan los costos asociados:

Tabla 7: Resumen de costos del SGC

Línea	Cantidad	Descripción	Costo unitario (US\$)	Costo total (US\$)
1	2	Servidor de producción: aplicación y base de datos (incluye licenciamiento de sistema operativo).	8,625.00	17,250.00
2	2	Computadora de usuario final (incluye licenciamiento de sistema operativo y office)	793.50	1,587.00

Línea	Cantidad	Descripción	Costo unitario (US\$)	Costo total (US\$)
3	1	Computadora de desarrollo (incluye licenciamiento de sistema operativo office)	1,150.00	1,150.00
TOTAL HARDWARE				19,987.00
4	1	SQL Server estándar 2019 (2 núcleos)	3,586.00	3,586.00
TOTAL SOFTWARE				3,586.00
5	810	Servicios de desarrollo software e implantación (Horas).	4.00	3,240.00
TOTAL SERVICIOS DE DESARROLLO				3,240.00
TOTAL DEL SISTEMA				26,813.00

Los costos totales de desarrollo e implementación del sistema para el Laboratorio serían de veintiséis mil ochocientos trece dólares de los Estados Unidos de América (US26,813.00).

5.1.2.3 Factibilidad Operativa

Esta dimensión del estudio analiza la viabilidad de poner en operación y mantener en servicio el sistema.

Se califica viable por cuanto el sistema propuesto se está diseñando de manera muy intuitiva, basándose en los estándares de diseño de interfaz de usuario adoptados actualmente en el Laboratorio, lo que hará que el usuario se sienta familiarizado y cómodo con la interfaz, permitiendo una curva de aprendizaje corta.

Adicionalmente se garantiza que el usuario tenga los conocimientos y destrezas necesarias para operar la aplicación, a través de capacitación y manual de usuario.

5.2 Fase de Elaboración

En esta fase se realizó el análisis y diseño del sistema, así como la implementación de la arquitectura ejecutable del mismo (programas base que establecen la forma en que se realizará el procesamiento de las diferentes funciones) a partir del patrón estructural empleado MVC (Modelo – Vista - Controlador) y los marcos de trabajo seleccionados para ello. Los resultados (entregables) de la misma son:

a) Análisis del sistema:

- i. Modelo de Casos de Uso.
- ii. Modelo de Dominio del Problema o Conceptual.

b) Diseño del Sistema:

- i. Modelo de Clases
- ii. Modelo de Interacción

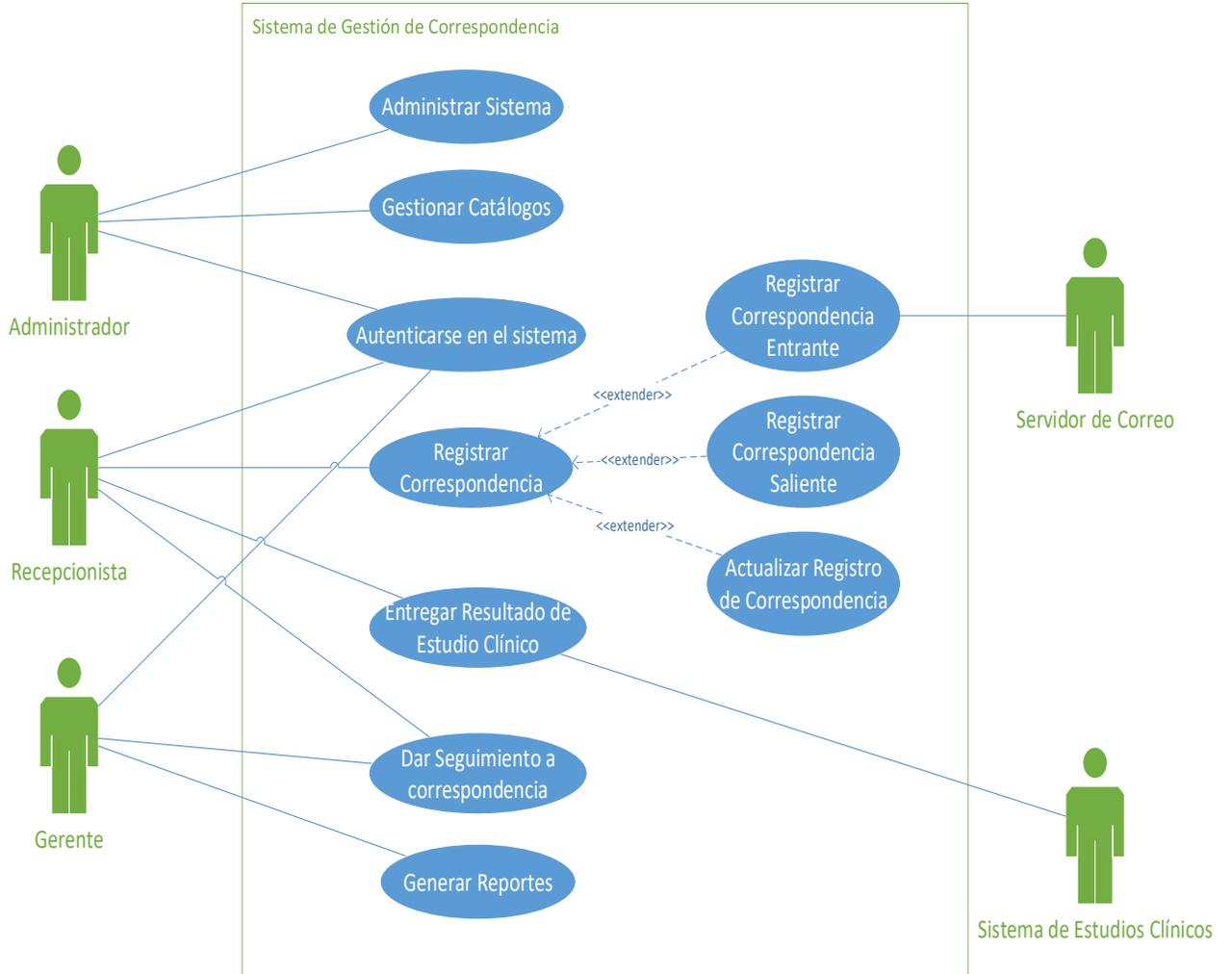
a) Análisis del Sistema:

5.2.1 Modelo de Casos de Uso

El modelo de Casos de Uso, también conocido como Modelo de Comportamiento describe los diferentes escenarios (éxitos y errores) de interacción de los actores (usuarios) con el sistema para la realización de los objetivos o funciones del mismo. Es un artefacto que define una secuencia de acciones que da lugar a un resultado de valor observable. A continuación, se presenta el diagrama del modelo de casos de uso. Este se compone a partir de actores (figuras de forma humana) y su interacción con las funcionalidades del sistema (óvalos):

5.2.1.1 Diagrama de Casos de Uso

Figura 5-1 Diagrama de Caso de Uso del Sistema



5.2.1.2 Actores

Para el SGC se han identificado los siguientes actores:

Actores Principales:³

- **Administrador:** Es el usuario encargado de la administración del Sistema. Tiene todos los permisos para la gestión dentro del aplicativo.
- **Recepcionista:** Es la persona que está a cargo del registro de la correspondencia y su seguimiento en el sistema. Únicamente goza de los permisos asignados a su cargo.
- **Gerente:** Es la persona que tiene la responsabilidad de supervisar el estado de la gestión de correspondencia. Únicamente goza de los permisos asignados a su cargo.

Actores Secundarios:⁴

- **Servidor de Correo:** Sistema de correo electrónico que emplea el sistema de correspondencia para notificar al destinatario la correspondencia recibida.
- **Sistema de Estudios Clínicos:** Corresponde al sistema de gestión de información que lleva el registro y control de los estudios contratados por los clientes y los resultados obtenidos. Interactúa con el SGC para entregar los resultados.

³ Se denominan **Actores Primarios o Principales** a los usuarios que interactúan directamente con la interfaz de usuario del sistema.

⁴ Se denominan **Actores Secundarios o Auxiliares** a aquellos componentes de cómputo o automatización con los que requiere interactuar el sistema para realizar sus funciones.

5.2.1.3 Especificación de Casos de Uso

➤ CU1: Autenticarse en el sistema

Tabla 8: CU1: Autenticarse en el sistema

Caso de Uso:	CU1: Autenticarse en el sistema
Autor:	Kerolin Alemán
Fecha:	12/07/2021
Requerimientos relacionados:	RNF3
Objetivo:	Garantizar que solo usuarios autorizados tengan acceso al sistema, asimismo, que éstos puedan utilizar las funcionalidades a las que se les ha otorgado permiso, conforme su rol en la organización.
Precondiciones:	El usuario ha sido debidamente registrado en el sistema.
Final exitoso:	El nombre de usuario y contraseña provistos por el actor coinciden con un usuario registrado en el sistema y éste permite el acceso a las funcionalidades del mismo conforme al rol y permisos configurados.
Final fallido:	<p>El nombre de usuario y/o contraseña provistos por el actor no coinciden con un registro de usuario en el sistema, o el actor ha seleccionado la opción de cancelar, y por tanto el sistema deniega el acceso, mostrando un mensaje de error de inicio de sesión.</p> <p>El actor puede reintentar autenticarse o abandonar el sistema.</p>

Flujo principal:

Actor (cualquier actor primario)	Sistema
1. Solicita acceso al sistema	2. Requiere al actor el nombre de usuario y la contraseña.

Actor (cualquier actor primario)	Sistema
<p>3. Provee la información solicitada</p>	<p>4. Verifica que la información provista por el actor se corresponde con un registro de usuario activo.</p> <p>De ser así: Concede al actor acceso al sistema con los permisos correspondientes conforme su rol.</p> <p>Caso contrario: Presenta mensaje de notificación de error y deniega el acceso al sistema.</p>

Flujo Alternativo:

Actor (cualquier actor primario)	Sistema
<p>1. Elige cancelar la solicitud de acceso en el paso #3 del flujo principal.</p>	<p>2. Cancela la operación, no concediendo el acceso al usuario.</p>

➤ **CU2: Administrar Sistema**

Tabla 9: CU2: Administrar Sistema

Caso de Uso:	CU2: Administrar Sistema
Autor:	Kerolin Alemán
Fecha:	12/07/2021
Requerimientos relacionados:	RNF3
Objetivo:	Gestionar los usuarios, la seguridad y la configuración del sistema.
Precondiciones:	Estar autenticado en el sistema.
Final exitoso:	En dependencia de la operación elegida, se da el alta, actualización o desactivación de un registro de usuario, seguridad o configuración del sistema.
Final fallido:	No se realiza la operación porque el actor la cancela.

Flujo principal:

Actor (Administrador)	Sistema
1. Selecciona una opción de administración del sistema.	2. Presenta el listado de los registros existentes de la opción de administración seleccionada. Permite Creación de un nuevo registro, o Actualización, Eliminación de un registro existente.
3. Selecciona la operación que desea realizar.	4. Si el actor seleccionó Creación: Solicita la información correspondiente. (Continúa en paso #5) Actualización: Presenta la información del registro y permite su edición. (Continúa en paso #7) Eliminación:

Actor (Administrador)	Sistema
	Presenta la información del registro y solicita confirmación de eliminación. (Continúa en paso #9)
5. Provee la información solicitada	6. Valida la información recibida y si es correcta conforme a las reglas establecidas, crea el nuevo registro con la información y finaliza. En caso contrario, notifica error y regresa al paso # 4.
7. Actualiza los campos correspondientes del registro y selecciona guardar los cambios.	8. Valida la información recibida y si es correcta conforme a las reglas establecidas, actualiza el registro. En caso contrario, notifica error y regresa al paso # 4.
9. Confirma eliminación del registro.	10. Elimina o marca el registro correspondiente como desactivado (en dependencia de la política establecida para la entidad correspondiente, y finaliza.

Flujo Alternativo:

Actor (Administrador)	Sistema
1. En alguno de los pasos del flujo principal que le competen, elige cancelar la operación.	2. Cancela la operación, no realizando la Creación, Actualización o Eliminación solicitada.

➤ **CU3: Gestionar Catálogos**

Tabla 10: CU3: Gestionar Catálogos

Caso de Uso:	CU3: Gestionar Catálogos
Autor:	Kerolin Alemán
Fecha:	12/07/2021
Requerimientos relacionados:	RF6

Objetivo:	Permitir el registro de la información base del sistema (catálogos) requerida para su funcionamiento.
Precondiciones:	Estar autenticado en el sistema.
Final exitoso:	Se realiza un nuevo registro, actualización de un registro existente o eliminación de un registro de un catálogo del sistema.
Final fallido:	No se realiza la operación porque el actor la cancela.

Flujo principal:

Actor (Administrador)	Sistema
1. Selecciona una opción de gestión de catálogos.	2. Presenta el listado de los registros existentes en el catálogo. Permite Creación de un nuevo registro, o Actualización, Eliminación de un registro existente.
3. Selecciona la operación que desea realizar.	4. Si el actor seleccionó Creación: Solicita la información correspondiente. (Continúa en paso #5) Actualización: Presenta la información del registro y permite su edición. (Continúa en paso #7) Eliminación: Presenta la información del registro y solicita confirmación de eliminación. (Continúa en paso #9)
5. Provee la información solicitada	6. Valida la información recibida y si es correcta conforme a las reglas establecidas, crea un nuevo registro en el catálogo y finaliza. En caso contrario, notifica error y regresa al paso # 4.

Actor (Administrador)	Sistema
7. Actualiza los campos correspondientes del registro y selecciona guardar los cambios.	8. Valida la información recibida y si es correcta conforme a las reglas establecidas, actualiza el registro y finaliza. En caso contrario, notifica error y regresa al paso # 4.
9. Confirma eliminación del registro.	10. Elimina el registro y finaliza.

Flujo Alternativo:

Actor (Administrador)	Sistema
1. En alguno de los pasos del flujo principal que le competen, elige cancelar la operación.	2. Cancela la operación, no realizando la Creación, Actualización o Eliminación del registro.

➤ **CU4: Registrar Correspondencia**

Tabla 11: CU4: Registrar Correspondencia

Caso de Uso:	CU4: Registrar Correspondencia
Autor:	Kerolin Alemán
Fecha:	12/07/2021
Requerimientos relacionados:	RF1
Objetivo:	Registrar la correspondencia física para su adecuado control y seguimiento en el sistema.
Precondiciones:	Se ha recibido correspondencia física entrante o saliente en la recepción del Laboratorio.
Final exitoso:	Registrada la información de la correspondencia física necesaria para el control de su gestión.
Final fallido:	No se realiza el registro de la correspondencia recibida porque el actor la cancela.

Flujo principal:

Actor (Recepcionista)	Sistema
1. Selecciona registrar correspondencia.	<p>2. Presenta listado de las correspondencias registradas. Permite las siguientes opciones:</p> <ul style="list-style-type: none">a) Registrar nueva correspondencia entrante.b) Registrar nueva correspondencia saliente.c) Actualizar, Consultar detalle o Eliminar correspondencia existente.
3. Selecciona la operación que desea realizar.	<p>4. Si el actor seleccionó:</p> <ul style="list-style-type: none">➤ Registrar correspondencia entrante: Invoca al caso de Uso CU4.1 “Registrar correspondencia entrante” y finaliza.➤ Registrar correspondencia saliente: Invoca al caso de Uso CU4.2 “Registrar correspondencia saliente” y finaliza.➤ Actualización del registro actual: Invoca al caso de Uso CU4.3 “Actualizar correspondencia” y finaliza.➤ Consulta de detalle del registro actual: Muestra detalle de la información de la correspondencia. (Continúa en paso #5)

Actor (Recepcionista)	Sistema
	<p>➤ Eliminación del registro actual:</p> <p>Presenta la información del registro y solicita confirmación de eliminación. (Continúa en paso #7)</p>
5. Selecciona regresar.	6. Regresa al paso #2.
7. Confirma eliminación del registro.	8. Elimina el registro y regresa al paso #2.

Flujo Alternativo:

Actor (Recepcionista)	Sistema
1. En el paso #3 del flujo principal elige cancelar.	2. Cancela el registro de correspondencia física y no realiza la operación.

➤ CU4.1: Registrar Correspondencia Entrante

Tabla 12: CU4.1: Registrar Correspondencia Entrante

Caso de Uso:	CU4.1: Registrar Correspondencia Entrante
Autor:	Kerolin Alemán
Fecha:	12/07/2021
Requerimientos relacionados:	RF1.1
Objetivo:	Registrar la información de la correspondencia física recibida para su adecuado control y seguimiento en el sistema, notificando además al destinatario vía correo electrónico.
Precondiciones:	El caso de uso CU4 ha invocado a este caso de uso.
Final exitoso:	Registrada la información de la correspondencia física recibida en el sistema y enviado correo electrónico de notificación al destinatario de la misma.

Final fallido:	<ul style="list-style-type: none"> • No se realiza el registro de la correspondencia recibida porque el actor la cancela. • Parcial: No se envía notificación por correo electrónico al destinatario debido a falla con este servicio. (La correspondencia se registra y el Actor debe notificar al destinatario por otro medio, por ejemplo, vía llamada telefónica).
-----------------------	---

Flujo principal:

Actor (Recepcionista)	Sistema
	1. Solicita la información básica de la correspondencia entrante (ver detalle en descripción de requerimiento RF1).
2. Provee la información solicitada.	<p>4. Valida la información recibida y si es correcta conforme a las reglas establecidas:</p> <ul style="list-style-type: none"> ➤ Crea un nuevo registro de correspondencia. ➤ Envía notificación de recepción de correspondencia al destinatario por correo electrónico. <p>En caso de que la información no sea correcta conforme las reglas establecidas notifica error y regresa al paso # 1.</p> <p>En caso de que no se logre enviar notificación al destinatario por fallo con el servicio de correo electrónico, notifica el error y finaliza.</p>

Flujo Alternativo:

Actor (Recepcionista)	Sistema
1. En el paso #2 del flujo principal elije cancelar.	2. Cancela el registro de correspondencia física recibida y no notifica al destinatario.

➤ **CU4.2: Registrar Correspondencia Saliente**

Tabla 13: CU4.2: Registrar Correspondencia Saliente

Caso de Uso:	CU4.2: Registrar Correspondencia Saliente
Autor:	Kerolin Alemán
Fecha:	12/07/2021
Requerimientos relacionados:	RF1.2
Objetivo:	Registrar la información de la correspondencia física saliente del Laboratorio necesaria para el control de su gestión.
Precondiciones:	El caso de uso CU4 ha invocado a este caso de uso.
Final exitoso:	Registrada la información de la correspondencia física saliente en el sistema.
Final fallido:	No se realiza el registro de la correspondencia enviada porque el actor la cancela.

Flujo principal:

Actor (Recepcionista)	Sistema
	1. Solicita la información básica de la correspondencia saliente (ver detalle en descripción de requerimiento RF1).
2. Provee la información solicitada.	3. Valida la información recibida y si es correcta conforme a las reglas establecidas crea un nuevo registro de correspondencia. En caso de que la información no sea correcta conforme las reglas establecidas notifica error y regresa al paso # 1.

Flujo Alternativo:

Actor (Recepcionista)	Sistema
1. En el paso #2 del flujo principal elije cancelar.	2. Cancela el registro de correspondencia física recibida y no notifica al destinatario.

➤ **CU4.3: Actualizar Correspondencia**

Tabla 14: CU4.3: Actualizar Correspondencia

Caso de Uso:	CU4.2: Registrar Correspondencia Saliente
Autor:	Kerolin Alemán
Fecha:	12/07/2021
Requerimientos relacionados:	RF1, RF1.2
Objetivo:	Actualizar la información de la correspondencia física saliente del Laboratorio necesaria para el control de su gestión, principalmente lo referido al registro de la evidencia de entrega de correspondencia saliente y cambios de estado de la correspondencia.
Precondiciones:	El caso de uso CU4 ha invocado a este caso de uso.
Final exitoso:	Actualizada la información de la correspondencia física en el sistema.
Final fallido:	No se realiza la actualización de la correspondencia porque el actor cancela la operación.

Flujo principal:

Actor (Recepcionista)	Sistema
	1. Presenta la información del registro y permite su edición.
2. Actualiza los campos correspondientes del registro y selecciona guardar los cambios.	3. Valida la información recibida y si es correcta conforme a las reglas establecidas, actualiza el registro y

Actor (Recepcionista)	Sistema
	finaliza. En caso contrario, notifica error y regresa al paso #1.

Flujo Alternativo:

Actor (Recepcionista)	Sistema
1. En el paso #2 del flujo principal elige cancelar.	2. Cancela el registro de correspondencia física recibida y no notifica al destinatario.

➤ **CU5: Entregar Resultado de Estudio Clínico**

Tabla 15: CU5: Entregar Resultado de Estudio Clínico

Caso de Uso:	CU5: Entregar Resultado de Estudio Clínico
Autor:	Kerolin Alemán
Fecha:	12/07/2021
Requerimientos relacionados:	RF4
Objetivo:	Registrar la información de entrega de los resultados de estudios clínicos a los clientes para su control.
Precondiciones:	<p>El cliente o su delegado llegó a la recepción solicitando retirar el informe de resultado de estudio clínico impreso.</p> <p>La recepcionista ha validado que:</p> <ul style="list-style-type: none"> a) La información de identidad del cliente y factura del estudio son correctos, b) El resultado físico se encuentra listo para entrega.
Final exitoso:	Registrada la información de la entrega del resultado.
Final fallido:	No se realiza el registro porque el actor cancela.

Flujo principal:

Actor (Recepcionista)	Sistema
1. Selecciona Entregar Resultado.	2. Presenta listado de los resultados entregados. Permite las siguientes opciones: <ul style="list-style-type: none">• Registrar entrega de resultado.• Actualizar, Consultar detalle o Eliminar registro existente.
3. Selecciona la operación que desea realizar.	4. Si el actor seleccionó: <ul style="list-style-type: none">➤ Registrar entrega de resultado:<p>Solicita la información correspondiente. (Continúa en paso #5).</p>➤ Actualización del registro actual:<p>Presenta la información del registro y permite su edición. (Continúa en paso #7).</p>➤ Consulta de detalle del registro actual:<p>Muestra detalle de la información de la entrega de resultado. (Continúa en paso #9)</p>➤ Eliminación del registro actual:<p>Presenta la información del registro y solicita confirmación de eliminación. (Continúa en paso #7)</p>
5. Provee la información solicitada.	6. Valida la información recibida y si es correcta conforme a las reglas establecidas, crea el registro de entrega de resultado y finaliza.

Actor (Recepcionista)	Sistema
	En caso de que la información no sea correcta conforme las reglas establecidas, notifica error y regresa al paso # 4.
7. Actualiza los campos correspondientes del registro y selecciona guardar los cambios.	8. Valida la información recibida y si es correcta conforme a las reglas establecidas, actualiza el registro. En caso contrario, notifica error y regresa al paso # 4.
9. Confirma eliminación del registro.	10. Elimina el registro y finaliza.

Flujo Alternativo:

Actor (Recepcionista)	Sistema
1. En alguno de los pasos del flujo principal que le competen, elige cancelar la operación.	2. Cancela la operación, no realizando el registro y finaliza.

➤ CU6: Dar Seguimiento a Correspondencia

Tabla 16: CU6: Dar Seguimiento a Correspondencia

Caso de Uso:	CU5: Dar Seguimiento a Correspondencia
Autor:	Kerolin Alemán
Fecha:	12/07/2021
Requerimientos relacionados:	RF3
Objetivo:	Permitir consultar la información de la correspondencia a través de diferentes filtros a fin de poder dar seguimiento a su gestión.
Precondiciones:	Existan registros de correspondencia en el sistema.
Final exitoso:	El Actor obtiene la información requerida para controlar la gestión de correspondencia.

Final fallido:	El Actor cancela la operación de consulta de información.
-----------------------	---

Flujo principal:

Actor (Recepcionista, Gerente)	Sistema
1. Solicita consulta de información de correspondencia.	2. Presenta las diferentes opciones de parámetros de filtro de la información.
3. Selecciona los valores deseados a los parámetros de filtro de la información.	4. Extrae y presenta la información que cumple con los parámetros de filtro seleccionados por el Actor.
5. Revisa la información y si desea, puede seleccionar la opción Exportar y el formato correspondiente.	6. Si el Actor selecciono exportar, entrega la información extraída en un documento en el formato seleccionado por el Actor.

Flujo Alternativo:

Actor (Recepcionista, Gerente)	Sistema
1. En alguno de los pasos del flujo principal que le competen, elije cancelar.	2. Cancela la operación de consulta de información.

➤ CU7: Generar Reportes

Tabla 17: CU7: Generar Reportes

Caso de Uso:	CU7: Generar Reportes
Autor:	Kerolin Alemán
Fecha:	12/07/2021
Requerimientos relacionados:	RF5

Objetivo:	Obtener información consolidada o a detalle de la correspondencia conforme la estructura establecida por el Laboratorio.
Precondiciones:	Existan registros de correspondencia en el sistema.
Final exitoso:	El Actor genera el reporte deseado.
Final fallido:	<ul style="list-style-type: none"> ➤ El reporte no se genera o resulta vacío. ➤ El actor cancela la operación de consulta de información.

Flujo principal:

Actor (Recepcionista, Gerente)	Sistema
1. Solicita generar un reporte.	2. Presenta las diferentes opciones de parámetros de filtro de la información.
3. Selecciona los valores deseados a los parámetros de filtro de la información.	4. Extrae y presenta la información que cumple con los parámetros de filtro seleccionados por el Actor.
5. Revisa la información y si desea, puede seleccionar la opción Exportar y el formato correspondiente.	6. Si el Actor selecciono exportar, entrega la información extraída en un documento en el formato seleccionado por el Actor.

Flujo Alternativo:

Actor (Recepcionista, Gerente)	Sistema
1. En alguno de los pasos del flujo principal que le competen, elije cancelar.	2. Cancela la operación de consulta de información.

5.2.2 Modelo de Dominio del Problema

El modelo de dominio del problema es un modelo conceptual de alto nivel que define los objetos del mundo físico sobre los que trata el problema. Es decir, las entidades de las que se procesará información. Documenta la relación entre las entidades y sus responsabilidades (Larman, UML y Patrones: Una introducción al análisis y diseño orientado a objetos y al proceso unificado, 2003).

En notación UML, un modelo de dominio se representa con un diagrama de clases a la que no se le definen ni atributos ni métodos. Las clases en el diagrama representan las entidades físicas del dominio del problema y cómo se relacionan entre sí.

5.2.2.1 Información base del Dominio del Problema

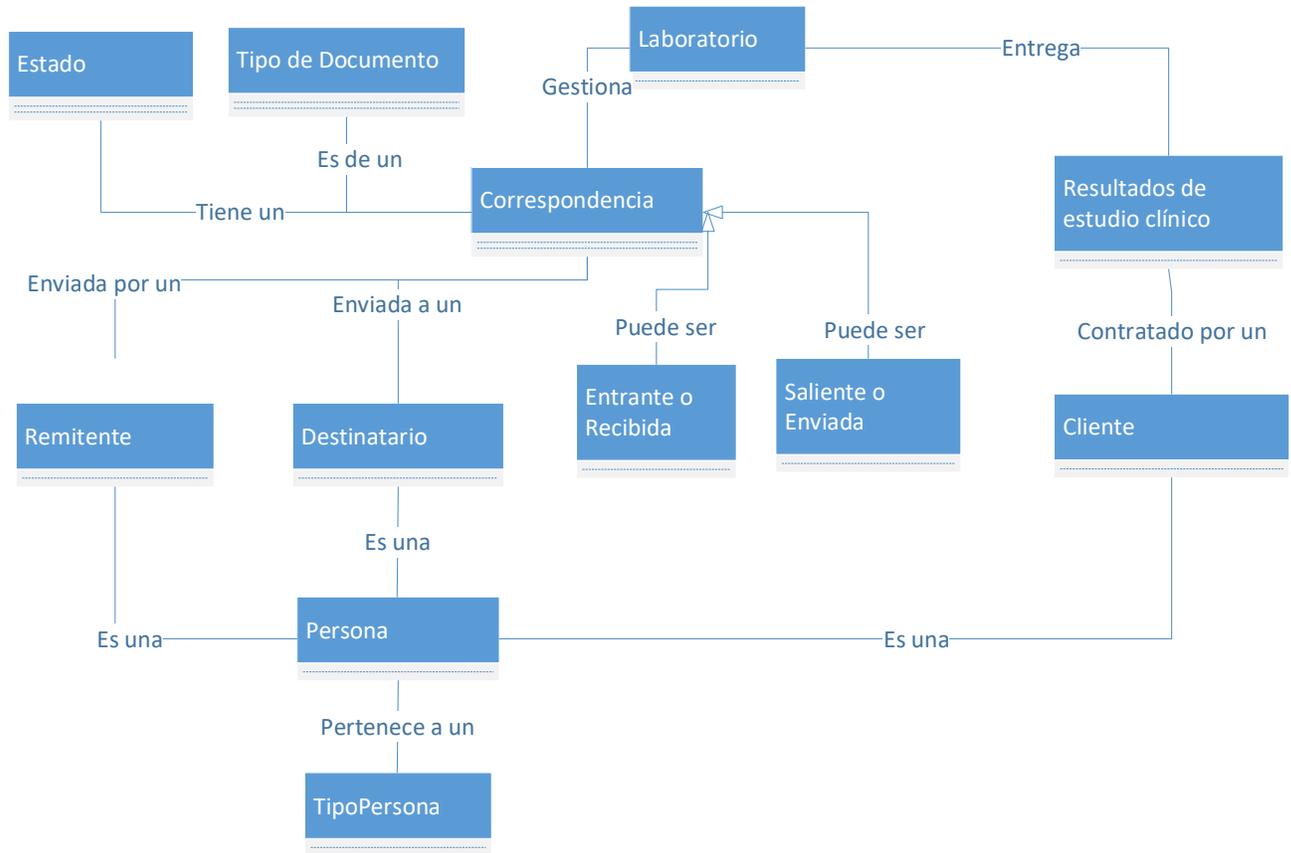
El Laboratorio Divino Niño requiere el sistema para gestionar de forma adecuada toda la correspondencia física, tanto la entrante o recibida como la saliente o enviada, para llevar control del estado en que esta se encuentra, así como el tipo de documento (carta, factura, cobro, etc.).

De cada registro de correspondencia se debe incorporar información de la persona que la envía o remitente como a la que está dirigida, o destinatario. Las personas se deben caracterizar por su relación con la empresa (empleados, externas o clientes).

Además de la correspondencia, se debe registrar la entrega de los resultados de estudios clínicos contratados por los clientes.

5.2.2.2 Diagrama de Dominio

Figura 5-2: Diagrama de Dominio



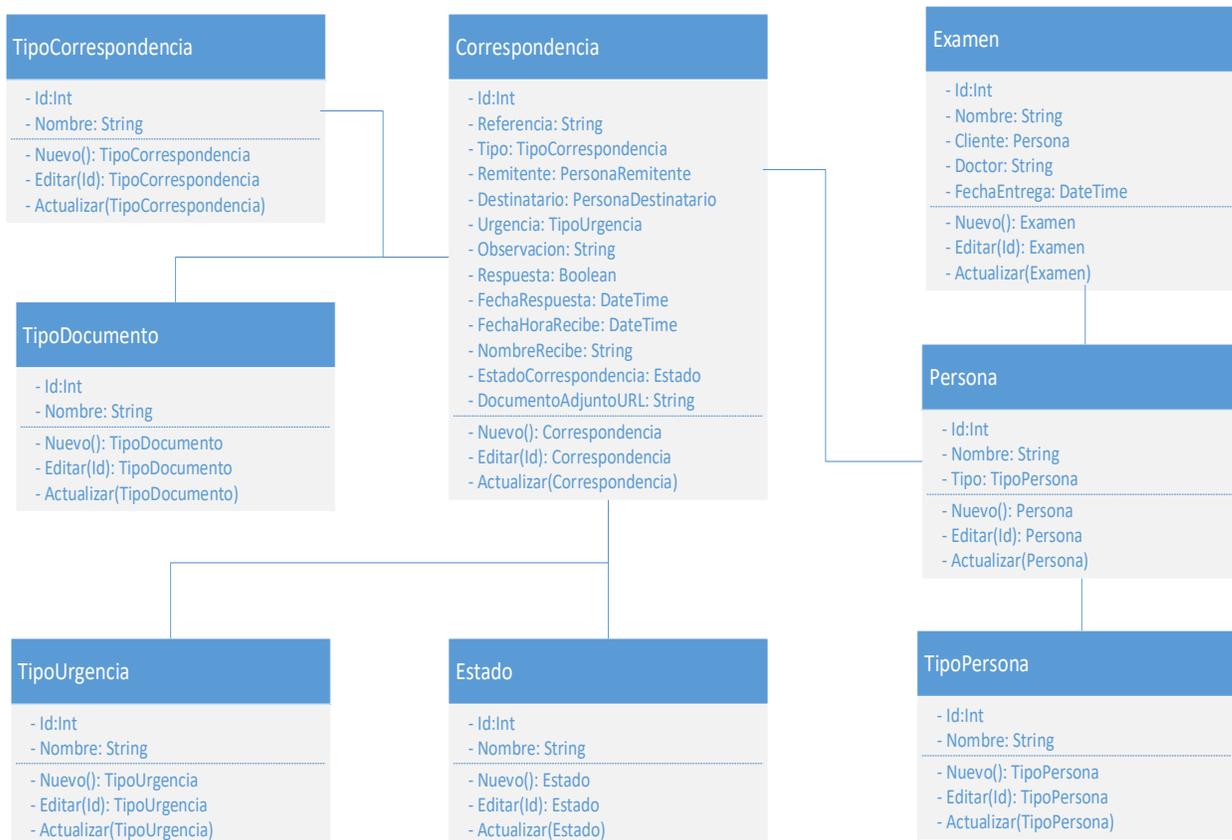
b) Diseño del Sistema:

5.2.3 Modelo de Clases

El modelado de clases es el centro del diseño de la aplicación en el paradigma de Programación Orientada a Objetos. Parte del modelo de dominio del problema y lo refina para obtener las entidades finales del sistema con sus relaciones. La definición de cada clase incluye los atributos (datos) que determinan la identidad y estado de cada objeto en el sistema y los métodos (operaciones) que las clases y sus instancias requieren para interactuar entre sí.

5.2.3.1 Diagrama de Clases

Figura 5-3: Diagrama de Clases



5.2.3.2 Descripción de Clases

Para el desarrollo del SGC, a partir de su diagrama de dominio y la definición de roles entidades / atributos, se identificaron las clases del dominio del sistema y sus relaciones.

- **Clases:**

1. **Correspondencia:** Representa a la correspondencia física en general.

2. **TipoCorrespondencia:** Categoriza la correspondencia por su tipo (saliente o entrante). Debido a que la información registrada en la correspondencia entrante y saliente es la misma, no se consideró crear jerarquía de clases de correspondencia sino una tipificación. En el caso de los Resultados de Estudios Clínicos, si bien es cierto, el Laboratorio lo consideró administrativamente “correspondencia física”, sus atributos de información son bastante diferentes, por lo que se decidió diseñarla como una clase aparte denominada Examen.
 3. **TipoDocumento:** Categoriza la correspondencia por el tipo del documento específico (carta, factura, cobro, etc.).
 4. **TipoUrgencia:** Clasifica la correspondencia en virtud de la urgencia o prioridad de la misma (normal, urgente, emergencia).
 5. **Estado:** Categoriza el estado de la correspondencia específica con relación a la etapa en que se encuentra en el ciclo de gestión del tipo de documento correspondiente, en el Laboratorio.
 6. **Persona:** Representa a la persona que juega un rol específico dentro de dominio del problema.
 7. **TipoPersona:** Categoriza la persona conforme el rol específico (empleado, externo, cliente) dentro del dominio del problema.
 8. **Examen:** Representa a toda persona que juega un rol específico dentro de dominio del problema.
- **Relaciones entre las clases:**
 1. La clase Correspondencia está asociada a la clase TipoCorrespondencia en cuanto cada Correspondencia se clasifica en un TipoCorrespondencia conforme sea entrante (recibida) o saliente (enviada).

2. La clase Correspondencia está asociada a la clase TipoDocumento en cuanto cada Correspondencia se corresponde con un TipoDocumento conforme sea carta, factura, cobro, etc.
3. La clase Correspondencia está asociada a la clase TipoUrgencia en cuanto a cada Correspondencia se le asigna un TipoUrgencia conforme la prioridad o urgencia de atención que su naturaleza implique.
4. La clase Correspondencia está asociada a la clase Estado en cuanto cada Correspondencia puede estar en un Estado conforme la etapa del ciclo de gestión de correspondencia en que se encuentre.
5. La clase Correspondencia está asociada a la clase Persona en cuanto cada Correspondencia tiene una Persona remitente y una Persona destinatario.
6. La clase Persona está asociada a la clase Tipo Persona en cuanto cada Persona se clasifica en un TipoPersona en virtud del rol que juega en la gestión de correspondencia.
7. La clase Examen está asociada a la clase Persona en cuanto cada Examen se corresponde a una Persona de tipo Cliente.

Métodos:

En primera instancia, para todas las clases se identificaron tres métodos básicos. Si en la fase de construcción se detectan nuevos métodos, éstos se incorporarán en el modelo de clases en desarrollo (del *Entity Framework*):

1. **Nuevo ()**: Método de creación de nuevo objeto de clase vacío que servirá para la operación de creación de una nueva instancia de cada entidad.
2. **Editar (Id)**: Método de recuperación de instancia de entidad a partir de su identificación para mostrado y editado.
3. **Actualizar (Clase)**: Método de actualización de datos para persistir la instancia de la entidad.

5.2.4 Diseño de la Base de Datos

En cumplimiento al requerimiento no funcional RNF2, que establece que el desarrollo del sistema debe adoptar las políticas y tecnologías establecidas para el desarrollo de software en el Laboratorio, el sistema debe persistir los datos en un gestor de base de datos relacional, específicamente Microsoft SQL Server, versión 2014. Asimismo, emplear el marco de trabajo Microsoft *Entity Framework* para el mapeo objeto-relacional con un enfoque *database o data-first*⁵, por lo cual fue requerido realizar un diseño de base de datos a partir del modelado de clases.

5.2.4.1 *Diagrama de Base de Datos*

El diagrama de base de datos se corresponde al esquema físico de un diagrama entidad-relación y muestra las diferentes tablas que conforman la base de datos, con sus atributos, así como sus relaciones. A continuación, se muestra el diagrama de base de datos del SGC. Este, a fines de mejor visualización se ha dividido en dos partes:

- a. Tablas de dominio de aplicación.
- b. Tablas para implementar la seguridad de usuarios.

Las tablas de dominio de aplicación, así como sus relaciones mantienen coherencia y correspondencia con las clases definidas en el modelo de clases, a fin de que el modelo generado por el *Entity Framework* a partir del diseño de Base de Datos fuera lo más cercano posible con el modelo de clases.

Las tablas para implementar la seguridad de usuario siguen un patrón simple de usuario-rol en el cual la autorización se realiza a opciones de menú conforme el rol del usuario.

⁵ El enfoque de implementación *database first* sobre el marco de trabajo de mapeo objeto-relacional de Microsoft

Diagrama de Tablas de Dominio de Aplicación

Figura 5-4: Diagrama de Tablas de Dominio de Aplicación

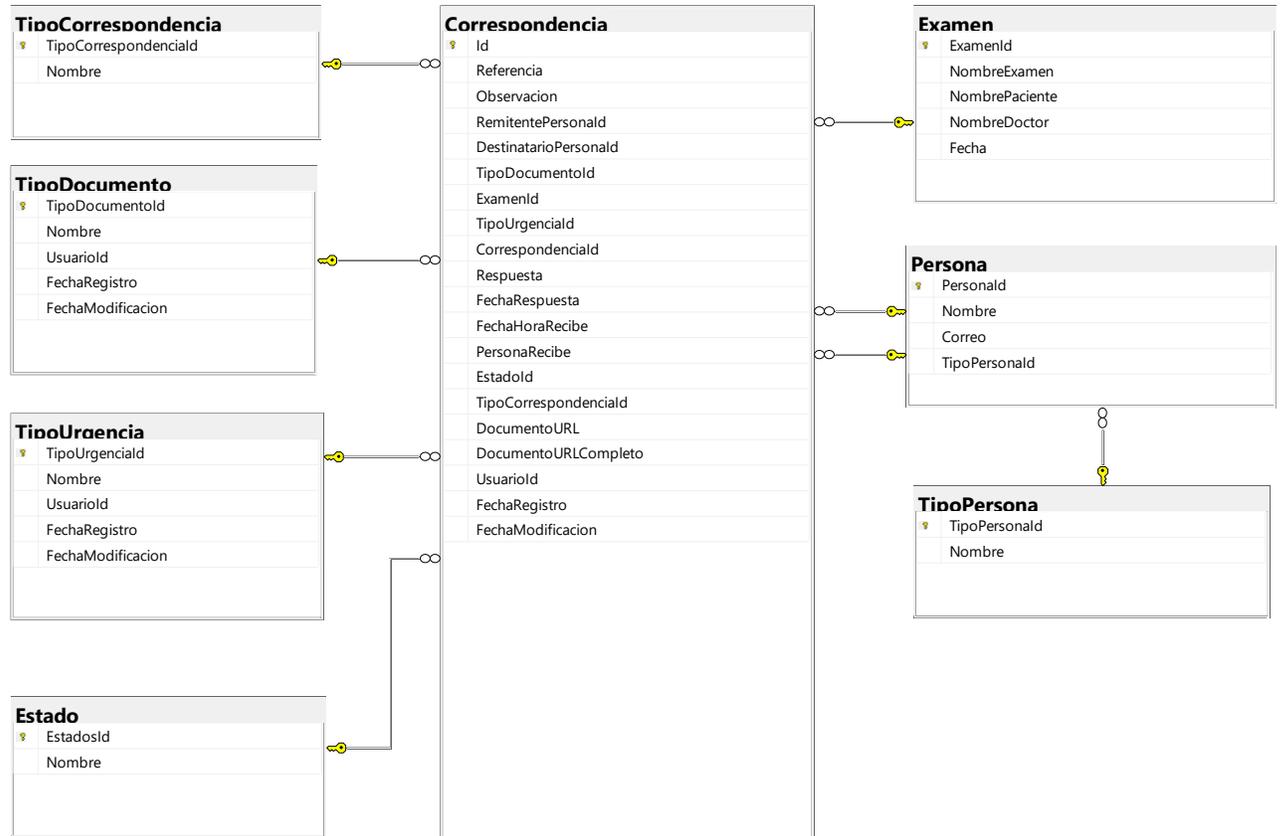
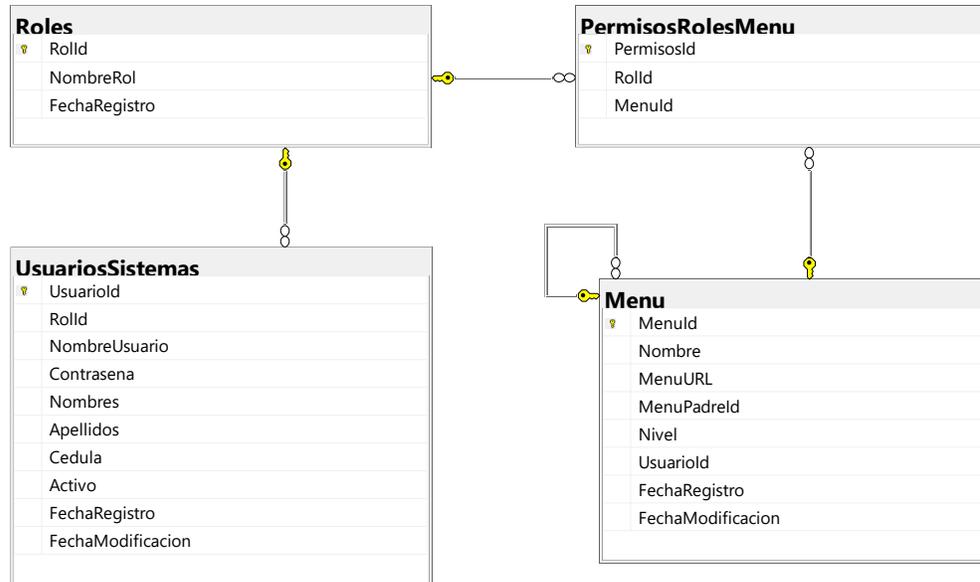


Diagrama de Tablas de Seguridad de Usuario

Figura 5-5: Diagrama de Tablas de Seguridad de Usuario



5.2.4.2 Diccionario de Datos

Tabla 18: Diccionario de Datos de Dominio de Aplicación

Tabla	Columna	Tipo	Tamaño	Descripción
Correspondencia	Id	int	4	Identificador, llave primaria de tabla. Autonumérico
	Referencia	nvarchar	240	Identificador lógico: Código secuencial o de referencia.
	Observacion	nvarchar	1000	Comentarios adicionales u observaciones con respecto a la correspondencia.
	RemitentePersonald	int	4	Identificador de Remitente. Llave foránea a tabla Persona.
	DestinatarioPersonald	int	4	Identificador de Destinatario. Llave foránea a tabla Persona.
	TipoDocumentold	int	4	Identificador de tipo documento. Llave foránea a tabla TipoDocumento.
	ExamenId	int	4	Identificador de Resultado de estudio clínico. Llave foránea a tabla Examen.
	TipoUrgenciald	int	4	Identificador de Tipo Urgencia o Prioridad de la correspondencia. Llave foránea a tabla TipoUrgencia.
	Correspondenciald	int	4	Referencia, opcional, a correspondencia relacionada si el campo Respuesta es verdadero.
	Respuesta	bit	1	Indica si la correspondencia requiere respuesta. De ser verdadero, el campo Correspondenciald almacena la referencia a la correspondencia original.
	FechaRespuesta	datetime	8	Fecha en que se debe responder a más tardar, si requiere respuesta.
	FechaHoraRecibe	datetime	8	Fecha y hora en que se recibe la correspondencia por parte del destinatario.
	PersonaRecibe	nvarchar	200	Nombre de persona que recibe la correspondencia en el destinatario.
	Estadold	int	4	Identificador de Estado de la Correspondencia. Llave foránea a tabla Estado.
	TipoCorrespondenciald	int	4	Identificador de Tipo de Correspondencia. Llave foránea a tabla TipoCorrespondencia.
	DocumentoURL	nvarchar	200	Localización del documento cargado al momento de registrar la correspondencia.
	DocumentoURLCompleto	nvarchar	600	Localización física del documento cargado al momento de registrar la correspondencia.
	Usuariold	int	4	Identificador de Usuario que registró la correspondencia. Llave foránea a tabla Usuario.
FechaRegistro	datetime	8	Fecha y hora en que se registró la correspondencia. Información de auditoría.	
FechaModificacion	datetime	8	Fecha y hora última en que se modificó registro de la correspondencia. Información de auditoría.	
Estado	EstadosId	int	4	Identificador, llave primaria de tabla. Autonumérico
	Nombre	varchar	50	Nombre o descripción del estado.
Examen	ExamenId	int	4	Identificador, llave primaria de tabla. Autonumérico
	NombreExamen	varchar	100	Nombre o descripción del estudio clínico o examen.
	NombrePaciente	varchar	50	Nombre del paciente al que se realizó el estudio.
	NombreDoctor	varchar	50	Nombre del médico tratante del paciente al que se realizó el estudio.
	Fecha	datetime	8	Fecha en que se realizó el estudio.

Tabla	Columna	Tipo	Tamaño	Descripción
Persona	Personald	int	4	Identificador, llave primaria de tabla. Autonumérico
	Nombre	varchar	200	Nombres y apellidos de la persona.
	Correo	varchar	200	Dirección de correo electrónico de la persona.
	TipoPersonald	int	4	Identificador de tipo persona. Llave foránea a tabla TipoPersona.
TipoCorrespondencia	TipoCorrespondenciald	int	4	Identificador, llave primaria de tabla. Autonumérico
	Nombre	varchar	50	Nombre o descripción del tipo de correspondencia.
TipoDocumentos	TipoDocumentold	int	4	Identificador, llave primaria de tabla. Autonumérico
	Nombre	varchar	50	Nombre o descripción del tipo de correspondencia.
	Usuariold	int	4	Identificador de Usuario que registró el Tipo de Documento. Llave foránea a tabla Usuario.
	FechaRegistro	datetime	8	Fecha y hora en que se registró el Tipo de Documento. Información de auditoría.
	FechaModificacion	datetime	8	Fecha y hora última en que se modificó registro. Información de auditoría.
TipoPersona	TipoPersonald	int	4	Identificador, llave primaria de tabla. Autonumérico
	Nombre	varchar	50	Nombre o descripción del tipo de persona.
TipoUrgencia	TipoUrgenciald	int	4	Identificador, llave primaria de tabla. Autonumérico
	Nombre	varchar	50	Nombre o descripción del tipo de urgencia.
	Usuariold	int	4	Identificador de Usuario que registró el Tipo de Urgencia. Llave foránea a tabla Usuario.
	FechaRegistro	datetime	8	Fecha y hora en que se registró el Tipo de Urgencia. Información de auditoría.
	FechaModificacion	datetime	8	Fecha y hora última en que se modificó registro. Información de auditoría.

Tabla 19: Diccionario de datos de Seguridad de Usuarios

Tabla	Columna	Tipo	Tamaño	Descripción
Menu	MenuId	int	4	Identificador, llave primaria de tabla. Autonumérico
	Nombre	varchar	50	Nombre o descripción de la entrada de menú.
	MenuURL	varchar	150	Ruta de la entrada de menú en la jerarquía.
	MenuPadreId	int	4	Referencia a entrada de menú padre.
	Nivel	tinyint	1	Nivel jerárquico de la entrada de menú.
	Usuariold	int	4	Identificador de Usuario que registró la entrada de menú. Llave foránea a tabla Usuario.
	FechaRegistro	datetime	8	Fecha y hora en que se creó el registro. Información de auditoría.
	FechaModificacion	datetime	8	Fecha y hora última en que se modificó registro. Información de auditoría.
PermisosRolesMenu	PermisosId	int	4	Identificador, llave primaria de tabla. Autonumérico
	RoleId	int	4	Nombre o descripción del Rol de usuario que tiene permiso de acceso a una entrada de menú. Llave foránea a tabla Roles.
	MenuId	int	4	Identificador de entrada de menú a la que el Rol de usuario definido en el campo anterior tiene acceso. Llave foránea a tabla Menu.
Roles	RoleId	int	4	Identificador, llave primaria de tabla. Autonumérico

Tabla	Columna	Tipo	Tamaño	Descripción
	NombreRol	varchar	50	Nombre o descripción del Rol de usuario.
	FechaRegistro	datetime	8	Fecha en que se creó el registro del rol.
UsuariosSistemas	Usuariold	int	4	Identificador, llave primaria de tabla. Autonumérico
	Rolld	int	4	Identificador del Rol de usuario. Llave foránea a tabla Roles.
	NombreUsuario	varchar	50	Nombre o alias del usuario.
	Contrasena	varchar	650	Contraseña del usuario, encriptada.
	Nombres	varchar	50	Nombres de pila del usuario.
	Apellidos	varchar	50	Apellidos del usuario.
	Cedula	varchar	16	Número de cédula ciudadana del usuario.
	Activo	bit	1	Indicador de que el usuario está activo.
	FechaRegistro	datetime	8	Fecha en que se creó el registro del Usuario.
	FechaModificacion	datetime	8	Fecha última en que se modificó el registro del Usuario.

5.2.5 Modelo de Interacción

El modelado de interacción parte del modelo de clases y define cómo se relacionan las diferentes clases entre sí y con los actores (usuarios) del sistema para el logro de sus objetivos (funciones). Es decir, describe el comportamiento del sistema conforme la arquitectura definida. Entre los diagramas que provee UML para modelar la interacción, el más empleado es el diagrama de secuencia.

En consideración a las restricciones derivadas del requerimiento no funcional RNF2, se adoptó el patrón de interacción de vistas de sistema definido para el desarrollo de software web en el Laboratorio Divino Niño. Dicho patrón general se basa en la siguiente especificación de presentación y acciones:

1. El sistema, una vez el usuario se autentica, le presenta la página principal con el menú de opciones que le da acceso a las funcionalidades que el usuario tiene autorizadas conforme su rol.
2. Al acceder a una opción de menú, el sistema presenta al usuario una vista con una lista de los registros activos del modelo correspondiente.

3. Asimismo, presenta una opción de Creación de un nuevo registro.
4. Cada registro permite las acciones de Edición, Consulta, Eliminación o Desactivación (según lo que aplique conforme la política definida).

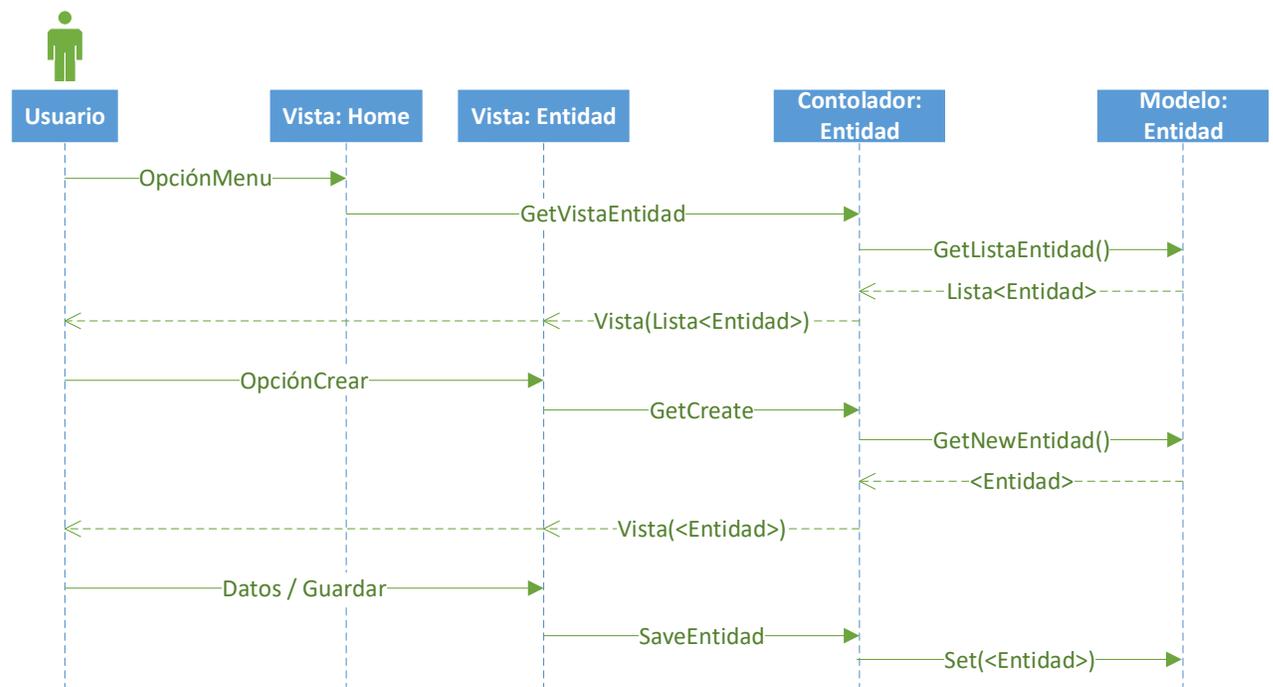
5.2.5.1 Diagrama de Secuencia

El diagrama de secuencia muestra los objetos participantes y el orden de su interacción en el intercambio de mensajes durante la ejecución del aplicativo (Larman, UML y Patrones: Una introducción al análisis y diseño orientado a objetos y al proceso unificado, 2003).

A continuación, se muestran los diagramas de secuencia generales del sistema, sobre la base del patrón de interacción descrito anteriormente.

- **Diagrama de secuencia para la operación Crear**

Figura 5-6: Diagrama de Secuencia para la operación Crear

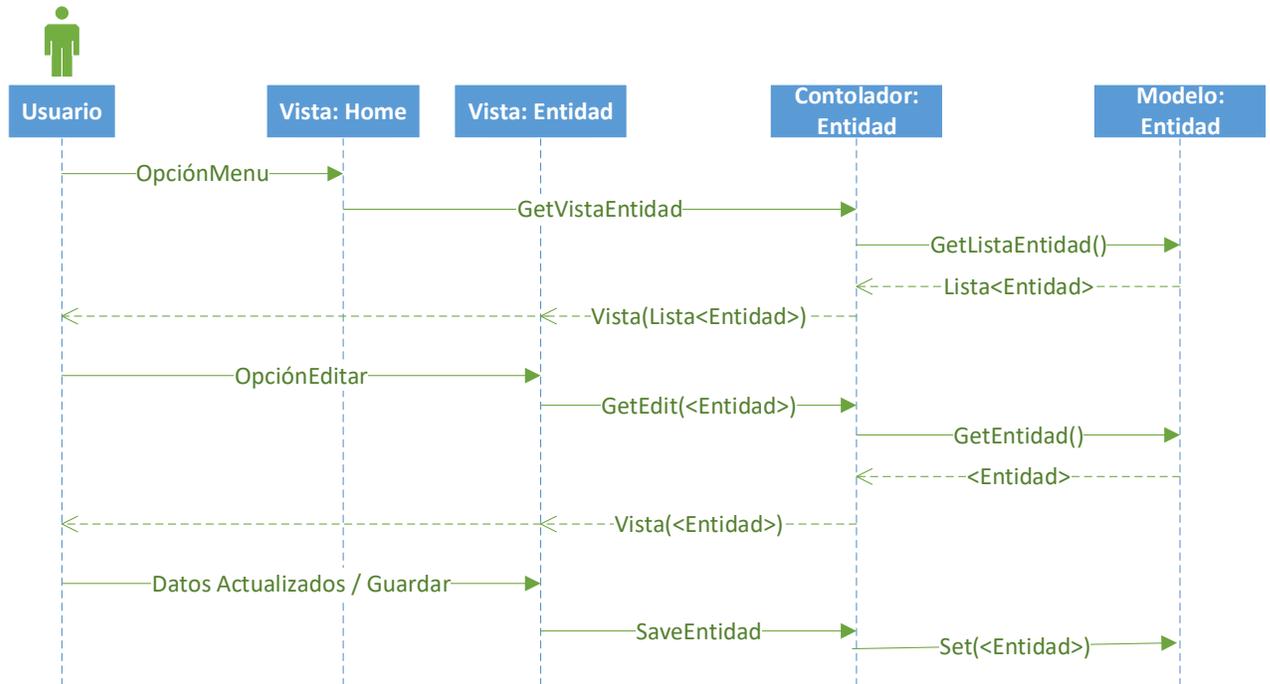


Secuencia de acciones para la operación Crear

1. El usuario selecciona una opción de menú desde la página principal (Vista Home).
2. La vista responde activando el método de vista del Controlador de la entidad correspondiente, el cual invoca al método correspondiente del Modelo de la entidad para obtener una lista de las instancias activas de éste.
3. Con la lista de instancias, el Controlador crea una vista de listado y la presenta al usuario.
4. El usuario selecciona la opción Crear.
5. La Vista responde activando el método de creación del Controlador de la entidad correspondiente, el cual invoca al método del Modelo de la entidad para crear una nueva instancia vacía.
6. Con la nueva instancia, el Controlador crea una vista de creación y la presenta al usuario.
7. El usuario provee a la vista los datos de los atributos de la entidad requeridos y selecciona guardar.
8. La vista responde activando el método de salvar cambios del Controlador de la entidad correspondiente, el cual invoca al método del Modelo de la entidad para guardar la información, pasando la instancia con los datos. El método guarda la información de la nueva instancia en la Base de Datos.

- **Diagrama de secuencia para la operación Editar**

Figura 5-7: Diagrama de Secuencia para la operación Editar



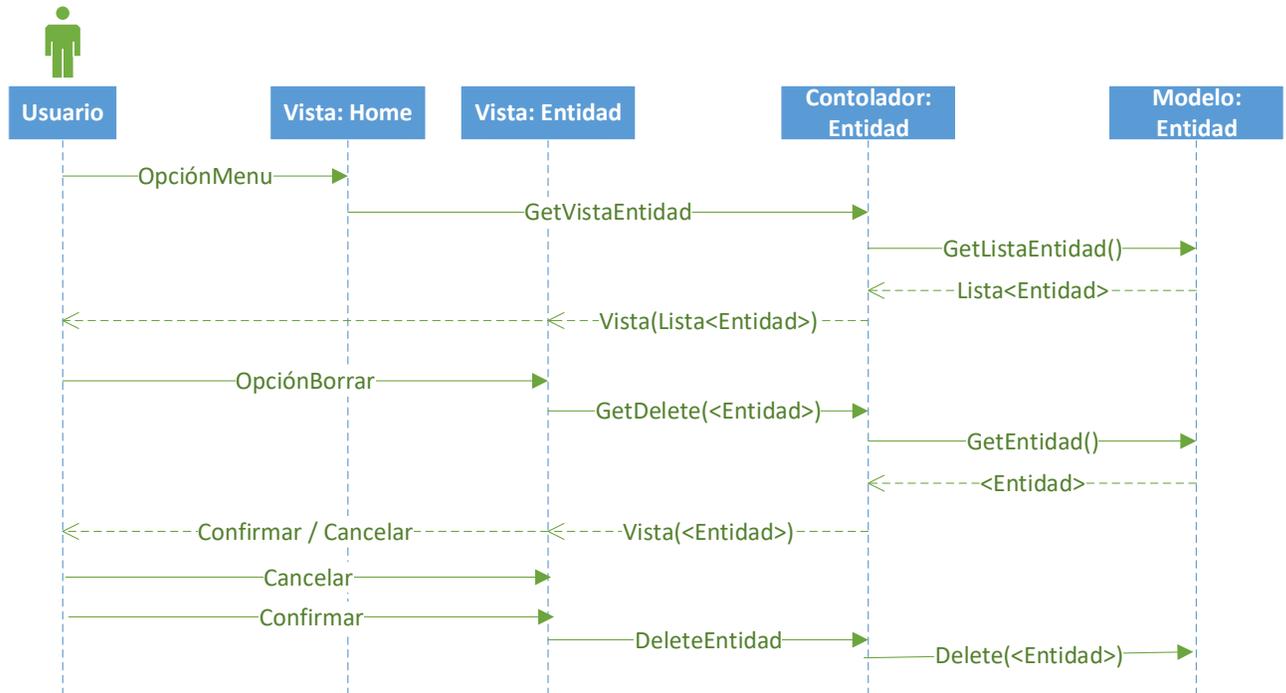
- **Secuencia de acciones para la operación Editar**

1. El usuario selecciona una opción de menú desde la página principal (Vista Home).
2. La vista responde activando el método de vista del Controlador de la entidad correspondiente, el cual invoca al método correspondiente del Modelo de la entidad para obtener una lista de las instancias activas de éste.
3. Con la lista de instancias, el Controlador crea una vista de listado y la presenta al usuario.
4. El usuario selecciona la opción Editar en uno de los miembros de la lista.

5. La Vista responde activando el método de edición del Controlador de la entidad correspondiente, pasándole la identificación del elemento de lista, el cual invoca al método del Modelo de la entidad para recuperar la información de la instancia.
6. Con la instancia obtenida, el Controlador crea una vista de edición y la presenta al usuario.
7. El usuario actualiza en la vista, los datos de los atributos de la entidad que desea editar y selecciona guardar.
8. La vista responde activando el método de salvar cambios del Controlador de la entidad correspondiente, el cual invoca al método del Modelo de la entidad para guardar la información, pasando la instancia con los datos. El método guarda la información de la nueva instancia en la Base de Datos.

- **Diagrama de secuencia para la operación Borrar**

Figura 5-8: Diagrama de Secuencia para la operación Borrar



- **Secuencia de acciones para la operación Borrar**

1. El usuario selecciona una opción de menú desde la página principal (Vista Home).
2. La vista responde activando el método de vista del Controlador de la entidad correspondiente, el cual invoca al método correspondiente del Modelo de la entidad para obtener una lista de las instancias activas de éste.
3. Con la lista de instancias, el Controlador crea una vista de listado y la presenta al usuario.
4. El usuario selecciona la opción Borrar en uno de los miembros de la lista.

5. La Vista responde activando el método de borrado del Controlador de la entidad correspondiente, pasándole la identificación del elemento de lista, el cual invoca al método del Modelo de la entidad para recuperar la información de la instancia.
6. Con la instancia obtenida, el Controlador crea una vista de consulta y la presenta al usuario, solicitándole confirme el borrado de la instancia.
7. El usuario revisa la información y confirma Borrar.
8. La vista responde activando el método de borrado del Controlador de la entidad correspondiente, el cual invoca al método del Modelo de la entidad para borrar la información, pasando la identificación de la instancia. El Modelo borra el registro de la Base de Datos.

5.3 Fase de Construcción

El SGC, conforme el requerimiento no funcional RNF1, se desarrolló como una aplicación web ASP.Net empleando el IDE Microsoft Visual Studio 2019 y construida conforme el paradigma de programación orientada a objetos, codificada en el lenguaje de programación C#.

5.3.1 Arquitectura del Sistema

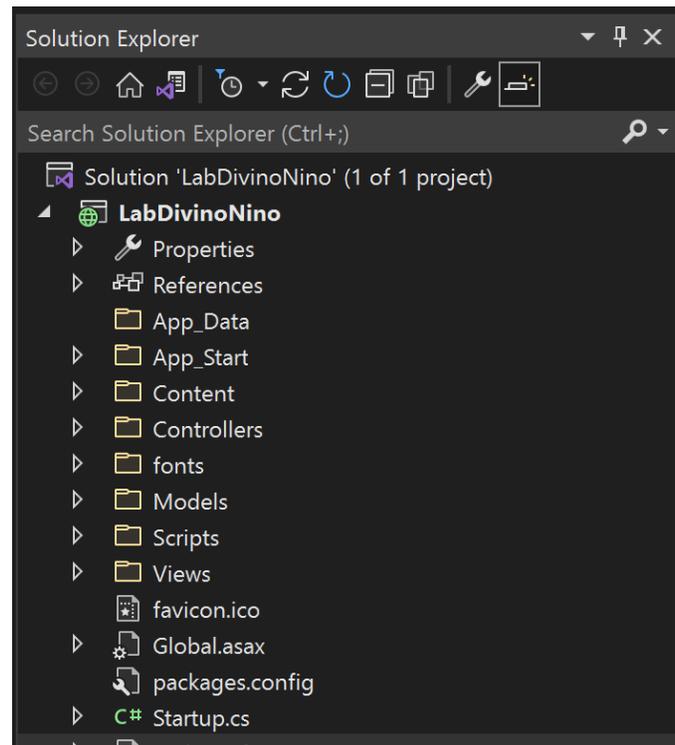
En atención a lo especificado por el requerimiento no funcional RNF2, la arquitectura del sistema se implementó sobre el patrón estructural Modelo-Vista-Controlador (MVC) empleando para esto el marco de trabajo MVC que integra ASP.Net, el cual incorpora las mejores prácticas de la industria para este patrón de arquitectura.

Para implementar dicha arquitectura como un prototipo ejecutable se aprovecharon los asistentes y plantillas que incorpora Visual Studio, para lo cual se creó un nuevo proyecto del tipo ASP.Net Web Application (.Net Framework) del tipo MVC.

La figura a la derecha muestra la estructura del proyecto creado por el asistente de Visual Studio.

El asistente incluye en el proyecto las referencias a los componentes correspondientes y la estructura de

Figura 5-9 Arquitectura del SGC



directorios que el marco de trabajo MVC requiere. Genera, además, código base de controladores y vistas para la ejecución del patrón MVC con algunas páginas base.

En una arquitectura basada en el patrón estructural Modelo-Vista-Controlador (MVC), las funcionalidades del sistema se presentan al usuario a través de vistas. Las vistas son páginas HTML construidas dinámicamente a partir de un método ActionResult() de un controlador. Para construir la vista, el controlador obtiene la información del modelo o modelos (entidades del dominio del problema) correspondientes. El usuario interactúa con la vista, pudiendo generar una nueva petición al sistema (a través de un botón de acción, por ejemplo), con lo cual la información de la página se envía actualizada al controlador correspondiente para su procesamiento, el que puede resultar en actualización del modelo y generación de una nueva vista al usuario.

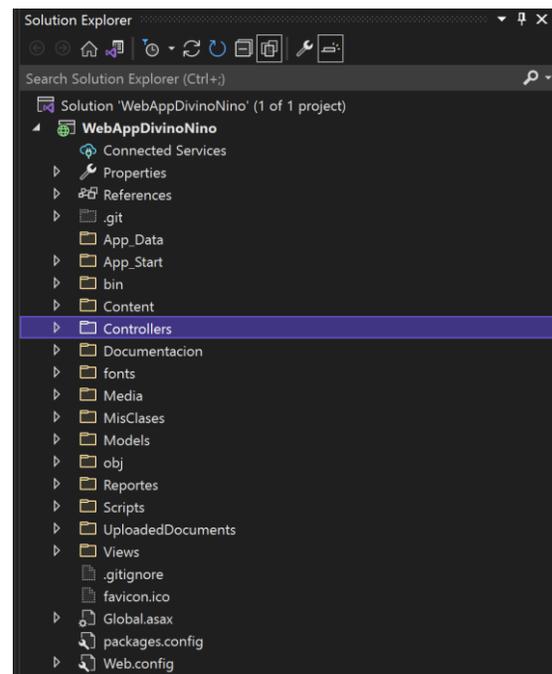
Esta secuencia de acciones conforma el ciclo de interacción del modelo MVC, a través del cual se realiza la ejecución de una aplicación basada en esta arquitectura.

5.3.2 Desarrollo del Sistema

El SGC, se desarrolló a partir del proyecto prototipo de arquitectura de ASP.Net MVC generada por Visual Studio 2019. La figura a la derecha, muestra la estructura final de la solución de Visual Studio empleada para el desarrollo del sistema.

Se modificó el prototipo para incluir un menú lateral izquierdo en lugar del menú horizontal superior que generó el asistente del Visual Studio para el prototipo. Esto a fin de cumplir con el requerimiento de compatibilidad con el resto de aplicaciones del Laboratorio.

Figura 5-10 Desarrollo del SGC



Para esto, se modificó la vista maestra del proyecto [_Layout.cshtml](#) a fin de que todas las páginas del sistema incluyeran la estructura definida del menú, y que éste estuviera disponible siempre para el usuario.

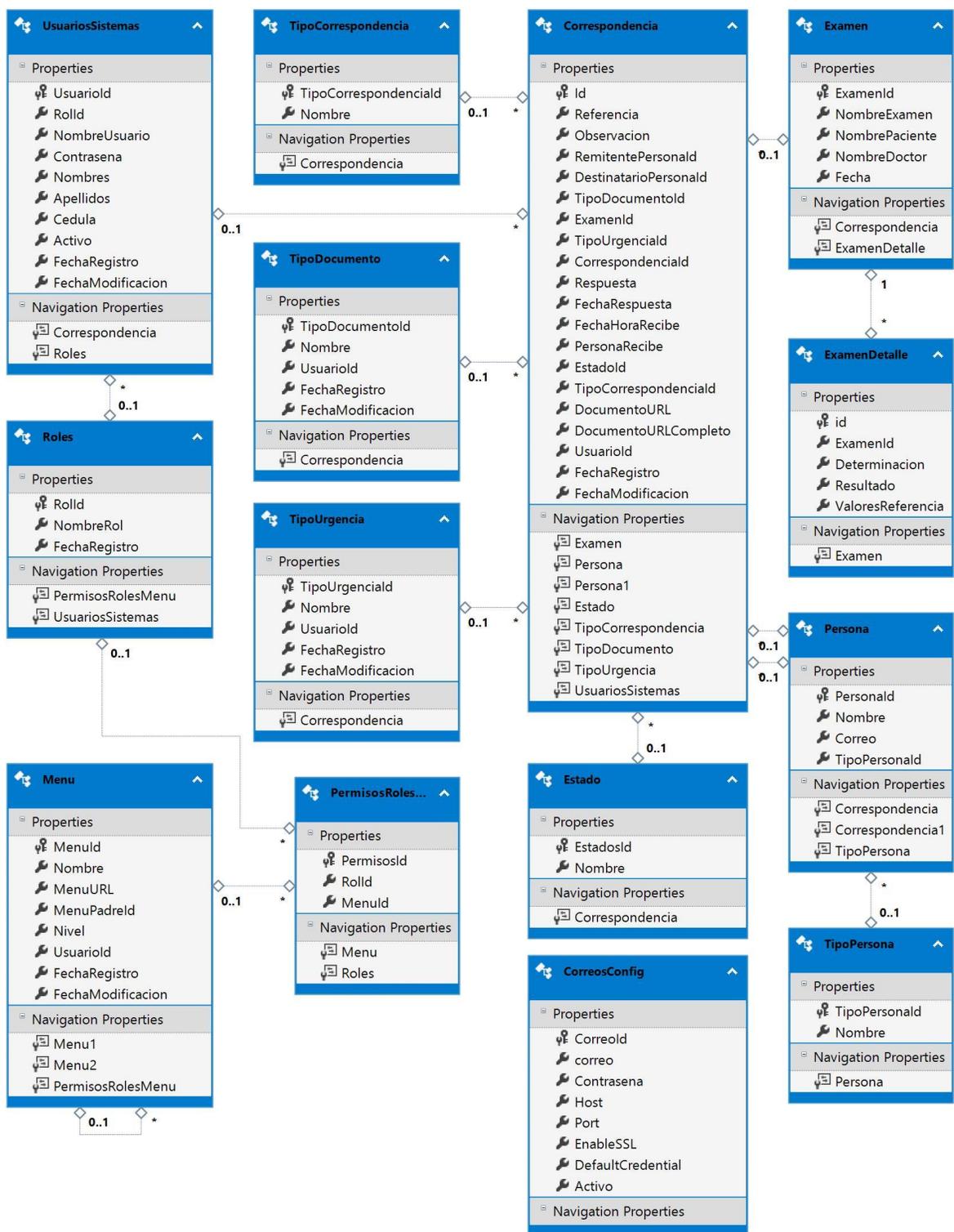
Asimismo, en atención a los requerimientos, para implementar la persistencia de los datos se empleó el ORM de Microsoft, *Entity Framework* con un enfoque *database first*.

El procesamiento del acceso a datos fue realizado por *Entity Framework* en conjunto con *LINQ to Entities*, que proporciona la capacidad de realizar consultas LINQ (Language-Integrated Query) contra el modelo de *Entity Framework* sin tener que usar un lenguaje de consulta independiente como SQL, con lo que se asegura la separación de responsabilidades y se mantiene alta cohesión y bajo acoplamiento.

La figura a continuación muestra el modelo de datos base empleado en el Entity Framework para el manejo de la persistencia de los datos.

Diagrama de modelo de datos del Entity Framework

Figura 5-11: Diagrama de modelo de datos del Entity Framework



5.3.2.1 Implementación de la Seguridad del Sistema

La seguridad del sistema se divide en dos grandes aspectos: a) la autenticación y b) la autorización.

La autenticación establece la identidad del usuario al momento de acceder al sistema, verificando que la información del nombre de usuario y contraseña provistos corresponden a un usuario debidamente registrado en el sistema.

La autorización, por otro lado, define a qué funcionalidades del sistema el usuario está autorizado a operar o acceder. La autorización se implementó a partir de la asignación de roles a los usuarios. A cada rol se le configura, a través del módulo de administración del sistema, a que funcionalidades (opciones del menú) el usuario tiene acceso.

Una vez que el usuario se autentica, el controlador principal HomeController provee un método del tipo ActionResult denominado SendMenu que genera las opciones de menú autorizadas al usuario correspondiente para su renderización en el menú lateral del sistema.

5.3.2.2 Implementación de las Funcionalidades del Sistema

La implementación de cada una de las funcionalidades del sistema partió de la codificación del método ActionResult del controlador correspondiente para responder a la opción de menú que la dispara.

Para cada método se generó la correspondiente vista de tipo listado, relacionada al modelo correspondiente.

Se agregaron a la vista los botones de acción de Crear y otros que aplicaran, así como las acciones de lista (Editar, Ver Detalle, etc).

Posteriormente se codificaron los métodos correspondientes para manejar cada una de estas acciones en los controladores correspondientes.

5.4 Fase de Implantación

Una vez concluida construcción del sistema y éste estuvo listo para su puesta en servicio a los usuarios finales, se realizó la fase de implantación en los servidores del Laboratorio Divino Niño. La fase de implantación comprendió las siguientes tareas:

- a) Instalación de instancia de capacitación a usuarios
- b) Capacitación y pruebas de usuario
- c) Evaluación del Uso del Sistema
- d) Instalación final del sistema en servidores de producción y puesta en servicio

5.4.1 Instalación de ambiente de capacitación a usuarios

Se utilizó como ambiente de capacitación y pruebas un equipo de escritorio que tenía previamente instalados Visual Studio y SQL Server Developer Edition.

Se realizó un respaldo de la base de datos del sistema en ambiente de desarrollo y se restauró en el equipo de ambiente de capacitación.

Se copió la carpeta de la solución del sistema en el ambiente de capacitación y se ajustó la cadena de conexión a base de datos local.

Se realizaron pruebas de verificación del funcionamiento, ejecutando la aplicación desde Visual Studio en el *application server* integrado.

5.4.2 Capacitación y pruebas de usuario

En el proceso de capacitación y pruebas solo participaron dos usuarios: a) Recepcionista y b) Gerente General. El proceso de capacitación y pruebas, conforme lo solicitado por el Laboratorio se realizó en sesiones separadas por roles.

Las sesiones de capacitación se organizaron de la siguiente manera:

- La desarrolladora realizó una demostración de cada una de las funcionalidades del sistema que le competen al usuario.
- El usuario realizó prueba de cada funcionalidad de forma asistida por la desarrolladora.
- La desarrolladora presentó y explicó al usuario el manual de usuario del sistema.

5.4.3 Evaluación del uso del sistema

A fines de dar cumplimiento al objetivo específico f), se planteó un marco de evaluación que considerara tanto aspectos cuantitativos como cualitativos a fin de verificar la efectividad del SGC en la solución del problema.

5.4.3.1 *Evaluación cuantitativa*

Permite verificar en qué medida se mejoró la problemática del Laboratorio con el uso del SGC. La metodología de evaluación seleccionada, parte de analizar el problema a fin de extraer las variables que lo caracterizan, y en función de la solución propuesta, definir los indicadores o métricas a utilizar para la evaluación.

Principales problemas identificados en situación sin sistema:

- a. Se ha olvidado entregar correspondencia recibida a tiempo.
- b. Se ha traspapelado correspondencia.
- c. No se ha enviado oportunamente información urgente.
- d. Se dificulta saber cuáles y cuántas comunicaciones se han gestionado.

Variables identificadas:

V_a = Cantidad de correspondencia no gestionada a tiempo

V_b = Cantidad de correspondencia extraviada

V_c = Tiempo de gestión de correspondencia

V_d = Tiempo para procesar información de volumen de correspondencia

Las variables V_a y V_b se corresponden a ocurrencia de eventos no deseables, las variables V_c y V_d son de tiempo de respuesta. La solución que implementa el SGC tiene por objetivo reducir al mínimo los valores de estas variables.

En este sentido, se establecen los siguientes indicadores o métricas de fiabilidad (Identificados con F) y desempeño (Identificados con D) para evaluar el comportamiento del sistema y la mejora sobre el procesamiento manual (situación sin sistema), en un período de tiempo de evaluación de la misma duración (un mes):

F_o = Mejora de fiabilidad en gestión oportuna

F_e = Mejora de fiabilidad ante extravío de correspondencia

D_c = Mejora de rendimiento de gestión de correspondencia

D_d = Mejora de rendimiento de obtención de información consolidada

Los cuales se definen de la siguiente manera:

$$F_o = F_{am} - F_{as}$$

Donde

F_{am} = Tasa de correspondencia no gestionada a tiempo manualmente

F_{as} = Tasa de correspondencia no gestionada a tiempo con sistema

De forma general:

$$F_a = V_a / V_g$$

Siendo V_g el total de correspondencia gestionada

$$F_e = F_{bm} - F_{bs}$$

Donde

F_{bm} = Tasa de correspondencia extraviada sin sistema

F_{bs} = Tasa de correspondencia extraviada con sistema

De forma general:

$$F_b = V_b / V_g$$

$$D_e = \left(\frac{V_{cm}}{V_{cs}} - 1 \right) * 100\%$$

Donde

V_{cm} = Tiempo medio de gestión de correspondencia sin sistema

V_{cs} = Tiempo medio de gestión de correspondencia con sistema

$$D_d = \left(\frac{V_{dm}}{V_{ds}} - 1 \right) * 100\%$$

Donde

V_{dm} = Tiempo medio para procesar información de volumen de correspondencia sin sistema

V_{ds} = Tiempo medio para procesar información de volumen de correspondencia con sistema

Una vez definidos los indicadores se procedió a coleccionar las mediciones de las variables correspondientes para cada una de las dos situaciones a comparar (sin sistema, con sistema) en el período de tiempo de un mes de operación en cada una y a realizar la evaluación.

- **Datos obtenidos de la medición:**

Tabla 20: Medida de las variables de fiabilidad.

Variable		Medición situación sin sistema				Medición situación con sistema			
		$V_{(a b)m}$	V_{gm-ca}	V_{gm-re}	V_{gm}	$V_{(a b)s}$	V_{gs-ca}	V_{gs-re}	V_{gs}
Nombre↓	Escenarios →								
Cantidad de correspondencia no gestionada a tiempo (V_a)		12	65	321	386	0	72	396	468
Cantidad de correspondencia extraviada (V_b)		3	65	321	386	0	72	396	468

Nota: V_g se muestra detallado por correspondencia estándar (V_{gm-ca} , V_{gs-ca}) y resultados de estudios clínicos (V_{gm-re} , V_{gs-re}) y consolidado (V_{gm} , V_{gs}).

Tabla 21: Medida de variables de desempeño.

Variable		Medición situación sin sistema		Medición situación con sistema	
		V _{cm} (Min)	V _{dm} (Min)	V _{cs} (Min)	V _{ds} (Min)
Nombre ↓	Escenarios →				
Tiempo medio de gestión de correspondencia (Vc)		54		21	
Tiempo medio para procesar información de volumen de correspondencia (Vd)			58		4

• **Evaluación del Sistema:**

Aplicando las definiciones de los indicadores, tenemos los siguientes resultados:

Tabla 22: Resultados de evaluación de fiabilidad.

Indicador o métrica	V _{(a b)m}	V _{gm}	F _{am}	V _{(a b)s}	V _{gs}	F _{as}	Fo Fe
Mejora de fiabilidad en gestión oportuna (Fo)	12	386	3.1%	0	468	0%	3.1%
Mejora de fiabilidad ante extravío de correspondencia (Fe)	3	386	0.8%	0	468	0%	0.8%

En el período evaluado, el sistema redujo al 0% los eventos no deseados, por lo que mejora en 3% la fiabilidad de la gestión en cuanto a la oportunidad y en 0.8% ante extravío. De mantenerse esta medida en régimen, podría afirmarse que el sistema es 100% fiable en estas dimensiones.

Tabla 23: Resultados evaluación de desempeño.

Indicador o métrica	V _{(c d)m}	V _{(c d)s}	D _(c d)
Mejora de rendimiento de gestión de correspondencia (Dc)	54	21	157%
Mejora de rendimiento de obtención de información consolidada (Dd)	58	4	1,350%

En el período evaluado, las medidas indican que el sistema redujo considerablemente el tiempo de respuesta, acelerando en 157% la gestión de correspondencia y en 1,350% la obtención de información consolidada.

5.4.3.2 Evaluación cualitativa

Permite medir la percepción del usuario sobre el SGC y su efectividad en resolver la problemática planteada, lo cual es clave para su adopción y consolidación, y por tanto su efectividad de uso.

Como metodología de evaluación se definió que al final de la capacitación y pruebas desarrolladas por los usuarios, se realizara entrevista a los participantes a fin de evaluar su percepción acerca los siguientes aspectos con relación al SGC:

1. ¿El sistema satisface los requerimientos planteados?
2. La información registrada, ¿le permitirá dar un efectivo seguimiento a la gestión de la correspondencia?
3. ¿Considera que el sistema tendrá un impacto positivo en mejorar la gestión de correspondencia en el Laboratorio?
4. ¿Considera que se respetaron y adoptaron de forma adecuada los estándares de diseño sistemas utilizados en el laboratorio?

A lo cual los usuarios contestaron⁶:

1. El sistema permite registrar la correspondencia recibida y enviada, así como la entrega de los resultados de los estudios clínicos. Tiene la opción de consultar la correspondencia registrada para darle seguimiento, por lo que cumple con los requerimientos.
2. Además del Remitente y Destinatario permite registrar cuándo una correspondencia espera respuesta. También permite registrar cuando se entrega la correspondencia enviada y quién la recibe. Permite definir

⁶ A partir del resumen de las notas tomadas por la autora de a entrevista realizada a los usuarios.

diferentes estados y a cada correspondencia se le puede asignar un estado. Es decir, el sistema tiene toda la información necesaria para el adecuado seguimiento de la correspondencia.

3. El sistema va a mejorar significativamente la gestión de la correspondencia en el Laboratorio. Considero que, al tener toda la información de interés registrada de forma centralizada para su seguimiento, se superarán todas las dificultades que se han tenido con el registro manual.
4. Efectivamente, el sistema se parece al resto de sistemas del Laboratorio, lo que nos facilita su uso, pues se nos hace familiar.

En este sentido, aunque no fue posible dimensionar cuantitativamente en qué medida se logró mejorar la situación problema del Laboratorio, los usuarios confirmaron que obtuvieron mejoras en la forma en que se procesa la información y por tanto, el sistema será efectivo en la solución del problema.

5.4.4 Instalación final del sistema en servidores de producción y puesta en servicio

Conforme lo dispuesto por el Laboratorio Divino Niño, la instalación del sistema corrió por cuenta del personal de informática del Laboratorio. La desarrolladora entregó la solución con el código fuente del sistema.

Para efectos de la instalación de la base de datos, fue requerido entregar un script de creación de la misma. El script de creación de la base de datos se adjunta en el Anexo No. 2.

6 CONCLUSIONES Y RECOMENDACIONES

Del análisis realizado a los resultados, se desprenden las siguientes conclusiones y recomendaciones:

6.1 Conclusiones

1. Se realizó con éxito el desarrollo del Sistema de Gestión de Correspondencia “SGC” para el Laboratorio de Investigaciones Clínicas Divino Niño, satisfaciendo los requerimientos técnicos definidos.
2. La identificación clara de los requerimientos del SGC, a pesar de las limitaciones de acceso y comunicación impuestas por la crisis sanitaria derivada de la pandemia de Covid-19 en el transcurso de 2020 y 2021, fue clave para lograr un desarrollo exitoso del sistema, por cuanto son las bases para el mismo.
3. Se logró adoptar, con bastante dificultad, pero exitosamente, las restricciones en cuanto a estándares de notación, codificación, herramientas y prácticas a observar en el desarrollo del proyecto, de acuerdo a las normativas establecidas por El Laboratorio, garantizando así la coherencia con sus otros sistemas informáticos y facilitando la apropiación y uso por parte del personal usuario.
4. Se aplicó exitosamente la metodología RUP para el desarrollo del SGC, verificándose que puede ser fácilmente adaptable y muy útil para la comprensión detallada de los requerimientos, necesidades y reglas del negocio. Asimismo, UML demostró ser una herramienta poderosa para la creación de artefactos que sustentan de manera efectiva los procesos de análisis y diseño de sistemas de información, a través de los modelos de casos de uso, dominio, clases y secuencia.
5. El patrón estructural Modelo-Vista-Controlador (MVC) y el marco de trabajo establecido en Visual Studio para implementar una arquitectura basada en MVC demostraron ser sumamente robustos, coherentes y de fácil uso para la codificación del sistema.

6. La constante comunicación e interacción del desarrollador con el cliente es clave para garantizar la fluidez del proceso de desarrollo. Las dificultades para lograr dicha interacción con el personal involucrado del Laboratorio, debido a las restricciones de acceso y comunicación impuestas por la crisis sanitaria derivada de la pandemia de Covid-19 en 2020 y 2021, impactó negativamente el avance de la codificación de la aplicación, retrasando el proceso de desarrollo en más de un año.
7. La capacitación a los usuarios principales del sistema (Recepcionista y Gerente General) para el uso de la aplicación, así como el apego en el diseño de interfaz de usuario del SGC a las políticas internas del Laboratorio facilitaron una rápida adopción del sistema.
8. A partir de lo manifestado por los usuarios en la implantación del sistema, con respecto a los criterios consultados para la evaluación del sistema, se puede concluir que el sistema tiene aceptación por parte de los mismos, se satisfacen los requerimientos, podrán apropiarse sin dificultades a su uso y que el sistema será de beneficio para la organización.

6.2 Recomendaciones

1. Se recomienda implementar una política de respaldo continuo de la información del sistema, tanto de la base de datos como de los documentos adjuntos a fin de evitar posibles pérdidas de información en caso fortuito de daño del almacenamiento de los servidores.
2. A futuro, una vez consolidado el uso del sistema, se recomienda ampliar el alcance del registro de las relaciones entre correspondencias a fin de tener mayor control y seguimiento de las conversaciones o hilos de correspondencia.
3. Asimismo, se recomienda implementar un dashboard para dar seguimiento a indicadores de gestión, como, por ejemplo, el cumplimiento de los tiempos de respuesta.

7 BIBLIOGRAFÍA

- Alvarez, M. A. (2019). *desarrolloweb.com*. Obtenido de <https://desarrolloweb.com/articulos/1325.php>
- Bustelo, C. (2015). *Normas técnicas y estándares relacionados con la gestión documental*. Barcelona: Universitat Oberta de Catalunya.
- Carmona, S., Sáenz, A., & Morales, J. (2013). *Repositorio Institucional UNAN-Managua*. Obtenido de <http://repositorio.unan.edu.ni/274/>
- FormaTalent. (01 de 07 de 2019). *La historia de Javascript*. Obtenido de FormaTalent Bussiness School: <http://formatalent.com/la-historia-de-javascript/>
- García, R., & Martín, B. (1 de 2 de 2017). *Herramientas para la Gestión de Documentos Electrónicos en los nuevos servicios de Información y Documentación*. Obtenido de Colegio de Documentación Bibliotecaria de Catalunya: <http://www.cobdc.org/jornades/7JCD/27.pdf>
- gestion.org. (2018). *gestion.org*. Obtenido de Tipos y gestión de la correspondencia: <https://www.gestion.org/tipos-y-gestion-de-la-correspondencia/>
- Gil, J. G. (2019). *OpenWebinars*. Obtenido de <https://openwebinars.net/blog/ques-postgresql/>
- Gómez, M. (2007). *La comunicación en las organizaciones para la mejora de la productividad: El uso de los medios como fuente informativa en empresas e instituciones andaluzas*. Málaga: Universidad de Málaga.
- Guarachi, C., & Rodrigo, A. (2014). *Repositorio Institucional UMSA*. Obtenido de <https://repositorio.umsa.bo/handle/123456789/7940>
- Guest. (12 de 04 de 2020). *Rockcontent*. Obtenido de <https://rockcontent.com/es/blog/bootstrap/>
- Hierro, J. (12 de 2019). *OpenProdoc: Gestor Documental*. Obtenido de http://jhierrot.github.io/openprodoc/index_ES.html
- Julián Pérez Porto, M. M. (2010). Obtenido de <https://definicion.de/carta/>

Larman, C. (2003). *UML y Patrones: Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Madrid, España: Pearson - Prentice Hall.

Larman, C. (2010). *UML y Patrones*. Madrid: Pearson Education.

Manene, L. (22 de 11 de 2013). *La Comunicación y su importancia en la Empresa*. Obtenido de Actualidad Empresa: <http://actualidadempresa.com/la-comunicacion-y-su-importancia-en-la-empresa/>

Microsoft. (01 de noviembre de 2017). *Documentacion de Sql*. Obtenido de <https://docs.microsoft.com/es-es/sql/sql-server/tutorials-for-sql-server-2016?view=sql-server-ver15>

Microsoft. (04 de 10 de 2018). Obtenido de <https://docs.microsoft.com/en-us/aspnet/mvc/overview/getting-started/introduction/getting-started>

Microsoft. (19 de 07 de 2018). *Microsoft Docs*. Obtenido de <https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ef/overview>

Microsoft. (28 de Enero de 2021). *Microsoft Docs*. Obtenido de Microsoft: <https://docs.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/>

Mozilla. (30 de 05 de 2019). *Introducción a Express/Node*. Obtenido de <https://developer.mozilla.org/>:
https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction

que es el lenguaje unificado de modelado uml. (2019). Obtenido de <https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>

Rodriguez, A., & Alfonso, C. (2015). *Repositorio Universidad Libre de Colombia*. Obtenido de https://repository.unilibre.edu.co/bitstream/handle/10901/8434/MONOGRAFIA-GESTION-DOCUMENTAL_.pdf?sequence=1&isAllowed=y

Rodríguez, M. D. (2013). Obtenido de https://dspace.unia.es/bitstream/handle/10334/2472/0431_Rodriguez.pdf;jsessionid=409ED3D7640971A96D5B08A6338177E8?sequence=1

RUP, M. (2019). *Metodología RUP*. Obtenido de <https://metodoss.com/metodologia-rup/>

S.L., O. D. (12 de 2019). *openkmKnowledgeManagement*. Obtenido de <https://www.openkm.com/es/>

School, T. N. (16 de Julio de 2018). *C#, ¿qué es? ¿Quién lo inventó? ¡Conoce este lenguaje de programación!* Obtenido de C#: <https://www.tokioschool.com/noticias/c-que-es/#:~:text=Andr%C3%A9s%20Hejlsberg%20decidi%C3%B3%20formar%20un,de%20programaci%C3%B3n%20orientado%20a%20objetivos.>

Trigos, Y., Claro, R., Claro, N., & Barbosa, S. (22 de 11 de 2019). *Manual de Archivo de Correspondencia*. Obtenido de <https://en.calameo.com/read/004741075872080896ed2>: <https://en.calameo.com/read/004741075872080896ed2>

UNE. (2005). *Proyecto UNE-ISO 15489-1: INFORMACION Y DOCUMENTACION. GESTION DE DOCUMENTOS*. Madrid: UNE.

8 ANEXOS

ANEXO1. Manual de Usuario

Sistema de gestión de correspondencia

Laboratorio Divino Niño

Manual de Usuarios

Contenido

1.	SISTEMA DE CONTROL Y SEGUIMIENTO DE CORRESPONDENCIAS.	- 4 -
2.	INICIAR SISTEMA	- 4 -
3.	INICIAR SESION	- 4 -
4.	PANTALLA PRINCIPAL	- 5 -
4.1.	OPCIONES DE MENÚ	- 6 -
4.2.	OPCIONES DEL USUARIO.....	- 6 -
5.	ACCIONES GENERALES (NUEVO, EDITAR, ELIMINAR, GUARDAR Y CANCELAR) PARA PANTALLAS DEL SISTEMA.....	- 7 -
5.1.	Acción botón Nuevo/Registrar.....	- 8 -
5.2.	Acción Editar	- 8 -
5.3.	Acción Eliminar.....	- 9 -
5.4.	Acción Botón Guardar.....	- 9 -
5.5.	Acción Botón Regresar.....	- 10 -
6.	ADMINISTRACIÓN DEL SISTEMA	- 11 -
6.1.	Permisos Roles	- 11 -
6.1.1.	Nuevo, Editar y Eliminar registro	- 11 -
6.2.	Usuarios.....	- 12 -
6.2.1.	Nuevo, Editar y Eliminar registro	- 13 -
6.3.	Roles.....	- 13 -
6.3.1.	Nuevo, Editar y Eliminar registro	- 14 -
6.4.	Menú del sistema.....	- 14 -

6.4.1.	Nuevo, Editar y Eliminar registro	- 15 -
6.5.	Configuración Correo.....	- 16 -
6.5.1.	Nuevo, Editar y Eliminar registro	- 16 -
7.	CATÁLOGOS DEL SISTEMA	- 17 -
7.1.	Catálogos de personas	- 17 -
7.1.1.	Nuevo, Editar y Eliminar registro	- 18 -
7.2.	Tipo de Persona	- 19 -
7.2.1.	Nuevo, Editar y Eliminar registro	- 19 -
7.3.	Tipos documentos	- 20 -
7.3.1.	Nuevo, Editar y Eliminar registro	- 20 -
7.4.	Tipo de Urgencia Correo.....	- 21 -
7.4.1.	Nuevo, Editar y Eliminar registro	- 21 -
7.5.	Dependencias	- 22 -
7.5.1.	Nuevo, Editar y Eliminar registro	- 22 -
7.6.	Estados Correspondencias.....	- 23 -
7.6.1.	Nuevo, Editar y Eliminar registro	- 23 -
8.	RECEPCIÓN DE CORRESPONDENCIAS.....	- 24 -
8.1.1.	Nuevo, Editar, Eliminar, Ver detalle, Ver documento adjunto.	- 24 -
8.1.2.	Campos en pantalla.....	- 25 -
1.	SEGUIMIENTO DE CORRESPONDENCIAS.....	- 27 -

Sistema de Gestión de correspondencia

El presente documento manual usuario brinda información para el manejo y administración del SGC del Laboratorio Divino Niño. Se presentarán generalidades y detalles de los pasos a seguir por medio de imágenes para el uso del sistema.

Se espera que como un manual del sistema sea un instrumento que apoye la labor que realizan las unidades administrativas para el cumplimiento y realización de sus funciones.

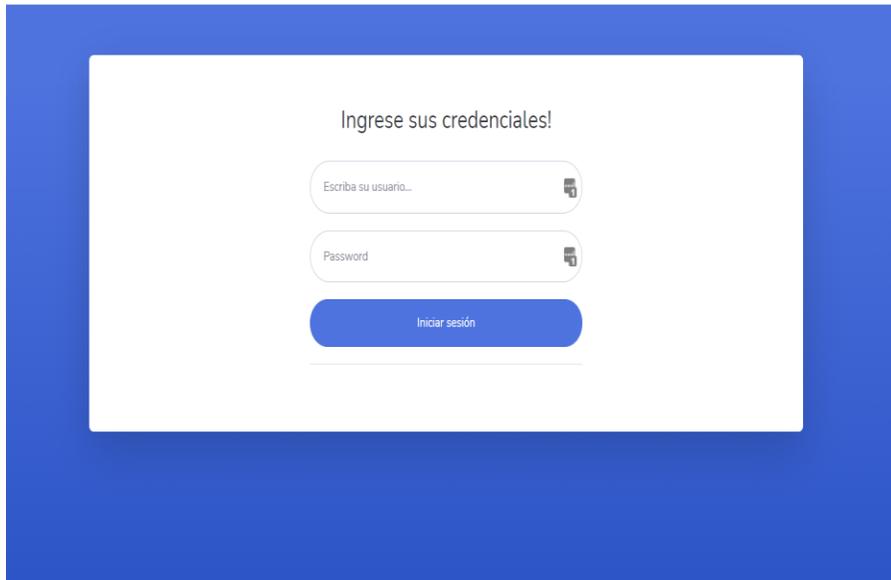
INICIAR SISTEMA

Para iniciar el sistema es necesario ingresar desde un explorador e ingresar la siguiente dirección: <servidor>/Correspondencia/

INICIAR SESION

Una vez iniciado el sistema, aparecerá en pantalla la solicitud para ingresar su usuario y contraseña.

Cuando el usuario hace clic en el botón **Iniciar sesión** o presione la tecla **ENTER**, el sistema consultará la existencia del usuario y generará una matriz de derechos que permitirá en lo sucesivo conocer a que opciones de menú tiene acceso de gestión.



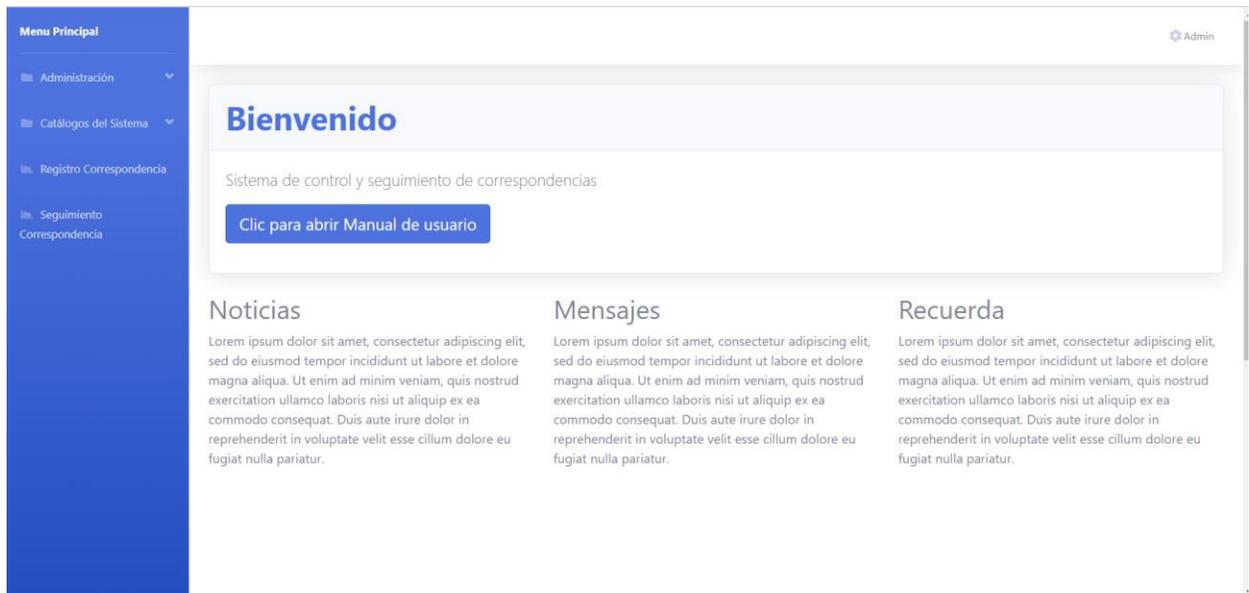
Si el Usuario no está registrado no podrá ingresar al sistema.

Si el recurso humano tiene dificultad en ingresar al sistema por contraseñas erróneas o no posee una contraseña actualizada, deberá enviar un correo al departamento de administración con la correspondiente solicitud de revisión de causa.

PANTALLA PRINCIPAL

Una vez terminada su autenticación, el sistema presenta la pantalla principal que contendrá:

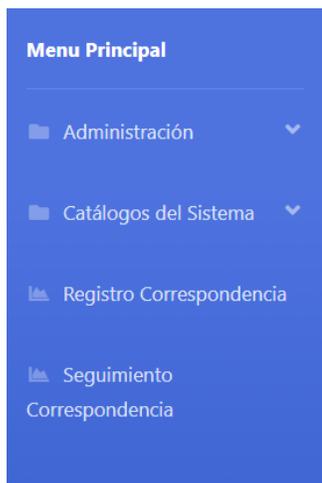
- Menú principal con sus opciones.
- Acciones en de perfil y cerrar Sesión.



A continuación, se detalla cada función.

OPCIONES DE MENÚ

En la parte izquierda de su pantalla aparecen las opciones de menú que usted posee en **dependencia de su rol**, si en un dado caso no aparece ninguno o cree usted que debería de tener mayores opciones de menú favor comunicarse con el administrador del sistema.



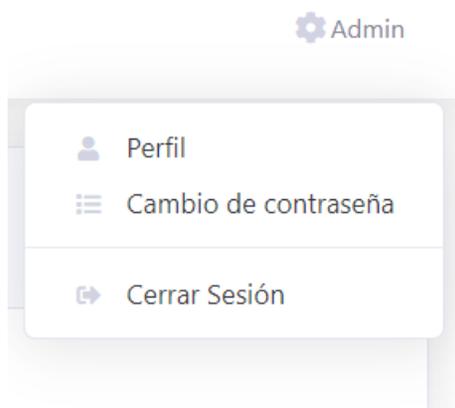
Usted podrá acceder a las opciones que desee haciendo clic sobre las letras en color blanco.

OPCIONES DEL USUARIO

Ubicado en la parte superior de la esquina derecha y sus funciones principales son:

- Administrar tu perfil
- Cambio de contraseña.
- Cerrar Sesión y finalizar el uso del sistema para evitar que personas no autorizadas puedan manipular el sistema en su computadora.

Debe de hacer clic en el nombre de usuario para desplegar las opciones.



Usted podrá cerrar sesión haciendo clic sobre las letras que indican cerrar sesión.

Nota: El sistema posee un procedimiento interno que al detectar 10 minutos de inactividad automáticamente el sistema cierra su sesión actual, muchas veces ocurrirá en el que el sistema pedirá de nuevo que inicie su sesión, eso es normal ya que es parte del procedimiento de seguridad.

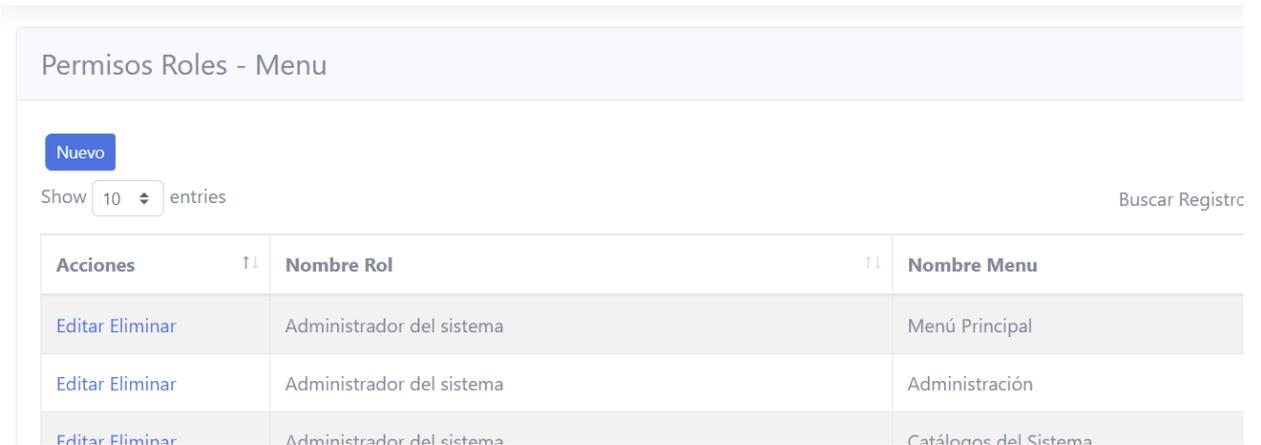
ACCIONES GENERALES (NUEVO, EDITAR, ELIMINAR, GUARDAR Y CANCELAR) PARA PANTALLAS DEL SISTEMA

Para casi todas las pantallas existen acciones y botones donde usted puede crear un nuevo registro, editar y eliminar.

Estas acciones cumplen con el mismo funcionamiento según la pantalla ubicado en el sistema, por tanto, se ocupará igualmente en casi todas las pantallas

Acción botón Nuevo/Registrar

Presione botón  para agregar un nuevo registro en el sistema



Permisos Roles - Menu

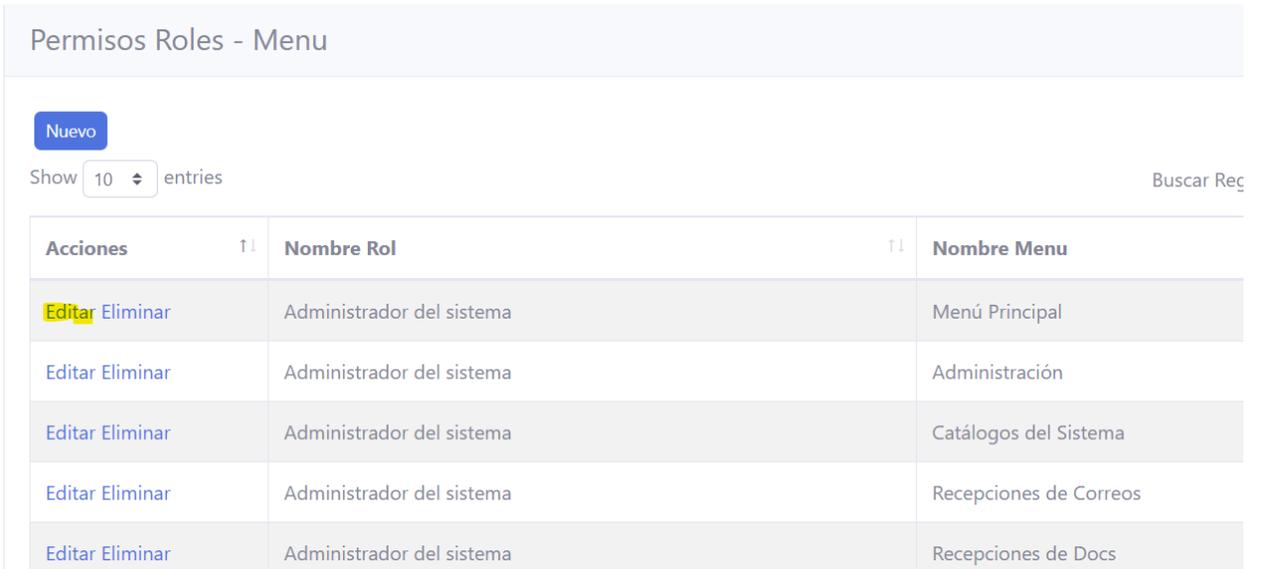
[Nuevo](#)

Show entries Buscar Registro

Acciones	Nombre Rol	Nombre Menu
Editar Eliminar	Administrador del sistema	Menú Principal
Editar Eliminar	Administrador del sistema	Administración
Editar Eliminar	Administrador del sistema	Catálogos del Sistema

Acción Editar

Presione la acción en la  fila del registro a continuación, se desplegará una pantalla para editar el registro.



Permisos Roles - Menu

[Nuevo](#)

Show entries Buscar Reg

Acciones	Nombre Rol	Nombre Menu
Editar Eliminar	Administrador del sistema	Menú Principal
Editar Eliminar	Administrador del sistema	Administración
Editar Eliminar	Administrador del sistema	Catálogos del Sistema
Editar Eliminar	Administrador del sistema	Recepciones de Correos
Editar Eliminar	Administrador del sistema	Recepciones de Docs

Acción Eliminar

Presione el botón **Eliminar** en la fila del registro a continuación el sistema mostrará una pantalla de confirmación.

Permisos Roles - Menu

[Nuevo](#)

Show entries Buscar Reg

Acciones	Nombre Rol	Nombre Menu
Editar Eliminar	Administrador del sistema	Menú Principal
Editar Eliminar	Administrador del sistema	Administración
Editar Eliminar	Administrador del sistema	Catálogos del Sistema
Editar Eliminar	Administrador del sistema	Recepciones de Correos
Editar Eliminar	Administrador del sistema	Recepciones de Docs

Eliminar Permiso

¿Quieres eliminar este registro?

Nombre

Administración

Nombre Rol

Administrador del sistema

[Regresar](#)

[Eliminar](#)

Acción Botón Guardar

Guardar

Presione el botón **Guardar** en el sistema para guardar en la base de datos un nuevo registro/ edición de registro o eliminar un registro.

Crear - Nuevo Permiso Rol - Menu

Rolld
Administrador del sistema

MenuId
Menú Principal

Regresar **Guardar**

Acción Botón Regresar

Regresar

Presione el botón **Regresar** en el sistema para cancelar una acción.

Crear - Nuevo Permiso Rol - Menu

Rolld
Administrador del sistema

MenuId
Menú Principal

Regresar **Guardar**

ADMINISTRACIÓN DEL SISTEMA

Permisos Roles

En esta pantalla usted podrá dar los permisos a los roles para el acceso al sistema.

Permisos Roles - Menu

[Nuevo Permiso](#)

Show entries Buscar Registros

Acciones	Nombre Rol	Nombre Menu
Editar Eliminar	Administrador del sistema	Menú Principal
Editar Eliminar	Administrador del sistema	Administración
Editar Eliminar	Administrador del sistema	Catálogos del Sistema
Editar Eliminar	Administrador del sistema	Recepciones de Correos
Editar Eliminar	Administrador del sistema	Recepciones de Docs
Editar Eliminar	Administrador del sistema	Permisos Usuarios

Nuevo, Editar y Eliminar registro

PARA LOS SIGUIENTES PUNTOS POR FAVOR LEA ACCIONES GENERALES (NUEVO, EDITAR, ELIMINAR, GUARDAR Y CANCELAR) PARA PANTALLAS DEL SISTEMA.

Al presionar el botón nuevo o editar el sistema mostrará en pantalla donde usted tendrá que ingresar o modificar los siguientes campos:

Crear - Nuevo Permiso Rol - Menu

RolId

Administrador del sistema

MenuId

Menú Principal

Regresar Guardar

Puede Guardar o cancelar el registro.

Usuarios

En esta pantalla usted podrá registrar los usuarios que harán uso del sistema al igual que su rol.

Usuarios Sistemas

Nuevo Usuario

Show 10 entries

Buscar Registros

Editar	Usuario	Nombres	Apellidos	Cedula	Rol	Activo	Fecha Creado	Ultima Modificaci
Editar	Admin	Admin	Admin	0011547487991P	Administrador del sistema	True	8/5/2021 02:21:12	18/5/2021 16:15:58
Editar	Test	Test	Test	123123123123	Administrador del sistema	True	8/5/2021 17:05:05	18/5/2021 16:16:06

Showing 1 to 2 of 2 entries

Previous 1 Next

Nuevo, Editar y Eliminar registro

PARA LOS SIGUIENTES PUNTOS POR FAVOR LEA ACCIONES GENERALES (NUEVO, EDITAR, ELIMINAR, GUARDAR Y CANCELAR) PARA PANTALLAS DEL SISTEMA.

Al presionar el botón nuevo o editar el sistema mostrará en pantalla donde usted tendrá que ingresar o modificar los siguientes campos:

Editar - Catálogo de usuarios del sistema

NombreUsuario	<input type="text" value="Admin"/>
Nombres	<input type="text" value="Admin"/>
Apellidos	<input type="text" value="Admin"/>
Cedula	<input type="text" value="0011547487991P"/>
Seleccione Rol	<input type="text" value="Administrador del sistema"/>
Activo	<input checked="" type="checkbox"/>

Roles

En esta pantalla usted podrá agregar un nuevo rol.

Roles

[Nuevo Rol](#)

Show entries Buscar Registros

Editar ↑↓	Nombre Rol	↑↓
Editar	Administrador del sistema	
Editar	Recepcionista	
Editar	Reportes	
Editar	Reportes 2	

Showing 1 to 4 of 4 entries Previous **1** Next

Nuevo, Editar y Eliminar registro

PARA LOS SIGUIENTES PUNTOS POR FAVOR LEA ACCIONES GENERALES (NUEVO, EDITAR, ELIMINAR, GUARDAR Y CANCELAR) PARA PANTALLAS DEL SISTEMA.

Al presionar el botón nuevo o editar el sistema mostrará en pantalla donde usted tendrá que ingresar o modificar los siguientes campos:

Crear - Rol

Nombre Rol

[Regresar](#) [Guardar](#)

Menú del sistema

Pantalla que se utiliza para grabar la estructura del menú del sistema.

Nota: Esta pantalla es de uso del administrador, ya que un cambio mínimo puede dañar la estructura del menú del sistema

Menu del sistema

[Nuevo Menu](#)

Show entries

Acciones ^{↑↓}	Nombre Menu ^{↑↓}	Menu URL ^{↑↓}	Menu Padre ^{↑↓}	Nivel ^{↑↓}
Editar	Menú Principal	/	Menú Principal	1
Editar	Administración	#	Menú Principal	2
Editar	Catálogos del Sistema	#	Menú Principal	2
Editar	Recepciones de Correos	#	Menú Principal	2
Editar	Recepciones de Docs	#	Menú Principal	2
Editar	Permisos Usuarios	#	Administración	3
Editar	Usuarios	/UsuariosSistemas	Administración	3
Editar	Menus	/Menus	Administración	3

Showing 1 to 8 of 8 entries

Nuevo, Editar y Eliminar registro

PARA LOS SIGUIENTES PUNTOS POR FAVOR LEA ACCIONES GENERALES (NUEVO, EDITAR, ELIMINAR, GUARDAR Y CANCELAR) PARA PANTALLAS DEL SISTEMA.

Al presionar el botón nuevo o editar el sistema mostrará en pantalla donde usted tendrá que ingresar o modificar los siguientes campos:

Editar Menu

Nombre

MenuURL

Menu Padre

Regresar
Guardar

Configuración Correo

Pantalla que se utiliza para grabar la configuración del correo que enviará los mensajes de confirmación a los remitentes o destinatario al ingresar una correspondencia del menú del sistema.

Nota: Mucho cuidado con la información ingresar ya que un cambio mínimo puede dañar el funcionamiento de envío de correos.

Configuración de Correos

Nueva configuración

Show entries Buscar Registros

Editar ↑↓	Correo ↑↓
Editar Eliminar	testinghvn@gmail.com

Showing 1 to 1 of 1 entries
Previous
1
Next

Nuevo, Editar y Eliminar registro

PARA LOS SIGUIENTES PUNTOS POR FAVOR LEA ACCIONES GENERALES (NUEVO, EDITAR, ELIMINAR, GUARDAR Y CANCELAR) PARA PANTALLAS DEL SISTEMA.

Al presionar el botón nuevo o editar el sistema mostrará en pantalla donde usted tendrá que ingresar o modificar los siguientes campos:

Editar catálogo de configuración de correos

Correo

Contraseña

Host

Port

EnableSSL

DefaultCredential

Activo

CATÁLOGOS DEL SISTEMA

Datos e información necesarias para ingresar para el funcionamiento del sistema.

Catálogos de personas

En esta pantalla es donde se ingresa información de las personas clientes, administrativas, instituciones, proveedores y que utilizarán para el ingreso de una correspondencia. **Muy distinta a la pantalla de usuarios del sistema.**

Catálogo de Personas - Clientes

Nuevo Catálogo de Personas - Clientes

Show 10 entries

Buscar Registros

Acciones	Nombre Completo	Correo	Tipo de Personal
Editar Eliminar	Dr. Prueba	drprueba@gmail.com	Empleado
Editar Eliminar	Dr.	Dr@hotmail.es	Exterior
Editar Eliminar	Empleado1	empleado1@hotmail.com	Empleado
Editar Eliminar	Empleado 2	empleado2@hotmail.com	Empleado
Editar Eliminar	Empleado 3	empleado1@hotmail.com	Empleado
Editar Eliminar	Empleado 4	empleado4@hotmail.com	Empleado
Editar Eliminar	Cliente1	cliente1@hotmail.com	Cliente
Editar Eliminar	Cliente2	cliente2@hotmail.com	Cliente

Nuevo, Editar y Eliminar registro

PARA LOS SIGUIENTES PUNTOS POR FAVOR LEA ACCIONES GENERALES (NUEVO, EDITAR, ELIMINAR, GUARDAR Y CANCELAR) PARA PANTALLAS DEL SISTEMA.

Al presionar el botón nuevo o editar el sistema mostrará en pantalla donde usted tendrá que ingresar o modificar los siguientes campos:

Crear - Personas

Nombre Completo

Correo

Tipo de personal

Tipo de Persona

Pantalla que guarda la clasificación de tipos de persona, dicha información es ocupada en la pantalla de catálogos de personas.

Tipos de Personas

[Nuevo Tipos de Personas](#)

Show entries

Acciones	Nombre Tipo Persona
Editar Eliminar	Empleado
Editar Eliminar	Exterior
Editar Eliminar	Cliente

Showing 1 to 3 of 3 entries Previous **1** Next

Nuevo, Editar y Eliminar registro

PARA LOS SIGUIENTES PUNTOS POR FAVOR LEA ACCIONES GENERALES (NUEVO, EDITAR, ELIMINAR, GUARDAR Y CANCELAR) PARA PANTALLAS DEL SISTEMA.

Al presionar el botón nuevo o editar el sistema mostrará en pantalla donde usted tendrá que ingresar o modificar los siguientes campos:

Editar - Catálogo de tipo de persona

Nombre

[Regresar](#) [Guardar](#)

Tipos documentos

Pantalla que guarda la clasificación de tipos de documentos de la correspondencia, dicha información es ocupada en la pantalla de correspondencias.

Tipos de Documentos

[Nuevo Documento](#)

Show entries Buscar Registros

Acciones ↑↓	Nombre Tipo Documento ↑↓	Usuario Registro ↑↓	Fecha Registro ↑↓	Fecha Última Modificación ↑↓
Editar Eliminar	Cobranza	Admin	9/5/2021 12:25:05	9/5/2021 12:26:16
Editar Eliminar	Consultas	Admin	9/5/2021 12:25:14	9/5/2021 12:26:21

Showing 1 to 2 of 2 entries Previous **1** Next

Nuevo, Editar y Eliminar registro

PARA LOS SIGUIENTES PUNTOS POR FAVOR LEA ACCIONES GENERALES (NUEVO, EDITAR, ELIMINAR, GUARDAR Y CANCELAR) PARA PANTALLAS DEL SISTEMA.

Al presionar el botón nuevo o editar el sistema mostrará en pantalla donde usted tendrá que ingresar o modificar los siguientes campos:

Editar catálogo de tipos de documentación

Nombre Tipo Documento

[Regresar](#) [Guardar](#)

Tipo de Urgencia Correo

Pantalla que guarda el tipo de correspondencia que se registra. Dicha información se ocupa en registro de correspondencia.

Catálogo de Tipo Urgencia

[Nuevo Catálogo de Tipo Urgencia](#)

Show entries Buscar Registros

Acciones ↑↓	Nombre Tipo Urgencia ↑↓	Usuario Registro ↑↓	Fecha Registro ↑↓	Fecha Última Modificación ↑↓
Editar Eliminar	Alto	Admin	9/5/2021 12:35:24	
Editar Eliminar	Medio	Admin	9/5/2021 12:35:32	
Editar Eliminar	Bajo	Admin	9/5/2021 12:35:46	

Showing 1 to 3 of 3 entries Previous **1** Next

Nuevo, Editar y Eliminar registro

PARA LOS SIGUIENTES PUNTOS POR FAVOR LEA ACCIONES GENERALES (NUEVO, EDITAR, ELIMINAR, GUARDAR Y CANCELAR) PARA PANTALLAS DEL SISTEMA.

Al presionar el botón nuevo o editar el sistema mostrará en pantalla donde usted tendrá que ingresar o modificar los siguientes campos:

Editar - Catálogo de tipos de urgencias correos

Nombre Tipo Urgencia

[Regresar](#) [Guardar](#)

Dependencias

Pantalla utilizada para guardar las dependencias. Dicha información se utiliza

Dependencias

[Nueva Dependencia](#)

Acciones	Nombre Dependencia	Usuario Registro	Fecha Registro	Fecha Última Modificación
Editar Eliminar	Informática	Admin	12/5/2021 11:48:18	
Editar Eliminar	RRHH	Admin	12/5/2021 11:48:25	12/5/2021 11:49:18

Nuevo, Editar y Eliminar registro

PARA LOS SIGUIENTES PUNTOS POR FAVOR LEA ACCIONES GENERALES (NUEVO, EDITAR, ELIMINAR, GUARDAR Y CANCELAR) PARA PANTALLAS DEL SISTEMA.

Al presionar el botón nuevo o editar el sistema mostrará en pantalla donde usted tendrá que ingresar o modificar los siguientes campos:

Editar - Dependencia

Nombre

[Regresar](#) [Guardar](#)

Estados Correspondencias

Pantalla que guardar los estados de las correspondencias con el fin de darle un seguimiento a las correspondencias

Estado Correspondencia

[Nuevo Estado](#)

Show entries Buscar Registros

Editar <small>T </small>	Nombre estado	<small>T </small>
Editar Eliminar	En espera de entrega	
Editar Eliminar	Recibido, pendiente de respuesta	
Editar Eliminar	Finalizado	
Editar Eliminar	Entregado, pendiente de respuesta	

Showing 1 to 4 of 4 entries [Previous](#) [1](#) [Next](#)

Nuevo, Editar y Eliminar registro

PARA LOS SIGUIENTES PUNTOS POR FAVOR LEA ACCIONES GENERALES (NUEVO, EDITAR, ELIMINAR, GUARDAR Y CANCELAR) PARA PANTALLAS DEL SISTEMA.

Al presionar el botón nuevo o editar el sistema mostrará en pantalla donde usted tendrá que ingresar o modificar los siguientes campos:

Crear - Estado Correspondencia

Nombre

[Regresar](#) [Guardar](#)

RECEPCIÓN DE CORRESPONDENCIAS

Esta pantalla es de una las principales, guarda toda correspondencia enviadas y recibidas a la institución.

Administración de correspondencias

[Registrar Correspondencia Recibida](#) [Registrar correspondencia Enviada](#)

Show entries Buscar Registros

Acciones	Identificador único	Referencia	Observación	Destinatario	Tipo de Correspondencia	Estado	Tipo de documento	Ver Documento
Editar Ver Detalle Eliminar	REC-11	Dr Hanssel	Prueba	drprueba@gmail.com	Recibido	En espera de entrega	Cobranza	Ver Documento

Showing 1 to 1 of 1 entries [Previous](#) [1](#) [Next](#)

Nuevo, Editar, Eliminar, Ver detalle, Ver documento adjunto.

PARA LOS SIGUIENTES PUNTOS POR FAVOR LEA ACCIONES GENERALES (NUEVO, EDITAR, ELIMINAR, GUARDAR Y CANCELAR) PARA PANTALLAS DEL SISTEMA.

En la pantalla principal aparecerá una lista de las correspondencias registradas.

A continuación, se explica cada acción que se encuentra en la pantalla principal.

Para crear una nueva correspondencia de tipo **recibida** deberá de dar clic en el botón

[Registrar Correspondencia Recibida](#)

Para crear una nueva correspondencia de tipo **enviada** deberá de dar clic en el botón

[Registrar correspondencia Enviada](#)

Una vez seleccionado el tipo de correspondencia **recibida/enviada** el sistema mostrará un formulario para ingreso de información

Registrar - Correspondencia Recibida

Referencia

Destinatario [Agregar Nuevo Personal](#)
[Actualizar Lista de correos](#)

Remitente

Observación

Seleccione tipo de documento Nivel de urgencia

¿Se requiere fecha de respuesta?

¿Es una respuesta de correspondencia anterior?

Estado

Adjuntar Imagen Digital (Opcional)
 No se eligió archivo

Campos en pantalla

En pantalla aparecerán los siguientes campos:

Referencia: Campo para escribir una referencia, puede ser el nombre de una persona, institución o algo en específico que se necesita como una referencia.

Destinatario y remitente: En este campo se puede seleccionar el correo de la persona destinatario y remitente.

En caso de que usted necesite agregar un nuevo destinatario o remitente puede hacerlo con la acción: “[Agregar Nuevo Personal](#)”, el sistema lo enviará a la pantalla para agregar un nuevo personal.

Utilice la acción “[Actualizar Lista de correos](#)” para actualizar la lista de destinatarios y remitentes.

Observación: Campo para agregar una observación.

Tipo de documento: Campo de selección utilizado para categorizar que tipo de documento se está recibiendo o enviando.

Nivel de Urgencia: Campo de selección utilizado para indicar el nivel de urgencia de la correspondencia.

¿Ser requiere fecha de respuesta?: Se utiliza para indicar si se necesita responder. Al seleccionar la fecha respuesta el sistema le presentará un nuevo campo para indicar cuando es la fecha límite de respuesta.

¿Se requiere fecha de respuesta?

FechaRespuesta

¿Es una respuesta de correspondencia anterior?: En caso de indicar que la correspondencia es una respuesta a una anterior puede utilizar este campo para seleccionar a que dicha correspondencia.

¿Es una respuesta de correspondencia anterior?

Seleccione correspondencia

Estado: Campo de selección que indica en que estado se encuentra la correspondencia. De alta importancia ya que se utilizará para el seguimiento de las correspondencias.

Adjuntar documentación: Utilizado para subir archivos de cualquier tipo documento e imágenes al sistema.

SEGUIMIENTO DE CORRESPONDENCIAS

Pantalla para realizar seguimiento a las correspondencias según su estado y con filtros de estado y destinatario.

Seguimiento de correspondencias

Estado: Destinatario:

Show entries

Acciones	Identificador único	Referencia	Destinatario	Estado
Modificar Estado Ver Historial Imprimir	REC-11	Dr Hanssel	hansselnavarro@gmail.com	Finalizado
Modificar Estado Ver Historial Imprimir	REC-12	Dr Hanssel	hansselnavarro@gmail.com	En espera de entrega

Showing 1 to 2 of 2 entries

Acciones de la pantalla

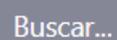
PARA LOS SIGUIENTES PUNTOS POR FAVOR LEA ACCIONES GENERALES (NUEVO, EDITAR, ELIMINAR, GUARDAR Y CANCELAR) PARA PANTALLAS DEL SISTEMA.

En la pantalla principal aparecerá una lista de las correspondencias registradas.

A continuación, se explica cada acción que se encuentra en la pantalla principal.

Filtrar Lista: La pantalla muestra campos para filtrar la lista de correspondencias

Estado: Destinatario:



Utiliza el botón para filtrar por los campos seleccionados.

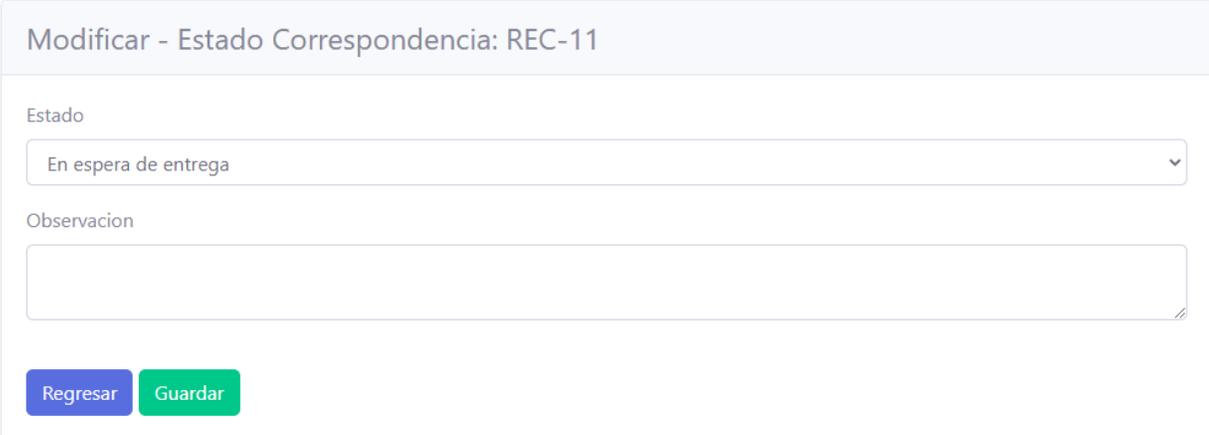


Utiliza el botón para limpiar filtros y ver todos los registros de correspondencia.

Modificar Estado

Pantalla utilizada para actualizar el estado de la correspondencia. Se puede agregar una observación al guardar al cambiar un estado.

Dicha pantalla almacena un historial de los cambios realizado a la correspondencia con su fecha en la que se realizó la modificación.



Modificar - Estado Correspondencia: REC-11

Estado

En espera de entrega

Observacion

Regresar Guardar

Ver Historial

Muestra todo el historial de cambios de estados:

Historial de Correspondencias

Buscar:

#	Estado	Destinatario	Correo	Fecha Respuesta	Fecha Modificación
1	En espera de entrega	Hanssel Navarro Murillo	hansselnavarro@gmail.com	14/05/2021	18/05/2021
2	Finalizado	Hanssel Navarro Murillo	hansselnavarro@gmail.com	14/05/2021	18/05/2021
3	Recibido, pendiente de respuesta	Hanssel Navarro Murillo	hansselnavarro@gmail.com	14/05/2021	18/05/2021
4	Finalizado	Hanssel Navarro Murillo	hansselnavarro@gmail.com	14/05/2021	18/05/2021
5	Entregado, pendiente de respuesta	Hanssel Navarro Murillo	hansselnavarro@gmail.com	14/05/2021	18/05/2021
6	En espera de entrega	Hanssel Navarro Murillo	hansselnavarro@gmail.com	14/05/2021	18/05/2021
7	Recibido, pendiente de respuesta	Hanssel Navarro Murillo	hansselnavarro@gmail.com	14/05/2021	18/05/2021
8	Finalizado	Hanssel Navarro Murillo	hansselnavarro@gmail.com	14/05/2021	18/05/2021
9	Finalizado	Hanssel Navarro Murillo	hansselnavarro@gmail.com	14/05/2021	18/05/2021
10	Finalizado	Hanssel Navarro Murillo	hansselnavarro@gmail.com	14/05/2021	18/05/2021

1 - 10, de 10 registros

< 1 >

Imprimir

Imprime todo los datos y detalle de cambios en modo de reporte.

Correspondencia	Estado	Destinatario	Destinatario
11		Hanssel Navarro Murillo	hansselnavarro@gmail.com
Referencia	Fecha Respuesta		
Dr Hanssel	14/5/2021 00:00:00		
Estado	FechaModificacion	Observacion	FechaRegistro
En espera de entrega	18/5/2021 13:57:00	test	18/5/2021 13:57:00
Finalizado	18/5/2021 15:38:00	test1	18/5/2021 15:38:00
Recibido, pendiente de respues	18/5/2021 15:41:00	test3	18/5/2021 15:41:00
Finalizado	18/5/2021 15:43:00	test4	18/5/2021 15:43:00
Entregado, pendiente de respue	18/5/2021 15:44:00	test5	18/5/2021 15:44:00
En espera de entrega	18/5/2021 15:44:00	tstst	18/5/2021 15:44:00
Recibido, pendiente de respues	18/5/2021 15:44:00	tssetett4	18/5/2021 15:44:00
Finalizado	18/5/2021 15:44:00	tessss	18/5/2021 15:44:00
Finalizado	18/5/2021 15:45:00		18/5/2021 15:45:00
Finalizado	18/5/2021 15:52:00	fgggg	18/5/2021 15:52:00

ANEXO 2. Scripts para instalación de Base de Datos

```
USE [master]
GO
/***** Object: Database [DivinoNinoBD] Script Date: 4/6/2021 16:07:20 *****/
CREATE DATABASE [DivinoNinoBD]
GO
ALTER DATABASE [DivinoNinoBD] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [DivinoNinoBD] SET ANSI_NULLS OFF
GO
ALTER DATABASE [DivinoNinoBD] SET ANSI_PADDING OFF
GO
ALTER DATABASE [DivinoNinoBD] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [DivinoNinoBD] SET ARITHABORT OFF
GO
ALTER DATABASE [DivinoNinoBD] SET AUTO_CLOSE ON
GO
ALTER DATABASE [DivinoNinoBD] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [DivinoNinoBD] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [DivinoNinoBD] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [DivinoNinoBD] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [DivinoNinoBD] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [DivinoNinoBD] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [DivinoNinoBD] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [DivinoNinoBD] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [DivinoNinoBD] SET ENABLE_BROKER
GO
ALTER DATABASE [DivinoNinoBD] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
```

```

GO
ALTER DATABASE [DivinoNinoBD] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [DivinoNinoBD] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [DivinoNinoBD] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [DivinoNinoBD] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [DivinoNinoBD] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [DivinoNinoBD] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [DivinoNinoBD] SET RECOVERY SIMPLE
GO
ALTER DATABASE [DivinoNinoBD] SET MULTI_USER
GO
ALTER DATABASE [DivinoNinoBD] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [DivinoNinoBD] SET DB_CHAINING OFF
GO
ALTER DATABASE [DivinoNinoBD] SET FILESTREAM( NON_TRANSACTED_ACCESS = OFF )
GO
ALTER DATABASE [DivinoNinoBD] SET TARGET_RECOVERY_TIME = 60 SECONDS
GO
ALTER DATABASE [DivinoNinoBD] SET DELAYED_DURABILITY = DISABLED
GO
ALTER DATABASE [DivinoNinoBD] SET QUERY_STORE = OFF
GO
USE [DivinoNinoBD]
GO
/***** Object: Table [dbo].[Estados] Script Date: 4/6/2021 16:07:21 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Estados](
    [EstadosId] [int] IDENTITY(1,1) NOT NULL,
    [Nombre] [varchar](50) NULL,
    CONSTRAINT [PK_Estados] PRIMARY KEY CLUSTERED

```

```

(
    [EstadosId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Personas] Script Date: 4/6/2021 16:07:21 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Personas](
    [PersonalId] [int] IDENTITY(1,1) NOT NULL,
    [Nombre] [varchar](200) NULL,
    [Correo] [varchar](200) NOT NULL,
    [TipoPersonalId] [int] NULL,
    CONSTRAINT [PK_Personas] PRIMARY KEY CLUSTERED
(
    [PersonalId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[UsuariosSistemas] Script Date: 4/6/2021 16:07:21 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[UsuariosSistemas](
    [UsuarioId] [int] IDENTITY(1,1) NOT NULL,
    [RolId] [int] NULL,
    [NombreUsuario] [varchar](50) NULL,
    [Contrasena] [varchar](650) NULL,
    [Nombres] [varchar](50) NULL,
    [Apellidos] [varchar](50) NULL,
    [Cedula] [varchar](16) NULL,
    [Activo] [bit] NULL,
    [FechaRegistro] [datetime] NOT NULL,
    [FechaModificacion] [datetime] NULL,
    CONSTRAINT [PK_UsuariosSistema] PRIMARY KEY CLUSTERED

```

```

(
    [Usuarioid] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

/***** Object: Table [dbo].[Correspondencias] Script Date: 4/6/2021 16:07:21 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Correspondencias](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Referencia] [nvarchar](120) NULL,
    [Observacion] [nvarchar](500) NULL,
    [RemitentePersonald] [int] NULL,
    [DestinatarioPersonald] [int] NULL,
    [TipoDocumentold] [int] NULL,
    [ExamenId] [int] NULL,
    [TipoUrgenciald] [int] NULL,
    [Correspondenciald] [int] NULL,
    [Respuesta] [bit] NULL,
    [FechaRespuesta] [datetime] NULL,
    [FechaHoraRecibe] [datetime] NULL,
    [PersonaRecibe] [nvarchar](100) NULL,
    [Estadold] [int] NULL,
    [TipoCorrespondenciald] [int] NULL,
    [DocumentoURL] [nvarchar](100) NULL,
    [DocumentoURLCompleto] [nvarchar](300) NULL,
    [Usuarioid] [int] NULL,
    [FechaRegistro] [datetime] NULL,
    [FechaModificacion] [datetime] NULL,
CONSTRAINT [PK_CorrespondenciaEnviadas] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

```

```

/***** Object: Table [dbo].[CorreosConfig] Script Date: 4/6/2021 16:07:21 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[CorreosConfig](
    [Correoid] [int] IDENTITY(1,1) NOT NULL,
    [correo] [varchar](70) NULL,
    [Contrasena] [varchar](250) NULL,
    [Host] [varchar](70) NULL,
    [Port] [int] NULL,
    [EnableSSL] [bit] NULL,
    [DefaultCredential] [bit] NULL,
    [Activo] [bit] NULL,
    CONSTRAINT [PK_correoconfig] PRIMARY KEY CLUSTERED
(
    [Correoid] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Dependencias] Script Date: 4/6/2021 16:07:21 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Dependencias](
    [Dependenciald] [int] IDENTITY(1,1) NOT NULL,
    [Nombre] [varchar](100) NULL,
    [Usuarioid] [int] NULL,
    [FechaRegistro] [datetime] NULL,
    [FechaModificacion] [datetime] NULL,
    CONSTRAINT [PK_Dependencias] PRIMARY KEY CLUSTERED
(
    [Dependenciald] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Examen] Script Date: 4/6/2021 16:07:21 *****/

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Examen](
    [ExamenId] [int] IDENTITY(1,1) NOT NULL,
    [NombreExamen] [varchar](100) NULL,
    [NombrePaciente] [varchar](50) NULL,
    [NombreDoctor] [varchar](50) NULL,
    [Fecha] [datetime] NULL,
    CONSTRAINT [PK_Examen] PRIMARY KEY CLUSTERED
(
    [ExamenId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[ExamenDetalle] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[ExamenDetalle](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [ExamenId] [int] NOT NULL,
    [Determinacion] [nvarchar](100) NULL,
    [Resultado] [nvarchar](100) NULL,
    [ValoresReferencia] [nvarchar](100) NULL,
    CONSTRAINT [PK_ExamenDetalles] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Menu] Script Date: 4/6/2021 16:07:21 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```

CREATE TABLE [dbo].[Menu](
    [MenuId] [int] IDENTITY(1,1) NOT NULL,
    [Nombre] [varchar](50) NULL,
    [MenuURL] [varchar](150) NULL,
    [MenuPadreId] [int] NULL,
    [Nivel] [tinyint] NULL,
    [Usuarioid] [int] NULL,
    [FechaRegistro] [datetime] NULL,
    [FechaModificacion] [datetime] NULL,
CONSTRAINT [PK_Menu] PRIMARY KEY CLUSTERED
(
    [MenuId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[PermisosRolesMenu] Script Date: 4/6/2021 16:07:21 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[PermisosRolesMenu](
    [PermisosId] [int] IDENTITY(1,1) NOT NULL,
    [RolId] [int] NULL,
    [MenuId] [int] NULL,
CONSTRAINT [PK_Permisos] PRIMARY KEY CLUSTERED
(
    [PermisosId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Roles] Script Date: 4/6/2021 16:07:21 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Roles](
    [RolId] [int] IDENTITY(1,1) NOT NULL,
    [NombreRol] [varchar](50) NULL,

```

```

        [FechaRegistro] [datetime] NULL,
    CONSTRAINT [PK_Roles] PRIMARY KEY CLUSTERED
    (
        [RolId] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]
GO
/***** Object: Table [dbo].[TipoCorrespondencias] Script Date: 4/6/2021 16:07:21 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[TipoCorrespondencias](
    [TipoCorrespondenciald] [int] IDENTITY(1,1) NOT NULL,
    [Nombre] [varchar](50) NULL,
    CONSTRAINT [PK_TipoCorrespondencias] PRIMARY KEY CLUSTERED
    (
        [TipoCorrespondenciald] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]
GO
/***** Object: Table [dbo].[TipoDocumentos] Script Date: 4/6/2021 16:07:21 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[TipoDocumentos](
    [TipoDocumentold] [int] IDENTITY(1,1) NOT NULL,
    [Nombre] [varchar](50) NULL,
    [Usuariold] [int] NULL,
    [FechaRegistro] [datetime] NULL,
    [FechaModificacion] [datetime] NULL,
    CONSTRAINT [PK_TipoDocumento] PRIMARY KEY CLUSTERED
    (
        [TipoDocumentold] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

```

```

GO
/***** Object: Table [dbo].[TipoPersonas] Script Date: 4/6/2021 16:07:21 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[TipoPersonas](
    [TipoPersonald] [int] IDENTITY(1,1) NOT NULL,
    [Nombre] [varchar](50) NULL,
    CONSTRAINT [PK_TipoPersonas] PRIMARY KEY CLUSTERED
(
    [TipoPersonald] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[TipoUrgencias] Script Date: 4/6/2021 16:07:21 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[TipoUrgencias](
    [TipoUrgenciald] [int] IDENTITY(1,1) NOT NULL,
    [Nombre] [varchar](50) NULL,
    [Usuariod] [int] NULL,
    [FechaRegistro] [datetime] NULL,
    [FechaModificacion] [datetime] NULL,
    CONSTRAINT [PK_TipoUrgencia] PRIMARY KEY CLUSTERED
(
    [TipoUrgenciald] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Correspondencias] WITH CHECK ADD CONSTRAINT
[FK_Correspondencias_Estados] FOREIGN KEY([Estadod])
REFERENCES [dbo].[Estados] ([EstadosId])
GO
ALTER TABLE [dbo].[Correspondencias] CHECK CONSTRAINT [FK_Correspondencias_Estados]
GO

```

```

ALTER TABLE [dbo].[Correspondencias] WITH CHECK ADD CONSTRAINT
[FK_Correspondencias_TipoCorrespondencias] FOREIGN KEY([TipoCorrespondenciald])
REFERENCES [dbo].[TipoCorrespondencias] ([TipoCorrespondenciald])
GO
ALTER TABLE [dbo].[Correspondencias] CHECK CONSTRAINT
[FK_Correspondencias_TipoCorrespondencias]
GO
ALTER TABLE [dbo].[Correspondencias] WITH CHECK ADD CONSTRAINT
[FK_Correspondencias_TipoDocumentos] FOREIGN KEY([TipoDocumentoid])
REFERENCES [dbo].[TipoDocumentos] ([TipoDocumentoid])
GO
ALTER TABLE [dbo].[Correspondencias] CHECK CONSTRAINT
[FK_Correspondencias_TipoDocumentos]
GO
ALTER TABLE [dbo].[Correspondencias] WITH CHECK ADD CONSTRAINT
[FK_Correspondencias_TipoUrgencias] FOREIGN KEY([TipoUrgenciald])
REFERENCES [dbo].[TipoUrgencias] ([TipoUrgenciald])
GO
ALTER TABLE [dbo].[Correspondencias] CHECK CONSTRAINT
[FK_Correspondencias_TipoUrgencias]
GO
ALTER TABLE [dbo].[Correspondencias] WITH CHECK ADD CONSTRAINT
[FK_Correspondencias_UsuariosSistemas] FOREIGN KEY([Usuarioid])
REFERENCES [dbo].[UsuariosSistemas] ([Usuarioid])
GO
ALTER TABLE [dbo].[Correspondencias] CHECK CONSTRAINT
[FK_Correspondencias_UsuariosSistemas]
GO
ALTER TABLE [dbo].[CorrespondenciasModificaciones] WITH CHECK ADD CONSTRAINT
[FK_CorrespondenciasModificaciones_Correspondencias] FOREIGN KEY([Correspondenciald])
REFERENCES [dbo].[Correspondencias] ([Id])
GO
ALTER TABLE [dbo].[CorrespondenciasModificaciones] CHECK CONSTRAINT
[FK_CorrespondenciasModificaciones_Correspondencias]
GO
ALTER TABLE [dbo].[CorrespondenciasModificaciones] WITH CHECK ADD CONSTRAINT
[FK_CorrespondenciasModificaciones_Estados] FOREIGN KEY([EstadosId])
REFERENCES [dbo].[Estados] ([EstadosId])
GO

```

```

ALTER TABLE [dbo].[CorrespondenciasModificaciones] CHECK CONSTRAINT
[FK_CorrespondenciasModificaciones_Estados]
GO
ALTER TABLE [dbo].[CorrespondenciasModificaciones] WITH CHECK ADD CONSTRAINT
[FK_CorrespondenciasModificaciones_UsuariosSistemas] FOREIGN KEY([Usuarioid])
REFERENCES [dbo].[UsuariosSistemas] ([Usuarioid])
GO
ALTER TABLE [dbo].[CorrespondenciasModificaciones] CHECK CONSTRAINT
[FK_CorrespondenciasModificaciones_UsuariosSistemas]
GO
ALTER TABLE [dbo].[ExamenDetalle] WITH CHECK ADD CONSTRAINT
[FK_ExamenDetalle_Examen] FOREIGN KEY([ExamenId])
REFERENCES [dbo].[Examen] ([ExamenId])
GO
ALTER TABLE [dbo].[ExamenDetalle] CHECK CONSTRAINT [FK_ExamenDetalle_Examen]
GO
ALTER TABLE [dbo].[Menu] WITH CHECK ADD CONSTRAINT [FK_Menu_Menu] FOREIGN
KEY([MenuPadreId])
REFERENCES [dbo].[Menu] ([MenuId])
GO
ALTER TABLE [dbo].[Menu] CHECK CONSTRAINT [FK_Menu_Menu]
GO
ALTER TABLE [dbo].[PermisosRolesMenu] WITH CHECK ADD CONSTRAINT
[FK_PermisosRolesMenu_Menu] FOREIGN KEY([MenuId])
REFERENCES [dbo].[Menu] ([MenuId])
GO
ALTER TABLE [dbo].[PermisosRolesMenu] CHECK CONSTRAINT [FK_PermisosRolesMenu_Menu]
GO
ALTER TABLE [dbo].[PermisosRolesMenu] WITH CHECK ADD CONSTRAINT
[FK_PermisosRolesMenu_Roles] FOREIGN KEY([RolId])
REFERENCES [dbo].[Roles] ([RolId])
GO
ALTER TABLE [dbo].[PermisosRolesMenu] CHECK CONSTRAINT [FK_PermisosRolesMenu_Roles]
GO
ALTER TABLE [dbo].[Personas] WITH CHECK ADD CONSTRAINT [FK_Personas_TipoPersonas]
FOREIGN KEY([TipoPersonalId])
REFERENCES [dbo].[TipoPersonas] ([TipoPersonalId])
GO
ALTER TABLE [dbo].[Personas] CHECK CONSTRAINT [FK_Personas_TipoPersonas]
GO

```

```

ALTER TABLE [dbo].[UsuariosSistemas] WITH CHECK ADD CONSTRAINT
[FK_UsuariosSistema_Roles] FOREIGN KEY([RolId])
REFERENCES [dbo].[Roles] ([RolId])
GO
ALTER TABLE [dbo].[UsuariosSistemas] CHECK CONSTRAINT [FK_UsuariosSistema_Roles]
GO
/***** Object: StoredProcedure [dbo].[sp_ReporteEstadistico] Script Date: 4/6/2021 16:07:21
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[sp_ReporteEstadistico]
@fi varchar(11),
@ff varchar(11),
@tipo varchar(1)
AS

if(@tipo = 'A')
begin
    select b.nombre TipoCorrespondencia, c.Nombre Estado, YEAR(Convert(date,
a.FechaRegistro,103)) Fecha, count(*) Cantidad from Correspondencias a
    inner join TipoCorrespondencias b on b.TipoCorrespondenciald =
a.TipoCorrespondenciald
    inner join Estados c on c.EstadosId = a.Estadold
    where convert(date, a.FechaRegistro,103) between convert(date, @fi, 103) and
convert(date, @ff,103)
    group by a.FechaRegistro, b.Nombre, c.Nombre
end

if(@tipo = 'M')
begin
    SET LANGUAGE SPANISH;
    select b.nombre TipoCorrespondencia, c.Nombre Estado, DATENAME(MONTH,
MONTH(Convert(date, a.FechaRegistro,103))) Fecha, count(*) Cantidad from Correspondencias
a
    inner join TipoCorrespondencias b on b.TipoCorrespondenciald =
a.TipoCorrespondenciald
    inner join Estados c on c.EstadosId = a.Estadold

```

```

        where convert(date, a.FechaRegistro,103) between convert(date, @fi, 103) and
convert(date, @ff,103)
        group by a.FechaRegistro, b.Nombre, c.Nombre
end

if(@tipo = 'F')
begin
    select b.nombre TipoCorrespondencia, c.Nombre Estado,Convert(date,
a.FechaRegistro,103) Fecha, count(*) Cantidad from Correspondencias a
        inner join TipoCorrespondencias b on b.TipoCorrespondenciald =
a.TipoCorrespondenciald
            inner join Estados c on c.EstadosId = a.Estadold
        where convert(date, a.FechaRegistro,103) between convert(date, @fi, 103) and
convert(date, @ff,103)
        group by a.FechaRegistro, b.Nombre, c.Nombre
end
GO
/***** Object: StoredProcedure [dbo].[sp_ReporteEstadistico_fecha_recibido] Script Date:
4/6/2021 16:07:21 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[sp_ReporteEstadistico_fecha_recibido]
AS

```

```

        select
            CONCAT(LEFT((UPPER(c.Nombre)),3), '-', CONVERT(nvarchar,a.id)) idunico
        , b.nombre TipoCorrespondencia
        , a.Referencia
        , c.Nombre Estado
        , a.Observacion
        , d.Correo destinatariocorreo
        ,Convert(date, a.FechaRegistro,103) FechaRegistro
        ,Convert(date, a.FechaHoraRecibe,103) FechaRecibeFisico
        ,DATEDIFF(DAY, a.FechaRegistro, ISNULL(a.FechaHoraRecibe,GETDATE())) AS
DiasRecibeFisico
        from Correspondencias a

```

```
        inner join TipoCorrespondencias b on b.TipoCorrespondenciald =
a.TipoCorrespondenciald
        inner join Estados c on c.EstadosId = a.Estadold
        inner join Personas d on d.Personald = a.DestinatarioPersonald
```

GO

```
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPane1', @value=N'[0E232FF0-B466-
11cf-A24F-00AA00A3EFFF, 1.00]
```

```
Begin DesignProperties =
```

```
  Begin PaneConfigurations =
```

```
    Begin PaneConfiguration = 0
```

```
      NumPanes = 4
```

```
      Configuration = "(H (1[40] 4[20] 2[20] 3) )"
```

```
    End
```

```
    Begin PaneConfiguration = 1
```

```
      NumPanes = 3
```

```
      Configuration = "(H (1 [50] 4 [25] 3))"
```

```
    End
```

```
    Begin PaneConfiguration = 2
```

```
      NumPanes = 3
```

```
      Configuration = "(H (1 [50] 2 [25] 3))"
```

```
    End
```

```
    Begin PaneConfiguration = 3
```

```
      NumPanes = 3
```

```
      Configuration = "(H (4 [30] 2 [40] 3))"
```

```
    End
```

```
    Begin PaneConfiguration = 4
```

```
      NumPanes = 2
```

```
      Configuration = "(H (1 [56] 3))"
```

```
    End
```

```
    Begin PaneConfiguration = 5
```

```
      NumPanes = 2
```

```
      Configuration = "(H (2 [66] 3))"
```

```
    End
```

```
    Begin PaneConfiguration = 6
```

```
      NumPanes = 2
```

```
      Configuration = "(H (4 [50] 3))"
```

```
    End
```

```
    Begin PaneConfiguration = 7
```

```
      NumPanes = 1
```

```

    Configuration = "(V (3))"
End
Begin PaneConfiguration = 8
    NumPanes = 3
    Configuration = "(H (1[56] 4[18] 2) )"
End
Begin PaneConfiguration = 9
    NumPanes = 2
    Configuration = "(H (1 [75] 4))"
End
Begin PaneConfiguration = 10
    NumPanes = 2
    Configuration = "(H (1[66] 2) )"
End
Begin PaneConfiguration = 11
    NumPanes = 2
    Configuration = "(H (4 [60] 2))"
End
Begin PaneConfiguration = 12
    NumPanes = 1
    Configuration = "(H (1) )"
End
Begin PaneConfiguration = 13
    NumPanes = 1
    Configuration = "(V (4))"
End
Begin PaneConfiguration = 14
    NumPanes = 1
    Configuration = "(V (2))"
End
ActivePaneConfig = 0
End
Begin DiagramPane =
Begin Origin =
    Top = 0
    Left = 0
End
Begin Tables =
    Begin Table = "cm"
    Begin Extent =

```

Top = 6
Left = 38
Bottom = 136
Right = 297
End
DisplayFlags = 280
TopColumn = 0
End
Begin Table = "c"
Begin Extent =
Top = 6
Left = 335
Bottom = 136
Right = 561
End
DisplayFlags = 280
TopColumn = 0
End
Begin Table = "p"
Begin Extent =
Top = 6
Left = 599
Bottom = 136
Right = 769
End
DisplayFlags = 280
TopColumn = 0
End
Begin Table = "e"
Begin Extent =
Top = 138
Left = 38
Bottom = 234
Right = 208
End
DisplayFlags = 280
TopColumn = 0
End
Begin Table = "us"
Begin Extent =

```

    Top = 138
    Left = 246
    Bottom = 268
    Right = 436
End
DisplayFlags = 280
TopColumn = 0
End
End
End
Begin SQLPane =
End
Begin DataPane =
    Begin ParameterDefaults = ""
    End
End
Begin CriteriaPane =
    Begin ColumnWidths = 11
        Column = 1440
        Alias = 900
        Table = 1170
        Output = 720
        Append = 1400
        NewValue = 1170
        SortType = 1350
        SortOrder = 1410
        GroupBy = 1350
        Filter = 1350
        Or = 1350
        Or = 1350
        Or = 1350
    End
End
End
', @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'ViewCorrespondenciaModificaciones'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPaneCount', @value=1 ,
@level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'ViewCorrespondenciaModificaciones'

```

```
GO
USE [master]
GO
ALTER DATABASE [DivinoNinoBD] SET READ_WRITE
GO
```