



UNIVERSIDAD NACIONAL DEL LITORAL
FACULTAD DE INGENIERÍA Y CIENCIAS HÍDRICAS



Desarrollo de módulo de guardia
ambulatoria con triaje automatizado
mediante inteligencia artificial en el sistema
web SaDER.

PROYECTO FINAL DE CARRERA
INGENIERÍA INFORMÁTICA

ESTUDIANTE: ANDINO DANIEL AGUSTÍN
DIRECTOR: CARIGNAN MARÍA AGUSTINA

SANTA FE, CAPITAL.

RESUMEN

La situación económica en Argentina se mantiene en caída desde la pandemia de COVID-19, ocurrida en el año 2020. Esto se ve reflejado en muchos sectores, como el de salud. A partir de la modificación de la Ley Nº 26682 “Marco Regulatorio de Medicina Prepaga”, las cuotas de obras sociales y servicios de medicina prepagas privados aumentaron en más de un 30 % a comienzos de 2024, por lo que resultan costosas para gran sector de la población del país (ObservatorioCiudad2024, 2024). Como consecuencia, la salud pública ha experimentado un aumento en la cantidad de pacientes que reciben atención año a año. Este aumento conlleva que los centros médicos se encuentren cercanos al límite o por encima de sus capacidades de funcionamiento, teniendo salas de esperas sobrecargadas de pacientes, profesionales fatigados por exceso de trabajo y uso desmedido de recursos e insumos públicos (Pagina12, 2024).

La salud pública en la Argentina carece de la infraestructura y de la cantidad de recursos necesarios para atender a la demanda del público (UnoEntreRios, 2024). Esta falencia es notable, principalmente, en el sector de las herramientas informáticas que utilizan los profesionales para realizar su trabajo. Propuesta en el año 2022 y aprobada en 2023, la Ley 27.706 (Congreso de la Nación Argentina, 2022) dicta que las historias clínicas pasan a ser digitales, de modo que las atenciones deben mantener un registro informático. Como consecuencia de esta ley, cada centro médico debe autogestionarse, es decir, se debe hacer responsable de obtener el software necesario para llevar a cabo sus tareas.

Dado este contexto, se plantea como propuesta de proyecto final de la carrera Ingeniería en Informática el desarrollo de un módulo de guardia ambulatoria, el cual incluye la clasificación automática de niveles de urgencia en los pacientes. Este módulo

es específico para el sistema web SaDER, sistema que propone el Ministerio de Salud de la Provincia de Entre Ríos de Argentina.

Para el desarrollo del proyecto se propone una metodología incremental con diez fases, cuyos entregables aportan un incremento en cuanto a las nuevas funcionalidades del sistema. Este desarrollo se lleva a cabo en las instalaciones del ente impulsor.

El periodo de desarrollo del proyecto es de cuatro meses y una semana, comenzando en agosto de 2024 y finalizando a principios de diciembre de 2024, según el calendario establecido. Se destinan cuatro horas diarias de trabajo. También se cuenta con un responsable de realizar las tareas de desarrollo, siendo el alumno que propone el proyecto final, así como tres profesionales: dos asesores y un director del proyecto.

En cuanto a los impactos del proyecto, se destacan dos aspectos clave: Por un lado, se automatizan tareas como el triaje, lo que contribuye a reducir posibles riesgos de errores humanos y disminuir la carga laboral del personal médico. Por otro lado, se mejorar la asistencia al médico durante la consulta, permitiéndole acceder de manera rápida y efectiva tanto a la información actual como a la histórica del paciente, todo dentro de un único sistema.

Adicionalmente, se destacan beneficios como la capacidad de monitoreo en tiempo real y la obtención de estadísticas actuales e históricas, tanto para el Ministerio como para cada efector individualmente.

Palabras clave: Aplicación Web - Triaje - Guardia Ambulatoria - Ley 27706 - Entre Ríos - Historia Clínica Electrónica.

Tabla de Contenido

1. Introducción	1
1.1. Contexto	1
1.2. Propuesta	5
1.3. Objetivos	7
1.3.1. Objetivo General	7
1.3.2. Objetivos Específicos	7
1.4. Alcance	8
2. Marco Teórico	11
2.1. Guardias Médicas	11
2.2. Triaje	14
2.3. Inteligencia Artificial en un sistema estructurado de triaje	17
2.3.1. Multinomial Logistic Regression o Regresión Logística Multinomial	18
2.3.2. Random Forest o Bosques Aleatorios	20
2.3.3. Support Vector Machine	22
2.3.4. Métricas de evaluación	24
3. Metodología	25
3.1. Especificación de requerimientos	25
3.1.1. Propósito	26

Tabla de Contenido

3.1.2. Ámbito del módulo	26
3.1.3. Definiciones, acrónimos, abreviaturas y referencias	27
3.1.4. Panorama general	28
3.1.5. Perspectiva del producto	28
3.1.6. Funciones del producto	30
3.1.7. Características del usuario	30
3.1.8. Limitaciones	31
3.1.9. Supuestos y dependencias	34
3.1.10. Requisitos específicos	35
3.1.11. Interfaces externas	48
3.1.12. Funciones	49
3.1.13. Requisitos de rendimiento	53
3.1.14. Restricciones de diseño	54
3.1.15. Atributos del sistema de software	54
3.2. Requerimientos de inteligencia artificial	56
3.2.1. Informes de sistemas de triaje	57
3.2.2. Desarrollo de ventajas y desventajas de modelos y comparación	59
3.2.3. Análisis y selección de modelos	63
3.3. Librerías de inteligencia artificial	65
4. Tecnologías seleccionadas	67
4.1. Tecnologías Frontend	67
4.2. Tecnologías Backend	68
4.3. Tecnologías de inteligencia artificial	69
4.4. Base de datos	70
4.5. Autorización y autenticación	72
4.6. Alternativas consideradas	73
5. Módulo de admisiones	75

Tabla de Contenido

5.1. Diseño de la base de datos	75
5.2. Desarrollo de la base de datos	78
5.3. Diseño del módulo de admisiones	81
5.3.1. Pantalla principal de la guardia	81
5.3.2. Nueva admisión de un paciente y edición de admisión	82
5.3.3. Registrar atención brindada por el profesional	84
5.4. Desarrollo del módulo de admisiones	84
5.4.1. Desarrollo de frontend	85
5.4.2. Desarrollo de backend	89
5.5. Pruebas del módulo de admisiones	92
5.5.1. Lista de funciones	92
5.5.2. Pruebas de las funciones	93
6. Módulo de triaje	104
6.1. Diseño de módulo de triaje	104
6.1.1. Diseño del formulario agregar datos y signos vitales	104
6.2. Desarrollo de modelos de inteligencia artificial	106
6.2.1. Dataset	106
6.2.2. Logistic Regression	110
6.2.3. Random Forest	112
6.2.4. Support Vector Machine	114
6.2.5. Comparación de los resultados y conclusiones	118
7. Módulo de estadísticas	122
7.1. Diseño del módulo estadístico	122
7.2. Desarrollo del módulo estadístico	128
7.2.1. Desarrollo de frontend	128
7.2.2. Desarrollo de backend	129
7.3. Pruebas del módulo estadístico	131

Tabla de Contenido

7.3.1. Lista de funciones	131
7.3.2. Pruebas de las funciones	132
8. Conclusión	135
9. Trabajos futuros	139
Referencias	143
Anexos	149

Índice de Ilustraciones

1.1.	Árbol de problema	2
2.1.	Diagrama de guardia médica	12
2.2.	Proceso Multinomial Logistic Regression	19
2.3.	Figura de un árbol de decisión y su dominio por regiones	20
2.4.	Predicción Random Forest	21
3.1.	Casos de Usos	30
5.1.	Diagrama Entidad Relacion	75
5.2.	Pantalla principal de la guardia	82
5.3.	Pantalla nueva/edición de admisión de paciente	83
5.4.	Pantalla de registrar atención de profesional	84
5.5.	Configuración de la barra de navegación	85
6.1.	Pantalla de agregar datos y signos vitales	105
6.2.	Distribución de clases en los datos	108
6.3.	Tasa de aciertos modelo Regresión Logística	111
6.4.	Tasa de aciertos para número de estimadores	113
6.5.	Tasa de aciertos para valores de random state	114
6.6.	Tasa de aciertos para los kernels	116
6.7.	Tasa de aciertos para valores de gamma	117
6.8.	Matrices de confusión	119
6.9.	Proceso de predicción automática de triaje	120

Índice de Ilustraciones

7.1.	Pantalla de estadísticas anuales	125
7.2.	Pantalla de estadísticas mensuales	126
7.3.	Pantalla de estadísticas semanales	126
7.4.	Pantalla de reporte de guardia	127
7.5.	Configuración de archivos de frontend sala de estadísticas	128

Índice de Tablas

2.1.	Niveles de priorización según SET	16
3.1.	Requerimiento funcional 1	36
3.2.	Requerimiento funcional 2	37
3.3.	Requerimiento funcional 3	38
3.4.	Requerimiento funcional 4	39
3.5.	Requerimiento funcional 5	40
3.6.	Requerimiento funcional 6	41
3.7.	Requerimiento funcional 7	42
3.8.	Requerimiento funcional 8	42
3.9.	Requerimiento funcional 9	43
3.10.	Requerimiento no funcional 1	44
3.11.	Requerimiento no funcional 2	45
3.12.	Requerimiento no funcional 3	45
3.13.	Requerimiento no funcional 4	46
3.14.	Requerimiento no funcional 5	47
3.15.	Requerimiento no funcional 6	47
3.16.	Comparación de modelos de inteligencia computacional	63
5.1.	Prueba P001	94
5.2.	Prueba P002	95
5.3.	Prueba P003	96

Índice de Tablas

5.4.	Prueba P004	97
5.5.	Prueba P005	98
5.6.	Prueba P006	99
5.7.	Prueba P007	100
5.8.	Prueba P008	101
5.9.	Prueba P009	102
5.10.	Prueba P010	103
6.1.	Distribución de triaje en el dataset	107
6.2.	Formato de un caso	109
6.3.	Número de iteraciones por modelo para converger	112
6.4.	Tasas de aciertos de los modelos predictivos	118
7.1.	Prueba P011	133
7.2.	Prueba P012	134
7.3.	Prueba P013	134

1 | Introducción

1.1. Contexto

El censo nacional argentino del año 2022 reveló que en la provincia de Entre Ríos un 38,2 % de la población tiene cobertura social proveniente de planes o programas de salud estatales o que no posee ningún tipo de cobertura (INDEC, 2022). Entre estos programas se destaca Sumar. El Programa Sumar es una política pública que promueve un acceso equitativo y de calidad a los servicios de salud para toda la población que no posee cobertura formal en salud (Ministerio de Salud de la Nación, 2023) Según datos del mes de junio de 2023, la cantidad de inscriptos al programa Sumar en la provincia de Entre Ríos es del 34,4 % de la población total (ProgramaSumar, 2024). A este número se le debe sumar aquellas personas que, aún teniendo cobertura, prefieren recibir atención en centros públicos.

Entre los distintos sectores de los centros clínicos, se destacan las guardias. Las guardias hospitalarias se consideran como la puerta de entrada de los pacientes a los efectores y el corazón de los hospitales, ya que son los lugares por donde más gente circula y donde se atienden los casos de urgencia. Por este motivo, la planificación y ejecución de una resolución efectiva e integral a los problemas en la atención resulta esencial, habida cuenta de la relevancia sanitaria (Busso, Bellavita, Heim, y Maidana, 2019). Generalmente, el proceso de atención comienza con la admisión, se registra al paciente y se le toman datos, del cual se deriva una clasificación de urgencia para

Contexto

poder ser atendido. El triaje es esta clasificación de los pacientes en diferentes niveles de urgencia o priorización de la asistencia (*MANUAL para la IMPLEMENTACIÓN de un SISTEMA de TRIAJE para los CUARTOS de URGENCIAS*, 2010).

La problemática de las guardias médicas, tanto a nivel país como de la provincia, tienen orígenes en sectores sociales, políticos y económicos, cuyas causas son numerosas. En Figura 1.1 se mencionan algunas de ellas. En primer lugar se destaca la falta de profesionales de la salud para cubrir los turnos de guardias, lo que lleva a repetir plazas, sobrecargando de trabajo a estos profesionales o dejando puestos vacíos. El alto nivel de demanda afecta a los profesionales, desgastando física y mentalmente y requiriendo de descansos, ocasionando congestión y demoras en las salas de espera, así como un deterioro en el desempeño laboral. También se han reportado que los centros públicos atienden con poco recursos, habiendo situaciones extremas donde no se poseen recursos básicos para realizar las atenciones. Estas causas repercuten en una menor calidad y cantidad de atención a los pacientes, en demoras para recibir la atención, entre otros efectos (Nöllmann, 2024) (Czubaj, 2023) (Vítolo, 2014).

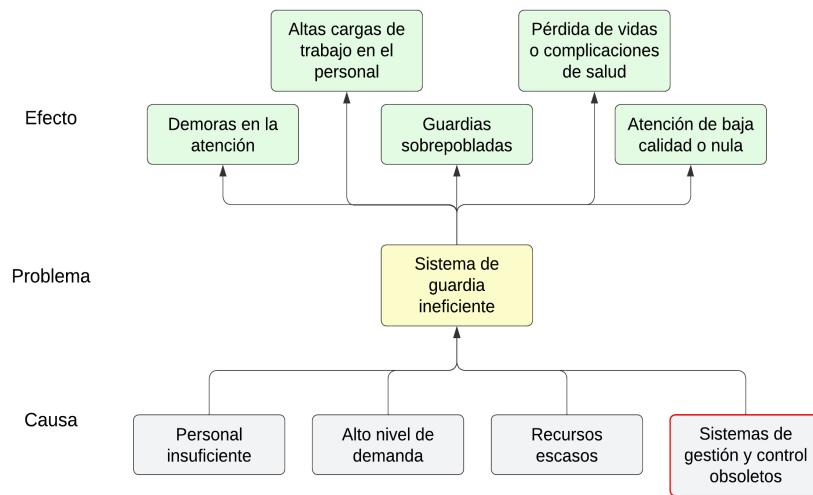


Figura 1.1: Árbol de problema

Con respecto a las guardias en la provincia de Entre Ríos, en el año 2020, durante el cual la pandemia de COVID-19 impactó en todo el mundo, se registraron un total

Contexto

de 1.922.288 consultas ambulatorias (Ministerio de Salud, 2022). De este número, un 27 % aproximadamente (518.808), corresponde a consultas realizadas en Emergencias/Urgencias. Ante la necesidad de poder atender esta cifra de demanda, el Sistema Informático de Registro y Gestión Sanitaria de la Provincia de Entre Ríos (Salud Digital Entre Ríos o SADER) brinda herramientas informáticas a los centros médicos de la provincia de manera gratuita. Impulsado también por la ley 27.706, la cual exige que los y las profesionales y auxiliares de la salud deben registrar toda intervención médico-sanitaria en la historia clínica electrónica, en otras palabras, toda consulta médica debe ser registrada en sistemas digitales.

El sistema SADER cuenta con cientos de efectores registrados a comienzos de 2024, muchos de los cuales poseen guardia. Actualmente estos centros médicos no pueden cumplir con sus tareas de guardia de forma digital de manera óptima, dado que SADER no posee un apartado enfocado a las guardias ambulatorias. Para suplir esta necesidad, algunos efectores han optado por utilizar el sistema de turnos de ambulatoria, también propio de SADER, para realizar la atención en guardias. Este módulo no está especializado para las necesidades que requiere un software de urgencia, haciendo ineficiente el trabajo del personal. Una de las principales consecuencias es la pérdida de tiempo: en las guardias se prioriza la rápida atención y el módulo de ambulatoria requiere de pasos obligatorios para registrar la consulta del paciente, de modo que parte del tiempo de atención del paciente se utiliza en brindar los datos al sistema. Otros centros médicos revelaron que utilizan herramientas tales como planillas digitales o impresas, donde es común cometer errores o perder información. Lo que demuestra una situación donde no existe una unificación en los sistemas de guardias para utilizar una única herramienta, sumado a que las que se utilizan son ineficientes y obsoletas.

Siendo entonces el principal problema que el Ministerio de Salud, como organización responsable de cubrir las necesidades de los efectores, no brinda un sistema

digital para las guardias. Sistema que debe estar adaptado exclusivamente, supliendo las necesidades de las guardias y mitigando al máximo los riesgos posibles por parte de los profesionales como consecuencias de los problemas mencionados previamente.

Como alternativas de solución existen sistemas informáticos de guardia en el mercado, algunos de los cuales poseen triaje. Se destaca el Historia de Salud Integrada (HSI), que es una herramienta gratuita promocionada por el Ministerio de Salud a nivel nacional (HSI, 2025). HSI, al igual que SADER, es un sistema web, que brinda distintos módulos para administrar centros médicos, entre ellos el de guardia que posee triaje. Considerando esta posibilidad de utilizar solo el módulo de guardia del HSI los usuarios se encuentran con una disyuntiva: utilizar un único módulo de un sistema nuevo para una tarea específica y emplear SADER para las demás, o utilizar el sistema HSI por completo. La primer opción conlleva todas las desventajas de no unificar un sistema para una misma organización, como lo son la incompatibilidad aislando la información en cada sistema; la segunda opción implica un mayor aprendizaje y capacitación por parte del personal, lo que conlleva pérdida de tiempo y esfuerzo, y aumenta la probabilidad de confundir al usuario y que cometa errores. Por otra parte, la clasificación de triaje en HSI debe hacerse siempre de forma manual. Al no tener automatizada esta tarea, dentro de este sistema pueden darse riesgos ocasionados por las causas de la Figura 1.1. Un ejemplo de esto es una incorrecta clasificación en el nivel de urgencia, que desencadene en un tiempo de espera mayor por parte del paciente, dependiendo del nivel de prioridad de atención, y que se agrave la situación del mismo. Por lo que, se descarta al HSI como una alternativa viable para el problema planteado, tratándose los motivos de que carece de interoperabilidad con otros programas de salud y, fundamentalmente, en la falta de automatización del triaje. Proceso que, cabe recalcar, se debe realizar de manera correcta independientemente del ambiente y situación en que se realice.

Otras alternativas son los sistemas adquiridos bajo licencias comerciales, las cuales

-salvo excepciones- sólo permiten al comprador ejecutar los programas, vedando el acceso al conocimiento de las instrucciones de los programas y, por ende, la libre adaptación de los productos a las necesidades del ente (Castello, Bollo, Gauna, Montes, y Vargas, 2010). Como una alternativa para solucionar el problema se puede considerar la adquisición de un software privado, corrigiendo así la falta de herramientas digitales en las guardias. Sin embargo, dado que muchos centros clínicos tienen su forma particular de organización y de funcionamiento, el poder adaptar los softwares privados a las medidas específicas de cada caso implicaría una mayor cantidad de recursos económicos, tecnológicos y humanos. Esfuerzo el cual los efectores no pueden enfrentar dados los recursos actuales que poseen. Entre estas dificultades se pueden mencionar la falta de acceso al código fuente, el pago extra para que la empresa que brinda el sistema pueda realizar las modificaciones o la negativa a realizar los cambios para mantener un software estándar. Ejemplos de estos software son Colmena (ITColmena, 2009), utilizado en algunos efectores de la provincia de Entre Ríos, o Geclisa (Geclisa, 2025).

1.2. Propuesta

El propósito de este proyecto es el de brindar un apartado de guardia especial para el sistema SADER, y que cuente con triaje automatizado por Inteligencia Artificial (IA). Este módulo cumplirá con las funcionalidades requeridas y está focalizado en mejorar la productividad y la eficiencia de los profesionales al disminuir el riesgo humano. Posibilitando así la mejora, adaptación, innovación y mantenimiento para próximas versiones del sistema. El principal motivo de realizar el proyecto sobre el sistema SADER recae en la dependencia laboral del responsable del presente proyecto con el organismo impulsor del sistema, en este caso, el Ministerio de Salud de la Provincia de Entre Ríos.

Propuesta

Entre los beneficios e impactos del proyecto propuesto se destaca la informatización del registro de las consultas en las guardias, respetando lo impuesto en la Ley 27.706. También se mejora la seguridad y confidencialidad con la que es tratada la información de los pacientes.

La implementación de la inteligencia artificial conlleva beneficios al disminuir los riesgos de errores humanos. Se automatizan tareas o procesos: las IAs pueden ejecutar tareas repetitivas y optimizar procesos sin la necesidad de la intervención humana. A su vez, la clasificación de los pacientes se ha de hacer en un tiempo corto, de forma ágil y efectiva (Jiménez, 2003). En otras palabras: el triaje se debe realizar de manera correcta independientemente del ambiente y situación en que se realice, así como del estado y capacidades de los profesionales de guardia. Al implementar inteligencias artificiales en el sistema se asiste al personal prediciendo los niveles de urgencias cumpliendo con esta condición, dado que las inteligencias no sufren de los riesgos de error humanos mencionados (NexusIntegra, 2023) (Neuroons, 2020). Las IAs también disminuyen el trabajo de los profesionales al automatizar la tarea de triaje e informatizar la comunicación de información, de forma que se privilegia el tiempo y los recursos.

Otra ventaja de los sistemas digitales recae en la asistencia al médico durante la consulta, al obtener datos actuales del paciente, así como históricos, de manera rápida y efectiva dentro de un mismo sistema, independientemente del centro médico en que el paciente reciba la atención.

El proyecto también impacta en algunos sectores específicos.

- **Ecológicamente:** al disminuir el uso de papel en los centros médicos y de insumos de administración, tales como impresoras y tintas, lapiceras, entre otros, posibilitando la migración de las atenciones basadas en fichas de papel a un sistema digital.
- **Económicamente:** evitando costos de licencias privadas en los centros médicos al utilizar un sistema de gestión gratuito, así como aquellos gastos que se

economizan al reemplazar historias clínicas impresas por un sistema digital.

- **Socialmente:** al mejorar la organización y eficiencia de las consultas a partir del orden de atención; se reduce el tiempo de espera del paciente desde su ingreso al centro de salud hasta su atención por parte del médico; se prioriza el tratamiento de los casos más urgentes, disminuyendo la pérdida de vidas, así como los posibles riesgos de salud por no recibir la atención a tiempo. También se mejora la comunicación entre el personal de guardia a partir de obtener información precisa en cada momento.

Por otra parte, impacta en el Ministerio de Salud mismo y cada efecto usuario de SADER, ya que los datos de las guardias podrán ser visualizados a través de estadísticas, los cuales pueden ser utilizados en futuros informes, propuestas, tomas de decisiones, monitoreo en tiempo real por parte de los directivos, entre otros propósitos.

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar un módulo de guardias ambulatorias con triaje automatizado mediante inteligencia artificial dentro sistema web Sader.

1.3.2. Objetivos Específicos

- Desarrollar un módulo de admisión, donde se inicie el proceso de atención, ingresando los datos, síntomas y signos del paciente.
- Desarrollar un módulo de triaje, que clasifique el nivel de urgencia de la atención del paciente a partir de una inteligencia artificial.
- Desarrollar un módulo de visualización estadística de datos de la aplicación de guardia ambulatoria.

1.4. Alcance

A continuación se describen algunas consideraciones para definir el alcance del proyecto.

El sistema soporta la inicialización del proceso de atención de las siguientes situaciones en las que el paciente ingrese al centro médico:

- Ingrasa consciente y es capaz de brindar la totalidad de la información que requiera el usuario clínico.
- Ingrasa inconsciente, pero un acompañante puede brindar información.

El proyecto no soporta otras posibles situaciones, las cuales seon consideradas para futuros desarrollos en próximas versiones del sistema en el cual será implementado. La inicialización de una nueva atención se lleva a cabo cuando el usuario clínico ingrese los datos del paciente.

En cuanto al usuario clínico, es éste quien indica al sistema si requiere de la asistencia del mismo para el triaje. Si no requiere asistencia debe ingresar la clasificación manualmente. Siendo el caso afirmativo, el sistema realiza la clasificación y la opción debe ser aprobada por el usuario encargado o por el médico responsable. Si la opción que brindó el sistema fue rechazada, alguno de los agentes nombrados deberá ingresar la clasificación pretendida.

El sistema actual Sader posee un apartado especial para análisis estadísticos de distinta índole. Se destacan los siguientes tiempos que brindan información relevante de la guardia. De estos tiempos, se opta por mantener el registro histórico de al menos uno durante el desarrollo de dicho módulo, considerando el nivel de criticidad e impacto de dicho tiempo:

1. Ingreso de los datos del paciente y la respuesta de triaje. Este tiempo sirve para que cada efector pueda controlar y analizar su capacidad y calidad operativa.
2. Inicio de pedido de clasificación y la respuesta de triaje. Este tiempo es de utilidad

para evaluar el rendimiento del clasificador.

3. Ingreso de los datos del paciente e inicio de la consulta por parte del médico (excepcionando los casos de urgencia).

También guarda el nivel de la clasificación brindada por el sistema, en cada nuevo ingreso de paciente, y si fue aprobada o no.

En cuanto a la clasificación de los niveles de urgencia se utiliza un modelo de inteligencia artificial, automatizando la tarea de decisión de triaje.

En cada efector se visualiza la información en tiempo real, es decir, los casos que han ocurrido en las últimas 24 horas dentro del apartado de admisiones. Los usuarios que tengan permisos de encargado de un efector tendrán acceso a la información histórica de la guardia.

Con estos objetivos, se pretende que el proyecto cumpla con los siguientes requisitos no funcionales:

- Rapidez de respuesta: en situaciones donde el tiempo es crucial al atender al paciente, no se acepta que se pierda tiempo esperando respuestas largas por parte del sistema.
- Estable en las clasificaciones de triaje: ante situaciones donde los datos ingresados son similares las clasificaciones no haya una diferencia mayor a uno entre ambos niveles.
- Software mantenable: el código es mantenable, posibilitando una rápida y fácil implementación de nuevas funcionalidades.
- Adaptable a las modificaciones: el sistema es lo suficientemente robusto de forma que si se agregan o cambian funcionalidades en un módulo, el resto de los módulos deben tener ningún impacto o el mínimo posible.
- Interfaz intuitiva: la interfaz de usuario está diseñada de modo que los usu-

rios logren utilizar sus funciones de manera intuitiva, evitando pasos extras que dificulten o demoren el uso del sistema.

- Seguridad de datos: la información del paciente será tratada con seguridad, de modo que solo el personal profesional pueda acceder a ésta.

Así mismo, para mantener un estándar de calidad, se realiza un análisis dinámico de código. Las pruebas son de tipo caja negra para cada funcionalidad desarrollada, ya que se pretende un desarrollo rápido que cumpla con los estándares de calidad y seguridad. De este modo, si las funcionalidades superan dichas pruebas son marcadas como realizadas. Posteriormente se diseñan y ejecutan pruebas específicas para cada módulo del proyecto.

En cuanto a la portabilidad, el uso del sistema está orientado a la aplicación en computadoras de escritorios, aunque también será posible utilizarlo en celulares y tablets. La accesibilidad de los usuarios es restrictiva, siendo únicamente el personal autorizado quien acceda a los datos y a las funcionalidades del sistema.

Por último se destaca el uso de un servidor durante el transcurso del proyecto. Durante el desarrollo de cada etapa, el servidor es utilizado para consultar la base de datos del sistema SADER. Luego de finalizar cada etapa se desplegará el código elaborado en el servidor, con el objetivo de realizar las pruebas pertinentes.

Criterios de aceptación:

- La tasa de aciertos del modelo de inteligencia artificial debe ser como mínimo del 75 % (setenta y cinco por ciento) para ser considerado como apto para su implementación dentro del sistema.
- El módulo de registro debe guardar, al menos, uno de los tiempos de relevancia establecidos.
- El sistema permitirá el ingreso de un paciente nuevo al completar el mínimo obligatorio de datos, los cuales son indicados en la sección de requerimientos.

2 | Marco Teórico

2.1. Guardias Médicas

Guillén (2018) resalta que la misión de la Medicina de urgencias es atender y resolver mediante el diagnóstico precoz y el tratamiento cualquier situación que ponga en riesgo la vida del paciente o suponga una amenaza vital para el mismo o para algunos de sus órganos.

La estructura de una guardia médica que atiende todo tipo de urgencias, incluyendo adultos, niños y obstetricia, es un componente crítico del sistema de salud. Este tipo de servicio debe estar diseñado para proporcionar atención rápida y efectiva a pacientes con diversas necesidades médicas, garantizando que se cumplan los estándares de calidad y seguridad. A continuación, se describe en detalle la organización y funcionamiento de una guardia médica integral.

Guillén también afirma que un medio de atención médica en Urgencias debe comprender:

- **Triaje:** es una clasificación de los pacientes con el fin de priorizar la asistencia sanitaria y asignar el recurso asistencial adecuado a los pacientes que acuden a los Sistemas de Urgencias Hospitalarios (SUH).
- **Atención:**
 - Atención a la urgencia vital (emergencia, politraumatizado, reanimación car-

- diopulmonar [RCP]).
- Atención a la patología urgente clínicamente objetivable.
 - Atención a la patología menos urgente y no urgente con respuesta adaptada a la solicitud.
 - Observación clínica: el funcionamiento de la observación es equiparable al de una unidad de hospitalización, donde se podrán realizar el tratamiento y la valoración de la evolución. Desde esta unidad se procederá al alta o al ingreso hospitalario.

El proceso que se realiza en las guardias médicas varía en diversos aspectos. No obstante, el funcionamiento, en rasgos generales y de manera simplificada, cumple con el siguiente diagrama:

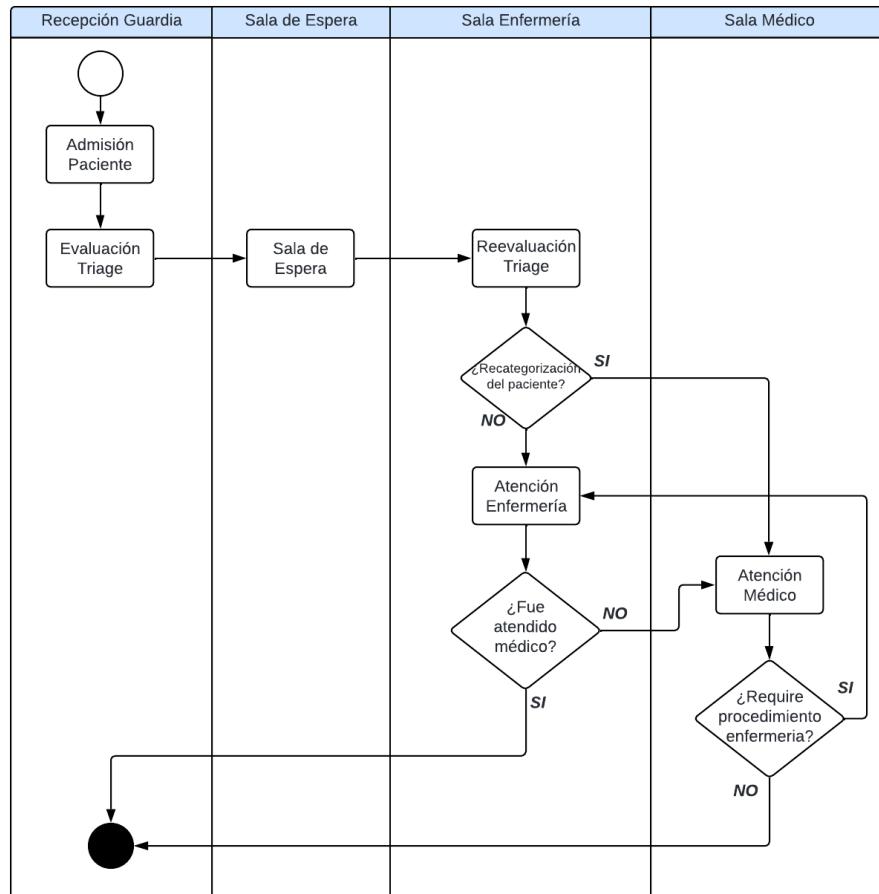


Figura 2.1: Diagrama de guardia médica

En el diagrama se observa el proceso establecido en muchos centros clínicos y hospitales. Se describen las tareas llevadas a cabo, así como las delimitaciones de las estructuras, llamadas salas físicas, y las responsabilidades del personal.

El proceso general comienza con la admisión del paciente, al cual se le solicitan los datos necesarios para registrarla dentro de la guardia. Estos datos pueden variar dependiendo del establecimiento, aunque, en muchos casos, es información personal, así como la razón de concurrir a la guardia. Esta tarea es realizada por el personal de recepción, quién puede ser un enfermero o un recepcionista sin conocimientos médicos. Posteriormente, el paciente recibe una primera evaluación de triaje, donde se determina la gravedad de su situación mientras se encuentra en la sala de espera.

El siguiente paso es una reevaluación de triaje por parte de un enfermero calificado. Esta reevaluación se realiza en base a la información recabada por el personal de recepción y a la observación del paciente. En base a esta reevaluación, se asigna un nivel de prioridad al paciente, el cual determina el tiempo máximo de espera para ser atendido, o, en caso de ser necesario, la atención inmediata del personal médico. En caso de no tratarse de una urgencia, el enfermero se encarga de tratar al paciente dependiendo del motivo de arrivamiento. Esta tarea puede requerir de la medición de datos antropométricos, como medición de la temperatura corporal, por mencionar uno.

Luego de ser atendido por el enfermero, el paciente es derivado a un médico, quién se encarga de realizar una evaluación más detallada y atender al paciente en base a sus necesidades.

Una vez que el paciente es atendido por el médico, se procede a realizar el diagnóstico y tratamiento correspondiente. En caso de ser necesario, el paciente es derivado a una sala de observación, donde el enfermero realizará un seguimiento de su evolución notificando al médico al respecto. Finalmente, el paciente es dado de alta o, en caso de ser necesario, es derivado a un hospital para su internación.

Cabe destacar que, para aquellos casos de urgencias, el proceso es más rápido, ya que el paciente es atendido de manera inmediata por el médico.

2.2. Triaje

Gómez Jiménez (2003) define el triaje como el proceso de valoración clínica preliminar, que ordena a los pacientes antes de la valoración diagnóstica y terapéutica completa, de forma que en una situación de saturación del servicio o de disminución de recursos, los pacientes más urgentes son tratados primeros.

El triaje es esencial para valorar la calidad de una guardia de urgencias y debe ser la puerta de entrada a una asistencia eficaz y eficiente. Debe ser estructurado y evaluar de forma rápida la gravedad de cada paciente, de una manera reglada, válida y reproducible. No solamente tiene que priorizarse la atención médica sino asignar el lugar y los medios apropiados para la atención.

Las funciones del triaje deben ser las siguientes:

- Identificar a los pacientes con riesgo vital.
- Priorizar en función del nivel de clasificación.
- Asegurar la reevaluación de los pacientes en espera.
- Asignar el área más adecuada a cada paciente.
- Aportar información acerca del proceso asistencial.
- Ayudar a corregir las situaciones de saturación del Sistema de Urgencias (SU) y disminuir la congestión de las áreas de tratamiento.

Para ello el sistema de triaje estructurado es el compuesto de una escala de triaje fiable, útil y válida, junto con una estructura física y una estructuración profesional

Triaje

y tecnológica en los servicios, dispositivos y centros donde se atienden las urgencias. Lo cual permita realizar el triaje a los pacientes en base a un modelo de calidad, que sea evaluable y mejorable.

Los actuales sistemas de triaje estructurado que más se utilizan están basados en escalas de 5 niveles. Actualmente existen 5 modelos de triaje estructurado, de los cuales este proyecto se basará en el sistema español de triaje (SET). Habitualmente, es el personal de enfermería el responsable de realizar el triaje, con o sin la colaboración del médico, aunque siempre debe realizarlo personal entrenado.

El *Manual para la implementación de un sistema de triaje para los cuartos de urgencias* (2010) explica que el SET clasifica los niveles de urgencia en las siguientes categorías:

- Nivel I (Resucitación): Situaciones que requieren resucitación, con riesgo vital inmediato.
- Nivel II (Emergencia): Situaciones de emergencia o muy urgentes, de riesgo vital previsible, la resolución de las cuales depende radicalmente del tiempo.
- Nivel III (Urgencia): Situaciones de urgencia, de potencial riesgo vital.
- Nivel IV (Menos urgente): Situaciones de menor urgencia, potencialmente complejas, pero sin riesgo vital potencial.
- Nivel V (No urgente): Situaciones no urgentes, que permiten una demora en la atención o pueden ser programadas, sin riesgo para el paciente.

Nivel de gravedad	Nivel de urgencia	Categoría	Tiempo de atención
I	Reanimación	Rojo	Inmediato
II	Emergencia	Naranja	Inmediato enfermería / 7 minutos médico
III	Urgente	Amarillo	45 minutos
IV	Menos urgente	Verde	120 minutos
V	No urgente	Azul	240 minutos

Tabla 2.1: Niveles de priorización según SET

Un sistema de triaje estructurado moderno debe poseer un enfoque integral de calidad para los servicios de urgencias, cuya implementación genera efectos positivos directos tanto para los pacientes como para los profesionales de la salud y el sistema sanitario en general.

Entre los principios fundamentales de un sistema de triaje estructurado, se pueden destacar algunos:

- debe ser un sistema de triaje de 5 niveles de priorización, normalizado y dotado de un programa informático de gestión del triaje, que permita el registro de la clasificación, el control de todos los pacientes dentro y fuera del servicio y el control de tiempos de actuación (“Timelines”).
- ha de ser un modelo dotado de un programa informático de ayuda a la decisión clínica en el triaje, evaluado y validado, con ayuda continua y con registro anamnésico del triaje.
- debe poder integrarse en un modelo global de historia clínica electrónica, integradora de la actividad médica y de enfermería, estandarizada y acorde con estándares de calidad, que permita un total control de la gestión clínica y administrativa del cuarto de urgencias.

2.3. Inteligencia Artificial en un sistema estructurado de triaje

IBM define la inteligencia artificial como “un conjunto de tecnologías que permiten a las máquinas simular la inteligencia humana, incluyendo el aprendizaje, el razonamiento y la auto-corrección” (IBM, 2020).

En la actualidad, la IA tiene un impacto significativo en diversas industrias, revolucionando la forma en que se procesan y analizan los datos. En el ámbito de la salud, su aplicación ha mejorado la precisión diagnóstica, optimizado procesos y transformado la atención al paciente.

Entre las diversas ramas que componen la inteligencia artificial, el aprendizaje estadístico se destaca al enfocarse en métodos que utilizan datos para inferir patrones y relaciones.

En *Una introducción al aprendizaje estadístico con aplicaciones en R* (2013) el aprendizaje estadístico se define como un conjunto de herramientas para modelizar y comprender conjuntos de datos complejos. A diferencia de los métodos estadísticos tradicionales, que a menudo se centran en la inferencia y la explicación de relaciones entre variables, el aprendizaje estadístico pone un mayor énfasis en la predicción y la generalización a partir de conjuntos de datos.

El aprendizaje estadístico abarca una variedad de técnicas, incluyendo:

- Regresión: Utilizada para modelar la relación entre una variable dependiente y una o más variables independientes.
- Árboles de Decisión: Herramientas que dividen los datos en subgrupos basados en características específicas, facilitando interpretaciones claras.
- Máquinas de Soporte Vectorial (SVM): Métodos que buscan encontrar el hiper-

plano óptimo que separa diferentes clases en un conjunto de datos.

- Redes Neuronales: Modelos inspirados en el funcionamiento del cerebro humano, capaces de aprender patrones complejos a partir de grandes volúmenes de datos.

El aprendizaje estadístico no solo busca hacer predicciones precisas, sino también entender las relaciones subyacentes entre variables. Esto es especialmente relevante en contextos donde la interpretación de los resultados es crucial, como la relación entre diagnósticos y signos vitales de un paciente.

2.3.1. Multinomial Logistic Regression o Regresión Logística Multinomial

El modelo de regresión logística es similar al clasificador de regresión lineal, pero éste se diferencia al definir la función de probabilidades logística. Esta ecuación es llamada Softmax Regression, la cual está definida de la siguiente manera:

$$\rho(X) = P_n(Y = 1/X) = \frac{e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n}}{1 + e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n}} \quad (2.1)$$

, donde $P_n(Y = 1/X)$ es la probabilidad de que la salida Y sea igual a 1 dado el vector de características o entradas X_n y β_n son los coeficientes de predicción.

La ecuación anterior es utilizada para calcular la probabilidad, con resultado entre 0 y 1, de que se pertenezca a dicha clase en base a las entradas X. El resultado de la misma es la respuesta para una única clase, por lo que, para predecir más de una clase se debe utilizar la fórmula de regresión logística múltiple.

Suponiendo una cantidad K de clases, la ecuación reescriba es:

$$P_n(Y = j/X) = \frac{e^{\beta_{j0} + \beta_{j1} X_1 + \beta_{j2} X_2 + \dots + \beta_{jn} X_n}}{\sum_{k=1}^K e^{\beta_{k0} + \beta_{k1} X_1 + \beta_{k2} X_2 + \dots + \beta_{kn} X_n}} \quad (2.2)$$

donde, $P_n(Y = 1/X)$ es la probabilidad de que la salida de Y corresponda a la

clase j, dado el vector de entradas o características X_n y β_{jn} son los coeficientes para la clase j, y β_{kn} los coeficientes para cada una de las K clases.

Para el método de regresión logística multinomial, ante cada entrada de X, se tendrán K probabilidades, es decir, una probabilidad por cada clase. Para predecir una única salida con este método, existen varias alternativas. En el presente informe se opta por utilizar el argumento máximo, el cual toma el mayor valor entre todas las opciones.

El funcionamiento del método es destacado en la Figura 2.2. El proceso comienza con las entradas X_n , a partir de las mismas se calculan las probabilidades $\rho(X_n)$, las cuales tienen valores entre 0 y 1 al ser calculadas con la función Softmax, por último se toma el argumento o probabilidad de clase máximo, siendo ésta la salida.

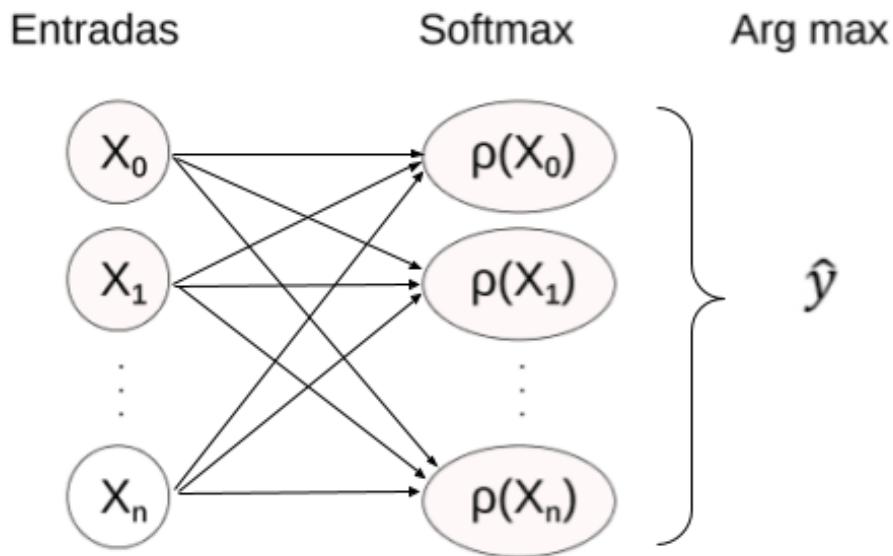


Figura 2.2: Proceso Multinomial Logistic Regression

2.3.2. Random Forest o Bosques Aleatorios

Los árboles de decisión son útiles para problemas de clasificación. Éstos se construyen a partir de la división o partición del espacio o dominio, generando diferentes regiones, donde cada punto de división tiene una condición o regla de decisión en base a la característica de entrada.

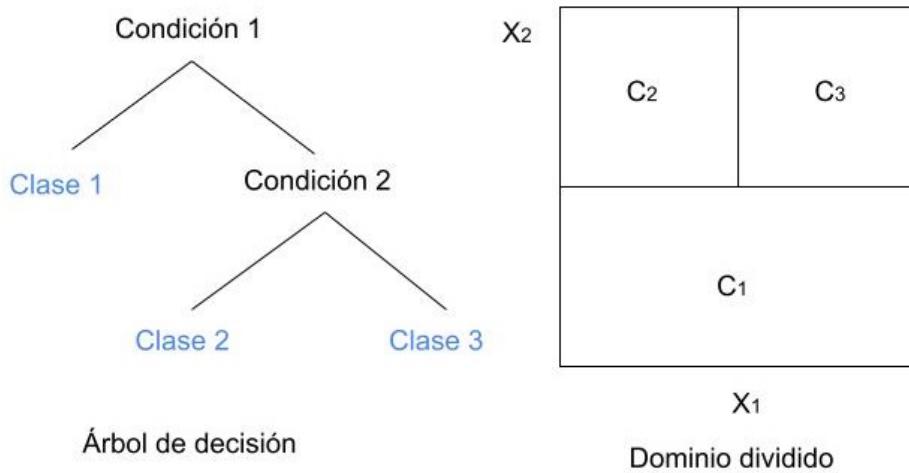


Figura 2.3: Figura de un árbol de decisión y su dominio por regiones

Para dividir los sectores, se utilizan distintos métodos lógicos. Entre los más utilizados actualmente, son destacable:

- Gini: mide la impureza de una división, es decir, que ésta será menor a medida que los sectores contengan cada vez más de una misma clase. El coeficiente de Gini está definido por la siguiente ecuación:

$$G = \sum_{k=1}^K \hat{\rho}_{mk} \cdot (1 - \hat{\rho}_{mk}) \quad (2.3)$$

, donde $\hat{\rho}_{mk}$ representa la proporción de observaciones de entrenamiento en la m-ésima región que pertenecen a la k-ésima clase.

- Entropía cruzada: mide el grado de mezcla o desorden de un sector, por lo que, una alta entropía implica un alto grado de desorden, contrariamente, un grado

pequeño indica poco desorden. La entropía cruzada está definida por la siguiente ecuación:

$$D = - \sum_{k=1}^K \hat{p}_{mk} \cdot \log(\hat{p}_{mk}) \quad (2.4)$$

, donde \hat{p}_{mk} es la probabilidad de ocurrencia del evento

Los árboles de decisión tienden al sobreajuste en problemas complejos o con ruido. Para corregir esta dificultad, se suelen utilizar los random forest.

Los random forest son un conjunto de árboles los cuales se construyen con el objetivo de ser más precisos y robustos en cuanto a las predicciones. Se mejora la precisión de los árboles de decisión individuales a través de la selección aleatoria de atributos. Los bosques aleatorios consideran, para cada división o nodo, un subconjunto de características para dividir los datos, en lugar de tomar todas las características como el método de Bagging. Esta condición genera una mayor diversidad en los árboles, reduciendo la correlación entre ellos y haciendo más fiable el método.

La predicción final en un Random Forest es el promedio de las predicciones individuales de cada árbol (regresión) o la votación mayoritaria (clasificación).

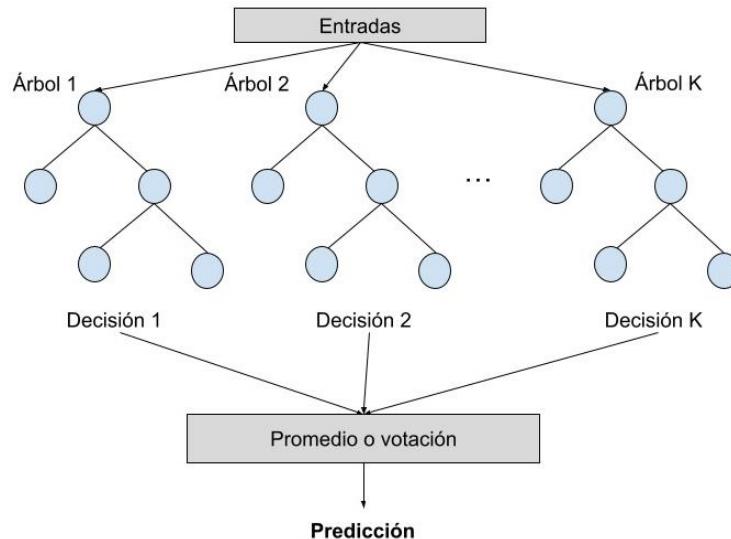


Figura 2.4: Predicción Random Forest

2.3.3. Support Vector Machine

El clasificador de soporte vectorial o support vector classifier es una técnica que utiliza el concepto de hiperplano de separación. El clasificador se basa en la clasificación dependiendo de qué lado de un hiperplano pertenezca los datos de entrada, para ello se debe encontrar el hiperplano que mejor logre separar los datos de las distintas clases.

La ecuación de un hiperplano de dimensión n es la siguiente:

$$\beta_{k0} + \beta_{k1}X_1 + \beta_{k2}X_2 + \dots + \beta_{kn}X_n = 0 \quad (2.5)$$

, donde, para valores distintos a cero de la ecuación, es decir menor o mayor a cero, las entradas X_n se sitúan a un lado u otro del hiperplano, clasificando así su clase.

Los hiperplanos tienen la particularidad de que existen infinitas posibilidades que separan las entradas. Para decidir qué hiperplano es el óptimo a utilizar, se utiliza el hiperplano de margen máximo, conocido en inglés como maximal margin hyperplane o optimal separating hyperplane.

El hiperplano de margen máximo es el hiperplano que tiene una distancia mínima más lejana a los datos de entrada. Para hallar el hiperplano, dado un conjunto de n observaciones de entrenamiento $x_1, \dots, x_n \in \mathbb{R}^p$, con las clases $y_1, \dots, y_n \in \{-1, 1\}$, la solución es la optimización del problema

$$\text{maximizar}_M(\beta_0, \beta_1, \dots, \beta_n) \quad (2.6)$$

sujeto a,

$$\sum_{j=1}^n \beta_j^2 = 1 \quad (2.7)$$

$$y_i(\beta_{k0} + \beta_{k1}X_1 + \beta_{k2}X_2 + \dots + \beta_{kn}X_n) \geq M(1 - \epsilon_i) \quad \forall i = 1, \dots, n \quad (2.8)$$

$$\epsilon_i \geq 0, \sum_{j=1}^n \epsilon_i \leq C \quad (2.9)$$

, donde C es un parámetro de ajuste no negativo, M es la distancia del margen, la cual se trata sea la mayor posible, ϵ_i son variables de holgura que permiten a las observaciones individuales estar en el lado equivocado del margen o del hiperplano.

El clasificador de soporte vectorial posee ventajas en casos de clasificación binaria o de dos clases. Para aquellas situaciones donde se tienen más de dos clases, es recurrente emplear máquinas de soporte vectorial o support vector machines.

Las máquinas de soporte vectorial (SVM) son una extensión del clasificador anteriormente desarrollado. Las SVM amplían el espacio de características de una manera particular utilizando núcleos o kernels. Las máquinas utilizan dicho kernel para transformar los datos a un espacio de mayor dimensión, al cual luego aplican un hiperplano lineal, logrando una separación no lineal en el espacio original.

Entre los distintos kernels, se destacan los siguientes:

- Lineal

$$K(x_i, x_{i'}) = \sum_{j=1}^n x_{ij} \cdot x_{i'j} \quad (2.10)$$

- Polinómico

$$K(x_i, x_{i'}) = \left(1 + \sum_{j=1}^n x_{ij} \cdot x_{i'j} \right)^d \quad (2.11)$$

, donde d es un entero positivo, el cual brinda el grado del polinomio.

- Radial

$$K(x_i, x_{i'}) = \exp \left(-\gamma \sum_{j=1}^n (x_{ij} - x_{i'j})^2 \right) \quad (2.12)$$

, donde γ es una constante positiva.

La función de optimización para el SVM es la siguiente:

$$\text{minimizar}(\beta_0, \beta_1, \dots, \beta_n) \left\{ \sum_{j=1}^K \max [0, 1 - y_i f(x_i)] + \lambda \sum_{j=1}^n \beta_j^2 \right\} \quad (2.13)$$

, donde λ es un parámetro de ajuste no negativo y $f(x_i) = \beta_0 + \beta_1 X_i + \beta_2 X_i + \dots + \beta_n X_i$.

2.3.4. Métricas de evaluación

Las métricas son fundamentales al trabajar con inteligencia artificial, ya que permiten evaluar objetivamente el rendimiento de los modelos. Existen muchas métricas, algunas más adecuadas que otras, dependiendo del tipo de problema que se esté tratando.

La tasa de aciertos o exactitud es una de las métricas, la cual se utiliza aquí como única métrica para verificar las predicciones de los modelos. La ecuación de su cálculo es la siguiente:

$$\text{Exactitud} = \frac{\text{Nº predicciones correctas}}{\text{Nº total predicciones}} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.14)$$

, donde TP son los Verdadero Positivos (True Positives), TN son los Verdaderos Negativos (True Negatives), FP son los Falsos Positivos (False Positives) y FN los Falsos Negativos (Flase Negatives).

Esta ecuación representa las predicciones realizadas correctamente por el modelo, que es la suma de los TP y TN, dividido la cantidad total de las predicciones realizadas, es decir, la suma de las correctas más las incorrectas. A medida que las equivocaciones, dadas por la suma de FP y FN, sean menores, la exactitud o tasa de aciertos será mayor para el modelo.

3 | Metodología

3.1. Especificación de requerimientos

Como se indica en el apartado Sección 1.3, el presente informe consta de tres módulos definidos, que pueden desarrollarse de forma independiente. No obstante, para su implementación final como un sistema completo, el resultado o las acciones de cada módulo están estrechamente vinculados entre sí, especialmente con el módulo de adquisición de datos del paciente.

En un ciclo de vida incremental, el entregable se produce a través de una serie de iteraciones que sucesivamente añaden funcionalidad dentro de un marco de tiempo predeterminado. El entregable contiene la capacidad necesaria y suficiente para considerarse completo sólo después de la iteración final (Project Management Institute, 2017).

Partiendo de la definición anterior, se plantea para este proyecto un desarrollo incremental. Esta decisión se basa principalmente en la prioridad de establecer un orden de desarrollo que permita obtener entregables de funcionalidades más reducidas y específicas que los módulos principales. Al optar por un desarrollo incremental, se facilita la determinación de tareas, la estimación y precisión de tiempos para su realización, y la implementación de cambios. Además, este enfoque con entregables funcionales reducidos posibilita analizar el impacto de las nuevas funcionalidades in-

corporadas mediante pruebas del sistema realizadas por los directivos y usuarios en todas las etapas de desarrollo, una iniciativa ampliamente recomendada en el entorno laboral donde se lleva a cabo las tareas.

Otro punto a destacar, es acerca de la realización de las tareas, las cuáles son llevadas a cabo por una sola persona, quién se encuentra en relación de dependencia laboral con la organización mencionada. Esta capacidad de recursos imposibilita la paralelización de tareas. Por ello, las tareas son realizadas de manera secuencial.

Por último, se menciona que el sistema Sader posee actualmente funcionalidades de las cuales el proyecto reutiliza y también son incorporadas al mismo los desarrollos logrados durante el transcurso de las actividades.

3.1.1. Propósito

El propósito de esta sección del documento es proporcionar una guía exhaustiva y detallada de los requisitos fundamentales del módulo web. Se centra en la descripción minuciosa y completa de las funcionalidades y características que el sistema debe implementar, asegurando una especificación precisa y clara para su desarrollo y cumplimiento.

3.1.2. Ámbito del módulo

El software desarrollado consiste en un módulo web que forma parte integral del sistema SADER, accesible a través de navegadores web en diversas plataformas, como computadoras, tabletas y dispositivos móviles con conectividad a internet. El acceso al mismo se realiza a través de la barra lateral de la interfaz gráfica del sistema SADER, donde se encuentra el enlace al módulo de guardia. Este módulo está diseñado para la administración y monitoreo de la guardia ambulatoria, con un enfoque específico en el registro y la gestión de pacientes admitidos, así como en la conservación y protección de su información personal. Las funcionalidades del módulo se limitan

exclusivamente a la admisión y atención de pacientes ambulatorios, excluyendo procesos adicionales como la internación o derivación de pacientes.

El objetivo principal del módulo es optimizar las tareas del personal sanitario, mediante la automatización de procesos como el triaje. Esto requiere que el software sea altamente eficiente, priorizando tanto la rapidez en las respuestas como la continuidad operativa ante posibles errores. El diseño del sistema se ha concebido para ser robusto y resiliente, garantizando un funcionamiento ininterrumpido y una experiencia fluida para el usuario.

El público objetivo del software está compuesto por profesionales de la salud capacitados, para quienes se han diseñado interfaces sencillas e intuitivas. Sin embargo, ciertas operaciones dentro del módulo requieren conocimientos médicos previos, como la toma de signos vitales (presión arterial, temperatura corporal, etc.), los cuales deben ser ejecutados de forma externa al sistema, que actúa únicamente como repositorio de los datos registrados. Es fundamental que los usuarios tengan una competencia básica en el manejo de herramientas informáticas, tales como el acceso al sistema web, el inicio de sesión y la navegación a través de sus diferentes funcionalidades.

3.1.3. Definiciones, acrónimos, abreviaturas y referencias

- Efector: entidad, organización o profesional que proporciona servicios de salud.
- RF: Requerimiento Funcional. Los requerimientos funcionales se centran en los procesos, tareas y operaciones que el software debe realizar..
- RNF: Requerimiento No Funcional. Los requerimientos no funcionales son características que no están directamente relacionadas con las funciones específicas del sistema, sino con cómo debe comportarse o funcionar.
- WEB: conjunto de protocolo y del lenguaje de diseño de páginas para la difusión de contenidos mediante la red Internet.

- RENAPER: Registro Nacional de las Personas. Organismo del Estado que depende del Ministerio del Interior. Su principal función es la identificación y registro de las personas físicas en el territorio argentino.
- REGPROF: Registro Federal de Profesionales de la Salud. Sistema que centraliza la información sobre los profesionales de la salud en Argentina.
- SISA: Sistema Integrado de Información Sanitaria Argentino. A continuación se listan los documentos que referencian esta sección:
- IEEE (1998). Recommended Practice for Software Requirements Specifications (IEEE Standard 830-1998).

3.1.4. Panorama general

Los requisitos se clasifican en dos categorías principales: Requerimientos Funcionales (RF) y Requerimientos No Funcionales (RNF). Los Requerimientos Funcionales detallan las funcionalidades específicas que el sistema debe implementar. Estos especifican las acciones, datos, y comportamientos que el sistema debe soportar, describiendo los servicios y tareas que el software proporcionará a los usuarios finales.

Por otro lado, los Requerimientos No Funcionales establecen los criterios bajo los cuales el sistema debe operar. Estos requisitos definen atributos de calidad del sistema, tales como la usabilidad, rendimiento, seguridad, escalabilidad, y otros factores de operación que no están directamente relacionados con la funcionalidad, pero que son críticos para evaluar su comportamiento y eficiencia en diversas condiciones de uso.

3.1.5. Perspectiva del producto

El módulo web forma parte del conjunto integral de módulos que constituyen el sistema web SADER. Este módulo es de acceso libre y gratuito para los efectores públicos que deseen utilizarlo, siempre que se encuentren bajo la jurisdicción del Ministerio de Salud de la Provincia de Entre Ríos. Para la validación tanto de pacientes

Especificación de requerimientos

como de profesionales de la salud, el módulo se integra con soluciones de terceros, asegurando la autenticidad y trazabilidad de los datos. Cada acción realizada en el sistema queda registrada, garantizando así su verificación en eventuales auditorías.

Al ser implementado, el módulo impacta directamente en el personal asignado a las guardias y en la administración de las mismas. Dada la recepción de numerosas solicitudes por parte de los efectores para disponer de un módulo específico para la gestión de guardias, se anticipa una rápida adopción del sistema desde su despliegue inicial.

El módulo proporciona una vista global del estado de la guardia para todos los usuarios con los permisos correspondientes, mientras que las funcionalidades específicas son habilitadas en función de la profesión del usuario. La gestión de permisos de acceso es un componente central del sistema. Al igual que el sistema SADER en su totalidad, este módulo mantiene la política de autonomía para cada efecto en la administración de sus usuarios. Un grupo reducido de usuarios designados en cada efecto tiene la capacidad de gestionar y asignar permisos dentro del sistema, permitiendo que la operación de las guardias funcione de manera independiente del equipo de desarrollo de SADER. La responsabilidad de asignar los permisos necesarios recae en el personal administrativo del efecto, asegurando así un control efectivo y autónomo sobre las operaciones del módulo.

Un caso especial es el de los usuarios de SADER que poseen privilegios extras. Entre ellos se destacan los directivos de los efectores, que tienen acceso permanente a la información del respectivo efecto. Otros usuarios destacados son las personas con cargos en el Ministerio de Salud, quienes pueden acceder a la información de todos los efectores.

3.1.6. Funciones del producto

A continuación, se muestra una representación gráfica de las funcionalidades ofrecidas por el sistema, especificando el acceso a cada una de ellas según el perfil del usuario. Es importante señalar que algunas acciones comunes a los sistemas web actuales, como la autenticación de usuarios, pertenecen a otros módulos del sistema SADER. Por lo tanto, en este documento se abordan exclusivamente las funciones específicas del módulo propuesto en este proyecto.

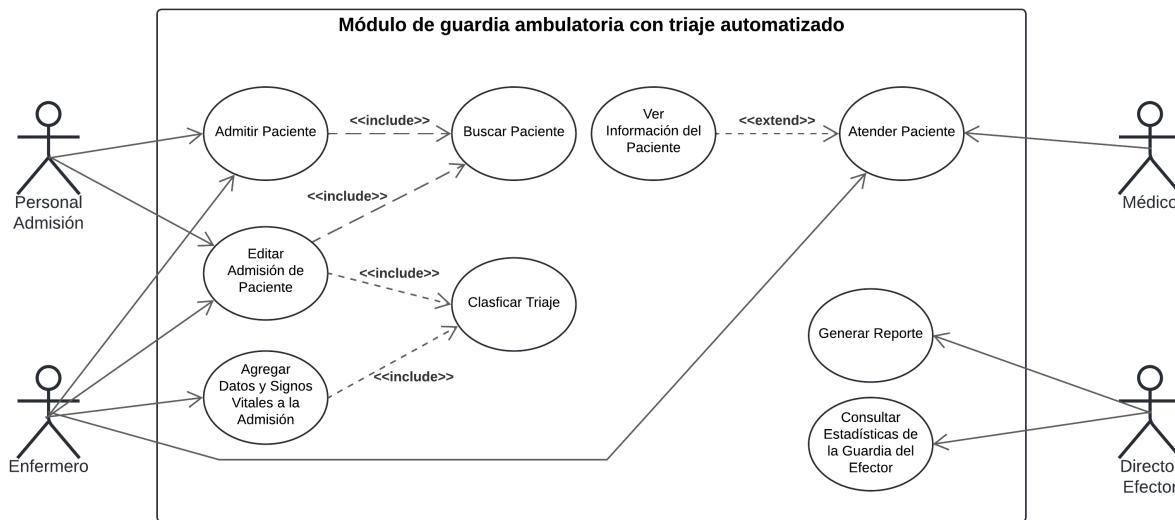


Figura 3.1: Casos de Usos

3.1.7. Características del usuario

El software está dirigido a médicos, enfermeros y personal administrativo de hospitales, por lo que los usuarios son profesionales con formación en el ámbito de la salud. El módulo ofrece herramientas específicas diseñadas para cada rol dentro de la guardia ambulatoria, lo que requiere que los usuarios posean conocimientos y experiencia en este tipo de entornos, asegurando así una utilización eficiente del sistema. Se espera que los usuarios operen las funcionalidades del módulo de manera responsable, dado

se encuentran gestionando información sensible y confidencial de los pacientes.

Cabe destacar a los usuarios que utilizan el sistema SADER previamente, quienes, al estar familiarizados con su funcionamiento, es esperable que logren adaptarse sin dificultad a este nuevo módulo. En cuanto a los usuarios con privilegios adicionales, como los directivos de los efectores, se prevé que utilicen el módulo principalmente con fines de monitoreo y supervisión, así como para la generación de informes y documentación relevante.

3.1.8. Limitaciones

- Limitaciones de hardware**

El módulo web está diseñado para ser compatible con una variedad de navegadores, alineado con el enfoque actual del sistema SADER. Los navegadores predominantes entre los usuarios son Google Chrome y Mozilla Firefox, seguidos en menor medida por Microsoft Edge y Safari.

El módulo está optimizado para ser escalable en función de la resolución de pantalla del dispositivo, permitiendo el acceso desde teléfonos móviles, tabletas y computadoras. No obstante, se recomienda una resolución mínima de 1366 x 768 píxeles para una visualización óptima.

De acuerdo con los datos del sistema SADER, más del 90 % de los usuarios operan en Windows 10, por lo que se sugiere cumplir con los requisitos del sistema para dicho sistema operativo. En términos de hardware, se recomiendan los siguientes requisitos mínimos para el funcionamiento del módulo en una computadora: procesador de 1 gigahercio (GHz) y 2 gigabytes (GB) de memoria RAM.

En cuanto a la conectividad, se recomienda una conexión a internet de banda ancha, preferiblemente a través de wifi o 4G, para asegurar una transferencia de datos eficiente y una experiencia de usuario óptima.

En cuanto al servidor y al entrenamiento de la inteligencia artificial, empresas como Intel (Intel, 2025) o NVIDIA (NVIDIA, 2025) recomiendan un procesador

con múltiples cores y multihilos, dependiendo del modelo de inteligencia artificial a utilizar. Además, se sugiere un acelerador dedicado para tareas de Inteligencia Artificial, como placas de video (GPU) o unidades de procesamiento tensorial (TPU).

- **Interfaces con otras aplicaciones**

El módulo, al igual que el resto del sistema, integra diversas aplicaciones de terceros para su funcionamiento y verificación. Un ejemplo de esto es el Registro Nacional de las Personas (RENAPER), que contiene los datos de todas las personas en Argentina y es utilizado por el sistema SADER para validar la veracidad de la información de los pacientes. Asimismo, se emplea REGPROF y SISA para la verificación de la información de los profesionales de la salud, cumpliendo una función equivalente.

- **Funciones de auditoría**

El sistema SADER incorpora diversas funciones de auditoría que son heredadas por el módulo, generando restricciones y lineamientos en cuanto al desarrollo del software, lo que lleva más tiempo y esfuerzo en la codificación. Entre estas funciones se destacan:

1. **Registro de eventos (Logging):** El sistema lleva un registro detallado de todas las acciones realizadas por los usuarios en relación con la atención de los pacientes, incluyendo la creación y edición de admisiones, así como la atención directa a los pacientes. Este registro proporciona un historial completo de las actividades de los usuarios, facilitando el análisis y resolución de errores, así como la realización de auditorías.
2. **Monitoreo en tiempo real:** Los usuarios autorizados pueden visualizar la información de la guardia en tiempo real, permitiendo un seguimiento continuo de los datos registrados digitalmente durante la operación de la

guardia.

3. **Generación de reportes:** El módulo incluye funciones para la generación de reportes, que se utilizan para la mejora continua del desarrollo del sistema y para la elaboración de informes detallados.

- **Funciones de control**

El módulo incorpora las funciones de control vigentes en el sistema SADER, que incluyen:

1. **Autenticación y autorización de usuarios:** Los usuarios del sistema son validados mediante las aplicaciones de terceros mencionadas y el personal de administración. Los permisos de acción están gestionados de manera que no todos los usuarios tienen acceso a todas las funcionalidades del módulo, garantizando un control adecuado sobre las operaciones disponibles para cada perfil.
2. **Control de datos:** El acceso y la modificación de datos están registrados y restringidos en función de los permisos asignados. Además, los datos son sometidos a procesos de verificación para asegurar su integridad antes de su registro en el sistema.
3. **Comunicación segura:** El sistema SADER implementa un mecanismo de comunicación segura para la transmisión de datos, utilizando encriptación para proteger la información durante su transferencia.

- **Requisitos de lenguaje de programación**

Los lenguajes de programación empleados son los mismos que se utilizan en el sistema SADER, los cuales son Javascript, Python y SQL. Esta elección tiene como objetivo preservar la compatibilidad del sistema y minimizar el riesgo asociado con la integración de aplicaciones desarrolladas en diferentes lenguajes de programación dentro del entorno web.

- **Protocolos de comunicación**

Para todas las conexiones con el servidor web el sistema SADER utiliza el protocolo de comunicación HTTPS, de modo que toda la comunicación permanezca segura.

- **Criticidad del módulo**

Según las estadísticas del sistema, el promedio de usuarios conectados al sistema SADER es de 1.500, con una capacidad comprobada para soportar sin dificultades a 1.000 usuarios realizando consultas simultáneamente. Se ha establecido un límite de 1.000 usuarios conectados simultáneamente para asegurar un rendimiento óptimo, basado en los recursos de hardware disponibles bajo condiciones normales de operación.

No obstante, este límite puede ser excedido en situaciones críticas. En tales casos, se implementan recursos adicionales como parte del plan de contingencia para manejar posibles sobrecargas en el rendimiento del sistema.

- **Consideraciones de seguridad y protección**

En cuanto a la seguridad, el sistema SADER administra la autorización a través de tokens virtuales para verificar y autenticar a los usuarios. Los tokens virtuales son generados y validados para asegurar que los usuarios tengan los permisos adecuados para acceder a los recursos o realizar acciones específicas dentro del sistema.

3.1.9. Supuestos y dependencias

Se asume que el equipo informático y la conexión a internet utilizados por los usuarios del sistema cumplen con los requisitos mínimos recomendados anteriormente, con el propósito de garantizar un funcionamiento estable y eficiente del módulo.

Todos los formularios digitales del módulo siguen los estándares establecidos por el Ministerio de Salud, y cualquier modificación está sujeta a las disposiciones de dicha entidad. Cada formulario e información asociada dentro del módulo está respaldada

por las normativas emitidas por el organismo competente.

En relación con la dependencia de aplicaciones, se reitera la utilización de aplicaciones de terceros para la verificación de la información. Además, el módulo logrará un rendimiento óptimo si la inteligencia artificial encargada de automatizar el triaje mantiene una tasa de aciertos igual o superior a los criterios de aceptación establecidos. Esta dependencia es crítica, ya que la ausencia de inteligencia artificial comprometería los beneficios esperados del sistema automático frente al manual. Finalmente, para alcanzar el pleno alcance del módulo, es esencial su integración con las funcionalidades de atención médica existentes en el sistema SADER. Esta integración es crucial, dado que el proceso completo de la guardia culmina con la atención del paciente y la combinación de estas herramientas optimiza la eficiencia y economiza recursos.

3.1.10. Requisitos específicos

A continuación se destacan las técnicas utilizadas para relevar y elaborar los requerimientos de este proyecto:

- Entrevistas con personal profesional de efectores que utilizan el sistema, como enfermeros y médicos.
- Análisis de documentación, tanto de herramientas informáticas como de informes y formularios en papel.
- Cuestionarios realizados a personal profesional de efectores.
- Observación de funcionamiento de la estructura de las guardias, así como de los sistemas implementados, tanto analógicos como digitales.

Requerimientos Funcionales

Identificación del requerimiento	RF 01
Nombre del Requerimiento	Admitir paciente
Características	Los usuarios registran la admisión de nuevos pacientes.
Descripción del requerimiento	El sistema permitirá al personal de admisión registrar la entrada de nuevos pacientes en la guardia ambulatoria. El formulario de admisión incluirá una función de búsqueda de pacientes que proporcionará información verificada sobre el individuo a ser atendido. Se contemplará la posibilidad de que el paciente no esté en condiciones físicas para proporcionar la información requerida. En tales casos, se omitirá la búsqueda del paciente y el ingreso de todos los datos del formulario de admisión. Estos casos se clasificarán como urgencias especiales, y se deberá completar un campo mínimo de información para su registro.
Requerimiento NO funcional	RNF 01, RNF 02, RNF 05, RNF 06.
Prioridad del requerimiento	Alta

Tabla 3.1: Requerimiento funcional 1

Identificación del requerimiento	RF 02
Nombre del Requerimiento	Editar admisión de pacientes
Características	Los usuarios editarán información de las admisiones de los pacientes.
Descripción del requerimiento	El personal podrá editar la información de admisión de los pacientes, actualizando datos incorrectos o añadiendo nueva información relevante. Se podrán editar únicamente aquellas admisiones que no hayan recibido atención por parte del médico. La excepción a esta regla son los casos especiales de urgencias, los cuales podrán ser editados luego de recibir la atención.
Requerimiento NO funcional	RNF 01, RNF 02, RNF 05, RNF 06.
Prioridad del requerimiento	Alta

Tabla 3.2: Requerimiento funcional 2

Identificación del requerimiento	RF 03
Nombre del Requerimiento	Búsqueda de pacientes

Características	El sistema debe proporcionar una funcionalidad para buscar pacientes registrados previamente.
Descripción del requerimiento	La búsqueda se realizará sobre el número del Documento Nacional de Identidad. La búsqueda retornará la información necesaria para iniciar la admisión del paciente. Para los casos en que el sistema no posea registros del paciente se brindará un mensaje de retorno con la opción de cargarlo al sistema.
Requerimiento NO funcional	RNF 01, RNF 02.
Prioridad del requerimiento	Alta

Tabla 3.3: Requerimiento funcional 3

Identificación del requerimiento	RF 04
Nombre del Requerimiento	Clasificación de triaje
Características	Funcionalidad para requerir una predicción de la inteligencia artificial del nivel de urgencia.

Descripción del requerimiento	Esta función proporciona los datos necesarios para que el modelo de inteligencia artificial estime el nivel de urgencia del paciente. Esta funcionalidad se aplicará de manera predeterminada a todos los casos, es decir, se ejecutará automáticamente al registrar la admisión del paciente y al ingresar los datos de signos vitales. Adicionalmente, se ofrecerá la opción de seleccionar manualmente el triaje si así se requiere.
Requerimiento NO funcional	RNF 01, RNF 02, RNF 03.
Prioridad del requerimiento	Alta

Tabla 3.4: Requerimiento funcional 4

Identificación del requerimiento	RF 05
Nombre del Requerimiento	Agregar datos y signos vitales a la admisión
Características	El sistema debe permitir al personal de enfermería agregar datos y signos vitales del paciente.

Descripción del requerimiento	En esta función el personal autorizado ingresará los datos obtenidos a través de las mediciones. Los campos se dividirán en dos: los obligatorios y los no obligatorios. La interfaz de usuario resaltará aquellos datos obligatorios, de modo que se realice la carga de datos únicamente si se completan todos estos campos.
Requerimiento NO funcional	RNF 02, RNF 05, RNF 06.
Prioridad del requerimiento	Alta

Tabla 3.5: Requerimiento funcional 5

Identificación del requerimiento	RF 06
Nombre del Requerimiento	Ver información del paciente
Características	El sistema debe permitir a los médicos ver toda la información de la historia clínica del paciente.

Descripción del requerimiento	En esta funcionalidad el médico accederá a la información de la historia clínica del paciente en el momento que comience con la atención del paciente. Esta información será únicamente de lectura, es decir, no podrá modificar ninguno de los datos que se muestran.
Requerimiento NO funcional	RNF 03, RNF 04, RNF 05, RNF 06.
Prioridad del requerimiento	Media

Tabla 3.6: Requerimiento funcional 6

Identificación del requerimiento	RF 07
Nombre del Requerimiento	Atender paciente
Características	Los médicos pueden registrar la atención proporcionada a los pacientes.
Descripción del requerimiento	El médico registra la información relacionada a la atención brindada al paciente. La atención se comienza sobre una admisión que posea un nivel de urgencia y los signos vitales cargados previamente. Se designan una cantidad de campos obligatorios mínimos a completar, tanto para el caso de urgencia como para casos normales.

Requerimiento NO funcional	RNF 03, RNF 04, RNF 05, RNF 06.
Prioridad del requerimiento	Alta

Tabla 3.7: Requerimiento funcional 7

Identificación del requerimiento	RF 08
Nombre del Requerimiento	Generación de reportes
Características	El sistema brindará la función de generar reportes detallados de las actividades y datos registrados en la guardia.
Descripción del requerimiento	Los usuarios con permisos podrán generar reportes a partir de seleccionar la información de guardia del efector. Esta información está acotada en el tiempo por una fecha de inicio y una fecha de fin. El reporte a generar contempla la información mostrada en Figura 7.4.
Requerimiento NO funcional	RNF 04, RNF 06.
Prioridad del requerimiento	Media

Tabla 3.8: Requerimiento funcional 8

Identificación del requerimiento	RF 09
Nombre del Requerimiento	Consultar estadísticas de la guardia
Características	Los usuarios con permisos pueden consultar estadísticas y métricas de rendimiento de la guardia.
Descripción del requerimiento	El módulo de estadísticas incluirá una sección dedicada a la información de guardia, con la capacidad de filtrar los datos según el tipo de usuario. Los usuarios pertenecientes a un efector visualizarán únicamente las estadísticas correspondientes a dicho efector. En cambio, los usuarios con privilegios del Ministerio de Salud tendrán acceso a una barra de búsqueda que les permitirá consultar las estadísticas de cualquier efector. Esta búsqueda se realizará ingresando el nombre del efector y seleccionando la opción correspondiente en los resultados.
Requerimiento NO funcional	RNF 02, RNF 04, RNF 05, RNF 06.
Prioridad del requerimiento	Alta

Tabla 3.9: Requerimiento funcional 9

Requerimientos No Funcionales

Identificación del requerimiento	RNF 01
Nombre del Requerimiento	Rendimiento
Características	El sistema garantizará un rendimiento óptimo en todo momento.
Descripción del requerimiento	Garantizar un alto nivel de funcionamiento en el módulo web. Los tiempos de respuesta del sistema siempre deben ser los menores posibles, especialmente en situaciones donde la guardia se encuentre al límite o por encima de sus capacidades de atención. Para ello se establece un máximo de 750 milisegundos como tiempo límite.
Prioridad del requerimiento	Alta

Tabla 3.10: Requerimiento no funcional 1

Identificación del requerimiento	RNF 02
Nombre del Requerimiento	Estabilidad
Características	El sistema será estable para cualquiera de las acciones que se realicen.

Descripción del requerimiento	La estabilidad del sistema se refiere a que el servicio brindado debe mantenerse libre de errores, en donde los fallos ocurridos sean mínimos y se consideren planes de contingencia de modo que no se afecte al uso por parte del usuario. La estabilidad es esperada en condiciones normales de uso y bajo cargas de trabajo esperadas.
Prioridad del requerimiento	Alta

Tabla 3.11: Requerimiento no funcional 2

Identificación del requerimiento	RNF 03
Nombre del Requerimiento	Mantenibilidad
Características	El sistema garantizará un desarrollo de modo que pueda ser actualizado, corregido y mejorado de manera eficiente a lo largo del tiempo.
Descripción del requerimiento	El software debe ser diseñado y desarrollado de tal manera que permita futuras modificaciones, correcciones de errores y mejoras de manera rápida y con el mínimo esfuerzo.
Prioridad del requerimiento	Alta

Tabla 3.12: Requerimiento no funcional 3

Identificación del requerimiento	RNF 04
Nombre del Requerimiento	Adaptabilidad
Características	El sistema soportará modificaciones teniendo conflictos mínimos o nulas.
Descripción del requerimiento	El software será lo suficientemente robusto de forma que al agregar o modificar funcionalidades en un módulo, el resto de los módulos deben tener ningún impacto o el mínimo posible.
Prioridad del requerimiento	Media

Tabla 3.13: Requerimiento no funcional 4

Identificación del requerimiento	RNF 05
Nombre del Requerimiento	Interfaz intuitiva
Características	La interfaz de usuario será intuitiva y de fácil uso, cuyo diseño brinde una experiencia confortable para los usuarios del sistema.

Descripción del requerimiento	El sistema debe tener una interfaz sencilla. Los usuarios deben ser capaces de utilizar el sistema de manera natural y priorizando un uso ágil. Con este fin, la interfaz contará con los campos mínimos requeridos para lograr su propósito en la guardia.
Prioridad del requerimiento	Alta

Tabla 3.14: Requerimiento no funcional 5

Identificación del requerimiento	RNF 06
Nombre del Requerimiento	Seguridad de información
Características	Al poseer datos personales delicados de cada paciente, el sistema garantizará la seguridad necesaria en cuanto a la información que se ingrese en el sistema.
Descripción del requerimiento	Garantizar la seguridad de los datos, siendo solo posible el acceso a éstos por usuarios registrados en el sistemas, siendo éstos los profesionales que trabajen en la guardia.
Prioridad del requerimiento	Alta

Tabla 3.15: Requerimiento no funcional 6

3.1.11. Interfaces externas

- **Interfaces de usuario**

La interfaz de usuario del módulo sigue las directrices establecidas por las políticas del sistema, mostrando únicamente la información mínima necesaria. Los botones y colores han sido seleccionados para generar un impacto positivo, evitando distracciones para el profesional, excepto en situaciones en las que sea necesario captar su atención en un área específica de la pantalla, como en la aparición de mensajes de error.

El acceso a la interfaz se realiza mediante un navegador web, garantizando facilidad de uso y compatibilidad con todos los dispositivos conectados a Internet.

- **Interfaces Hardware**

Para asegurar un rendimiento óptimo, es esencial que los equipos informáticos cumplan mínimamente con los requisitos establecidos en Subsección 3.1.8. Estas especificaciones de hardware aseguran que los usuarios experimenten un desempeño fluido y eficaz al interactuar con la plataforma. También se requieren periféricos estándar en el caso de las computadoras, como mouse, teclado y monitor.

- **Interfaces Software**

El módulo es compatible con los navegadores establecidos en Subsección 3.1.8. Así mismo es accesible a través de los sistemas operativos de computadoras como Windows, Linux o MacOS, de igual manera con los sistemas operativos de dispositivos móviles como Android o iOS. Esta compatibilidad permite al usuario la libertad de poder conectarse al sistema independientemente del medio en que lo realice.

- **Interfaces de comunicación**

Se utilizan protocolos estándares de Internet, como HTTPS, para la comunicación entre los clientes y servidores del sistema. El intercambio de información y archivos se realiza a través de medios seguros, contando con el certificado de

seguridad HTTPS para el sistema SADER.

3.1.12. Funciones

1. Función 1: Admitir paciente

- **Casos de Urgencia:** Se refiere a la generación de admisiones para situaciones de emergencia en las que, debido a riesgos de salud o incapacidad física, el paciente no pueda proporcionar su número de documento. En estos casos, se requerirá completar un campo mínimo de información, con el objetivo de poder editar y actualizar los datos posteriormente.
- **Casos Generales:** El módulo permite registrar la admisión de nuevos pacientes, siendo obligatoria la identificación del paciente. Para ello, se dispondrá de un campo de búsqueda en el que se deberá ingresar el número de documento de la persona a admitir.

En base a las técnicas mencionadas en la sección 3.1.10, el diseño del formulario de admisión se desarrolla a partir del consenso obtenido mediante el análisis de los datos registrados en los libros de guardia de los distintos efectores relevados. Adicionalmente, el formulario es validado por profesionales de dichos efectores, quienes aportan su aprobación o sugerencias tras la revisión del mismo.

Los campos del formulario común son los siguientes:

- Nombre paciente.
- Número de Documento Nacional de Identidad (DNI) del paciente.
- Motivos de la consulta. Mínimo uno.
- Tipo de guardia. Siendo las opciones: Adulto, Pediatría, Ginecología y obstetricia.

- Forma de ingreso. Siendo las opciones: Caminando, En Silla de Ruedas, En Ambulancia Sin Médico, En Ambulancia Con Médico.
- Intervención policial.
- Observaciones o referencias en cuanto al paciente.

Para el caso de las urgencias el formulario de admisión contempla únicamente un campo de texto para la descripción del paciente y/o del estado de arrivamiento.

2. Función 2: Editar admisión de pacientes

Las admisiones registradas en el efecto pueden ser editadas si se detecta que al menos uno de los datos es diferente a los ingresados inicialmente durante el proceso de admisión. Un requisito adicional para permitir la edición es que la admisión no haya sido marcada como atendida por el médico. Esta restricción no aplica a las admisiones de urgencia, ya que la atención médica puede realizarse, permitiendo posteriormente la edición de la admisión para completar la información faltante. Los campos del formulario son los mismos definidos en la Función 1.

3. Función 3: Búsqueda de pacientes

Función que recibe el número de documento de la persona, verifica su autenticidad en la base de datos y devuelve la información requerida del afiliado.

4. Función 4: Agregar datos y signos vitales a la admisión

El módulo permite a los usuarios con los permisos adecuados agregar datos y signos vitales. Esta función es distinta de la opción de editar, ya que se enfoca en añadir información nueva al registro en lugar de modificar los datos existentes. Mientras que la función de edición contiene los mismos campos que en la admisión inicial, esta funcionalidad incluye campos adicionales únicos y no editables. Los campos están divididos en dos categorías: los mínimos requeridos y los no

requeridos. El registro se efectua tan pronto se hayan completado todos los campos obligatorios.

Los campos del formulario son los siguientes:

- Frecuencia cardíaca.
- Frecuencia respiratoria.
- Temperatura corporal.
- Saturación de oxígeno.
- Presión sistólica.
- Presión diastólica.
- Escala de dolor. Valorizada de 0 a 10.
- Tipo de triaje: manual o automático. En caso de ser manual se debe ingresar el nivel.
- Observaciones extra: información que el personal de admisión no registró y será de utilidad para la atención, por ejemplo, automedicación por parte del paciente.

5. Función 5: Clasificación de triaje

La función recibe los datos necesarios que sirven como entrada para la inteligencia artificial, la cual realiza una predicción del triaje a partir de éstas.

Los campos de entrada para predicción de la inteligencia artificial son los enumerados en el informe correspondiente.

Los niveles de triaje seleccionados a partir del Sistema Español de Triaje (SET), por lo que las predicciones que brinda la inteligencia son:

- 1 (Nivel 1) - Reanimación
- 2 (Nivel 2) - Emergencia
- 3 (Nivel 3) - Urgencia

- 4 (Nivel 4) - Prioritario
- 5 (Nivel 5) - No urgente

6. Función 6: Ver información del paciente

Esta función tiene como objetivo recuperar la información relacionada con la historia clínica del paciente. Retorna el historial completo registrado en el sistema, incluyendo todos los datos almacenados en cualquier efector donde el paciente haya sido atendido.

7. Función 7: Atender paciente

El módulo permite registrar la atención brindada por el médico y por el enfermero como dos registros distintos. El registro de la atención es posible una vez que se hayan completado todos los campos obligatorios.

La atención comienza para los casos:

- Urgencias: luego de la admisión de urgencia del paciente.
- Otros: luego de haber completado la admisión con los datos y signos vitales de la función 4.

Para los casos de urgencias se permite la posibilidad de registrar el profesional que realizó la atención, con el objetivo de optimizar el tiempo de los profesionales en la atención.

La búsqueda del profesional que atendió al paciente se realiza a través del Número de Documento Nacional de Identidad (DNI) o Nombre del profesional.

8. Función 8: Generación de reportes

La generación de reportes esta disponible exclusivamente para usuarios con permisos específicos y requiere la modificación del módulo existente de reportes del sistema SADER.

La generación del informe incluye parámetros obligatorios, como:

- a) Fecha inicial
- b) Fecha final
- c) Efector seleccionado

Adicionalmente, se pueden seleccionar otras características opcionales, como los tiempos mencionados en la Función 10, el número de pacientes atendidos durante el período seleccionado y la cantidad de casos de urgencia registrados en cada nivel. La selección de estos campos se realiza mediante la interfaz gráfica, seleccionando las columnas correspondientes en la tabla de datos.

9. Función 9: Consulta estadísticas de la guardia

Esta función implica la modificación del módulo de estadística actual del sistema SADER. En él se pueden ver las estadísticas indicadas por los campos mencionadas en la Función 8.

10. Función 10: Registro de tiempo

La función discrimina dos tipos de tiempos:

- a) Tiempo de espera desde la toma de datos hasta la atención: duración entre el ingreso de los datos del paciente y la respuesta de triaje.
- b) Tiempo de espera desde la admisión hasta la atención: duración entre el registro de la admisión del paciente e inicio de la consulta por parte del médico.

3.1.13. Requisitos de rendimiento

Se prioriza la velocidad de respuesta sobre cualquier otro requerimiento no funcional. El diseño de las consultas y cualquier proceso debe garantizar que no se afecte el rendimiento de la base de datos ni se cause un impacto negativo en el tráfico de red que pudiera generar congestiones.

Se busca optimizar el uso eficiente y rápido de las funcionalidades, garantizando una experiencia fluida para los usuarios al enviar y recibir únicamente los datos esenciales.

3.1.14. Restricciones de diseño

El diseño de los formularios digitales esta alineado con los campos establecidos por los estándares del Ministerio de Salud. En cuanto al modelo de tablas, botones y demás componentes de la interfaz de usuario, se mantienen los patrones de diseño presentes en el sistema SADER. Este enfoque tiene como objetivo asegurar la compatibilidad y mantener la familiaridad del usuario con el módulo, facilitando la asociación de colores, tamaños y acciones, y promoviendo una experiencia de uso sencilla e intuitiva.

3.1.15. Atributos del sistema de software

1. Fiabilidad

Establezca procedimientos y estrategias detalladas para la recuperación rápida en caso de fallos graves. Esto incluye la realización de copias de seguridad periódicas de la información estadística de los efectores, asegurando que estas copias estén disponibles para su restauración en situaciones de emergencia que requieran la recuperación de datos.

2. Disponibilidad

Dada la naturaleza operativa de las guardias ambulatorias, que operan durante todos los días del año, el módulo esta disponible de manera continua, ofreciendo sus servicios en cualquier momento y durante todos los días.

Se implementa un diseño modular que incluya planes de contingencia para situaciones en las que se produzcan errores que impidan el uso del módulo, como la restauración a versiones anteriores funcionales durante la resolución de dichos errores, o mensajes informativos de 'acción no disponible' en la interfaz de usuario cuando sea necesario realizar mejoras de desarrollo en alguna función específica.

Estos planes garantizan que la operatividad del módulo se mantenga limitada, pero no se detenga por completo.

3. Seguridad

La implementación de permisos y registros de usuarios asegura la confiabilidad y seguridad en el acceso y almacenamiento de datos. La información almacenada está disponible para su consulta de manera continua, siempre que se cuente con los permisos necesarios. El registro de acciones (logging) proporciona un respaldo al almacenar digitalmente todos los movimientos realizados por los usuarios.

La utilización de protocolos seguros para la transmisión de datos a través de Internet, junto con un sistema de autorización de usuarios, es fundamental para la seguridad web, ya que añade una capa adicional de protección. En entornos físicos de alta concurrencia, como las guardias médicas, implementar estas características en el módulo garantiza una mayor protección, minimizando el riesgo de intervención por parte de actores no autorizados en las actividades digitales de la guardia.

4. Mantenibilidad

La documentación debe contener toda la información necesaria, de modo que las actualizaciones se realicen de manera sencilla, utilizando la menor cantidad de tiempo y recursos posibles.

5. Portabilidad

Al ser un módulo web, el acceso a la aplicación requiere únicamente una conexión a Internet, lo que ofrece varios beneficios. Entre estos, se incluye la posibilidad de utilizar dispositivos móviles, como teléfonos celulares o tabletas, para acceder al módulo. Además, al no requerir la instalación de software adicional, ya que solo se necesita un navegador web, las actualizaciones y mejoras del sistema se

implementan sin necesidad de modificaciones en los equipos de los usuarios.

La implementación de tecnologías de programación, tanto en lenguajes como en bibliotecas, asegura una compatibilidad con la mayoría de los navegadores web, garantizando que la capacidad tecnológica de los usuarios no sea un impedimento para el uso del módulo.

3.2. Requerimientos de inteligencia artificial

A continuación se desarrollan los resultados de la búsqueda de bibliografía de informes relacionados a la utilización de inteligencia artificial en sistemas de guardia. Posteriormente se describen los modelos más comunes entre los mismos, explicando sus ventajas y desventajas. Finalmente, en base a las características destacadas, se determinan los modelos seleccionados en base a las necesidades y el análisis de cada modelo.

En primer lugar, se toma como principal restricción la necesidad de que el modelo predictor del módulo inteligencia artificial tenga la posibilidad de ser implementado en un servidor que no se encuentra adaptado, en cuanto a sus recursos de hardware, para ejecutar modelos de inteligencia artificial complejos, como redes neuronales. Ya que el servidor, actualmente, cuenta con un procesador Intel Xeon Silver 4114, con un total de 136 gb de ram y sin placa de video, hardware crítico en la implementación de inteligencia artificial.

Estos recursos durante los horarios de uso promedio, es decir, fuera de los picos de consumo por parte de los usuarios, presentan una utilización superior al 50 %. Por lo tanto, si se emplea un modelo de inteligencia artificial con alto costo computacional, este uso adicional podría deteriorar significativamente la calidad del servicio brindado por la organización, afectando la experiencia, y por lo tanto el trabajo, del usuario, así como la estabilidad misma del sistema.

Por otra parte, la complejidad de los modelos debe ser la mínima posible, considerando un escenario donde un programador que posea conocimientos básicos de inteligencia artificial sea capaz de modificarlo. Este requisito se basa en evitar la limitación de personal disponible para trabajar en el módulo, es decir, que no sea un grupo selecto de profesionales los únicos capaces de comprender y corregir el código de los modelos. Este requisito busca promover la accesibilidad y la sostenibilidad del desarrollo, permitiendo que más integrantes del equipo puedan colaborar y garantizar la continuidad de las actividades en caso de rotación de personal.

3.2.1. Informes de sistemas de triaje

En esta sección, se destacan algunos papers e informes que tienen como objetivo, o incluyen, la implementación de inteligencia artificial en la predicción de triaje. El propósito de esta sección es exhibir las opciones más adecuadas a implementar en el módulo de triaje, de modo que funcione como referencia para la selección de los modelos.

1. *Emergency department triage prediction of clinical outcomes using machine learning models* (2019).

- **Logistic Regression with Lasso Regularization:** Modelo de clasificación binaria que predice probabilidades. Lasso aplica penalización para reducir coeficientes irrelevantes, mejorando la interpretabilidad y evitando sobreajuste.
- **Random Forest:** Modelo de conjunto basado en varios árboles de decisión. Promedia o vota predicciones, mejorando precisión y reduciendo sobreajuste.
- **Gradient Boosted Decision Tree (GBDT):** Modelo de árboles de decisión secuenciales. Cada árbol corrige errores previos, logrando alta precisión, pero con riesgo de sobreajuste.
- **Deep Neural Network (DNN):** Red con múltiples capas de neuronas,

efectiva en problemas complejos. Aprende características abstractas y maneja relaciones no lineales, ideal para grandes datos.

2. *Machine learning based triaje to identify low severity patients with a short discharge length of stay in emergency department (2022).*

- **CatBoost:** Algoritmo de boosting basado en árboles de decisión, optimizado para manejar datos categóricos sin necesidad de preprocesamiento extenso. Es eficiente y evita el sobreajuste, proporcionando buenos resultados en tareas de clasificación y regresión.
- **XGBoost:** Modelo de boosting que construye árboles secuencialmente, corrigiendo errores previos. Es altamente eficiente y preciso, muy popular en competiciones de aprendizaje automático por su capacidad de ajuste fino y velocidad.
- **Random Forest:** Modelo explicado en el punto anterior.
- **Logistic Regression:** Modelo citado en el punto anterior, pero sin penalización.

3. *An Artificial Intelligence Based Application for triage Nurses in Emergency Department, Using the Emergency Severity Index Protocol (2022).*

- **Fuzzy Logic:** Modelo basado en lógica difusa que maneja incertidumbre y ambigüedad en datos, usando grados de verdad en lugar de verdadero/falso. Es útil en sistemas de control y toma de decisiones.

4. *Artificial intelligence algorithm to predict the need for critical care in prehospital emergency medical services (2020).*

- **Deep Neural Network (DNN):** Modelo explicado anteriormente en el punto 1.

3.2.2. Desarrollo de ventajas y desventajas de modelos y comparación

En base a la bibliografía de los distintos papers destacados en Subsección 3.2.1, son destacables los siguientes modelos, los cuales se desarrollan sus las ventajas y desventajas.

1. Multinomial Logistic Regression o Regresión Logística Multinomial

El modelo de regresión logística es utilizado en tareas de clasificación, tanto binaria como multinomiales, como el triaje. A continuación se describen brevemente sus ventajas y desventajas (Analytics Lane, 2018) (AWS, 2024b).

Ventajas:

- Simple: tanto su interpretación matemática como su implementación en código es simple.
- Eficiente: es eficiente en términos computacionales, por lo que es una opción factible cuando se poseen recursos de hardware limitados.
- Interpretable: los coeficientes que se calculan indican la relación entre las distintas variables de entrada (signos vitales), así como la probabilidad de que se encuentre dentro de un nivel u otro.
- Versatilidad de datos: el modelo puede manejar variables numéricas como no numéricas, por lo que se pueden adaptar a nuevos tipos de datos de importancia en contextos de salud, como un síntoma.

Desventajas:

- Sobreajuste: el modelo es propenso al sobreajuste de los datos, dado que si no son suficientes o tienen sesgos considerables, el rendimiento se verá afectado negativamente.

- Limitaciones de relaciones no lineales: dado que la decisión es lineal, el modelo es incapaz de modelar relaciones no lineales. El rendimiento del modelo se verá afectado por la complejidad de las fronteras no lineales.

Si bien el modelo de regresión lineal posee algunas desventajas, destacables en el ámbito médico, como la limitación de relaciones no lineales, las ventajas descritas, adicionalmente al hecho de que el modelo sea utilizados en algunas de las bibliografías mencionadas Subsección 3.2.1, lo que demuestra la capacidad de aplicación del mismo, hacen que el modelo sea considerado como apto para su desarrollo.

2. Random Forest o Bosques Aleatorios

El modelo de random forest es utilizado para predicción y clasificación en problemas donde los árboles de decisión presentan sobreajuste durante el entrenamiento o los datos de entrada poseen cierto nivel de ruido (IBM, 2024a).

Al igual que en el modelo anterior, seguidamente se presentan las ventajas y desventajas del random forest:

Ventajas:

- Robustez: al tener una cantidad de árboles de decisión en un random forest, el modelo se ajustará de sobremanera al modelo, ya que el promedio de árboles no correlacionados reduce la varianza general y el error de la predicción.
- Estimador de valores: es eficaz para estimar valores perdidos o faltantes sin perder precisión, lo cual es útil en ciertos contexto, como el de una admisión de guardia, donde pueden no siempre obtenerse la totalidad de los datos de entrada.
- Precisión: el modelo es capaz de procesar grandes conjuntos de datos, logrando predicciones precisas.

- Evaluación de características: el modelo facilita la evaluación de la importancia de las variables de entradas, de modo que se puedan identificar entradas, en este caso los signos vitales, que presenten mayor relevancia.

Desventajas:

- Altos costos computacionales: la carga del modelo, así como su ejecución, requiere de mayor cantidad de recursos computacionales que de un único árbol de decisión. A medida que el tamaño del árbol y el conjunto de datos a procesar aumenten, la cantidad de recursos necesarios también lo hará.
- Largos tiempos de entrenamiento: similar al punto anterior, las dimensiones del árbol y de los datos de entradas conllevan a que el tiempo de entrenamiento sea más largo que el de otros modelos.
- Pérdida de la interpretación: al aumentar la cantidad de árboles, así como la profundidad de los mismos, la interpretabilidad de la predicción del modelo por parte de las personas disminuye.

Cabe destacar que los puntos débiles del modelo requieren de una capacidad de cómputo considerable, sin embargo, el alcance de los casos y los datos utilizados aquí no son considerados de una complejidad suficiente que implique una implementación y entrenamiento dificultoso del modelo. Por otra parte, es mencionable que en los papers mencionados en Subsección 3.2.1 utilizan random forest o algún modelo optimizado del mismo, lo que destaca el empleamiento del modelo para problemas de clasificación similares.

3. **Support Vector Machine o Máquinas de Soporte Vectorial** Las máquinas de soporte vectorial (SVM) son otro modelo con aprendizaje automático empleado en problemas de clasificación y regresión. Algunas de sus ventajas y desventajas destacables son las siguientes (IBM, 2024b).

Ventajas:

- Flexibilidad en fronteras no lineales: la utilización de kernels o núcleos no lineales, como polinómicos o radiales, permiten transformaciones en espacios de mayores dimensiones donde alguno de estos kernels logren mejores separaciones que una lineal. Es usual que en los escenarios que presenten fronteras no lineales las predicciones del modelo tengan resultados positivos.
- Robustez al ruido: los SVM son robustos ante valores atípicos o ruido en los datos de entradas, esto se logra al ajustar el margen mediante el parámetro de coste C(véase Ecuación 2.9).
- Eficiente computacionalmente: al utilizar un subconjunto de los datos de entrenamiento para definir el hiperplano, el modelo permite un uso más eficiente de la memoria así como una mayor velocidad de predicción.
- Efectivas en espacios de dimensiones altas: las SVM son, generalmente, efectivas en contexto que se trabajen con espacios de dimensiones altas, como el caso de este proyecto al trabajar con múltiples signos vitales.

Desventajas:

- Dificultad de interpretación: al ajustar el modelo a un conjunto de datos, dependiendo del entrenamiento así como de la dimensión de los datos y el kernel utilizado, la interpretación de la predicción es menor a la de los otros modelos presentados.
- Dependencia del kernel: la elección de un kernel incorrecto influye en la capacidad de predicción del modelo. En otras palabras, el rendimiento del modelo es dependiente del kernel que se elija, lo que, comúnmente, no es sencillo de realizar.

Las SVM ofrecen ventajas significativas en problemas de predicción como el triaje. Por otra parte, sus desventajas están relacionadas con la complejidad y depen-

dencia de los parámetros utilizados, los cuales son considerados al momento de su implementación y entrenamiento.

En la siguiente tabla se comparan los modelos listados anteriormente, mostrando las características de los mismos, consideradas como más destacables por el alumno.

Modelo	Ventajas	Desventajas	Costo Computacional	Interpretable
Logistic Regression	Simple de implementar, Eficiente, Versatil	Sobreajuste, Limitado en relaciones no lineales	Bajo	Alta
Random Forest	Robustez al sobreajuste, Preciso, Estima valores faltantes	Costoso computacionalmente comparado con un árbol de decisión, Largos tiempos de entrenamiento	Medio	Medio
Support Vector Machine	Robustez al ruido en datos, Maneja espacios con altas dimensiones, Flexible en problemas no lineales	Dependencia del kernel y parámetros	Medio	Bajo

Tabla 3.16: Comparación de modelos de inteligencia computacional

3.2.3. Análisis y selección de modelos

A partir de la comparación de los modelos, así como sus características más importantes, la elección de los modelos se basa en los siguientes criterios, los cuales, en

el veredicto del alumno que desarrolla los modelos, son los más adecuados para el presente documento:

1. Son modelos empleados actualmente, es decir, son aplicados a soluciones de problemas vigentes, como se muestra en los papers citados en Subsección 3.2.1.
2. Tienen un costo computacional bajo o moderado, comparado con modelos más costosos como redes neuronales, lo que permite su implementación en hardware con recursos limitados. Este punto es crucial, ya que permite una posible implementación en el servidor actual de la organización, el cual no cuenta con los recursos necesarios de un servidor para ejecutar tareas de predicciones con modelos complejos.
3. Tienen alto grado de desempeño en problemas de clasificación, como el triaje, lo que garantiza una precisión aceptable en la predicción de los niveles de triaje manteniendo un equilibrio con la velocidad.
4. Son modelos interpretables, a excepción del SVM, lo que facilita la comprensión de los resultados y la identificación de posibles errores o mejoras, así como el análisis en la importancia de los datos de entrada.
5. Son mantenibles, lo que permite a los desarrolladores del proyecto realizar ajustes y correcciones sin necesidad de conocimientos avanzados en inteligencia artificial.
6. Son escalables, permitiendo la adición o modificación de datos de entrada en base a la evolución de los requerimientos.

Habiendo analizado los distintos papers citados en Subsección 3.2.1, contextualizando los distintos modelos, así como sus ventajas y desventajas, se elaboró la siguiente lista de modelos predictivos:

1. Multinomial Logistic Regression (Softmax Regression)
2. Random Forest

3. Support Vector Machine (SVM)

Esta lista representa una primera instancia de los tipos de modelos seleccionados. Durante el desarrollo de esta fase, se evaluarán versiones optimizadas y alternativas de los modelos propuestos para equilibrar la precisión predictiva con las restricciones de hardware existentes, siendo posibles casos de Logistic Regression o Random Forest.

3.3. Librerías de inteligencia artificial

Teniendo en cuenta que las tecnologías de backend son Django REST y Django, la compatibilidad de funcionamiento está asegurada al utilizar el mismo lenguaje de programación, es decir, Python. Al momento de realizar este informe existen múltiples herramientas para desarrollar inteligencias artificiales, algunas de las más utilizables en el mercado y destacables actualmente son:

- **TensorFlow**: Un framework de machine learning desarrollado por Google, optimizado para modelos de deep learning y aplicaciones de inteligencia artificial a gran escala. Se destaca por su facilidad de despliegue en producción, soporte para dispositivos móviles y herramientas avanzadas como TensorFlow Serving y TensorFlow Lite.
- **PyTorch**: Una librería de deep learning flexible y eficiente, muy popular en investigación debido a su diseño dinámico de gráficos computacionales. Ofrece una curva de aprendizaje amigable y es ideal para tareas avanzadas como redes neuronales recurrentes y convolucionales.
- **Scikit-learn**: Una biblioteca de machine learning clásica, enfocada en modelos rápidos y ligeros como regresión, clasificación y clustering. Es ideal para proyectos que no requieren Deep Learning, destacándose por su facilidad de uso, rendimiento eficiente y excelente integración con otras herramientas del ecosistema Python.

Considerándose entonces modelos como regresión logística, random forest y lógica difusa, y esperando un rendimiento rápido con tiempos de predicción, menores a 1 segundo en condiciones normales del hardware, se opta por Scikit-learn como la librería más adecuada. Se destacan algunas de las características de Scikit-learn que justifican esta decisión:

- Es altamente eficiente para implementar modelos de machine learning clásicos como regresión logística y random forest.
- Mayor facilidad de aprendizaje y de mantenimiento de código con respecto a las demás herramientas.
- Está optimizado para velocidad y puede manejar predicciones en tiempo real rápidamente, especialmente en un entorno de producción con modelos ligeros.
- Puede integrarse fácilmente con Django REST para ofrecer predicciones en tiempo casi instantáneo.
- Familiaridad o conocimiento de la herramienta por parte del integrante del proyecto, quien desarrollará e implementará los modelos.

4 | Tecnologías seleccionadas

En esta sección se detallan las tecnologías empleadas para el desarrollo del módulo web. Para la implementación del frontend, backend y base de datos, se mantuvieron las mismas tecnologías actualmente utilizadas por el sistema SaDER al momento de realizar este informe.

Esta decisión se basa en asegurar la compatibilidad e integración del nuevo módulo con los componentes preexistentes del sistema. También se prioriza un desarrollo rápido y eficiente, evitando situaciones comunes que surgen a partir de la integración de nuevas librerías, tecnologías o lenguajes de programación en un proyecto existente. Entre las cuales se pueden mencionar: errores de incompatibilidad, diferencia de rendimiento o diferencias visuales.

4.1. Tecnologías Frontend

Para el desarrollo del frontend, encargado de las interfaces con las que los usuarios interactúan dentro de la plataforma web, se utilizaron las siguientes tecnologías:

- **React**: es una biblioteca JavaScript utilizada para la construcción de interfaces de usuario dinámicas y componentes reutilizables. React permite la creación de aplicaciones web eficientes, proporcionando una gestión óptima del estado y una actualización rápida del DOM virtual.
- **Material UI**: es un framework de componentes de interfaz de usuario basado en React que implementa las pautas de diseño de Google Material Design. Material

UI facilita el desarrollo de interfaces modernas y accesibles, ofreciendo una amplia gama de componentes preconstruidos y personalizables.

- **Bootstrap:** es un conjunto de herramientas de código abierto utilizado para el diseño de interfaces de usuarios de páginas web.
- **jQuery:** es una biblioteca de JavaScript utilizada para tareas como el recorrido y la manipulación de documentos HTML, la gestión de eventos, la animación y Ajax. Esta biblioteca sigue siendo útil para mantener compatibilidad con sistemas heredados y mejorar la interoperabilidad entre navegadores.

Se destaca que estas tecnologías son heredadas del sistema web actual de SaDER. Siendo seleccionadas originalmente por los primeros desarrolladores, ya que permiten una construcción eficiente y optimizada de la interfaz de usuario, garantizando compatibilidad con diferentes navegadores y dispositivos, sin comprometer la experiencia del usuario. Para el módulo estadístico se utilizó jQuery junto con Bootstrap, mientras que en los demás módulos se emplearon React y Material UI.

4.2. Tecnologías Backend

Para el desarrollo del backend, aquella parte del sistema que brinda los servicios necesarios al frontend y que el usuario no puede visualizar, se utilizan las siguientes tecnologías:

- **Django:** es un framework web de Python, diseñado para el desarrollo rápido y limpio de aplicaciones web. Se destaca por proporcionar herramientas completas para gestionar bases de datos, autenticación, formularios, entre otros, facilitando así la creación de aplicaciones web escalables y seguras.
- **Django REST:** es una extensión de Django diseñada para construir API RESTful. Facilita la creación de servicios de backend con serialización de datos, autenticación robusta y manejo eficiente de solicitudes HTTP, destacándose por su

flexibilidad para integrar frontend y backend en aplicaciones modernas.

Al igual que las tecnologías de frontend, esta elección recae en la herencia de las utilizadas en el sistema web. Se destaca la utilización del framework Django para el apartado estadístico y de Django REST para los demás módulos.

Inicialmente, estas herramientas fueron seleccionadas por los programadores que comenzaron el desarrollo de SaDER dada su capacidad de crear aplicaciones fácilmente, al implementar las herramientas que incorpora el framework, se prioriza un desarrollo limpio y robusto. También es destacable la gran comunidad de desarrolladores a nivel internacional que aportan al mantenimiento y al crecimiento del framework, así como su amplia documentación disponible en internet.

4.3. Tecnologías de inteligencia artificial

Para el desarrollo e implementación de los modelos de inteligencia artificial se utiliza la biblioteca Scikit-learn.

Es mencionable la utilización de librerías y bibliotecas complementarias:

- **Numpy**: es una biblioteca de Python, utilizada para manipular grandes matrices y realizar operaciones matemáticas eficientes. Se destaca su uso en el campo del cálculo numérico al ser rápida y fácilmente integrable con otras librerías.
- **Matplotlib**: es una librería de visualización que permite generar gráficos en Python de manera fácil.
- **Pandas**: es una librería de Python enfocada en la manipulación y análisis de datos, ofreciendo estructuras de datos y funciones similares a las planillas de cálculo.

Como se menciona en Sección 3.3, las herramientas para la inteligencia artificial son utilizadas basadas en las necesidades, tanto de los datos, los modelos y las especificaciones del hardware.

4.4. Base de datos

Continuando con los lineamientos del sistema SaDER, se emplea PostgreSQL para el almacenamiento, tratamiento y gestión de los datos, el cual cumple actualmente con dichas tareas en el sistema web.

PostgreSQL es un potente sistema de base de datos relacional por objetos de código abierto que utiliza y amplía el lenguaje SQL. Entre algunas de sus características se destacan:

- **Tipos de datos:** permite gestionar distintos tipo de datos, como los primitivos; estructuras más complejas como fechas; documentos como JSON; entre otros.
- **Integridad de los datos:** para proteger y asegurar la integridad de los datos se destacan las propiedades de unicidad, datos no nulos, llaves primarias y foráneas.
- **Concurrencia y rendimiento:** está optimizado con indexación comunes y avanzadas, también posee control de concurrencia multiversión (MVCC), paralelización de consultas de lectura, partición de tablas, compilación Just-in-time (JIT) de expresiones.
- **Fiabilidad, recuperación ante desastres:** posee registro de escritura anticipada (Write-ahead Logging o WAL), replicación asíncrona, síncrona y lógica.
- **Seguridad:** entre las medidas de seguridad se pueden mencionar la autenticación, un sistema robusto de control y acceso, seguridad a nivel de columnas y filas de las tablas y autenticación multifactor con certificados y un método adicional.
- **Soporte:** PostgreSQL cuenta con una comunidad activa a nivel internacional, brindando soporte por parte de usuarios y desarrolladores.

En el módulo de estadísticas, con el objetivo de optimizar el rendimiento y minimizar la carga de trabajo en el servidor, las consultas SQL que generan los gráficos de los distintos apartados se implementan utilizando vistas materializadas. Las vistas materializadas son un tipo particular de vista, la cual almacena en una tabla el

resultado de una consulta, en lugar de calcular el resultado en tiempo real como lo realizan las vistas normales (PostgreSQL, 2025). Esto permite almacenar los resultados precomputados de consultas complejas, reduciendo el tiempo de respuesta y la carga computacional durante la generación de los informes y visualizaciones.

Las vistas materializadas son útiles en aplicaciones de análisis y generación de informes, donde los datos no necesitan ser actualizados en tiempo real y el rendimiento es una prioridad. Entre las principales ventajas de las vistas materializadas que justifican su uso, se destacan las siguientes:

- **Almacenamiento físico:** Al almacenar los resultados de las consultas de forma persistente, se elimina la necesidad de realizar cálculos en cada acceso, lo que reduce significativamente el tiempo de obtención de los datos al consultar directamente la tabla materializada.
- **Actualización o refresco automático:** La automatización del proceso de actualización de las vistas materializadas garantiza que los datos almacenados se mantengan actualizados. Este proceso suele programarse para ejecutarse en intervalos periódicos, especialmente durante períodos de baja demanda por parte de los usuarios, aprovechando así los recursos del servidor de forma eficiente. PostgreSQL permite la programación automática de dichas actualizaciones mediante herramientas en sistemas operativos como Linux o Windows.
- **Reutilización en consultas:** Las vistas materializadas pueden ser utilizadas como cualquier otra vista en consultas adicionales, lo que contribuye a un aumento significativo en el rendimiento. Al tener los resultados precomputados, las consultas ejecutadas sobre ellas son considerablemente más rápidas, especialmente en escenarios con volúmenes de datos elevados o en operaciones complejas.

Es mencionable la frecuencia automática de actualización de las vistas, acción que se efectúa alrededor de las doce de la noche, horario en el cual el tráfico de usuario, y, por lo tanto, la demanda de trabajo del servidor, es baja.

PostgreSQL cumple un rol fundamental, al permitir almacenar y obtener de manera segura y confiable los datos, tanto de los usuarios que utilizan el sistema web como de las personas registradas en el mismo. De igual manera, se pueden realizar consultas complejas accesible de manera rápida, lo que permitirá a los usuarios obtener información detallada en el apartado estadístico, así como la posibilidad de generar informes a partir de estos.

4.5. Autorización y autenticación

Para la autorización y autenticación en este proyecto se utiliza el módulo de usuarios de Django, ya que esta herramienta permite hacer ambas cosas en un mismo tiempo.

El sistema de autenticación de Django maneja tanto la autenticación como la autorización. La autenticación verifica que un usuario es quien dice ser y la autorización determina qué se le permite hacer a un usuario autenticado.

El sistema de autenticación consta de:

- Usuarios
- Permisos: Banderas binarias (sí/no) que designan si un usuario puede realizar una determinada tarea.
- Grupos: Una forma genérica de aplicar etiquetas y permisos a más de un usuario.
- Un sistema de hash de contraseñas configurable
- Formularios y herramientas de visualización para el registro de usuarios o la restricción de contenidos.
- Un sistema backend conectable

El sistema de autenticación en Django pretende ser muy genérico y no proporciona algunas características que se encuentran comúnmente en los sistemas de autentica-

ción web. Para suplantar estas necesidades, se utilizan herramientas como expresiones regulares (REGEX), limitación de intentos de inicio de sesión, por mencionar algunos.

4.6. Alternativas consideradas

En cuanto al frontend, se consideran otras opciones para el desarrollo de las interfaces de usuario, incluyendo Vue.js y Angular como alternativas viables. Si bien ambas son tecnologías robustas y consolidadas, la decisión de utilizar React se fundamenta principalmente en la experiencia y el conocimiento técnico del equipo de desarrollo de frontend de la organización.

En cuanto a las tecnologías de inteligencia artificial, se evalua PyTorch y TensorFlow. Ambas herramientas ofrecen un alto potencial para desarrollar e implementar modelos de IA, desde los más simples hasta los más avanzados. Cabe destacar la experiencia previa del alumno en el uso de PyTorch, con proyectos realizados anteriormente a este informe.

No obstante, ambas tecnologías demandan mayores requerimientos de hardware en comparación con Scikit-learn. Por ello, su uso se plantea para futuras propuestas que involucren el desarrollo de modelos predictivos más complejos, como las redes neuronales.

Para el backend, muchas organizaciones prefieren utilizar frameworks robustos que ofrecen una amplia gama de herramientas, facilitando un desarrollo más eficiente y seguro. Algunos de estos frameworks integran tanto el desarrollo de funcionalidades de backend como el mapeo de bases de datos, además de incluir plantillas HTML, lo que permite gestionar todo el proceso web de forma integral. Esta aproximación agiliza la creación de aplicaciones web, reduciendo la curva de aprendizaje en comparación con el uso de herramientas individuales por separado.

Alternativas consideradas

Se destacan algunos frameworks, como Flask, que al igual que Django utiliza Python; Laravel, cuyo lenguaje de programación es PHP; y Spring MVC, que emplea Java y utiliza motores de plantillas como JSP (JavaServer Pages) o Thymeleaf para generar contenido HTML dinámico en las vistas.

5 | Módulo de admisiones

5.1. Diseño de la base de datos

El diseño de la base de datos se basa en un enfoque relacional, utilizando tablas para representar las entidades principales del sistema. Se utiliza el gestor de base de datos PostgreSQL para implementar y administrar la base de datos.

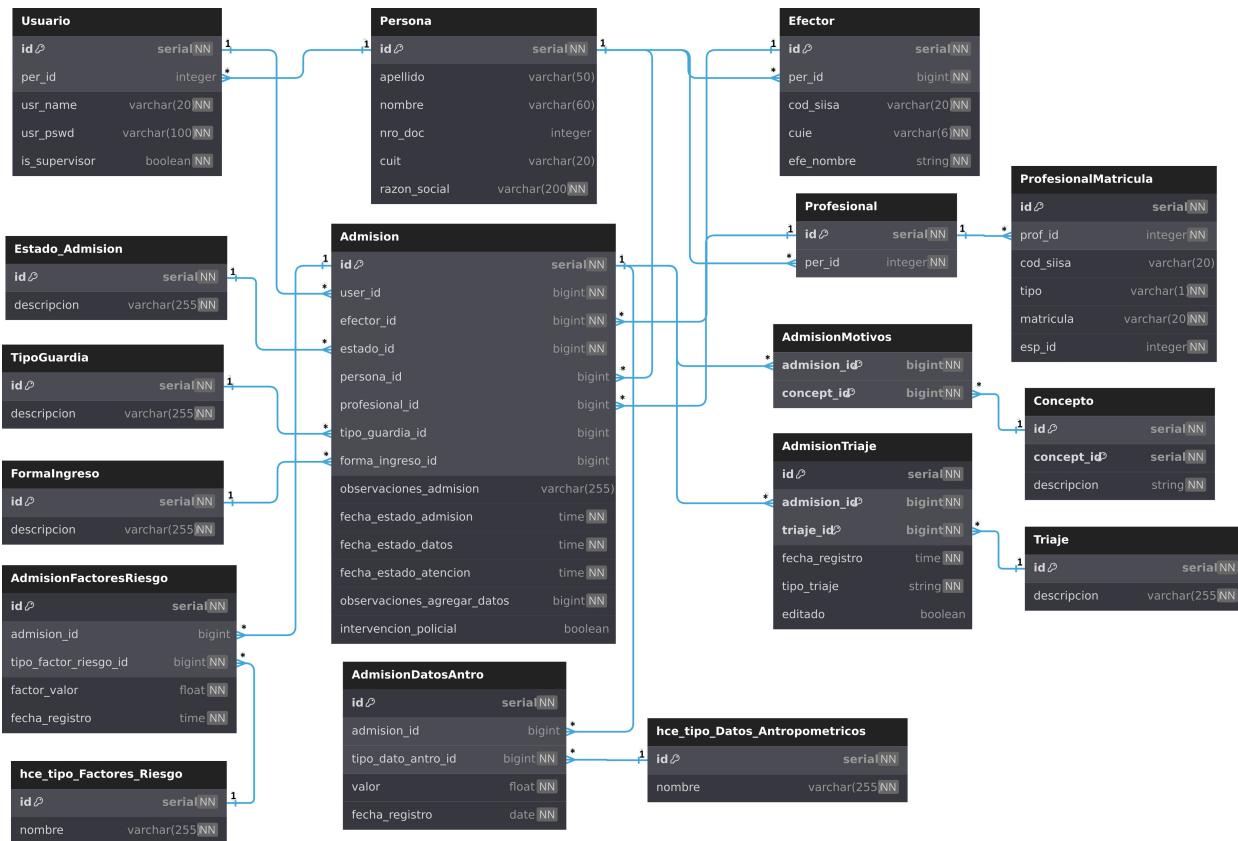


Figura 5.1: Diagrama Entidad Relacion

Las tablas nuevas son: Admision, Estado_Admision, TipoGuardia, FormaIngreso, AdmisionFactoresRiesgo, AdmisionDatosAntro, AdmisionMotivos, AdmisionTriaje y Triaje.

Se destaca la implementación de relaciones con tablas ya existentes, reutilizando tanto las entidades como sus datos y funcionalidades asociadas. Un ejemplo de esto es la tabla de usuarios, que resulta clave al incluir una serie de permisos que, según el usuario en cuestión, permiten realizar determinadas acciones dentro del sistema.

La tabla de usuarios está relacionada con la tabla de personas, lo que significa que cada usuario se asocia de manera única con una persona, de acuerdo con la política del sistema. Esta relación se justifica en la necesidad de garantizar la autenticidad de la persona, mediante la verificación en aplicaciones de terceros, asegurando que los registros en el sistema tengan un responsable real.

Por su parte, la tabla Admisión es el centro del módulo, ya que tiene relación con otras tablas o posee los datos necesarios para abarcar toda la información necesaria en este proyecto. Las admisiones tienen:

- fecha y hora para cada proceso: admisión, registro de datos para el triaje y registro de la atención.
- un efector, es decir, el centro médico o lugar físico donde es registrada la admisión;
- una forma de ingreso, que representa el estado de arrivamiento del paciente;
- un tipo de guardia, que diferencia las guardias físicas que posea el efector para atender;
- un estado, por el cual cambiará a medida que se realice el proceso;
- un usuario, que representa la persona que crea la admisión;

- lista de motivos o síntomas;
- un profesional, que representa el médico que atendió al paciente; uno o más niveles de triaje

Con respecto a las tablas Triaje y Admisión, existe una relación de muchos a muchos. En primer lugar, es importante señalar que una admisión puede ser creada y finalizada sin la necesidad de registrar un nivel de triaje, ya que este paso, aunque recomendado, no es obligatorio dentro del flujo de trabajo.

Por otro lado, la posibilidad de tener múltiples niveles de triaje para una misma admisión se justifica por la variabilidad en los procedimientos de atención entre los distintos efectores. En escenarios donde el proceso de guardia involucra múltiples etapas, es posible que el nivel de triaje se modifique. Un ejemplo de esto ocurre cuando la realización de estudios complementarios revela la necesidad de ajustar el nivel de triaje del paciente a uno más crítico.

La tabla Concepto posee una relación muchos a muchos con la admisión. Esta tabla representa los conceptos de SNOMED.

SNOMED es una terminología clínica de alcance global, que funciona como un vocabulario controlado, y permite estandarizar la información que se vuelve en cualquier tipo de registro médico electrónico (SNOMED, 2025). Es mantenida por SNOMED Internacional, que representa un estándar global para informes de salud. En el año 2018, Argentina se incorporó a dicha organización internacional, de modo que su acceso es gratuito en todo el país.

Cada Concepto representa un significado clínico único, al que se hace referencia mediante un identificador SNOMED CT único, numérico y legible por máquina. El identificador proporciona una referencia única e inequívoca a cada concepto y no tiene ningún significado atribuido interpretable por el ser humano (SNOMED, 2025). SNOMED posee más de 300 000 conceptos clínicos, de modo que permite a los médi-

cos registrar los datos con mayor precisión y coherencia. Otro punto importante es la frecuencia de actualización mensual a nivel internacional y semestral para Argentina, lo que permite mantener los términos actuales utilizados por los clínicos.

El diseño de modelos de datos con estándares permite una homogeneidad en los registros, recurso de suma importancia en medicina, donde el detalle de la información es crucial para garantizar un diagnóstico preciso y un tratamiento adecuado. La consistencia en los registros facilita la interoperabilidad entre sistemas, mejora la calidad de los datos y asegura que los profesionales de la salud puedan acceder a información completa y precisa, lo cual es fundamental para la toma de decisiones clínicas.

Las tablas Datos Antropométricos y Factores Riegos son utilizadas para registrar los distintos tipos de datos necesarios para realizar el triaje. Entre la información de la primera tabla se destacan la talla de altura, el peso, índice de masa corporal, entre otros. La segunda tabla posee datos como frecuencia cardíaca, temperatura corporal, presión sistólica y diastólica, por mencionar algunos.

5.2. Desarrollo de la base de datos

El proceso de desarrollo de la base de datos es fundamental en cualquier proyecto de software, y adquiere una importancia crítica en sistemas que operan en tiempo real, como los sistemas de producción. En estos casos, no se construye un modelo de datos completamente nuevo, sino que se realiza una modificación y ampliación sobre el modelo existente.

Para esta fase, se utiliza como base la estructura de datos de la plataforma Sa-DER, que gestiona las historias clínicas electrónicas. A partir de dicha estructura, se incorporan nuevas tablas, relaciones e índices, garantizando la integridad referencial

y la compatibilidad entre los datos preexistentes y los datos que se integrarán en el módulo de guardia. Se opta por ocultar el diseño de la base de datos de SaDER, dada las políticas de confidencialidad de la organización, por lo que, se puede observar únicamente el modelo propuesto para este informe en Figura 5.1.

La implementación se centra en desarrollar los modelos de las tablas utilizando las herramientas proporcionadas por el framework backend. Tanto Django como Django REST Framework interactúan y gestionan los datos mediante objetos Python denominados modelos. Estos modelos definen la estructura de las tablas, incluyendo columnas, tipos de datos, atributos, referencias a claves foráneas y otras características. Esta metodología abstrae la escritura de SQL (Structured Query Language), permitiendo trabajar con las tablas como objetos, lo que optimiza el tiempo de desarrollo y reduce posibles errores, dado que el compilador de Django genera automáticamente las tablas, relaciones e índices, advirtiendo en caso de que ocurrieran conflictos de código.

Es importante destacar que Django ofrece soporte para relaciones de muchos a muchos como atributos en los modelos. En este diseño, dichas relaciones se implementan mediante tablas intermedias, es decir, modelos o tablas específicos que contienen referencias a las correspondientes involucradas, como se muestra en la Figura 5.1. Este enfoque mantiene la coherencia con la estructura de base de datos preexistente, siguiendo la misma línea de diseño y desarrollo actual de la organización.

En cuanto a los datos, se trabaja con copia de una base de datos correspondiente al mes de Agosto del 2024. De ella se mantienen los datos de las tablas correspondientes a las siguientes tablas:

- Usuario
- Persona

- Profesional
- ProfesionalMatricula
- Efector
- Concepto
- Datos_Antropometricos
- Factores_Riesgo

Las tablas EstadoAdmision, TipoGuardia, FormaIngreso y Triaje cuentan con un fixture propio, cada uno en formato JSON (JavaScript Object Notation). Los fixtures son archivos que poseen el contenido de las tablas y son de gran utilidad en Django REST debido a su capacidad para facilitar y agilizar la carga de datos en una base de datos. Una de sus principales ventajas es la automatización del proceso de carga, lo que permite insertar rápidamente los datos predefinidos en las tablas.

El uso de fixtures tiene como objetivo establecer un conjunto de datos para estas tablas, inicialmente invariables, los cuales pueden transferirse fácilmente y cargarse en diferentes bases de datos físicas a lo largo de las distintas etapas o actividades. Esto simplifica la migración y sincronización de datos entre entornos de desarrollo, pruebas y producción, asegurando que la información en las tablas sea consistente en cada fase del desarrollo.

La base de datos para el módulo de guardia es un elemento crítico dentro del mismo, siendo su mayor implicancia el registro, administración, gestión y obtención de la información a través de aplicaciones web. De modo que se logre, no solo registrar y monitorear la actividad de las guardias de los distintos efectores en tiempo real, a partir del módulo de admisiones, sino que también se pueda realizar análisis, evaluaciones, informes y proyecciones de información histórica por los usuarios que tengan acceso a dicha información en el módulo estadístico, logrando así una mayor autonomía de las organizaciones a con SaDER y optimizando los tiempos de elaboración de

dichas tareas.

5.3. Diseño del módulo de admisiones

Para el diseño del módulo de guardia las interfaces deben ser sencillas e intuitivas, lo que conlleva a respetar ciertos aspectos en cuanto al diseño. Para ello la paleta de colores se mantiene idéntica a la del sistema actual, teniendo colores claros para los fondos, con bordes marcados sutilmente para dividir la información mostrada en pantalla. La información mostrada en todo momento es concisa, dado que no se pretende abrumar al usuario con datos innecesarios.

En base a este lineamiento, los formularios donde los usuarios ingresan datos e información son claros, remarcando las acciones que realiza el usuario. Un ejemplo de ello son los mensajes correctivos, presentes cuando el usuario ingresa incorrectamente un dato y que pretenden guiar al mismo.

5.3.1. Pantalla principal de la guardia

La pantalla principal es la carta de presentación con el módulo de guardia. En ella se listan las admisiones de pacientes específicos del efector en forma de tabla. En la parte superior de la misma se muestra el contador de la cantidad de admisiones actualizado, el nombre del efector y el botón para registrar nuevas admisiones, cuya acción presenta el formulario de Figura 5.3.

Esta tabla mantiene cierta relación con las planillas impresas utilizadas en los efectores de la provincia de Entre Ríos, por lo que se destacan los datos más relevantes de las admisiones: nombre y número de documento del paciente, nivel de urgencia, tiempo de espera y acciones.

Dentro de la columna de acciones se observan botones donde cada uno realiza una función específica. Se detallan los siguientes:

- Editar admisión. Su acción presenta el mismo formulario de Figura 5.3.
 - Agregar datos y signos vitales. Su acción presenta el formulario de Figura 6.1.
 - Registrar atención del profesional. Su acción presenta el formulario de Figura 5.4.
- Esta acción debe estar permitida únicamente para las admisiones registradas como urgencias y aquellas que tengan nivel de triaje 1 o 2.

The screenshot shows a user interface for managing admissions. On the left, there is a vertical sidebar with icons for user profile, search, and calendar. The main area has a header with three horizontal bars. Below the header is a table titled 'Nombre del Efector' (Effector Name). The table has columns for 'Nombre del paciente' (Patient Name), 'Nivel Triaje' (Triage Level), 'Tiempo Espera' (Waiting Time), and 'Acciones' (Actions). A blue button labeled 'Nueva Admisión' (New Admission) is located at the top right of the table. The table contains two rows of patient data, each with a '...'. To the right of the table, there are two small grey buttons labeled 'Botón 1' and 'Botón 2'. Below the table, there are three dots indicating more data.

Figura 5.2: Pantalla principal de la guardia

5.3.2. Nueva admisión de un paciente y edición de admisión

El formulario de registro de nuevas admisiones posee los siguientes campos que debe ingresar el usuario:

- Paciente: en este campo se ingresa el número de documento del paciente que será admitido.
- Botón de urgencia: este botón habilita la admisión para casos de urgencia, donde por diversos motivos no se puede identificar al paciente.

- Motivos/Síntomas: en este campo se ingresan los nombres de los síntomas que posea el paciente.
- Tipo de guardia: se selecciona una opción de la lista de tipos de guardia. Las opciones son: Adulto, Pediatría y Ginecología y Obstetricia.
- Forma de ingreso: se selecciona una opción de la lista de forma de ingreso. Las opciones son: Caminando, En silla de ruedas, Ambulancia con médico y Ambulancia sin médico.
- Intervención policial: una opción binaria, donde puede optarse por no seleccionar ninguna.
- Nivel de triaje: lista de opciones de los niveles de triaje, desde 1 hasta 5, para elegir manualmente el nivel de triaje de la admisión. Este campo no es obligatorio.
- Observaciones: campo de texto sin límite de tamaño, donde el profesional puede agregar observaciones que considere importantes, como pueden ser si el paciente se automedica recientemente.

Nueva Admisión				
Paciente:	<input type="text"/>		Urgencia	
Motivos/ Síntomas	<input type="text"/>			
Tipo Guardia <input type="text"/>	Forma Ingreso <input type="text"/>	Intervención Policial <input type="radio"/> Si <input type="radio"/> No	Nivel de Triage <input type="text"/>	
Observaciones <input type="text"/>				
Cancelar		Guardar		

Figura 5.3: Pantalla nueva/edición de admisión de paciente

5.3.3. Registrar atención brindada por el profesional

El formulario para registrar la atención brindada por el profesional en las admisiones de urgencias muestra la información relacionada a la admisión de dicho paciente, registrada en el Figura 5.3, de igual manera mostrará el nivel de triaje.

Se presenta un campo en el cual se debe ingresar el nombre o número de documento del profesional que se desea registrar. Debajo del buscador se listan la información de los profesionales buscados en el campo anterior. Se detalla el nombre del mismo, el tipo y número de matrícula.

Registrar atención del profesional

Información de la admisión

Nivel de Triaje

Buscar profesional

- Lista de profesionales que realizaron la atención
- ...

Cancelar Guardar

Figura 5.4: Pantalla de registrar atención de profesional

5.4. Desarrollo del módulo de admisiones

5.4.1. Desarrollo de frontend

El desarrollo del frontend del módulo de guardia se inicia estableciendo un componente de React propio para el módulo. Es decir, un componente dentro del Router-Browser, de modo que se le asigne una dirección propia para poder renderizar dicho componente. El objetivo es diferenciar el componente de Historia Clínica de Ambulatoria al de la Guardia, así como sus respectivas direcciones.

Archivos HTML

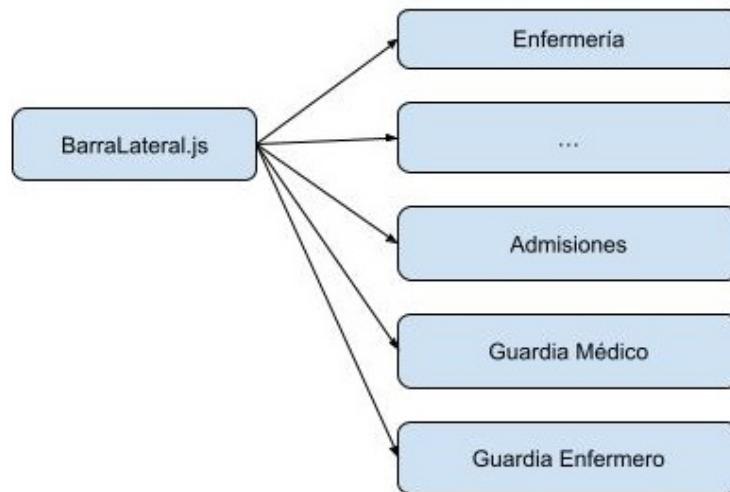


Figura 5.5: Configuración de la barra de navegación

El componente Guardia debe ser accesible mediante la interfaz de usuario. Por lo que se procede a desarrollar la pestaña designada a dicho módulo dentro del componente que renderiza la Barra Lateral, el cual tiene el nombre de “Admisiones”. A partir de la Barra Lateral se puede acceder a los distintos módulos del sistema web, navegando de manera más intuitiva y fluida.

Es destacable describir las particularidades de las pestañas de guardia, tanto para el médico como para el enfermero. Estas pestañas no se encuentran de manera accesible directamente dentro de la barra, como la pestaña de Admisiones, sino que, al

momento en que el profesional realiza la acción de Atender, Requisito Funcional 7, la barra de navegación destaque que se encuentra en el módulo de “guardia” específico de su profesional. Ya que, de no realizar esta señalización, los profesionales podrían llegar a confundirse, creyendo que se encuentran en el módulos atención ambulatoria, cuya interfaz gráfica es similar al de guardia.

Como se observa en la Figura 5.2, el componente Guardia se encuentra estructurado de manera que la cabecera contenga la información del contador de admisiones, el nombre del efector y el botón para crear una nueva admisión. En cuanto a la frecuencia de actualización de la tabla, se opta por agregar un botón a la izquierda del que se encuentra en el diseño original, la funcionalidad de éste es la de poder actualizar la información de la tabla inferior. Este enfoque prioriza una menor cantidad de peticiones por parte de los usuarios y disminuyen tanto el tráfico de información, como la carga de trabajo en el servidor, problema que causa la actualización automática.

La tabla presenta la información relativa de cada admisión de manera tabular, filtrando en varios órdenes las mismas. El orden se realiza de manera descendente, mostrando en la parte superior de la tabla las admisiones con mayor prioridad.

En primer lugar, se ordena según el tipo de admisión creada, es decir, se muestran primero las admisiones de urgencias y luego las normales. En segundo lugar, las admisiones normales se ordenan en base al nivel del triaje, si es que poseen, el cual puede cambiar constantemente, lo que modifica el orden también. Por último y en ambos casos, las admisiones de urgencia y normales, si se encuentran con dos o más en las mismas condiciones, donde no se pueda ordenar en base a las condiciones previamente mencionadas, se ordena según el tiempo de espera sea mayor. El tiempo de espera es considerado como la diferencia entre la hora actual del sistema y la fecha y hora en el que se registró la admisión, siempre que la admisión no tenga asociada ninguna consulta de historia clínica registrada. Para mantener y registrar los tiempos

de esperas se utilizó la librería Moment JS, la cual provee varias funcionalidades como el cálculo de diferencia entre tiempos.

El formulario de registrar una nueva admisión presenta los campos de la Figura 5.3. La búsqueda del paciente, Tabla 3.3 Requerimiento Funcional 3, se realiza al ingresar el número de documento de la persona. El resultado se muestra como una única opción a seleccionar, dado el diseño de unicidad en la tabla Persona.

El campo de los motivos requiere ingresar por texto el nombre del concepto. Se listan los primeros diez conceptos hallados en base a la información escrita por el usuario.

Para estos dos campos se desarrollan los respectivos buscadores de manera que se realicen las peticiones con ciertas restricciones. Un ejemplo de ello es la denegación de ingresar caracteres especiales en el campo de paciente, tales como el signo de exclamación, coma o punto.

Los campos de listas de selecciones ubicados en la tercera fila del formulario son solicitados al momento de que se presiona el botón para crear la nueva admisión. Una función asíncrona recibe los datos de las respectivas tablas, de modo que se puedan mostrar la información guardada en la base de datos. Este enfoque de desarrollo tiene como objeto desligar desde el frontend la información a mostrar en la interfaz de usuario, en otras palabras, se configura la información que recibe dicha función a partir del control del backend. Evitando así corregir código en ambos aspectos, frontend y backend, cuando se requiera modificar el formulario.

Para el caso de las admisiones de urgencia se alerta a través de mensajes en color rojo que se está creando una admisión de urgencias. Este tipo de admisión requiere una menor cantidad de campos obligatorios con respecto a una normal. Estos campos son remarcados con un mensaje en color rojo, en caso de no ser completados, en el cual se especifica el campo a ingresar.

En cuanto a la funcionalidad del botón editar, el formulario presentado es el mismo

de la Figura 5.3. La diferencia recae en la información mostrada, donde esta pantalla presenta los datos actuales de la admisión que se está editando.

Entre los botones de opciones mencionados en Subsección 5.3.1 se encuentra el de registrar la atención por parte del profesional. Desde el registro de la admisión, se diferencia tanto visual como funcionalmente dicho botón.

Para las admisiones creadas con un paciente verificado por el sistema, es decir, independientemente de que se trate de una admisión de urgencia o una normal, el botón muestra una leyenda “Atender” y redirecciona hacia el apartado de SaDER que permite la creación y carga de nuevas consultas en los pacientes. Por otra parte, para las urgencias, donde el campo de paciente no fue ingresado, el botón se llama “Asignar”, el cual abre el formulario Figura 5.4.

Al momento de realizar este documento el sistema web posee la política de registros, donde las consultas solo son creadas para los afiliados federados y verificados por RENAPER. A partir de esta limitación, se opta por desarrollar un formulario específico para estos casos en donde la admisión no se registra junto con un paciente. Permitiendo así a los usuarios mantener un registro de las acciones ocurridas en la guardia, así como el responsable de atender al paciente. Esta restricción fue comentada a los clientes interesados y tenido como mejora en futuras versiones del mismo, los cuales se encuentran por fuera del alcance de este proyecto.

Para realizar las cargas de guardia, tanto para los médicos como para los enfermeros, es necesario redireccionar al apartado mencionado anteriormente. Este es un componente de React que se encarga de almacenar toda la información necesaria, la cual es enviada al backend para registrar la nueva consulta. A su vez, el componente tiene distintas instancias dependiendo de los permisos que se le otorgan al profesional.

El sistema de gestión de los permisos fue desarrollado con anterioridad al presente informe. Sin embargo, éstos son críticos en el sistema, en la seguridad del mismo y

en la interacción con las interfaces de usuarios. Los permisos específicos para este módulo se separan en los siguientes:

- Admisiones
- Guardia médico
- Guardia enfermero

El permiso de Admisiones autoriza la visualización del componente que renderiza la interfaz mostrada en Figura 5.2. Los otros dos permiten al usuario acceder al componente para cargar las consultas médicas. Esta diferenciación es requerida dado que la función de backend reutilizada para el registro de consultas diferencia las creadas por un médico a las de un enfermero.

Basado en el cronograma de actividades, el desarrollo de las interfaces de usuarios fueron desarrolladas en primer lugar. Por ello, para probar el funcionamiento de las distintas acciones en este módulo, se utilizaron datos de prueba para simular la obtención y registro de información.

En el momento en que se realizaron las funcionalidades del backend, se modificaron las funciones encargadas de realizar las peticiones de obtención y de pedido de guardado de datos. No se encontró con mayores dificultades durante el proceso, lo que no causó conflictos en cuanto al tiempo designado para completar el desarrollo del módulo de guardia.

5.4.2. Desarrollo de backend

En primer lugar, se desarrolla la codificación de los modelos de Django que representan a las tablas Estado_Admision, TipoGuardia, FormaIngreso, Triaje y Admision. Como se menciona anteriormente, estos modelos son representaciones de las tablas almacenadas en la base de datos, de modo que, para cada modelo, se debe mantener la especificación de cada atributo de su tabla correspondiente.

Se incia por los modelos mencionados, dado que estos son totalmente nuevos dentro del sistema, a diferencia de otros modelos ya existentes, como el de Usuario por destacar uno.

A su vez, se codifican los fixtures correspondientes a los primeros cuatro modelos destacados en el párrafo anterior. Los fixtures optimizan el tiempo durante el desarrollo de todas las etapas. Esto se debe a que se desarrollan otros módulos en paralelo por parte de la organización, lo que implica cambiar constantemente la base de datos y la instancia de backend del servidor de prueba. Al reanudar la instancia propia de este proyecto en el servidor, se necesita cargar los datos correspondientes a las tablas, lo cual se agiliza al utilizar los comandos de Django útiles para cargar dichos fixtures.

A continuación se desarrollan los modelos de las tablas intermedias, aquellos que representan las relaciones de muchos a muchos. Se codifican funciones específicas para dichos modelos, las cuales son de utilidad para buscar relaciones específicas a partir de atributos de alguna de las tablas o de la misma tabla intermedia.

Desarrollado los modelos de Django, se prosigue con la codificación de los serializadores. Un serializador o serializer, en inglés, permite una interacción entre los formatos de las interfaces web y los de la aplicación de backend. Es utilizado para convertir datos con formatos sencillos JSON o XML y transformarlos en modelos de Django, y viceversa.

En cada serializador se desarrolla la función de validación, la cual comprueba que los datos recibidos sean correctos. También se codifica una función “to_representation”, la cual permite “deserializar” los datos, es decir, convertir los atributos de los modelos a un formato de tipo JSON, manteniendo así un control de qué datos retornar a las interfaces de usuario y cómo hacerlo.

Luego de los serializadores, el desarrollo continua con las vistas de cada modelo. En

Django las vistas se encargan de recibir las peticiones del frontend, procesar los datos y retornarlos. En cuanto a las endpoints de las vistas, es decir, las direcciones donde la interfaz de usuario interactúa con la aplicación de backend, se codifican inicialmente las peticiones de obtención de datos o GET en todas las tablas, tanto para una fila en específica como para la totalidad de las filas. En otras palabras, se puede buscar un objeto específico, a través de su número de identificación o ID, u obtener la lista completa de objetos.

Para la vista correspondiente al modelo de Admisión se desarrollan endpoints adicionales. Un endpoint utilizado para crear las nuevas admisiones, siendo reutilizado para crear admisiones normales o de urgencia. El serializador de Admisión se encarga de distinguir entre ambos tipos de admisiones y validar los datos obligatorios específicos de cada tipo. Otro endpoint es utilizado para actualizar los valores de cada admisión, cuyo fin es posibilitar la edición por parte del usuario. Por último, un endpoint de actualización parcial, la cual utiliza una petición de tipo PATCH, la cual únicamente es empleada para actualizar el estado de la admisión.

También es destacable mencionar la implementación de un logger en cada vista. Esta implementación sigue un estándar de la organización, con el objetivo de mantener un registro de las peticiones recibidas. Dentro del logger se registra, no solo la petición, sino que también el resultado de dicha petición. El resultado puede ser los datos requeridos o un mensaje de error. Esta práctica es de utilidad para mantener un registro de las acciones del usuario, así como para realizar un seguimiento para los casos en que ocurran errores, con el fin de descubrir el fallo y lograr corregirlo.

5.5. Pruebas del módulo de admisiones

A continuación, se detalla el diseño y los resultados de las pruebas ejecutadas en las funcionalidades del módulo de admisión de pacientes descritas en el documento de Especificación de Requerimientos.

Como herramienta auxiliar se utiliza Postman para realizar las pruebas individuales, específicamente en las tareas de backend. Postman es una herramienta utilizada durante el proceso de desarrollo, de pruebas y de documentación de aplicaciones. Facilita las tareas de probar endpoints al realizar peticiones REST de manera rápida y eficiente.

5.5.1. Lista de funciones

A continuación se listan las funciones específicas del apartado de admisión de pacientes, correspondientes al módulo de guardia ambulatoria.

1. Función 1: Admitir paciente

El módulo permite registrar la admisión de nuevos pacientes, siendo obligatoria la identificación del paciente. También se generan admisiones para situaciones de emergencia en las que, debido a riesgos de salud o incapacidad física, el paciente no pueda proporcionar su número de documento.

2. Función 2: Editar admisión de pacientes

Las admisiones registradas en el efecto pueden ser editadas si se detecta que al menos uno de los datos es diferente a los ingresados inicialmente durante el proceso de admisión. Un requisito adicional para permitir la edición es que la admisión no haya sido marcada como atendida por el médico.

3. Función 3: Búsqueda de pacientes

Función que recibe el número de documento de la persona, verifica su autenticidad

en la base de datos y devuelve la información requerida del afiliado.

4. Función 4: Agregar datos y signos vitales a la admisión

El módulo permite a los usuarios agregar datos y signos vitales medidos por el personal de enfermería.

5. Función 6: Ver información del paciente

Esta función tiene como objetivo recuperar la información relacionada con la historia clínica del paciente. Devuelve el historial completo registrado en el sistema, incluyendo todos los datos almacenados en cualquier efector donde el paciente haya sido atendido.

6. Función 7: Atender paciente

El módulo permite registrar la atención brindada por el médico y por el enfermero como dos registros distintos. El registro de la atención será posible una vez que se hayan completado todos los campos obligatorios.

7. Función 10: Registro de tiempo

La función discrimina dos tipos de tiempos:

- a) Tiempo de espera desde la toma de datos hasta la atención: duración entre el ingreso de los datos del paciente y la respuesta de triaje.
- b) Tiempo de espera desde la admisión hasta la atención: duración entre el registro de la admisión del paciente e inicio de la consulta por parte del médico.

5.5.2. Pruebas de las funciones

1. Función 1: Admitir paciente

Identificación de la prueba	P001
Título	Verificar admisión de paciente.
Descripción	Esta prueba se encarga de generar la admisión de un paciente y comprobar su funcionamiento.
Pasos a ejecutar	<ul style="list-style-type: none"> a) Hacer clic sobre el botón de “Nueva Admisión”. b) Ingresar el documento del paciente y seleccionar el resultado encontrado. c) Ingresar y seleccionar uno o más motivos. d) Completar el resto de los campos. e) Hacer clic en el botón “Guardar”. f) Verificar el mensaje de guardado exitoso.
Resultado esperado	Se espera que al realizar las acciones en su respectivo orden el sistema muestre un mensaje de guardado exitoso.
Resultado	Exitoso
Observaciones	Se permite la acción de guardado aún cuando no se cumplen con los requisitos mínimos del formulario. Esta acción debe ser corregida para asegurar un correcto funcionamiento.

Tabla 5.1: Prueba P001

Identificación de la prueba	P002
Título	Verificar admisión de urgencia.

Descripción	Esta prueba se encarga de generar la admisión de un caso de urgencia y comprobar su funcionamiento.
Pasos a ejecutar	<ul style="list-style-type: none"> a) Hacer clic sobre el botón de “Nueva Admisión”. b) Hacer clic en el botón “Urgencia” c) Completar el resto de campos. d) Hacer clic en el botón “Guardar” e) Verificar el mensaje de guardado exitoso.
Resultado esperado	Se espera que al realizar las acciones en su respectivo orden el sistema muestre un mensaje de guardar exitoso.
Resultado	Exitoso
Observaciones	Al igual que en la prueba P001, se permite la acción de guardado aún cuando no se cumplen con los requisitos mínimos del formulario. Misma recomendación que dicha prueba.

Tabla 5.2: Prueba P002

2. Función 2: Editar admisión de pacientes

Identificación de la prueba	P003
Título	Validar la edición de una admisión
Descripción	Esta prueba evalúa el funcionamiento de la edición de la información asociada a una admisión existente.

Pasos a ejecutar	<ul style="list-style-type: none"> a) Hacer clic en el botón “Editar” sobre alguna de las admisiones mostrada en la tabla de la página inicial. b) Agregar un motivo de consulta nuevo. c) Seleccionar opciones distintas a las originales en el resto de los campos. d) Hacer clic en el botón “Guardar”. e) Comprobar el mensaje de edición exitosa.
Resultado esperado	Se espera que al editar una admisión el sistema muestre un mensaje de edición exitosa.
Resultado	Exitoso
Observaciones	No se observaron comportamientos anómalos en la prueba.

Tabla 5.3: Prueba P003

Identificación de la prueba	P004
Título	Validar la edición de una admisión de urgencia
Descripción	<p>Esta prueba evalúa el funcionamiento de la edición de la información asociada a una admisión de urgencia existente. Se encarga de verificar que se haya ingresado información nueva, como un paciente, motivos u otros campos.</p>

Pasos a ejecutar	<ul style="list-style-type: none"> a) Hacer clic en el botón “Editar” sobre alguna de las admisiones de urgencia mostrada en la tabla de la página inicial. b) Agregar un motivo de consulta. c) Seleccionar opciones distintas a las originales en el resto de los campos. d) Hacer clic en el botón “Guardar”. e) Comprobar el mensaje de edición exitosa.
Resultado esperado	Se espera que al editar una admisión el sistema muestre un mensaje de edición exitosa.
Resultado	Exitoso
Observaciones	No se observaron comportamientos anómalos en la prueba.

Tabla 5.4: Prueba P004

3. Función 3: Búsqueda de pacientes

Identificación de la prueba	P005
Título	Verificar la búsqueda de pacientes
Descripción	La prueba asegura que la información detallada del paciente sea correcta.

Pasos a ejecutar	<ul style="list-style-type: none"> a) Hacer clic en el botón “Nueva Admisión” o en “Editar” sobre una admisión de urgencia. b) Ingresar el número de documento en el campo de paciente. c) Verificar la autenticidad de los datos mostrados en pantalla.
Resultado esperado	Se espera que al ingresar un número de documento validado en la Argentina se consiga la información correcta.
Resultado	Exitoso
Observaciones	La búsqueda retorna únicamente información solo si el número de documento es válido, pero no se muestra ninguna interacción que indique cuando éste es inválido.

Tabla 5.5: Prueba P005

4. Función 4: Agregar datos y signos vitales a la admisión

Identificación de la prueba	P006
Título	Validar carga de datos y signos vitales en una admisión
Descripción	Esta prueba se encarga de comprobar el correcto funcionamiento del registro de datos y signos vitales a una admisión.

Pasos a ejecutar	<ul style="list-style-type: none"> a) Hacer clic sobre el botón “Aregar Datos”. b) Ingresar los datos en los campos. c) Hacer clic en el botón “Manual” de triaje. d) Seleccionar una opción. e) Hacer clic en el botón “Guardar” f) Verificar el mensaje de guardado exitoso.
Resultado esperado	Se espera que al cargar al menos un campo y seleccionar el nivel de tiraje el sistema responda con un mensaje de guardado exitoso.
Resultado	Fallido
Observaciones	Luego de ingresar valores en todos los campos de signos y vitales, el sistema no guardó los datos y reveló un mensaje de error.
Cambios realizados	Se modificó el código correspondiente de modo que se solucione el error. Se realizaron cambios en el componente que recibe la entrada de los datos, de modo que reciba solamente valores numéricos y no los interprete como texto.

Tabla 5.6: Prueba P006

5. Función 6: Ver información del paciente

Identificación de la prueba	P007
------------------------------------	------

Título	Verificar la información de búsqueda del paciente
Descripción	La prueba tiene como objetivo comprobar que la información del paciente sea verídica y se muestre de manera precisa.
Pasos a ejecutar	<ul style="list-style-type: none"> a) Realizar los pasos correspondiente a la prueba P009. b) Comprobar que la información que aparece en pantalla corresponde a la del paciente asociado a la admisión.
Resultado esperado	Se espera que la información del paciente asociada entre la admisión y la mostrada en la página web de la Historia Clínica sea la misma.
Resultado	Exitoso
Observaciones	No se observan comportamientos anómalos entre la comunicación del módulo de admisión de guardia y el de Historia Clínica Electrónica.

Tabla 5.7: Prueba P007

6. Función 7: Atender paciente

Identificación de la prueba	P008
Título	Confirmar el registro de la atención del profesional en la admisión de urgencia
Descripción	Esta prueba se encarga de verificar que se registre la atención de un profesional para los casos cuyo nivel de triaje es menor o igual a 2, aquellos considerados como urgencias.

Pasos a ejecutar	<ul style="list-style-type: none"> a) Hacer clic en el botón “Atender” sobre una admisión de urgencia b) Ingresar el número de documento o nombre del profesional a asociar a la atención. c) Verificar que la información del profesional que buscó el sistema sea correcta. d) Hacer clic sobre el botón “Guardar”. e) Comprobar que el sistema muestre un mensaje de guardado exitoso.
Resultado esperado	Se espera que al ingresar correctamente la información del profesional se guarde correctamente.
Resultado	Exitoso
Observaciones	Se observa que el formulario permite la acción de guardar aún sin haber seleccionado a ningún profesional.

Tabla 5.8: Prueba P008

Identificación de la prueba	P009
Título	Confirmar el registro de la atención del profesional en la admisión
Descripción	Esta prueba se encarga de verificar que se registre la atención de un profesional para los casos cuyo nivel de triaje es mayor a 2.

Pasos a ejecutar	<ul style="list-style-type: none"> a) Hacer clic en el botón “Atender” sobre una admisión con nivel de urgencia de 3 o mayor. b) Comprobar que el sistema redirecciona a la página web de Historia Clínica Electrónica del sistema Sader.
Resultado esperado	Se espera que al realizar los pasos correspondiente se redireccione a la página web de Sader.
Resultado	Exitoso
Observaciones	No se observa ningún comportamiento anómalo.

Tabla 5.9: Prueba P009

7. Función 10: Registro de tiempo

Identificación de la prueba	P010
Título	Confirmar el registro de los tiempos de registro de la función 10.
Descripción	Prueba que registra el tiempo en que fue guardada correctamente una nueva admisión, comenzando el tiempo de espera.
Pasos a ejecutar	<ul style="list-style-type: none"> a) Generar una nueva admisión siguiendo los pasos de la prueba P001 o P002. b) Comprobar que el contador de tiempo de espera aumente en base a la hora registrada de la creación de la admisión.

Resultado esperado	Se espera que el registro de la fecha y hora sea correcto, de modo que el contador de tiempo de espera aumente tomando como inicio dicha hora. La fecha y hora registrada no debe ser superior a la actual.
Resultado	Exitoso
Observaciones	No se observa ningún comportamiento anómalo.

Tabla 5.10: Prueba P010

6 | Módulo de triaje

En esta sección del informe se describen las etapas de diseño del módulo de triaje, así como el desarrollo de la inteligencia artificial y las pruebas realizadas, tanto para la aplicación web como para la inteligencia.

6.1. Diseño de módulo de triaje

6.1.1. Diseño del formulario agregar datos y signos vitales

Al igual que en los formularios detallados en Figura 5.2 y Figura 5.3, el formulario de agregar datos y signos vitales posee un diseño basado en los requerimientos obtenidos con las técnicas destacas en Subsección 3.1.10, en base a la cooperación de los profesionales de la salud quienes participaron como fuente de dichos requerimientos.

El formulario para agregar datos y signos vitales posee los siguientes campos que debe ingresar el usuario profesional luego de realizar las mediciones correspondientes:

- Frecuencia cardíaca
- Frecuencia respiratoria
- Temperatura corporal
- Saturación de oxígeno
- Presión sistólica

Diseño de módulo de triaje

- Presión diastólica
- Escala de dolor. Valor de 0 a 10.
- Lesión o herida. Si o No.
- Observaciones: campo de texto sin límite de tamaño, donde se puede agregar observaciones que considere relevantes para la atención.
- Triage: se debe seleccionar alguno de los dos botones. En caso de que se seleccione Manual, se despliegan cinco opciones, en la que el usuario seleccionará el nivel de urgencia. Por defecto se mantiene en Automático.

La figura muestra una captura de pantalla de una aplicación móvil. El encabezado es 'Datos y Signos Vitales'. Hay seis cuadros para ingresar datos: 'Frecuencia Cardíaca', 'Frecuencia Respiratoria', 'Temperatura Corporal', 'Saturación de Oxígeno', 'Presión Sistólica' y 'Presión Diastólica'. Abajo de estos cuadros hay un control deslizante para la 'Escala dolor' (que va de 0 a 10) y un cuadro para marcar la 'Lesión o herida' (con un checkbox). Una sección para 'Observaciones' incluye un cuadro grande para escribir texto. A continuación, hay tres botones: 'Triage: Automático' (destacado en azul), 'Manual' (en gris) y 'Cancelar' (en gris). Un botón 'Guardar' (en azul) está en la parte inferior derecha.

Figura 6.1: Pantalla de agregar datos y signos vitales

Estos datos que sirven como entradas de la IA son ingresados a través del formulario anterior, el cual se puede acceder a partir de las opciones de la tabla mostrada en Figura 5.2.

6.2. Desarrollo de modelos de inteligencia artificial

En esta sección se describe el desarrollo de los distintos modelos de inteligencia artificial. En base a las tecnologías mencionadas en Sección 4.3, se codifican, entrenan y prueban los modelos seleccionados en el apartado Subsección 3.2.2. La calidad de los resultados obtenidos son dependientes, entre otros factores, del conjunto de datos utilizado. Por lo que, de realizar las mismas acciones que aquí son mencionadas, los logros pueden llegar a ser distintos.

Es destacable mencionar que no se utilizan técnicas para evaluar los rendimientos durante el entrenamiento, como validación cruzada, ya que, dado el escaso tiempo empleado para el desarrollo y entrenamiento de los distintos modelos, se prioriza la implementación completa de los modelos, obviando el uso de estas técnicas.

6.2.1. Dataset

El dataset fue obtenido de Kaggle, que es un sitio web en el que tanto profesionales de inteligencia artificial como científicos de datos comparten recursos, guías y bancos de datos, entre otras cosas (Yildiz, 2021).

Este conjunto de datos cuenta con 1267 casos. Para lograr un entrenamiento óptimo, y consecuentemente un rendimiento mayor, se procede a realizar un aumento de datos.

Las técnicas de aumento de datos consisten en generar artificialmente nuevos datos a partir de datos existentes. El aumento de datos aumenta artificialmente el conjunto de datos al realizar pequeños cambios en los datos originales (AWS, 2024a).

Entre las diferentes técnicas de aumento de datos, aquí se emplea el ruido gaussiano. Esta técnica toma las características originales y le añade un pequeño ruido aleatorio al valor numérico. Es decir, para cada valor se agrega el error aleatorio toma-

do de una distribución normal, cuya media es 0 y su desviación estándar es pequeña. En este proyecto se opta por utilizar un 0.01 de desviación estándar, o un 1 %.

El conjunto de datos o dataset obtenido con el aumento de datos cuenta con más de 2600 casos, los cuales son filtrados obteniendo un resultado final de 1683 casos. Este filtro se basa en quitar todos las filas que contengan al menos un dato incorrecto, es decir, aquellos mal registrados. Se consideró un dato mal registrado a aquellos que posean caracteres especiales, como puntos sucesivos, asteriscos, texto y signos gramaticales dentro campos numéricos.

En cuanto a los datos, se dividieron en un 20 % para datos de pruebas, siendo un total de 337, y el resto para ser utilizados en el entrenamiento, a los que corresponden 1346.

La siguiente tabla muestra la distribución de las clases del dataset completo:

Triaje	Nivel 1	Nivel 2	Nivel 3	Nivel 4	Nivel 5
Porcentaje	0,36 %	25,67 %	38,5 %	32,27 %	3,2 %

Tabla 6.1: Distribución de triaje en el dataset

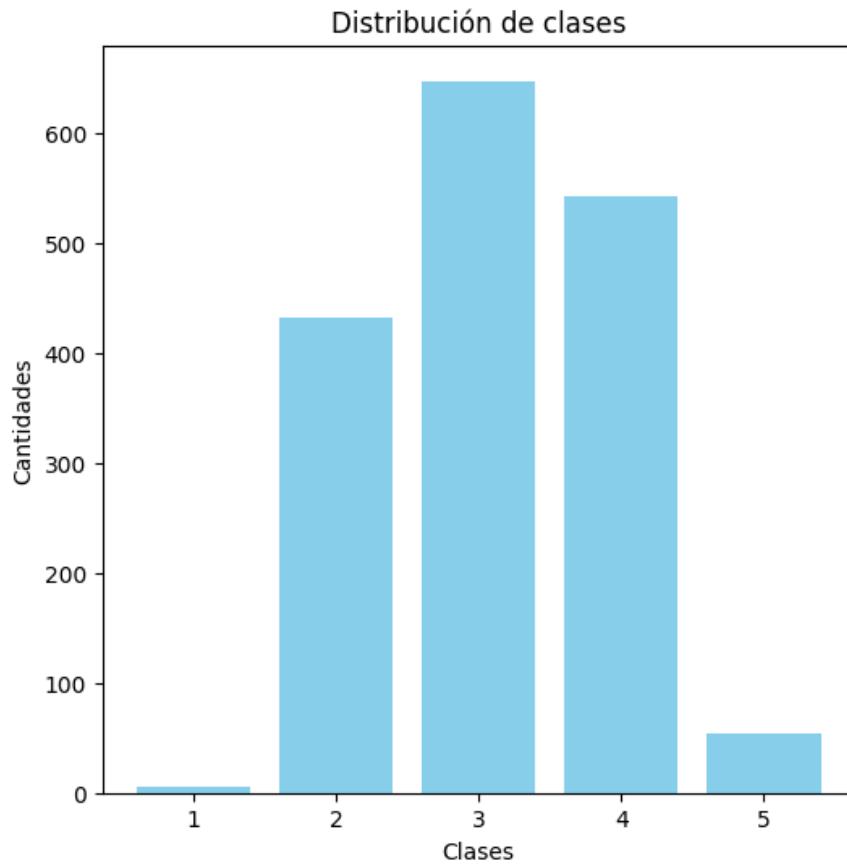


Figura 6.2: Distribución de clases en los datos

Como se observa, la distribución de las clases es despareja en cuanto a los niveles extremos, teniendo mayor concentración en el nivel 3 y 4. En base a esta disposición, los modelos seleccionados son entrenados teniendo en cuenta las variables o parámetros de “desbalance” de clases, lo que permite un entrenamiento distinto al que se realiza con un dataset balanceado. Este entrenamiento se centra en dar mayor peso a los casos que menor porcentaje de apariciones tengas, como el nivel 1 o 5, tanto para los aciertos como para las equivocaciones.

A continuación se muestra tabularmente la composición de un caso o fila del conjunto, con el nombre del campo, la representación o tipo de dato informático, una descripción del mismo y posibles opciones:

Nombre campo	Tipo dato	Descripción	Opciones
Edad	Numérico y entero	Representa la edad del paciente	0 años en adelante
Modo de arrivamiento	Numérico y entero	Forma de ingreso a la guardia médica	1. Caminando 2. Ambulancia
Herida	Numérico y entero	Paciente presenta una herida o lesión	0. Si 1. No
Dolor	Numérico y entero	Paciente presenta dolor	1. No 2. Si
Escala de dolor	Numérico y entero	Representa el grado de dolor	De 1 a 10, si posee dolor. En otro caso 0.
Presión sistólica	Numérico y entero	Representa la presión arterial sistólica del paciente	Expresado en mmHg
Presión diastólica	Numérico y entero	Representa la presión arterial diastólica del paciente	Expresado en mmHg
Frecuencia cardíaca	Numérico y flotante	Representa la cantidad de latidos por minuto	Expresado en valores por minuto
Frecuencia respiratoria	Numérico y entero	Representa la cantidad de respiraciones por minuto	Expresado en valores por minuto
Temperatura corporal	Numérico y flotante	Representa la temperatura del cuerpo.	Expresado en grados Celsius
Saturación de oxígeno	Numérico y entero	Representa el porcentaje de saturación de oxígeno en sangre	Expresado en porcentaje, del 0 al 100.
Triaje	Numérico y entero	Representa el valor de triaje registrado por un enfermero	Valores del 1 al 5

Tabla 6.2: Formato de un caso

Habiendo descrito el conjunto de datos y la conformación de los mismos, se procede a describir el entrenamiento y las pruebas de los distintos modelos predictivos seleccionados.

6.2.2. Logistic Regression

Para desarrollar la regresión logística se utiliza la clase LogisticRegression de sklearn. Esta clase requiere de ciertos parámetros, los cuales son detallados a continuación.

- **penalty**: función de penalidad o regularización, ayuda a prevenir el sobreajuste al penalizar valores altos en los coeficientes. Se utiliza 'l2', dado que soportaba la predicción de más de 2 clases.
- **C**: inverso al valor de regularización, valores grandes indican menor regularización y viceversa. Se utiliza por defecto 1.0.
- **max_iter**: cantidad máxima de iteración, se utiliza como valor definido 500.
- **solver**: algoritmo cuyo fin es optimizar el modelo. Afecta qué opciones de penalización están disponibles.
- **class_weight**: peso de clases para manejar datasets desbalanceados.
- **multi_class**: especifica el manejo de clasificación de clases múltiples, se utiliza 'multinomial' dado los 5 triajes.
- **random_state**: semilla de aleatoriedad, se utiliza 42 por defecto.
- **tol**: tolerancia para que el modelo converja, se utiliza 1e-3 por defecto.

Teniendo en cuenta que la calidad de entrenamiento del modelo depende de estos parámetros, se realiza una búsqueda, luego de la codificación del modelo, para saber los valores óptimos. Ciertos parámetros son definidos por defecto, en base a la disposición del dataset, como `penalty`, `class_weight` y `multi_class`, o para mantener una

consistencia en el desarrollo de todos los modelos, como el valor de C, max_iter y tol.

La búsqueda de hiperparámetros es el proceso de probar y evaluar diferentes combinaciones de parámetros configurables de los modelos, los cuales no se aprenden durante el entrenamiento del mismo. Este proceso tiene como objetivo identificar la combinación de hiperparámetros que maximiza el rendimiento del modelo. En este desarrollo se utiliza el método exhaustivo para la búsqueda, el cual consiste en probar todas las opciones disponibles y seleccionar el que mejor rendimiento otorga. Este enfoque es costoso computacionalmente, por lo que se limitaron los hiperparámetros a uno o dos por modelo.

Por ello se realiza una búsqueda para obtener la función solver óptima. Las opciones de funciones son 'newton-cg', 'lbfgs', 'sag' y 'saga'. Se entrena el modelo variando la función, obteniendo los valores de tasa de aciertos individualmente, tanto para el conjunto de datos entrenamiento como para el de prueba.

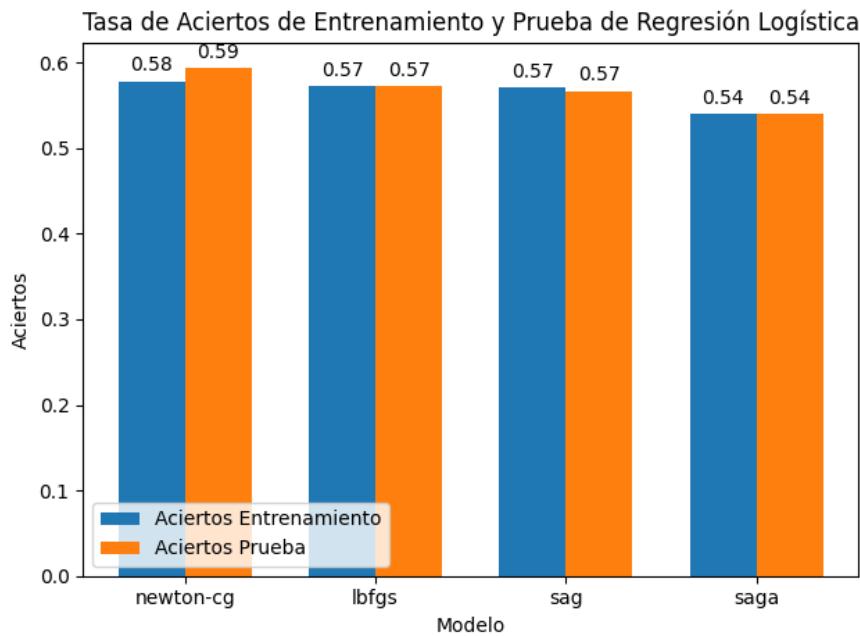


Figura 6.3: Tasa de aciertos modelo Regresión Logística

Por otra parte, se muestra a continuación el número de iteraciones dados en el proyecto de cada función para converger a la solución, es decir, que la mejora entre iteraciones está por debajo del umbral o tolerancia seleccionado.

Modelo	newton-cg	lbfgs	sag	saga
Nº iteraciones	7	24	111	148

Tabla 6.3: Número de iteraciones por modelo para converger

En base a las tasas de aciertos, así como al número de iteraciones por modelo, el mejor solver hallado fue el newton-cg.

Finalizada la etapa de desarrollo y entrenamiento, el modelo de regresión lógica final está compuesto por los parámetros por defecto, así como la función newton-cg. Es destacable mencionar que el mejor modelo de regresión no cumple con el criterio de aceptación, es decir, el umbral mínimo de tasa de aciertos del 75 %.

6.2.3. Random Forest

Para desarrollar el modelo random forest o bosque aleatorio se utiliza la clase RandomForestClassifier de sklearn. Para dicha clase se deben definir los siguientes parámetros:

- **class_weight**: peso de clases para manejar datasets desbalanceados.
- **n_estimators**: representa el número de árboles en el bosque.
- **max_depth**: controla la profundidad máxima de cada árbol.
- **random_state**: semilla de aleatoriedad.

Para este modelo se opta por realizar una búsqueda de valores óptimos para el número de estimadores o n_estimators, en primer lugar, para continuar con el coeficiente de random_state. Por otra parte, el class_weight continúa siendo desbalanceado y la profundidad del árbol o max_depth se mantuvo en 'None', de modo que, a partir de

una mayor trabajo de cómputo, no se limita la profundidad, permitiendo un mejor resultado en el entrenamiento.

La obtención del número de estimadores óptimo se realiza al variar el número desde 1 hasta 42. En cada iteración se entrena el modelo con el parámetro de estimador correspondiente, así como su tasa de aciertos para los datos de entrenamientos.

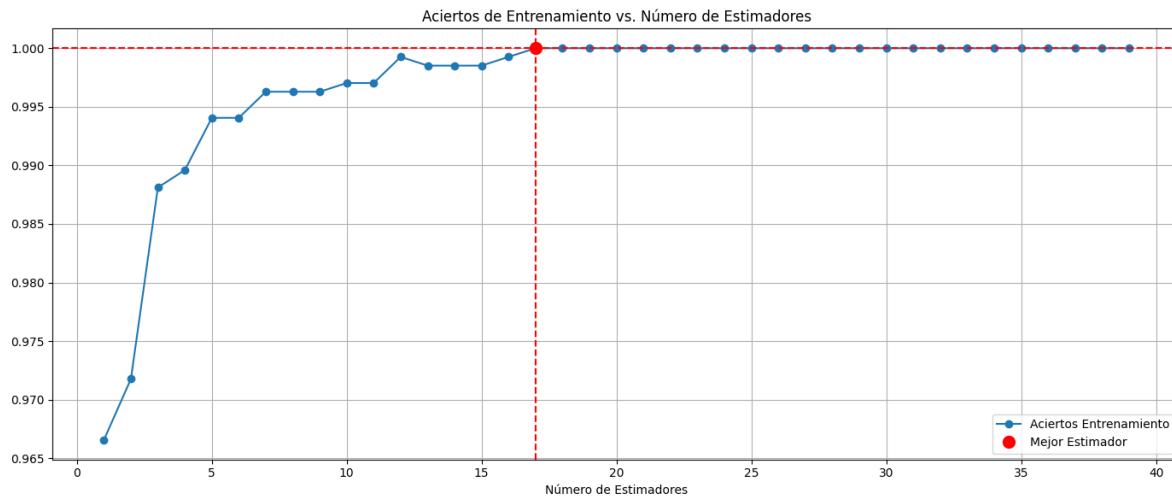


Figura 6.4: Tasa de aciertos para número de estimadores

En el gráfico anterior se muestra la varianza de la tasa de aciertos contra los números de estimadores. El número óptimo hallado es el 17, el cual está marcado con la línea roja, cuya tasa de aciertos de entrenamiento es 98,21 %. Así mismo, la tasa de aciertos de los estimadores mayores a 17 no superaron la diferencia de la tolerancia, es decir, no se lograba una mejora significativa. Por ello se opta por seleccionar el 17.

Continuando la búsqueda de parámetros, se realiza el mismo proceso para obtener el valor óptimo de random_state o semilla de aleatoriedad. Para ello se utilizan valores desde 1 hasta 49, con el número de estimadores hallado en el paso anterior.

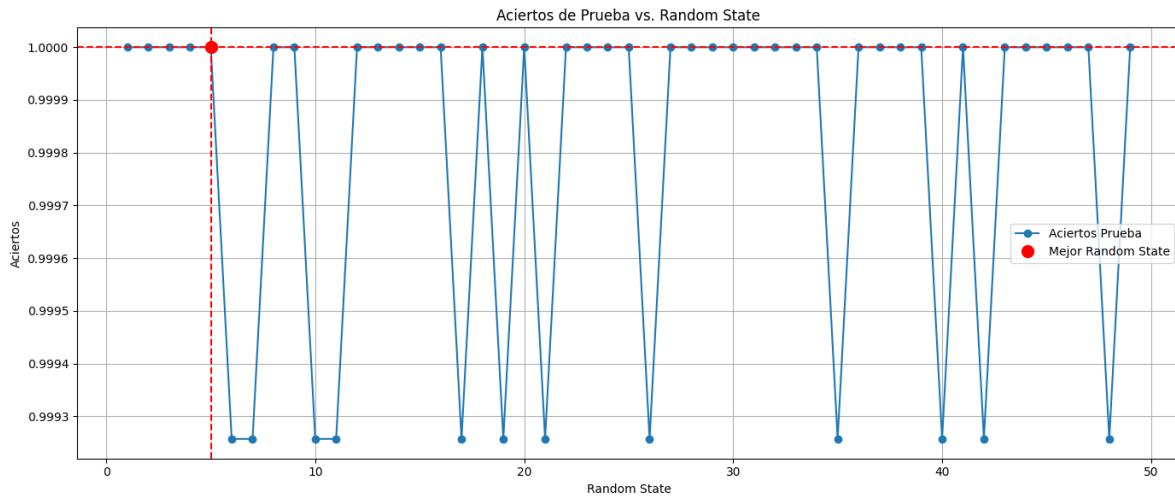


Figura 6.5: Tasa de aciertos para valores de random state

El valor óptimo, marcado con línea roja, es el random_state 5. De igual manera, los valores siguientes no presentan una mejora superior a la tolerancia, por ello se opta por este valor.

Por último, se entrena un modelo con los parámetros óptimos. Este modelo brinda una tasa de aciertos de 99,1 % para el conjunto de datos de pruebas.

6.2.4. Support Vector Machine

El modelo de support vector machine se desarrolla utilizando la clase svm de sklearn. Para dicha clase se definieron los siguientes parámetros:

- **kernel**: función del núcleo o kernel. Este parámetro define la función del hiperplano separador en el espacio de dimensiones mayores.
- **C**: inverso al valor de regularización, valores grandes indican menor regularización y viceversa. Se utiliza por defecto 1.0.
- **gamma**: parámetro que define la influencia de una muestra sobre las demás.
- **degree**: grado del polinomio, en caso de elegir el kernel 'poly'.
- **tol**: tolerancia para que el modelo converja, se utiliza 1e-3 por defecto.

- **max_iter**: se utiliza -1, valor que no limita la cantidad de iteraciones.
- **decision_function_shape**: se utiliza 'ovr' (one-vs-rest) para calcular la función de problemas teniendo una clasificación de múltiples clases.

Para este modelo se opta por realizar la búsqueda del valor óptimo para el kernel y gamma. Los kernels seleccionados son 'linear', 'poly', 'rbf' y 'sigmoid'. La búsqueda de un kernel óptimo es esencial, ya que permite un mejor ajuste a los datos del conjunto.

En primer lugar, se comienza por hallar el kernel óptimo. Al igual que en los modelos anteriores, esto se realiza al entrenar el modelo svm con los parámetros requeridos y variando el kernel. En cada variación se obtienen los valores de la tasa de aciertos para el conjunto de datos de entrenamiento.

Para aquellos casos, como el kernel 'poly', que requieren de parámetros específicos, como el grado del polinomio o degree, se utilizan los valores por defecto. Si bien, esta restricción afecta el rendimiento del modelo, se opta proceder de esta manera con el objetivo de optimizar el tiempo y lograr un entrenamiento para todos los kernels.

La Figura 6.6 muestra los resultados para cada kernel. La mejor tasa de aciertos la logró el kernel 'rbf', cuya diferencia con 'poly' es apenas del 2 %, demostrando un nivel de entrenamiento similar para ambos kernels. Los demás modelos, a pesar de no tener un máximo de iteraciones, convergen en base a la tolerancia definida, logrando valores de tasa de aciertos por debajo de los kernels mencionados anteriormente.

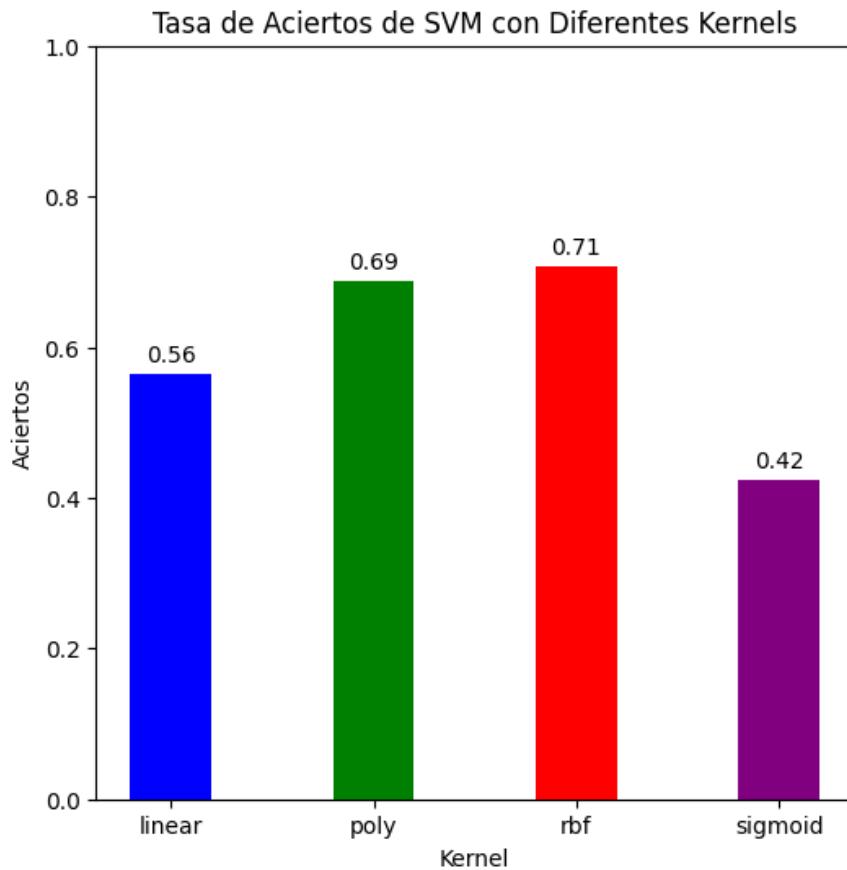


Figura 6.6: Tasa de aciertos para los kernels

Si bien los kernels 'poly' y 'rbf' logran los mejores resultados en el entrenamiento, no superan el mínimo establecido como criterio de aceptación. Para suplir esta falencia, se procede a realizar una optimización en cuanto al parámetro gamma. El kernel seleccionado como el óptimo es el 'rbr', dado que 'poly' requiere de otro parámetro importante, el grado del polinomio, para el cual también debe buscarse la variable óptima. Esto implicaba un mayor uso de cómputo y tiempo.

En cuanto a la búsqueda del parámetro óptimo gamma, se utiliza el kernel mencionado anteriormente y se varia sobre las opciones 'auto', 'scale', 0.001, 0.01, 0.1, 1 y 10. El valor 'auto' permite establecer el valor de gamma como el inverso del número de características del dataset (1 número de muestras), asignando igual influencia a todas las características. El cual es adecuado cuando las características tienen rangos

similares o están escaladas de manera uniforme. Por otro lado, el valor 'scale' ajusta el gamma considerando tanto el número de características como su varianza, mediante la fórmula ($1 / \text{número de muestras} * \text{varianza}$). Este enfoque adapta el modelo a la dispersión de los datos, lo que lo hace más robusto en datasets con características de diferentes escalas o distribuciones heterogéneas.

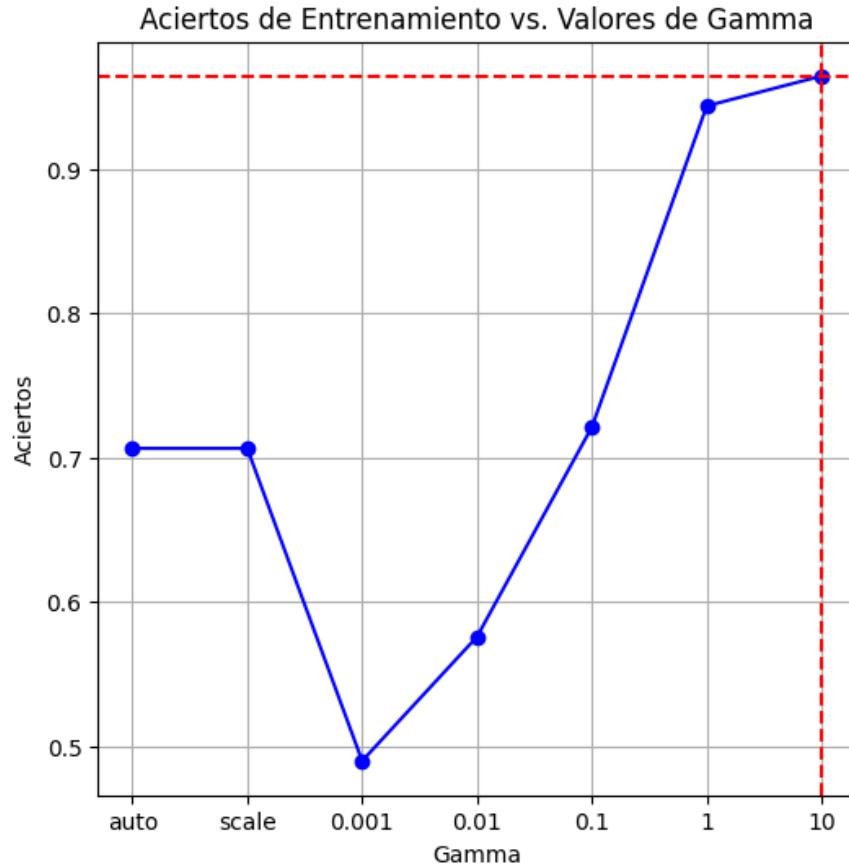


Figura 6.7: Tasa de aciertos para valores de gamma

En la Figura 6.7 se observa los valores de tasa de aciertos para el modelo entrenado con los distintos valores de gamma. Como se puede observar, los parámetros de búsqueda automática no son los mejores, ya que, al entrenarse con valores colocados manualmente se obtienen tasas más altas. En este caso se opta por detener la lista en el valor 10, el cual logró una precisión del 96,4 %, superando al gamma 1, cuya tasa de aciertos fue de 93,4 %.

Finalmente, habiendo encontrado los valores de parámetros óptimos, se entrena un modelo con dichos parámetros. Luego del entrenamiento se prueba dicho modelo con el conjunto de datos de pruebas, obteniendo una tasa de aciertos de 99,1 %.

6.2.5. Comparación de los resultados y conclusiones

En esta sección se comparan los resultados de los modelos de inteligencia artificial, la selección del mejor modelo y se desarrolla el proceso llevado a cabo para implementarlo en conjunto con el backed.

1. Comparación de los resultados

Habiendo entrenado los modelos con los valores óptimos específicos de cada uno, se examinan los resultados de los mismos en base a las pruebas realizadas, en la siguiente tabla.

El modelo con mayor tasa de aciertos es el random forest, seguido por el support vector machine, con solo una diferencia del 2,7 %. Por último, el desempeño es del logistic regression.

Estos valores reflejan las características no solo del entrenamiento, sino que también del propio conjunto de datos. Por lo que, en caso de utilizarse un dataset distinto, o lograr una composición de datos distintas en base a las necesidades y requerimientos, los resultados pueden diferenciarse a los obtenidos y mostrados en el presente informe.

Modelo	Tasa de aciertos en pruebas
Logistic Regression	57,5 %
Random Forest	99,1 %
Support Vector Machine	96,4 %

Tabla 6.4: Tasas de aciertos de los modelos predictivos

A continuación se muestra la matriz de confusión, para cada modelo seleccionado. La matriz de confusión es útil para evaluar el rendimiento de un modelo de clasificación. La matriz muestra las clases verdaderas para los datos en las filas y las clases que el modelo predijo en las columnas. Es mencionable el eje horizontal de las tablas, el cual, por motivos de codificación, comienza en 0, siendo el respectivo para el nivel 1 del triaje y así sucesivamente.

A partir de la matriz de confusión se puede detectar visualmente las predicciones que realiza el modelo y si se equivocó o no, así como en qué valor de triaje lo hacen. Esta herramienta es útil para la realización de evaluaciones a partir de ciertas métricas como la tasa de aciertos, que es el más común y sencillo de implementar, y otros más complejos como la precisión, sensibilidad o recall, f1 score, por mencionar algunos.

Por motivos de escasez de tiempo, se obvia un análisis más profundo de las pruebas de los modelos, optando basarse únicamente en la métrica de tasa de aciertos para seleccionar el modelo a implementar.

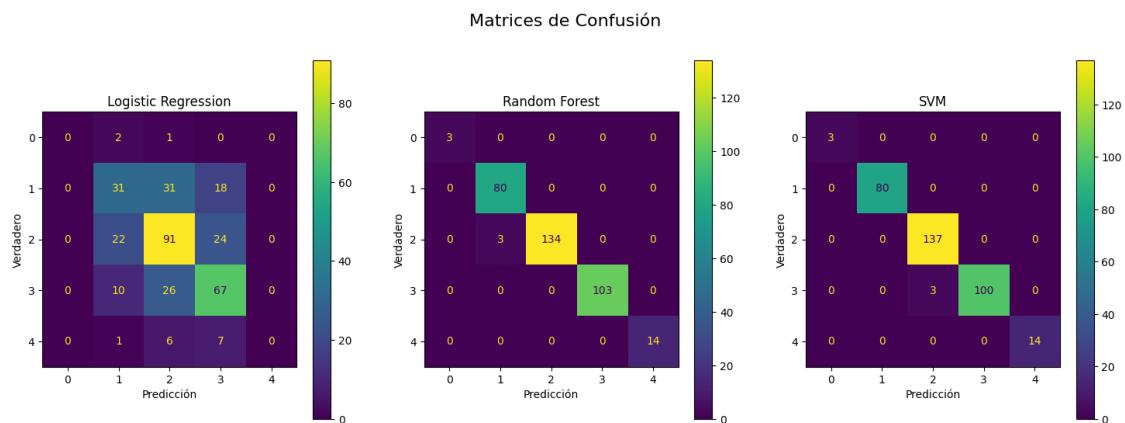


Figura 6.8: Matrices de confusión

En cuanto a las matrices de confusión del random forest y svm se observan las similitudes en cuanto a las predicciones. Lo que demuestra la tasa de aciertos obtenidas en ambos modelos. La diagonal principal es similar para ambas matrices,

salvo algunas predicciones erróneas entre los niveles 2 y 3.

El modelo de regresión logística presenta una matriz de confusión dispareja en todos los niveles. Donde se destaca las predicciones erróneas en los extremos, tanto para el nivel 1 como para el nivel 5. También es mencionable que el nivel 2 tiene la mayor cantidad de predicciones equivocadas, con 49 predicciones erróneas.

2. Conclusiones e integración de la inteligencia artificial

En base a los datos y gráficos mencionados anteriormente, se opta seleccionar al random forest como el modelo a ser integrado en el backend del sistema web.

Esta decisión no es final, ya que se puede cambiar el modelo por distintas circunstancias a solicitud de terceros. El cambio puede ser resuelto siguiendo los pasos realizados, los cuáles son detallados a continuación.

Se detalla el proceso que se realiza para predecir automáticamente el triaje, mostrado en Figura 6.9.

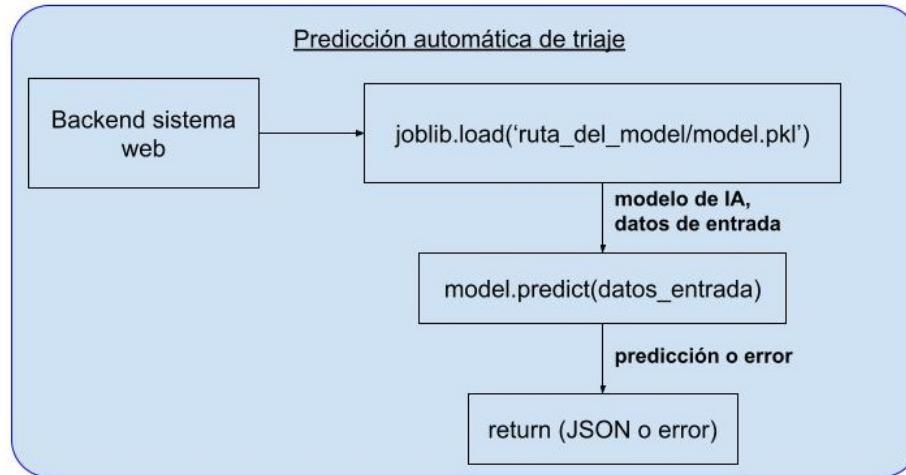


Figura 6.9: Proceso de predicción automática de triaje

Para implementar un modelo de predicción de sklearn en el sistema web, se debe utilizar una herramienta auxiliar para exportar el modelo entrenado. En este proyecto se utiliza joblib, que es un conjunto de herramientas gratuito y que po-

seee actualizaciones frecuentes. Joblib provee una función, dump(), que permite guardar objetos de python y ser exportados con una extensión '.pkl'.

En cuanto a las modificaciones en el backend, se agrega un endpoint, el cual recibe los datos ingresados en el formulario de la Figura 6.1, los demás datos, como la edad de la persona o la forma de ingreso o arrivamiento, son obtenidos de la información del paciente o de la admisión, registrada en sus respectivas tablas. Obtenido los datos que son las entradas al modelo predictivo, se utiliza la función load(), que permite importar un objeto de python, el cual debe coincidir con el nombre dado en la función dump(). Se prosigue a predecir el triaje pasando los parámetros al modelo, la cual se almacena en una variable. Por último, procede a quitar de la memoria del sistema el modelo de inteligencia con la función del de python y se retorna la predicción en formato JSON o el error en caso de que ocurriera.

Con esta sucesión de pasos, se desarrolla la implementación del modelo de inteligencia artificial en el sistema web. De requerirse implementar otro modelo se debe cambiar el nombre del mismo en la ruta dentro del joblib.load(), recompilando el entorno del backend.

7 | Módulo de estadísticas

7.1. Diseño del módulo estadístico

Esta sección del informe tiene como objetivo describir el diseño del módulo de estadísticas, su desarrollo, tanto de frontend como de backend y sus posteriores pruebas y resultados.

La sala de estadística es un módulo que permite a los usuarios autorizados de SaDER visualizar y analizar datos estadísticos de los efectores de salud de la provincia de Entre Ríos. Este módulo es accesible a través de la barra lateral de la interfaz de usuario, pestaña que se habilita a los usuarios que posean el permiso necesario. Los datos son diferenciados en base a los distintos módulos de SaDER, como turnos registrados, cantidad de consultas registradas en la historia clínica electrónica, y mostrados en gráficos y tablas. Dentro de la sala de estadísticas se desarrolla el módulo de estadísticas de admisiones.

El módulo de estadísticas de admisiones se enfoca en la visualización de datos de admisiones registradas en las guardias de los efectores de salud de la provincia de Entre Ríos que utilizan SaDER. Las estadísticas de las admisiones son consideradas en base al relevamiento de requerimientos, teniendo en cuenta estadísticas relevantes de otras herramientas informáticas utilizadas en los efectores relevados.

Optando por dichas estadísticas, los siguientes datos son tenidos en cuenta para el

módulo:

- Cantidad de admisiones totales registradas: normales y de urgencias por separado
- Forma de ingreso a la guardia registrada en la admisión
- Tipo de guardia registrada en la admisión
- Cantidad de admisiones registradas con intervención policial
- Cantidad de consultas de historia clínica registradas a partir de la admisiones de guardia
- Fecha y hora de registro de admision
- Fecha y hora de registro de consulta médica

En primer lugar, se destaca que el apartado de estadísticas conlleva un formato específico, en el que se muestran tanto los gráficos como tablas con diseños visuales definidos, como tamaños y colores. Esto tiene como objetivo mantener una unicidad entre los distintos apartados dentro del módulo de estadísticas, pretendiendo que el usuario se focalice en los datos mostrados y no pierda su atención al notar un impacto en el cambio visuales, tales como colores, diseños o gráficos.

También es importante resaltar la diferenciación existente en la sala de estadísticas entre los usuarios autorizados del efecto, tales como el director del mismo, y aquellos usuarios del Ministerio que poseen permisos. La sala de estadísticas predefinida para los usuarios de los efectores únicamente exhibe la información registrada en dicho efecto, lo que permite un monitoreo y control de registros históricos específicos de la guardia correspondiente.

Por otro lado, la interfaz destinada a los demás usuarios presenta la información agregada de toda la provincia de Entre Ríos. En esta pantalla se incorporan filtros de ubicación especializados, que permiten restringir la información según los siguientes

criterios:

- **Región de la provincia:** La provincia se divide en cuatro regiones, clasificadas según los puntos cardinales.
- **Departamentos:** Subdivisiones administrativas dentro de cada región.
- **Localidades:** Municipios o ciudades específicas dentro de cada departamento.
- **Efector:** Instituciones de salud u otros efectores específicos.

La aplicación de estos filtros puede llevarse a cabo mediante la interacción directa con el mapa de la provincia o a través de las opciones de selección proporcionadas en los filtros mencionados. Este diseño facilita una navegación intuitiva y una segmentación precisa de los datos, permitiendo a los usuarios autorizados acceder a la información relevante de manera eficiente y efectiva.

La información se presenta en tres gráficos temporales diferenciados: anual, mensual y semanal. Esta división permite al usuario autorizado analizar los datos aplicando filtros en función del gráfico visualizado, seleccionando de manera precisa el año o mes de interés en el gráfico correspondiente.

Todos los gráficos aquí mencionados representan de forma detallada la cantidad total de admisiones registradas, desglosadas en dos categorías: admisiones normales y admisiones de urgencia.

Adicionalmente, se utiliza una tabla complementaria para mostrar estos datos de manera tabular. La tabla consta de dos columnas, donde una representa el tipo de dato y la otra la cantidad asociada. La columna de tipos de datos refleja la lista previamente mencionada. Este enfoque permite al usuario realizar un análisis más estructurado y detallado de los patrones de admisión a lo largo del tiempo, las formas de ingresos y tipo de guardia admitidas.

En cuanto a los gráficos se mencionan los de gráficos de línea para el apartado

anual (véase Figura 7.1) y mensual (véase Figura 7.2), ya que es comúnmente utilizado para visualizar la evolución de datos a lo largo del tiempo. De este modo, los usuarios pueden observar el comportamiento de las admisiones registradas en las guardias, detectando tanto picos como descensos a lo largo del eje temporal, sea el anual o mensual, pudiendo así tomar decisiones a partir de ello.

En cuanto a la Figura 7.1, se observan la cantidad de admisiones totales, admisiones de urgencia y admisiones normales a lo largo del año seleccionado. El eje horizontal corresponde a los meses del año y el eje vertical la cantidad de admisiones. De igual manera, la Figura 7.2 muestra datos similares, teniendo en el eje horizontal los días del mes seleccionado y en el eje vertical la cantidad de admisiones para cada día.

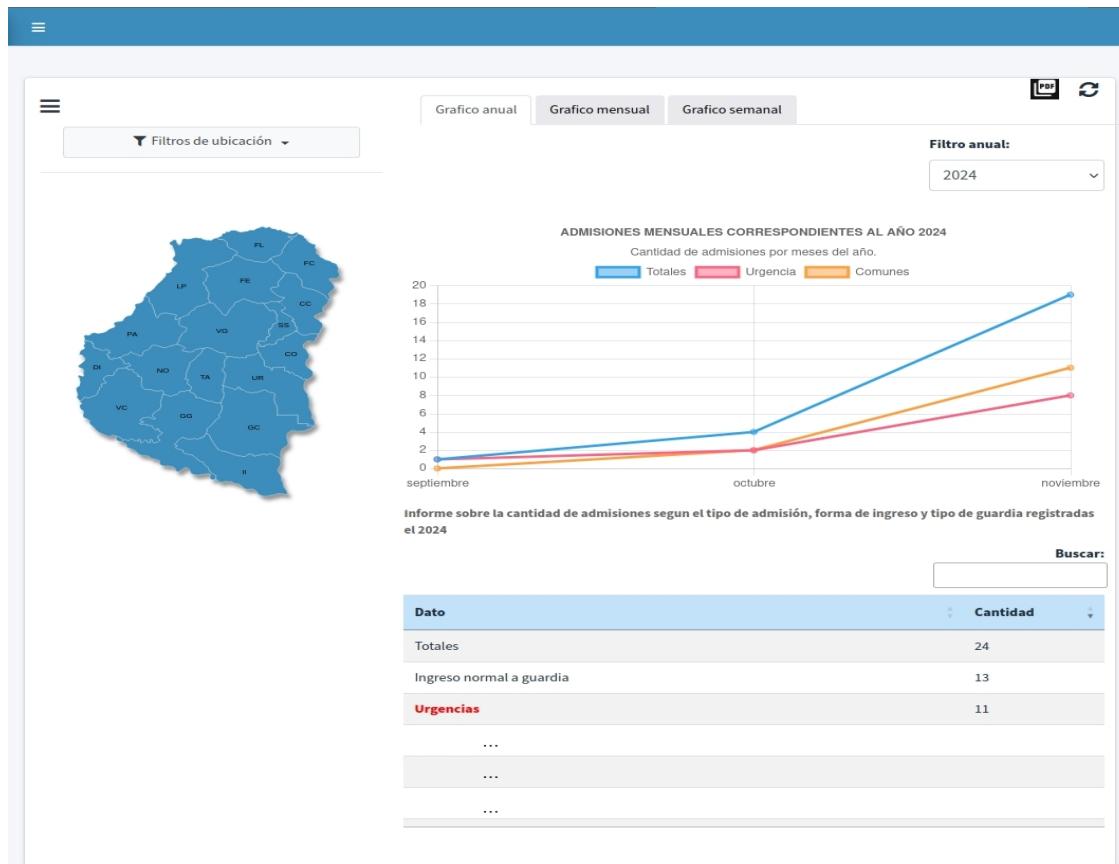


Figura 7.1: Pantalla de estadísticas anuales

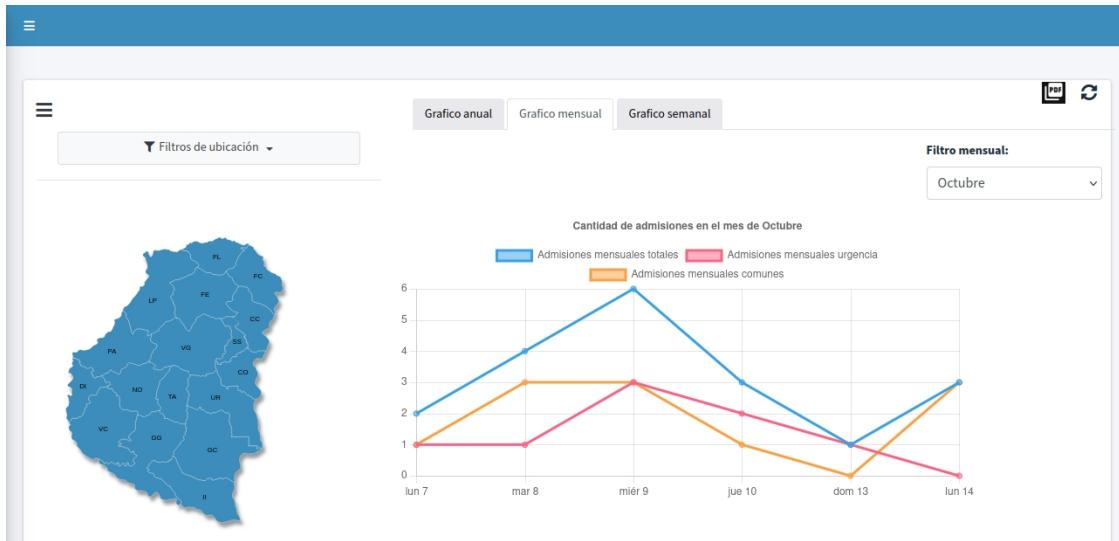


Figura 7.2: Pantalla de estadísticas mensuales

Por su parte, la interfaz de semanal Figura 7.3 utiliza un gráfico de barras. Los gráficos de barras son ampliamente utilizados para comparar diferentes categorías o grupos de datos de manera visual y clara. El eje horizontal corresponde a los días de la semana y el eje vertical la cantidad. En este gráfico se compara las admisiones de urgencias respecto de las admisiones totales, teniendo una aproximación visual del porcentaje entre ambas cantidades.

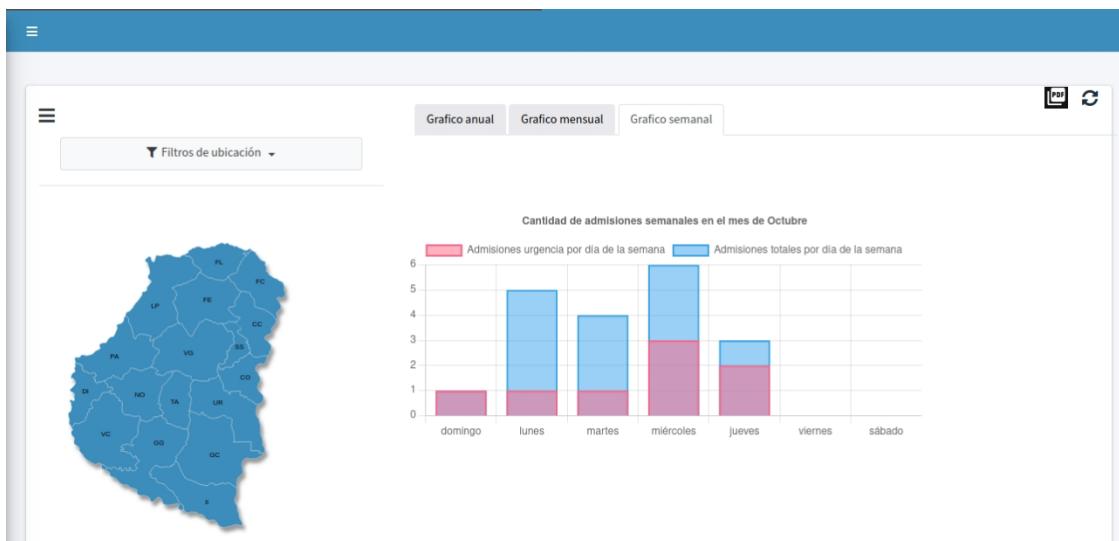


Figura 7.3: Pantalla de estadísticas semanales

En cuanto a la generación de reporte, el botón ubicado en la parte superior derecha de las pantallas Figura 7.1, Figura 7.2 y Figura 7.3 redirecciona a una nueva pestaña. En ésta se genera un vista de las estadísticas relevantes, la cual se puede optar por imprimir a través del botón ubicado en la parte superior izquierda.

	<h2>Estadísticas de Admisiones de Guardia</h2>			
Nombre del efector				
Información del período anual/mensual/semanal				
<u>Cantidad de admisiones</u>				
Totales:	Urgencias:	Normales:		
<u>Tipo de guardia</u>				
Adulto:	Pediatria:	Ginecología y Obstetricia:		
<u>Forma de ingreso</u>				
Total:	Caminando:	Ambulancia s/médico Ambulancia c/médico		
<u>Intervención policial</u>				
<u>Triaje</u>				
Nivel 1:	Nivel 2:	Nivel 3:	Nivel 4:	Nivel 5:
<u>Tiempos Promedios</u>				
Tiempo de espera	Tiempo de atención			

Figura 7.4: Pantalla de reporte de guardia

En la figura anterior se observa el diseño de la pantalla que accede el usuario al generar el reporte. El mismo tiene la particularidad de que, dependiendo de la pestaña en la que se encuentre, es decir, en el gráfico anual, gráfico mensual o semanal, se obtienen los datos e información de dicho periodo. El botón ubicado en la esquina superior izquierda permite imprimir el reporte generado.

7.2. Desarrollo del módulo estadístico

7.2.1. Desarrollo de frontend

El desarrollo de la sala de estadísticas comienza con la modificación de los componentes de dicho módulo, añadiendo los cambios nuevos del presente proyecto. El módulo de estadística se encuentra desarrollado con templates Django, cuya interfaz de usuario está modificada con jQuery. Esta configuración posee un conjunto de componentes de la siguiente manera:

- Un archivo HTML llamado “main”, el cual cumple la función de base de la plantilla o template de Django. Éste posee algunos componentes comunes en cada módulo de la sala de estadística. Entre ellos se encuentran la barra de navegación lateral, el mapa de la provincia de Entre Ríos, así como los filtros de ubicación.
- Un archivo Javascript “main”, que posee las funciones que brindan las interacciones de la interfaz de usuario. Dentro de las funcionalidades, la barra de navegación lateral indica cual es el componente HTML a renderizar, es decir, indican los gráficos a mostrar del módulo especificado.

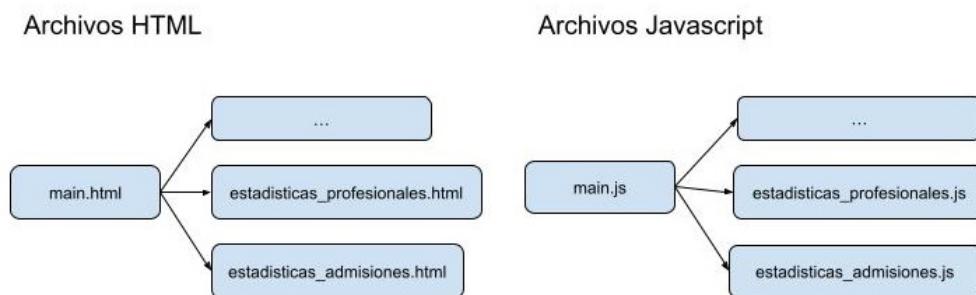


Figura 7.5: Configuración de archivos de frontend sala de estadísticas

A partir de esta configuración, se modifica ambos archivos, de modo que se contengan los nuevos componentes propios del módulo de guardia.

El archivo HTML de las estadísticas de guardia, es nombrado como 'estadisticas_admisiones.html', así como el Javascript, dado que el objeto principal en el que se basan los gráficos y las estadísticas son las admisiones de guardia.

En base al diseño de Figura 5.5, se codifican tres pestañas para las estadísticas de admisiones con un gráfico específico en cada una. El archivo Javascript modifica el gráfico, así como la tabla de estadísticas, en base a la pestaña en que se encuentra la interfaz. También se encarga de recibir los filtros seleccionados y aplicarlos a los datos, mostrando sólo aquellos filtrados.

La información necesaria para cada uno de los gráficos del módulo es obtenida en el componente 'main' al interactuar con la barra lateral, lo que realiza una petición al endpoint específico para las admisiones de guardia. Esta información es suministrada dinámicamente al gráfico correspondiente a mostrar y la tabla de estadísticas.

En cuanto a la generación de reporte de estadísticas, se toma como periodo de tiempo aquel seleccionado en la pestaña que se encuentre el usuario en el momento de querer generar el reporte. Es decir, si se encuentra en la pestaña anual, con un año específico, el reporte contendrá los datos de dicho periodo.

La pestaña del reporte se encuentra desarrollada con HTML, definiendo los estilos de letra, tipografía, tamaño y alineación de los mismos a través de CSS.

7.2.2. Desarrollo de backend

Se comienza la codificación de las funcionalidades de backend al definir los modelos de tablas. Estos modelos tienen similitudes a los utilizados anteriormente, pero el objetivo de su implementación es otro, dado que los datos, tanto de las tablas como de los gráficos, fueron obtenidos utilizando vistas materializadas.

En cuanto a las vistas materializadas, como se menciona en Sección 4.4, tienen

muchas ventajas, siendo la principal la velocidad y eficiencia al obtener datos. Se desarrolla el modelo “AdmisionMV”, el cual cumple la función de recibir y hacer coincidir los datos almacenados en la vista materializada con los atributos del modelo. Cabe destacar que el modelo solo realiza las acciones mencionadas anteriormente, delegando la responsabilidad y lógica de qué datos y cómo mostrarlos a la vista materializada.

Por otra parte, la interfaz de usuario posee un botón de actualización, el cual realiza una petición para actualizar la vista materializada con última información disponible al momento de realizar la petición.

El desarrollo prosigue con el desarrollo del endpoint al cual se solicita la información desde el apartado de frontend. En primera instancia, el endpoint está protegido por permisos, denegando cualquier acceso no autorizado. Si se encuentra autorizado el usuario que realiza la petición, la vista procede a formatear la información.

Para ello se tiene en cuenta una lista con tres objetos: tabla, curvas y filtros.

El objeto de la tabla contiene información de los siguientes campos, respetando un formato de nombre del dato y cantidad:

- Número total de admisiones registradas
- Número de admisiones de tipo urgencia
- Número de admisiones normales
- Cantidad de consultas de historia clínica registradas a partir de las admisiones de guardia
- Cantidad de admisiones registradas para cada tipo de guardia
 - Número de adulto
 - Número de pediatría
- Cantidad de admisiones que se registraron con intervención policial

El objeto de las curvas contiene información dividida según el tipo de gráfico y se dan en distintas listas para cada uno respectivamente. Para el gráfico anual y mensual se agrupan tres listas, la curva de total de admisiones, la curva de admisiones de urgencia y la curva de admisiones normales. Por otra parte, el gráfico semanal únicamente agrupa la información en admisiones totales y de urgencias, dado el tipo de gráfico que se utiliza.

Por último, el objeto de filtros. En las imágenes de los gráficos de Sección 7.1, se destacan un conjunto de filtros, cuya funcionalidad ya se encuentra desarrollada al momento de realizar este módulo. Sin embargo, se modifica la función para que reciba la información proveniente de la vista de admisiones y pueda realizar las acciones respectivas, retornando los nuevos datos filtrados.

7.3. Pruebas del módulo estadístico

A continuación, se detalla el diseño y los resultados de las pruebas de las funcionalidades del módulo de estadísticas. Se detallan las funciones, el diseño de las pruebas y el resultado de su realización.

7.3.1. Lista de funciones

A continuación se listan las funciones específicas del apartado, correspondientes al módulo de sala de estadísticas de este proyecto.

1. Función 8: Generación de reportes

El sistema permitirá a los usuarios con permisos generar reportes detallados de las actividades y datos de la guardia, seleccionando información del efector.

2. Función 9: Consultar estadísticas Los usuarios de un efector visualizarán solo

estadísticas de su efector, mientras que los usuarios del Ministerio de Salud podrán buscar información general de la provincia o específica de cualquier efector mediante la utilización de los filtros.

7.3.2. Pruebas de las funciones

1. Función 9: Consultar estadísticas

Identificación de la prueba	P011
Título	Validar gráficos y tablas del módulo de admisiones.
Descripción	Esta prueba se encarga de comprobar la obtención y generación de los gráficos y tablas para el apartado de guardia, dentro de la sala de estadísticas.
Pasos a ejecutar	<ul style="list-style-type: none"> a) Ingresar al apartado 'admisiones' de sala de estadísticas. b) Validar el gráfico anual y la tabla de datos. c) Hacer clic sobre la pestaña de 'Gráfico mensual'. d) Verificar el gráfico mensual. e) Hacer clic sobre la pestaña de 'Gráfico semanal'. f) Validar el gráfico semanal.
Resultado esperado	Se espera que al realizar las acciones en el orden establecido el sistema muestre los gráficos con sus respectivos datos.
Resultado	Exitoso

Observaciones	Las curvas y datos mostrados en los gráficos y tablas están sometidos a una primera versión propuesta para el proyecto, la limitación de los datos no permite mostrar más estadísticas, por lo que, en caso de requerirse, se debe optar por elegir otro tipo de gráfico o ayuda visual más adecuada.
----------------------	---

Tabla 7.1: Prueba P011

Identificación de la prueba	P012
Título	Verificar el funcionamiento de los filtros del módulo.
Descripción	Esta prueba se encarga de ingresar distintos tipos de filtros y comprobar el funcionamiento de los gráficos y las tablas a partir de ellos.
Pasos a ejecutar	<ul style="list-style-type: none"> a) Hacer clic sobre el botón de 'Filtros de ubicación' b) Seleccionar una región. c) Seleccionar un departamento. d) Seleccionar una localidad. e) Seleccionar un efector. f) Comprobar la corrección de los datos e información mostrada en cada uno de los gráficos y en la tabla.
Resultado esperado	Se espera que al realizar las acciones en su respectivo orden, tanto los gráficos como la tabla reflejen la información filtrada correctamente.
Resultado	Exitoso

Observaciones	Sin observaciones.
----------------------	--------------------

Tabla 7.2: Prueba P012

2. Funcion 8: Generación de reportes

Identificación de la prueba	P013
Título	Validar gráficos y tablas del módulo de admisiones.
Descripción	Esta prueba se encarga de comprobar la obtención y generación de los gráficos y tablas para el apartado de guardia, dentro de la sala de estadísticas.
Pasos a ejecutar	<ul style="list-style-type: none"> a) Hacer clic en el botón que genera el reporte. b) Verificar la autenticidad de los datos. c) Hacer clic sobre el botón de imprimir. d) Realizar la impresión del reporte. e) Validar la impresión.
Resultado esperado	Se espera que al realizar las acciones en el orden establecido el sistema muestre los gráficos con sus respectivos datos.
Resultado	Exitoso
Observaciones	El gráfico suele mostrar diferencias en los tamaños, tipos de letra y alineamiento de los objetos dependiendo del navegador web que se utilice al momento de generar el reporte.

Tabla 7.3: Prueba P013

8 | Conclusión

En este proyecto final de carrera se desarrolló e implementó un módulo de guardia médica dentro del sistema web SaDER. El módulo aborda la problemática de gestionar y monitorear una guardia médica a través de la utilización de herramientas de software, obteniendo información y datos de manera rápida y accesible.

En cuanto al desarrollo del proyecto es destacable mencionar el cumplimiento de los objetivos propuestos, desarrollando los módulos de registro de admisiones, clasificación de triaje automatizada mediante inteligencia artificial y visualización de estadísticas. Siendo tanto el primero como el último módulo implementados en el sistema web real como una herramienta digital, el cual, al momento de la realización de este informe, utilizan algunos efectores de la provincia de Entre Ríos. Cabe destacar que, en su mayoría, los efectores no utilizaban software para la gestión de las guardias médicas previamente a este proyecto, siendo delegadas todas las tareas de registro, control y administración a medios analógicos, como los informes en papel.

A continuación se destacan algunos de los beneficios e impactos identificados del proyecto.

En primer lugar, la optimización en el registro de admisiones y el orden de atención de guardia, al utilizar un sistema integrado al cual los involucrados en la guardia puedan acceder a la información de manera rápida, segura y confidencial, teniendo así más herramientas para la toma de decisiones.

Institucionalmente, también es beneficioso el complemento de la historia clínica electrónica con las admisiones, relacionando las atenciones recibidas en la guardia con la historia clínica del paciente, respetando así la ley de historia clínica electrónica.

Por otra parte, se contribuye con el medio ambiente, al lograr una disminución en la utilización de papel tanto para las tareas de los profesionales médicos y enfermeros, como la utilización de libros de guardias o la generación de reportes digitales.

En cuanto a los impactos sociales, mediante entrevistas y cuestionarios, ciertos efectores manifestaron su aprobación en cuanto a la mejora de eficiencia y organización de las consultas de los pacientes a partir del ordenamiento de las atenciones. Este impacto fue relevado en efectores de nivel de atención primaria, es decir, en centros de salud, donde la concurrencia de personas a las guardias no supera las cien personas diarias. Por lo que, se espera que el impacto de la organización en hospitales, departamentales y provinciales, utilizando de manera correcta la herramienta, sea aún mayor que en el nivel de atención primaria.

Es relevante señalar los beneficios económicos para los efectores de la provincia. Muchos centros médicos anexan diferentes sistemas digitales para cumplir con distintas tareas, entre ellos el sistema web SaDER, dentro de los cuáles sólo aquellos efectores con mayor presupuesto pueden afrontar el gasto de un software privado. Al brindar una herramienta para las guardias médicas totalmente gratuita beneficia a los de menor capacidad económica e incita a los demás a evitar gastos en adquirir software. Esta posibilidad sin costos impulsó la comunicación por parte de algunos usuarios, quienes realizaron comentarios y sugerencias de posibles mejoras dentro dichos módulos, las cuáles son descritas en Capítulo 9 Trabajos futuros.

Adicionalmente, algunos usuarios, tanto profesionales como efectores, declararon que comenzaron a utilizar el sistema web completo atraídos por el módulo de guardia médica. Esto demuestra un persistente cambio de paradigma en cuanto al trabajo de los profesionales de salud que comenzó hace años con la implementación de historia

clínica electrónica y que continúa incrementando ante la disponibilidad de nuevas herramientas digitales.

Un punto relevante es la omisión del módulo de inteligencia artificial, es decir, la automatización del triaje, en el entorno real de implementación. Esta decisión tiene sustento en la política de la organización que desarrolla el sistema, la cual se basa en documentos reales y legales para desarrollar módulos. Por lo que, a falta de un documento del procedimiento genérico que sea utilizado en todos los efectores para calcular el triaje, se optó por implementar únicamente los otros dos módulos. Sin embargo, cabe destacar que el módulo de triaje se encuentra funcionalmente disponible, siendo posible su utilización en futuros casos de pruebas o como base para versiones mejoradas. Esto se debe a que fue desarrollado basado en criterios ingenieriles, así como en beneficios y ventajas de sistemas implementados a nivel mundial. De modo que, el modelo permanece como un aporte por parte del integrante del proyecto o un recurso utilizable hasta que la organización gubernamental garantice las condiciones legales y operativas que permitan la implementación de la misma.

Durante el desarrollo del proyecto ocurrieron los siguientes riesgos: el retraso del cronograma y el cambio en los requerimientos.

El primer riesgo ocurrió en varias etapas del proyecto, siendo la más significativa el retraso al relevar los requerimientos, lo que dilató casi una semana la tarea. Para gestionar el impacto de este riesgo se optó por aumentar la cantidad de horas de trabajo. Paralelamente se comunicó con los interesados en el proyecto, con el fin de informar acerca de la situación del proyecto y la modificación de las próximas entregas. De esta manera, se logró finalizar la actividad sin mayores complicaciones.

En cuanto al otro riesgo ocurrido, las sugerencias de cambios de requerimientos se dieron durante el desarrollo del módulo de estadísticas. Estas sugerencias estaban orientadas al módulo de admisiones, específicamente los primeros dos puntos desta-

cados en Capítulo 9, por lo que fueron relevadas y documentadas adecuadamente, proponiéndolas para futuras versiones del sistema.

Desde el punto de vista personal del integrante del proyecto, la experiencia fue positiva y enriquecedora en varios sentidos. Principalmente en el sentido de lograr diseñar, desarrollar y probar un proyecto de software con un alcance de proyecto final de carrera, el cual requirió de la utilización e integración de muchos conocimientos previamente estudiados a lo largo de la carrera. Por otro lado, como experiencia al ver el impacto del proyecto al ser implementado en un entorno real, así como el aprendizaje profesional e individualmente logrado durante el transcurso y desarrollo del proyecto.

9 | Trabajos futuros

El desarrollo logrado en este proyecto presenta la propuesta inicial, la cual puede ser mejorada y expandida en futuras versiones. Las oportunidades de mejoras tienen el objetivo de ampliar y aportar beneficios no logrados en esta instancia, donde la retroalimentación de los usuarios juega un papel crítico.

Las ideas identificadas están consideradas por fuera de este proyecto, por lo que no se encuentran en el alcance del mismo. Algunas de estas propuestas fueron identificadas por el autor del proyecto, mientras que otras fueron sugeridas por los usuarios del sistema web SaDER. Las mismas aportan valor al sistema, ampliando las herramientas y funcionalidades, así como mejorando la experiencia de los usuarios.

A continuación se destacan algunas propuestas o ideas a ser consideradas en trabajos posteriores.

- Listado de admisiones finalizadas durante el último día.

Dentro del módulo de admisiones, la interfaz gráfica permitirá acceder al listado de las admisiones finalizadas durante las últimas veinticuatro horas del efecto. Esto permitirá al personal de los centros clínicos un acceso rápido y sencillo al registro, manteniendo un control de las actividades, así como del trabajo del personal, llevadas a cabo durante el último día.

- Admisiones de personas no identificadas o extranjeras.

Las admisiones, así como el registro de historia clínica electrónica, se encuentra restringido a personas afiliadas registradas en el sistema web, es decir, personas

con documentación argentina. Si bien las admisiones de urgencia no requieren obligatoriamente un documento, poseen la desventaja de que el personal médico y de enfermería no puede registrar la historia clínica en los mismos.

Por ello se propone modificar la lógica de registro de personas, ampliando el alcance para aquellas personas de origen extranjero a Argentina y personas que, dada la situación específica, no posea o no sea capaz de entregar el documento nacional. Esta propuesta agrega un valor significativo, ya que permitirá a los profesionales de la salud registrar la historia clínica de los pacientes no afiliados al sistema, agregando valor al mismo sistema SaDER, el cual no cuenta con dicha posibilidad de registro.

- Registro y estadística de todos los tiempos promedios.

La modificación consiste en registrar todos los tiempos promedios significativos para un control de calidad del servicio de guardia. Dado que en el informe solo se contempló únicamente el registro de la admisión y de la primera consulta médica cargada, es decir, el tiempo de espera del paciente.

Se propone adicionar los mecanismos necesarios para registrar el tiempo de respuesta del servidor al calcular el triaje, lo que contempla el tiempo de transmisión de los datos junto con el que le toma al modelo realizar la predicción. De igual manera con el tiempo ocurrido desde la admisión hasta el cálculo del triaje, el cual tiene como inicio el registro de admisión y como final la respuesta del servidor, siendo el cálculo automático, o el registro de los datos y signos vitales del paciente, cuando el triaje es ingresado manualmente.

El valor de esta idea recae en que permitirá a los efectores tener un control de calidad del servicio de guardia, pudiendo identificar tiempos anormales y áreas para mejorar el proceso de atención de los pacientes.

- Evaluación en profundidad de los modelos predictivos implementados con métricas más adecuadas para cada modelo.

La evaluación de un modelo de inteligencia artificial es crucial para comprender la calidad de su entrenamiento y de las predicciones que logrará. Para ello se utilizan distintas métricas, las cuales brindan distinta información. Algunos modelos son más adecuados a ser evaluados con ciertas métricas en específicas, por lo que se propone realizar un exámen exhaustivo con más de una métrica por modelo. Esta evaluación permitirá no solo un mejor entendimiento de cómo las entradas influyen en cada modelo, sino que también pueden cambiar la elección sobre el modelo seleccionado para realizar la automatización del triaje. Esta propuesta también abarca la posibilidad de implementar nuevos campos dentro del formulario que son de relevancia para un sistema de triaje, tales como el problema o síntoma del paciente.

- Implementación de predicciones de triaje con inteligencia artificial en el sistema de producción.

La automatización del triaje en el sistema de producción posibilita beneficios como la reducción de riesgo de errores humanos y la disminución de la carga laboral del personal médico. Sin embargo, la implementación de un modelo de inteligencia artificial en un entorno real de producción requiere de un análisis exhaustivo, tanto a nivel de sistemas como jurídicos y legales. Por lo que se propone realizar una implementación en un entorno controlado, como un efecto dispuesto a realizar las pruebas del software, donde se pueda evaluar el impacto del modelo en la toma de decisiones del personal médico y de enfermería, así como la calidad del modelo de inteligencia artificial.

- Monitoreo de la ubicación del paciente dentro de la guardia.

La tabla de admisiones de guardia posee información relevante para ordenar las atenciones a brindar a los pacientes. En efectores que posean instalaciones físicas de grandes magnitudes, donde una sala se encuentre separada físicamente de otras, pero que se realicen trabajos en conjunto es crucial brindar información

de la ubicación del paciente. Un ejemplo de ello puede ser una sala de enfermería o médica apartada de la sala de estudios, donde la comunicación entre el personal no es directa. Por lo que aquí el servicio de guardia se puede hacer las siguientes preguntas: ¿El paciente se encuentra en la sala de espera? ¿El paciente aún permanece en la sala de estudio o se realizó efectivamente el estudio? ¿El paciente volvió a la sala de enfermería a realizarse los tratamientos indicados por el médico? Adicionalmente, se destaca la utilidad de la tabla, la cual puede ser mostrada en una pantalla dentro de la sala de espera, lo que no solo mantiene informado al personal del efecto, sino a personas que acompañen a los pacientes.

Esta propuesta agrega valor al sistema, permitiendo un mejor control y seguimiento de los pacientes, así como una mejor comunicación entre el personal médico y de enfermería. Estas mejoras son cruciales en efectores de gran tamaño, donde la comunicación entre los profesionales no es directa, y puede lograrse de manera efectiva y sencilla a través de la mejora del software aquí desarrollado.

- Consulta del estado actual de la guardia de otro efecto.

Esta herramienta permitirá a los usuarios de un efecto o centro clínico ver información destacable del estado actual de la guardia de otro efecto. Entre la información a compartir, en una primera instancia, se destaca la cantidad de admisiones que posee, las admisiones finalizadas, así como el tiempo de espera y de atención promedio. El beneficio de esta implementación recae en que situaciones donde un efecto tenga una demanda por encima de sus capacidades con respecto a la guardia médica, pudiendo así derivar a pacientes no registrados a otros efectores cercanos en base a sus estados actuales. De igual manera, el alcance de la herramienta puede aplicarse desde los propios vehículos del efecto hasta los sistemas de emergencias médicas que coordinan los trasladados de las ambulancias.

Referencias

- Analytics Lane. (2018). *La regresión logística* (n.º 23). <https://www.analyticslane.com/2018/07/23/la-regresion-logistica/>.
- AWS. (2024a). *¿qué es el aumento de datos?* (n.º 28). <https://aws.amazon.com/es/what-is/data-augmentation/>.
- AWS. (2024b). *¿qué es la regresión logística?* (n.º 24). <https://aws.amazon.com/es/what-is/logistic-regression/>.
- Busso, L., Bellavita, A., Heim, S., y Maidana, C. (2019). ImplementaciÓn del sistema triaje en los hospitales pÚblicos de la provincia de buenos aires. (4).
- Castello, D., Bollo, C., Gauna, C., Montes, I., y Vargas, C. R. (2010). Políticas de software libre en el estado argentino. *Universidad Nacional de Córdoba*(10), 3.
- Congreso de la Nación Argentina. (2022). *Ley 27.706: Ley de presupuesto general de la administración nacional para el ejercicio 2023* (n.º 29). <https://www.argentina.gob.ar/nORMATIVA/nACIONAL/LEY-27706-380710/TEXTO>. (Publicada en el Boletín Oficial el 26 de diciembre de 2022)
- Czubaj, F. (2023). “está en riesgo la seguridad del paciente” Éxodo de médicos y servicios que bajan su nivel en un sistema sanitario que cruce (n.º 7). <https://www.lanacion.com.ar/sociedad/esta-en-riesgo-la-seguridad-del-paciente-exodo-de-medicos-y-servicios-de-menor-nivel-en-un-sistema-nid18052023/>. La Nación.

Referencias

- Geclisa. (2025). *Geclisa* (n.º 12). <https://geclisa.com/>.
- HSI. (2025). *Historia de salud integrada* (n.º 9). <https://hsipladema.net/>.
- IBM. (2024a). *¿qué es el random forest?* (n.º 25). <https://www.ibm.com/mx-es/topics/random-forest>.
- IBM. (2024b). *¿qué son las máquinas de vector de soporte (svm)?* (n.º 26). <https://www.ibm.com/es-es/topics/support-vector-machine>.
- INDEC. (2022). *Censo nacional de población, hogares y viviendas 2022. cuadro 1.8. provincia de entre ríos. población total en viviendas particulares. cobertura de salud según departamento* (n.º 1).
- Intel. (2025). *Artificial intelligence (ai) servers* (n.º 30). <https://www.intel.com/content/www/us/en/learn/ai-servers.html>.
- ITColmena. (2009). *It colmena group s.r.l.* (n.º 11). <https://www.itcolmena.com/Castellano/index.html>.
- Jiménez, J. G. (2003). Clasificación de pacientes en los servicios de urgencias y emergencias: Hacia un modelo de triaje estructurado de urgencias y emergencias. (13).
- Manual para la implementaciÓn de un sistema de triaje para los cuartos de urgencias* (n.º 5). (2010). Washington, DC.
- Ministerio de Salud de la Nación. (2023). *Total de beneficiarios inscriptos al programa sumar con cobertura efectiva básica (ceb)* (n.º 2). <https://datos.gob.ar/ca/dataset/salud-total-beneficiarios-inscriptos-al-programa-sumar-con-cobertura-efectiva-basica-ceb>.
- Neuroons. (2020). *Razones para incorporar la inteligencia artificial a tu empresa* (n.º 15). <https://neuroons.com/es/razones-incorporar-inteligencia-artificial-empresa/>.
- NexusIntegra. (2023). *Ventajas y desventajas de la inteligencia artificial en em-*

Referencias

- presas (n.º 14). <https://nexusintegra.io/es/ventajas-y-desventajas-de-la-inteligencia-artificial/>.
- NVIDIA. (2025). *Multi-node deep learning training with tensorflow* (n.º 31). <https://docs.nvidia.com/ai-enterprise/deployment/multi-node/latest/requirements.html>.
- Nöllmann, M. (2024). “*todo es demasiado estresante*”. *la fuga de médicos de guardia en hospitales profundiza la crisis sanitaria* (n.º 6). <https://www.lanacion.com.ar/sociedad/todo-es-demasiado-estresante-la-fuga-de-medicos-de-guardia-en-hospitales-profundiza-la-crisis-nid17022024/>.
- ObservatorioCiudad2024. (2024). *Informe sobre el impacto en los aumentos de las prepagas de la resolución nº 1/2024* (n.º 18). <https://observatoriociudad.org/informe-sobre-el-impacto-en-los-aumentos-de-las-prepagas-de-la-resolucion-n-1-2024/>.
- Pagina12. (2024). *Más demanda en hospitales públicos del conurbano* (n.º 16). <https://www.pagina12.com.ar/727339-mas-demanda-en-hospitales-publicos-del-conurbano>.
- PostgreSQL. (2025). *Materialized views postgresql* (n.º 21). www.postgresql.org/docs/current/rules-materializedviews.html.
- ProgramaSumar. (2024). *¿qué es el programa sumar?* (n.º 3). <https://www.argentina.gob.ar/salud/sumar>.
- SNOMED. (2025). *What is snomed ct?* (n.º 19). <https://www.snomed.org/what-is-snomed-ct>.
- UnoEntreRios. (2024). *Hospitales: Aumentan las consultas por subas en prepagas* (n.º 17). <https://www.unoentrerios.com.ar/la-provincia/hospitales-aumentan-consultas-subas-prepagas-n10111205.html>.
- Vítolo, D. F. (2014). *SaturaciÓn de las guardias de emergencia parte ii. posibles*

Referencias

- soluciones* (n.º 8). Noble Compañía de Seguros.
- Yildiz, I. (2021). *Emergency service - triage application (kaggle)* (n.º 27).
<https://www.kaggle.com/datasets/ilker-yildiz/emergency-service-triage-application>.

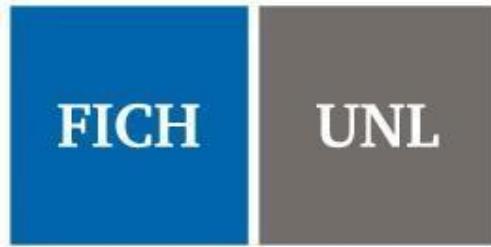
Bibliografía

- Project Management Institute. (2017). A Guide to the Project Management Body of Knowledge (PMBOK® Guide) (6th ed.). Newtown Square, PA: Project Management Institute.
- Gómez Jiménez, J., Ramón-Pardo, P., & Rua Moncada, C. (2010). Manual para la implementación de un sistema de triaje para los cuartos de urgencias. Organización Panamericana de la Salud/Organización Mundial de la Salud (OPS/OMS).
- Busso Leonardo, Bellavita Adriano, Heim Sigrid, Maidana Clara. (2020). Implementación Del Sistema Triaje En Los Hospitales Públicos De La Provincia De Buenos Aires.
- Dr. Castello, Cr. Bollo, Cr. Gauna, Ing. Montes, Cr. Rocha Vargas (2010). Políticas de software libre en el estado argentino. Universidad Nacional de Córdoba.
- J. Gómez Jiménez (2003). Clasificación de pacientes en los servicios de urgencias y emergencias: Hacia un modelo de triaje estructurado de urgencias y emergencias.
- George Kipourgos, Anastasios Tzenalis, Vasiliki Diamantidou, Constantinos Koutsojannis, Ioannis Hatzilygeroudis (2022). An Artificial Intelligence Based Application for triage Nurses in Emergency Department, Using the Emergency Severity Index Protocol.
- Study Shervin Farahmand, Omid Shabestari, Meghdad Pakrah, Hooman Hosseini-

Referencias

- Nejad, Mona Arbab, Shahram Bagheri-Hariri (2017). Artificial Intelligence-Based triaje for Patients with Acute Abdominal Pain in Emergency Department; a Diagnostic Accuracy.
- Honorable Congreso de la Nación Argentina. (2023). Ley 27706: Programa Federal Único de Informatización y Digitalización de Historias Clínicas de la República Argentina.
 - James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning with applications in R. Springer.
 - Guillén, C. B. (2018). Manual de urgencias (3ra ed.). Médica Panamericana.

Anexos



UNIVERSIDAD NACIONAL DEL LITORAL
Facultad de Ingeniería y Ciencias Hídricas

**PROPUESTA DE PROYECTO FINAL DE CARRERA
INGENIERÍA INFORMÁTICA**

**Desarrollo de módulo de guardia ambulatoria con triaje
automatizado mediante inteligencia artificial en el sistema web
SaDER.**

Alumno: Andino, Daniel Agustín

Director: María Agustina Carignan

Co-Director:

Asesor temático:

Santa Fe, Agosto de 2024

Índice

Resumen	3
Palabras Claves	3
Justificación	3
Objetivo	6
Alcance	7
Metodología	8
Plan de tareas	13
Cronograma de actividades	17
Puntos de seguimiento	19
Riesgos	20
Recursos	24
Presupuesto	25
Plan de comunicaciones	28
Bibliografía	30
Referencias	31

Resumen

La situación económica en Argentina se mantiene en caída desde la pandemia de COVID-19, ocurrida en el año 2020. Esto se ve reflejado en muchos sectores, como el de salud. A partir de la modificación de la Ley N° 26682 “Marco Regulatorio de Medicina Prepaga”, las cuotas de obras sociales y servicios de medicina prepagas privados aumentaron en más de un 30 % a comienzos de 2024, por lo que resultan costosas para gran sector de la población del país [18]. Como consecuencia, la salud pública ha experimentado un aumento en la cantidad de pacientes que reciben atención año a año. Este aumento conlleva que los centros médicos se encuentren cercanos al límite o por encima de sus capacidades de funcionamiento, teniendo salas de esperas sobrecargadas de pacientes, profesionales fatigados por exceso de trabajo y uso desmedido de recursos e insumos públicos [16].

La salud pública en la Argentina carece de la infraestructura y de la cantidad de recursos necesarios para atender a la demanda del público [17]. Esta falencia es notable, principalmente, en el sector de las herramientas informáticas que utilizan los profesionales para realizar su trabajo. En el año 2023 se aprobó la Ley 27706, en la cual se dicta que las historias clínicas pasarán a ser digitales, de modo que las atenciones deben mantener un registro informático. Como consecuencia de esta ley, cada centro médico debe autogestionarse, es decir, se debe hacer responsable de obtener el software necesario para llevar a cabo sus tareas .

Dado este contexto, se plantea como propuesta de proyecto final de la carrera Ingeniería en Informática el desarrollo de un módulo de guardia ambulatoria, el cual incluye la clasificación automática de niveles de urgencia en los pacientes. Este módulo será específico para el sistema web que propone el Ministerio de Salud de la Provincia de Entre Ríos de Argentina.

Para el desarrollo del proyecto se propone una metodología incremental con diez fases, cuyos entregables aportan un incremento en cuanto a las nuevas funcionalidades del sistema. Este desarrollo se llevará a cabo en las instalaciones del ente impulsor.

El proyecto se realizará durante cuatro meses y una semana, comenzando en agosto de 2024 y finalizando a principios de diciembre de 2024, según el calendario establecido. Se destinarán cuatro horas diarias de trabajo. También se contará con un responsable de realizar las tareas de desarrollo, siendo el alumno que propone el proyecto final, así como tres profesionales: dos asesores y un director del proyecto.

En cuanto a los impactos del proyecto, se destacan dos aspectos clave: Por un lado, se automatizarán tareas como el triaje, lo que contribuirá a reducir posibles riesgos humanos y disminuirá la carga laboral del personal médico. Por otro lado, se mejorará la asistencia al médico durante la consulta, permitiéndole

acceder de manera rápida y efectiva tanto a la información actual como a la histórica del paciente, todo dentro de un único sistema.

Adicionalmente, se destacan beneficios como la capacidad de monitoreo en tiempo real y la obtención de estadísticas actuales e históricas, tanto para el Ministerio como para cada efector individualmente.

Palabras Claves

Aplicación Web - Triaje - Guardia Ambulatoria - Ley 27706 - Entre Ríos - Historia Clínica Electrónica.

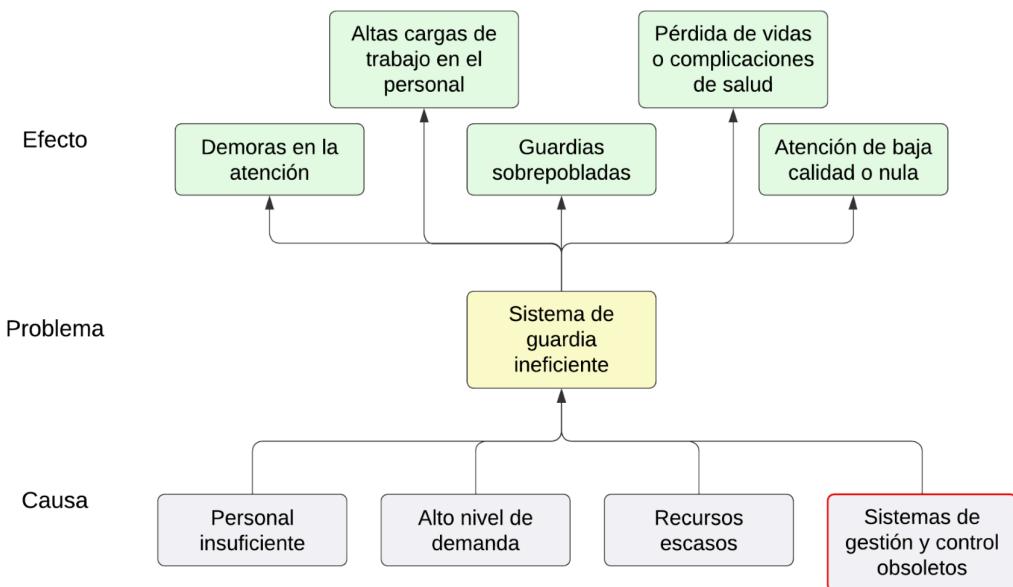
Justificación

El Censo Nacional de Población, Hogares y Viviendas del año 2022 reveló que en la provincia de Entre Ríos un 38,2 % de la población tiene cobertura social proveniente de planes o programas de salud estatales o que no posee ningún tipo de cobertura [1]. Entre estos programas se destaca Sumar. El Programa Sumar es una política pública que promueve un acceso equitativo y de calidad a los servicios de salud para toda la población que no posee cobertura formal en salud [2]. Según datos del mes de junio de 2023, la cantidad de inscriptos al programa Sumar en la provincia de Entre Ríos es del 34,4 % de la población total [3]. A este número se le deben sumar aquellas personas que, aun teniendo cobertura, prefieren recibir atención en centros públicos.

Entre los distintos sectores de los centros clínicos, se destacan las guardias. Las guardias hospitalarias se consideran como la puerta de entrada de los pacientes a los efectores y el corazón de los hospitales, ya que son los lugares por donde más gente circula y donde se atienden los casos de urgencia. Por este motivo, la planificación y ejecución de una resolución efectiva e integral a los problemas en la atención resulta esencial, habida cuenta de la relevancia sanitaria [4]. Generalmente, el proceso de atención comienza con la admisión, se registra al paciente y se le toman datos. De este proceso se deriva el triaje, que consiste en una clasificación de los pacientes en diferentes niveles de urgencia o priorización de la asistencia [5].

La problemática de las guardias médicas, tanto a nivel país como de la provincia, tienen orígenes en sectores sociales, políticos y económicos, cuyas causas son numerosas. En la Figura 1 se mencionan algunas de ellas. En primer lugar, se destaca la falta de profesionales de la salud para cubrir los turnos de guardias, lo que lleva a repetir plazas, sobrecargando de trabajo a estos profesionales o dejando puestos vacíos. El alto nivel de demanda ocasiona un desgaste físico y mental en los profesionales, que exige la realización de descansos y, en consecuencia, se producen congestionamiento y demoras en las salas de espera, así como un deterioro en el desempeño laboral. También se ha reportado que los centros públicos atienden con pocos recursos, incluso registrando

situaciones extremas donde no se poseen recursos básicos para realizar las atenciones. Estas causas repercuten en una menor calidad y cantidad de atención a los pacientes, en demoras para recibir la atención, entre otros efectos [6] [7] [8].



(Figura 1. Árbol de problemas)

Con respecto a las guardias en la provincia de Entre Ríos, en el año 2020, durante el cual la pandemia de COVID-19 impactó en todo el mundo, se registró un total de 1 922 288 consultas ambulatorias (Ministerio de Salud, 2022). De este número, un 27 % aproximadamente (518 808) corresponde a consultas realizadas en emergencias/urgencias. Ante la necesidad de atender esta cifra de demanda, el sistema informático Salud Digital Entre Ríos o (SaDER) brinda herramientas informáticas a los centros médicos de la provincia de manera gratuita. Este sistema fue impulsado por la Ley 27706, la cual exige que los profesionales y auxiliares de la salud deben registrar toda intervención médico-sanitaria en la historia clínica electrónica; en otras palabras, toda consulta médica debe ser registrada en sistemas digitales.

El sistema SaDER cuenta con más de 500 efectores registrados a comienzos de 2024, muchos de los cuales poseen guardia. Estos centros médicos, actualmente, no pueden cumplir con sus tareas de guardia de forma digital de manera óptima, dado que SaDER no posee un apartado enfocado a las guardias ambulatorias. Para suplir esta necesidad, algunos efectores han optado por utilizar el sistema de turnos de ambulatoria, también propio de SaDER, para realizar la atención en guardias. Sin embargo, este módulo del sistema no está especializado para las necesidades que requiere un software de urgencia, haciendo ineficiente el trabajo del personal.

Una de las principales consecuencias de las falencias mencionadas es la pérdida de tiempo: en las guardias se prioriza la rápida atención y el módulo de ambulatoria requiere de pasos obligatorios para registrar la consulta del paciente, de modo que parte del tiempo de atención del paciente se utiliza en brindar los datos al sistema. Otros centros médicos revelaron que utilizan herramientas, tales como planillas digitales o impresas, donde es común cometer errores o perder información. Esta situación evidencia la ausencia de un sistema unificado en las guardias, además de que las herramientas tecnológicas empleadas son ineficientes y obsoletas.

De esta manera, se evidencia que uno de los principales problemas por resolver es que el Ministerio de Salud, como organización responsable de cubrir las necesidades de los efectores, no brinda un sistema digital para las guardias. Sistema que debe estar adaptado, exclusivamente, para suplir las necesidades de las guardias y mitigar al máximo los riesgos posibles por parte de los profesionales como consecuencias de los problemas mencionados previamente.

Como alternativas de solución existen sistemas informáticos de guardia en el mercado, algunos de los cuales poseen triaje. Se destaca el de la Historia de Salud Integrada (HSI), que es una herramienta gratuita promocionada por el Ministerio de Salud a nivel nacional [9]. HSI, al igual que SaDER, es un sistema web, que brinda distintos módulos para administrar centros médicos, entre ellos, el de guardia que posee triaje. Considerando esta posibilidad de utilizar sólo el módulo de guardia del HSI, los usuarios se encuentran con una disyuntiva: utilizar un único módulo de un sistema nuevo para una tarea específica y emplear SaDER para las demás, o utilizar el sistema HSI por completo. La primera opción conlleva todas las desventajas de no unificar un sistema para una misma organización, como la incompatibilidad de datos en cada sistema; la segunda opción implica un mayor aprendizaje y capacitación por parte del personal, lo que conlleva pérdida de tiempo y esfuerzo, y aumenta la probabilidad de confundir al usuario y que cometa errores. Por otra parte, la clasificación de triaje en HSI debe hacerse siempre de forma manual. Al no tener automatizada esta tarea, dentro de este sistema pueden ocurrir riesgos relacionados con las causas mostradas en la Figura 1. Un ejemplo de esto es una incorrecta clasificación en el nivel de urgencia, que desencadene un tiempo de espera mayor por parte del paciente, dependiendo del nivel de prioridad de atención, y que se agrave la situación del mismo. Por lo tanto, se descarta al HSI como una solución viable para el problema planteado por dos motivos principales. Primero, carece de interoperabilidad con otros programas de salud, lo que limita su efectividad en un entorno integrado. Segundo, no cuenta con la automatización del proceso de triaje, el cual es fundamental que se realice correctamente, sin importar el entorno o las circunstancias en las que se lleve a cabo.

Otras alternativas son los sistemas adquiridos bajo licencias comerciales, las cuales -salvo excepciones- sólo permiten al comprador ejecutar los programas, vedando el acceso al conocimiento de las instrucciones de los programas y, por ende, la libre adaptación de los productos a las necesidades del ente [10]. Como una alternativa para solucionar el problema se puede considerar la adquisición de

un software privado, corrigiendo así la falta de herramientas digitales en las guardias. Sin embargo, debido a la variabilidad en la organización y funcionamiento de los centros clínicos, adaptar los softwares privados a las necesidades específicas de cada caso resultaría complicado. Además, surgen otros obstáculos como la falta de acceso al código fuente, costos adicionales para modificaciones, o la negativa de las empresas a realizar cambios para preservar un software estándar. Ejemplos de estos software son Colmena [11], utilizado en algunos efectores de la provincia de Entre Ríos, o Geclisa [12].

El propósito de este proyecto es brindar un apartado de guardia especial para el sistema SaDER, que cuente con triaje automatizado por Inteligencia Artificial (IA). Este módulo cumplirá con las funcionalidades requeridas y estará focalizado en mejorar la productividad y la comodidad de los profesionales al disminuir el riesgo humano. De esta manera, se logrará la mejora, adaptación, innovación y mantenimiento para próximas versiones.

Cabe destacar que el principal motivo de realizar el proyecto sobre el sistema SaDER recae en la dependencia laboral del responsable de este proyecto con el organismo impulsor del sistema, en este caso, el Ministerio de Salud de la Provincia de Entre Ríos.

Entre los beneficios e impactos del proyecto propuesto se destaca la informatización del registro de las consultas en las guardias, respetando lo impuesto en la Ley 27706. También se mejora la seguridad y confidencialidad con la que es tratada la información de los pacientes.

La implementación de la Inteligencia Artificial (IA) conlleva beneficios al disminuir los riesgos de errores humanos. Se automatizan tareas o procesos: las IA pueden ejecutar tareas repetitivas y optimizar procesos sin necesidad de la intervención humana. A su vez, la clasificación de los pacientes se ha de hacer en un tiempo corto, de forma ágil y efectiva [13]. En otras palabras: el triaje se debe realizar de manera correcta, independientemente del ambiente y la situación, así como del estado y de las capacidades de los profesionales de guardia. Al implementar inteligencias artificiales en el sistema, se asiste al personal prediciendo los niveles de urgencias, cumpliendo así con la condición mencionada anteriormente, dado que las inteligencias no sufren los riesgos de errores humanos mencionados [14] [15]. Las IA también disminuyen el trabajo de los profesionales al automatizar la tarea de triaje e informatizar la comunicación de información, de forma que se privilegia el tiempo y los recursos. Otra ventaja de los sistemas digitales recae en la asistencia al médico durante la consulta, al obtener datos actuales del paciente, así como históricos, de manera rápida y efectiva, dentro de un mismo sistema, independientemente del centro médico en el que el paciente reciba la atención.

El proyecto también impacta en algunos sectores específicos.

- Ecológicamente: al reducir el uso de papel en los centros médicos y de insumos administrativos, como impresoras, tintas y lapiceras, entre otros, se

- facilita la migración de las atenciones basadas en fichas de papel hacia un sistema digital
- Económicamente: al utilizar un sistema de gestión gratuito, se evitan costos de licencias privadas en los centros médicos, así como aquellos gastos que se economizan al reemplazar historias clínicas impresas por un sistema digital.
- Socialmente: al mejorar la organización y eficiencia de las consultas a partir del orden de atención, se reduce el tiempo de espera del paciente desde su ingreso al centro de salud hasta su atención por parte del médico; se prioriza el tratamiento de los casos más urgentes, disminuyendo la pérdida de vidas, así como los posibles riesgos de salud por no recibir la atención a tiempo. También se mejora la comunicación entre el personal de guardia por la obtención de información precisa en cada momento.

Por otra parte, impacta en el Ministerio de Salud y en cada efecto usuario de SaDER, ya que los datos de las guardias podrán ser visualizados a través de estadísticas, así como ser utilizados en futuros informes, propuestas, tomas de decisiones, monitoreo en tiempo real por parte de los directivos, entre otros propósitos.

Objetivo

Objetivo general

Desarrollar un módulo de guardias ambulatorias con triaje automatizado mediante inteligencia artificial dentro del sistema web SaDER.

Objetivos específicos

- Desarrollar un módulo de admisión, donde se inicie el proceso de atención, ingresando los datos, síntomas y signos del paciente.
- Desarrollar un módulo de triaje, que clasifique el nivel de urgencia de la atención del paciente a partir de una inteligencia artificial.
- Desarrollar un módulo de visualización estadística de datos de la aplicación de guardia ambulatoria.

Alcance

El sistema soportará la inicialización del proceso de atención de las siguientes situaciones en las que el paciente ingrese al centro médico:

- Ingresa consciente y es capaz de brindar la totalidad de la información que requiera el usuario clínico.
- Ingresa inconsciente, pero un acompañante puede brindar información.

El proyecto no soportará otras posibles situaciones, las cuales serán consideradas para futuros desarrollos en próximas versiones del sistema en el cual será implementado. La inicialización de una nueva atención se llevará a cabo cuando el usuario clínico ingrese los datos del paciente.

En cuanto al usuario clínico, será éste quien indique al sistema si requiere de la asistencia del mismo para el triaje. Si no requiere asistencia, deberá ingresar la clasificación manualmente. Siendo el caso afirmativo, el sistema realizará la clasificación y la opción deberá ser aprobada por el usuario encargado o por el médico responsable. Si la opción que brindó el sistema fue rechazada, alguno de los agentes nombrados deberá ingresar la clasificación pretendida.

El sistema actual SaDER posee un apartado especial para análisis estadísticos de distinta índole. Para este proyecto, el módulo encargado mantendrá un registro histórico de los tiempos entre:

1. Ingreso de los datos del paciente y la respuesta de triaje. Este tiempo servirá para que cada efector pueda controlar y analizar su capacidad y calidad operativa.
2. Inicio del pedido de clasificación y la respuesta de triaje. Este tiempo será de utilidad para evaluar el rendimiento del clasificador.
3. Ingreso de los datos del paciente e inicio de la consulta por parte del médico (excepcionando los casos de urgencia).

También guardará el nivel de la clasificación brindada por el sistema, en cada nuevo ingreso de paciente, y si fue aprobada o no.

En cuanto a la clasificación de los niveles de urgencia, se utilizará un modelo de inteligencia artificial, automatizando la tarea de decisión de triaje.

En cada efector se visualizará la información en tiempo real, es decir, los casos que han ocurrido en las últimas 24 horas. Los usuarios que tengan permisos de encargado de un efector tendrán acceso a la información histórica de la guardia. Con estos objetivos, se pretende que el proyecto cumpla con los siguientes requisitos no funcionales:

- Rapidez de respuesta: en situaciones donde el tiempo es crucial al atender al paciente, no se aceptará que se pierda tiempo esperando respuestas largas por parte del sistema.
- Estable en las clasificaciones de triaje: ante situaciones donde los datos ingresados son similares a las clasificaciones no habrá una diferencia mayor a uno entre ambos niveles.
- Software sostenible: el código será fácil de sostener, permitiendo una rápida y sencilla implementación de nuevas funcionalidades..
- Adaptable a las modificaciones: el sistema será lo suficientemente robusto de forma que, si se agregan o cambian funcionalidades en un módulo, el resto de módulos no debe sufrir ningún impacto o el mínimo posible.

- Interfaz intuitiva: la interfaz de usuario estará diseñada de modo que los usuarios logren utilizar sus funciones de manera intuitiva, evitando pasos extras que dificulten o demoren el uso del sistema.
- Seguridad de datos: la información del paciente será tratada con seguridad, de modo que sólo el personal profesional pueda acceder a ésta.

Así mismo, para mantener un estándar de calidad, se realizará un análisis dinámico de código. Las pruebas serán de tipo caja negra para cada funcionalidad desarrollada, ya que se pretenderá un desarrollo rápido que cumpla con los estándares de calidad y seguridad. De este modo, si las funcionalidades superan dichas pruebas, serán marcadas como realizadas. Posteriormente, se diseñarán y ejecutarán pruebas específicas para cada módulo del proyecto.

En cuanto a la portabilidad, el uso del sistema estará orientado a la aplicación en computadoras de escritorios, aunque también será posible utilizarlo en celulares y tablets. La accesibilidad de los usuarios será restrictiva, siendo únicamente el personal autorizado quien acceda a los datos y a las funcionalidades del sistema. Por último, se destaca el uso de un servidor durante el transcurso del proyecto. Durante el desarrollo de cada etapa, el servidor será utilizado para consultar la base de datos del sistema SaDER. Luego de finalizar cada etapa, se desplegará el código elaborado en el servidor, con el objetivo de realizar las pruebas pertinentes.

Criterios de aceptación:

- El total de clasificaciones que brinde el sistema deben ser aprobadas en un 75 % (setenta y cinco por ciento).
- El tiempo de respuesta de triaje debe ser menor a 2 segundos, bajo condiciones normales¹ de la infraestructura del sistema.
- El módulo de registro debe guardar, al menos, uno de los tiempos establecidos.
- El sistema permitirá el ingreso de un paciente nuevo, si se completó el mínimo obligatorio de datos, los cuales serán indicados en el documento de requerimientos.

1. *Condiciones normales*: establecen el entorno ideal de software y hardware en el que el sistema puede funcionar correctamente sin experimentar problemas significativos.

Metodología

Como se menciona en los objetivos generales, este proyecto propone tres módulos definidos, los cuales pueden desarrollarse de manera independiente. Sin embargo, para su posterior implementación como sistema completo, el resultado o acción de cada módulo depende directamente de los demás, principalmente del módulo de adquisición de datos del paciente.

En un ciclo de vida incremental, el entregable se produce a través de una serie de iteraciones que, sucesivamente, añaden funcionalidad dentro de un marco de tiempo predeterminado. El entregable contiene la capacidad necesaria y suficiente para considerarse completo sólo después de la iteración final [1].

Partiendo de la definición anterior, se propone para este proyecto un desarrollo incremental. Esta elección se basa, principalmente, en la prioridad de establecer un orden de desarrollo donde se obtengan entregables de funcionalidades más reducidas y específicas que los módulos principales. Al utilizar un desarrollo incremental, se mejora la determinación de tareas, la estimación y precisión de tiempos de realización de cada una y la implementación de cambios. Así mismo, un desarrollo incremental con entregables funcionales reducidos posibilita el análisis del impacto de las nuevas funcionalidades, incorporadas a través de la prueba del sistema por parte de los directivos y usuarios durante todas las etapas de desarrollo; iniciativa que es muy recomendada en el lugar de trabajo donde se realizará dicho proyecto.

Otro punto a destacar es que la realización de las tareas de este proyecto serán llevadas a cabo por una sola persona, imposibilitando la paralelización de tareas. Por ello, las mismas serán realizadas de manera secuencial.

Por último, se destaca que, actualmente, el sistema SaDER posee funcionalidades que el proyecto utilizará y, también, serán incorporadas al mismo los desarrollos que se logren durante el transcurso de este proyecto.

El proyecto se dividirá en fases a partir de un conjunto de paquetes de trabajo, donde cada paquete contará con su respectivo entregable.

Fase 1: Definición de requerimientos

Se realizará una reunión con un médico asesor con el fin de obtener los requisitos de funcionamiento actual en guardias de ambulatoria, datos necesarios para la admisión de un paciente y los métodos de triaje.

Entregable:

- Especificación de requerimientos de software (SRS).

Criterio de aceptación:

- El documento de requerimientos debe incluir mínimamente una sección de requerimientos funcionales, requerimientos no funcionales y un diagrama de actividades UML o Casos de Uso.

Fase 2: Desarrollo del módulo de admisión

Se desarrollará el software correspondiente a la admisión del paciente, tanto las vistas del frontend como la lógica del backend.

Entregable:

- Diseño de modelo de datos.
- Diseño de interfaz de usuario.
- Software de interfaz de usuario.
- Software de backend.

Criterio de aceptación:

- El diseño del modelo de datos debe contener un diagrama de entidad-relación con nombres, atributos, cardinalidades y relaciones.
- El diseño de la interfaz de usuario debe cumplir con los requerimientos establecidos en el SRS para el módulo.
- Ambos diseños deben cumplir con los requisitos funcionales y no funcionales especificados para el módulo.

Fase 3: Pruebas del módulo de admisión

En esta fase se realizarán las pruebas correspondientes con el objetivo de evaluar el funcionamiento del software.

Entregable:

- Informe de pruebas.

Criterio de aceptación:

- El informe debe contener al menos dos pruebas por cada requerimiento funcional.
- En cada prueba se deberá nombrar el requerimiento, la serie de pasos realizados, datos proveídos al sistema, en el caso de ingresarlos, y el resultado de la prueba, siendo positiva siempre que se haya encontrado un error.

Hito 1:

- Se finaliza el módulo de admisión de paciente con la interfaz de usuario y las funcionalidades de backend, con sus respectivas pruebas, así como el documento de requerimientos.

Fase 4: Investigación de sistemas de triaje y diseño del modelo clasificadorio.

En esta fase se realizará un análisis de aquellos sistemas de triaje de acceso libre con el objetivo de obtener las funcionalidades que coincidan con las sugerencias del profesional asesor y definir una referencia del software.

También se investigará sobre modelos de clasificación de inteligencia artificial y se diseñará el modelo de clasificación. Se incluye la selección de una base de datos de entrenamiento, en caso de que el modelo seleccionado requiera de entrenamiento.

Entregable:

- Informe de sistemas de triaje de libre acceso.
- Informe de modelo clasificadorio seleccionado.
- Base de datos de entrenamiento (opcional).

Criterio de aceptación:

- El informe de sistemas de triaje de libre acceso debe contener una descripción del sistema, una lista de funcionalidades en cada pantalla y la dirección de acceso al código fuente.

Fase 5: Desarrollo del módulo de triaje

Se desarrollará el software correspondiente a las funcionalidades de frontend y backend de la clasificación. También se desarrollará la inteligencia artificial que se encargará de clasificar los niveles de triaje.

Entregable:

- Diseño de modelo de datos.
- Diseño de interfaz de usuario.
- Software de interfaz de usuario.
- Software de backend.
- Software del modelo clasificadorio entrenado.

Criterio de aceptación:

- El diseño del modelo de datos debe contener un diagrama de entidad relación con nombres de entidades, atributos, relaciones y cardinalidades.
- El diseño de la interfaz de usuario debe cumplir con los requerimientos establecidos en el SRS para el módulo.
- Ambos diseños deben cumplir con los requisitos funcionales y no funcionales especificados para el módulo.
- El modelo clasificadorio debe lograr una tasa de aciertos mayor o igual a 80 % durante su entrenamiento.

Fase 6: Pruebas del módulo de clasificación

En esta fase se realizarán las pruebas pertinentes para las funcionalidades de frontend y backend y para la clasificación de la inteligencia artificial.

Entregable:

- Informe de pruebas.

Criterio de aceptación:

- El informe debe contener al menos dos pruebas por cada requerimiento funcional.
- En cada prueba se deberá nombrar el requerimiento, la serie de pasos realizados, datos proveídos al sistema, en el caso de ingresarlos, y el resultado de la prueba, siendo positiva siempre que se haya encontrado un error.

Hito 2:

- Se completa el desarrollo del módulo de triaje con la interfaz de usuario y las funcionalidades de backend. Se seleccionó el modelo clasificadorio en base a los requerimientos y realizó el modelo de inteligencia artificial, así como sus respectivas pruebas.

Fase 7: Integración del módulo de atención médica

En esta fase se realizará la integración del software actual de atención médica que posee SaDER con los módulos desarrollados en este proyecto.

Entregable:

- Software del módulo integrado.

Criterio de aceptación:

- Los módulos del proyecto que requieran del uso de alguna de las funcionalidades integradas deben hacerlo sin fallas o errores.

Fase 8: Desarrollo de módulo de registro estadístico

Se realizará el software correspondiente a la obtención de las estadísticas mencionadas en el alcance del proyecto.

Entregable:

- Diseño de interfaz de usuario.
- Software de interfaz de usuario.
- Software de backend.

Criterio de aceptación:

- El diseño de la interfaz de usuario debe cumplir con los requerimientos establecidos en el SRS para el módulo.

- Ambos diseños deben cumplir con los requisitos funcionales y no funcionales especificados para el módulo.

Fase 9: Pruebas del módulo de registro estadístico

En esta fase se realizarán las pruebas para el módulo de registro estadístico.

Entregable:

- Informe de pruebas.

Criterio de aceptación:

- El informe debe contener al menos dos pruebas por cada requerimiento funcional.
- En cada prueba se deberá nombrar el requerimiento, la serie de pasos realizados, datos proveídos al sistema, en el caso de ingresarlos, y el resultado de la prueba, siendo positiva siempre que se haya encontrado un error.

Hito 3:

- Se completa el desarrollo del módulo estadístico con la interfaz de usuario y las funcionalidades de backend, con sus respectivas pruebas.

Fase 10: Integración del módulo en producción.

En esta fase se integrará el proyecto desarrollado al sistema de producción, de modo que esté disponible para uso por parte de todos los usuarios.

Entregable:

- Software del sistema completo desarrollado.
- Informe final.

Criterio de aceptación:

- El sistema en producción debe cumplir con todos los requerimientos funcionales y no funcionales de cada módulo.

Hito 4:

- Se concluye el proyecto, logrando una integración en el sistema de producción que no altere el funcionamiento previo del sistema.

Plan de tareas

Se detallan las actividades para cada fase con la estimación de duración para cada una. Para este proyecto se dedicarán 4 horas diarias, desde los días lunes hasta viernes, sin considerar aquellos días feriados y no laborables. Se estima una duración total de 340 horas. De este modo se inicia el desarrollo de las tareas el día 07/08/2024 y finaliza el día 4/12/2024.

Nombre de la tarea	Duración (horas)	Responsable
1.1 Requerimientos de guardia	8	
1.1.a Realizar entrevista con el asesor profesional	2	Andino, Daniel Agustín
1.1.b Examinar documentación provista	4	Andino, Daniel Agustín
1.1.c Realizar informe de requerimientos de guardia	2	Andino, Daniel Agustín
1.2 Informe de gestión de triaje	10	
1.2.a Investigar sobre modelos de triaje	6	Andino, Daniel Agustín
1.2.b Examinar documentación provista	2	Andino, Daniel Agustín
1.2.c Realizar informe de modelo de triaje	2	Andino, Daniel Agustín
2.1 Desarrollo de frontend del módulo de admisión	24	
2.1.a Diseño de la interfaz de usuario	6	Andino, Daniel Agustín
2.1.b Desarrollo de la interfaz de usuario	18	Andino, Daniel Agustín
2.2 Desarrollo de backend del módulo de admisión	30	
2.2.a Diseño de modelo de datos	12	Andino, Daniel Agustín
2.2.b Desarrollo de funcionalidades de backend	18	Andino, Daniel Agustín
3 Pruebas del módulo de admisión	22	
3.a Diseñar pruebas de módulo de admisión	2	Andino, Daniel Agustín

3.b Ejecutar pruebas de módulo de admisión	4	Andino, Daniel Agustín
3.c Elaborar informe de prueba	4	Andino, Daniel Agustín
3.d Elaborar informe de avance 1	6	Andino, Daniel Agustín
3.e Corregir errores	6	Andino, Daniel Agustín
Hito 1: Se finaliza el módulo de admisión de paciente con la interfaz de usuario y las funcionalidades de backend, con sus respectivas pruebas, así como el documento de requerimientos.	0	

4.1 Investigación del sistemas de triaje gratuitos	14	
4.1.a Investigar sistemas de triaje gratuitos	4	Andino, Daniel Agustín
4.1.b Examinar código de sistema de triaje gratuitos	8	Andino, Daniel Agustín
4.1.c Elaborar informe sobre sistemas de triaje gratuitos	2	Andino, Daniel Agustín
4.2 Investigación de modelos de clasificación de inteligencia artificial	18	
4.2.a Investigar sobre modelos de inteligencia artificial clasificatorios	10	Andino, Daniel Agustín
4.2.b Definir modelo clasificatorio de inteligencia artificial	6	Andino, Daniel Agustín
4.2.c Elaborar informe de modelo elegido	2	Andino, Daniel Agustín
5.1 Desarrollo de funcionalidades de frontend del módulo de clasificación	20	
5.1.a Diseño de la interfaz de usuario	6	Andino, Daniel Agustín
5.1.b Desarrollo de la interfaz de usuario	14	Andino, Daniel Agustín
5.2 Desarrollo de las funcionalidades del backend del módulo de clasificación	28	
5.2.a Diseño de modelo de datos	8	Andino, Daniel Agustín
5.2.b Desarrollo de funcionalidades de backend	14	Andino, Daniel Agustín

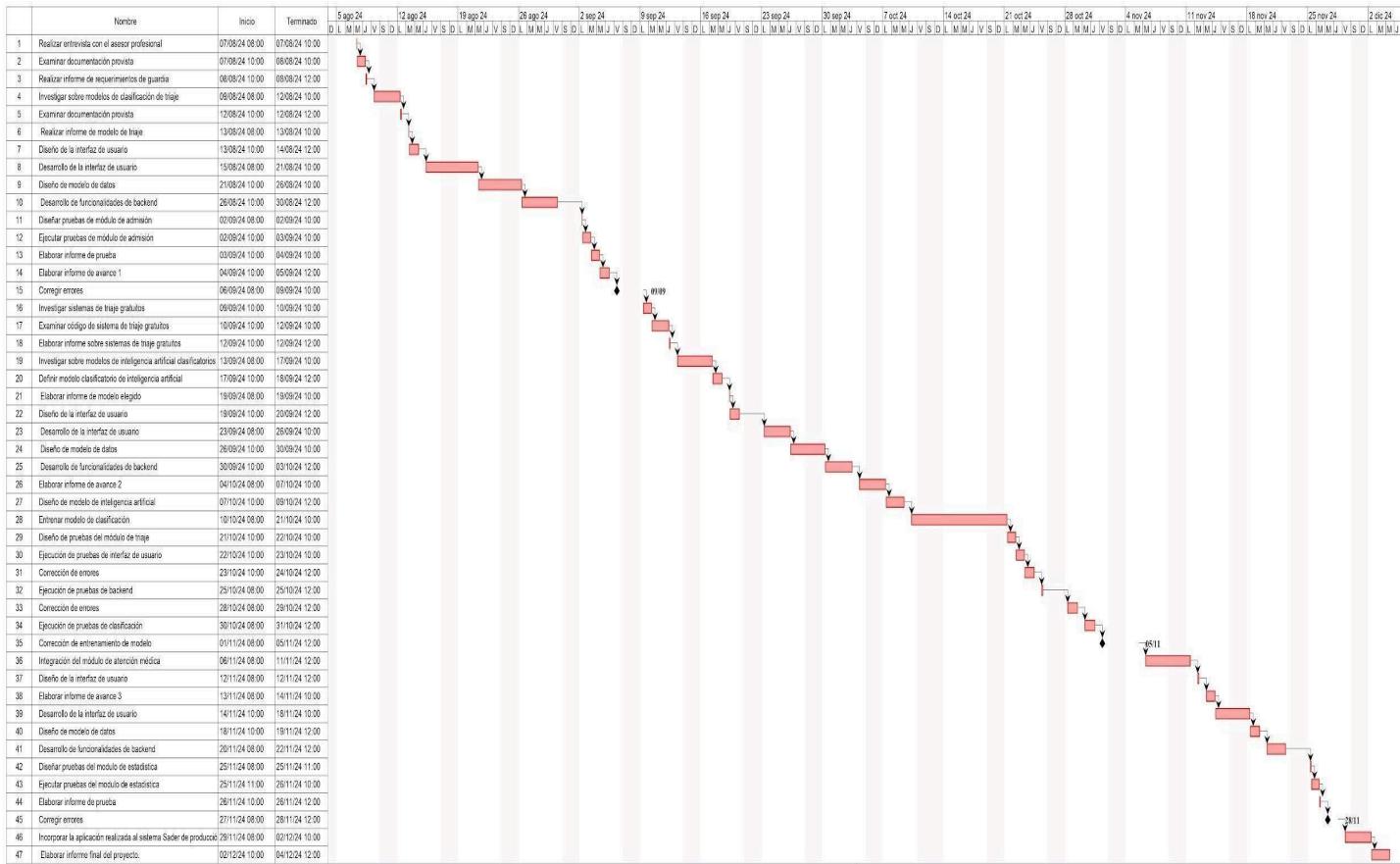
5.2.c Elaborar informe de avance 2	6	Andino, Daniel Agustín
5.3 Desarrollo de la inteligencia artificial	36	
5.3.a Diseño de modelo de inteligencia artificial	10	Andino, Daniel Agustín
5.3.b Entrenar modelo de clasificación	26	Andino, Daniel Agustín
6.1 Pruebas de las funcionalidades de frontend	14	
6.1.a Diseño de pruebas del módulo de triaje	4	Andino, Daniel Agustín
6.1.b Ejecución de pruebas de interfaz de usuario	4	Andino, Daniel Agustín
6.1.b Corrección de errores	6	Andino, Daniel Agustín
6.2 Pruebas de las funcionalidades de backend	12	
6.2.a Ejecución de pruebas de backend	4	Andino, Daniel Agustín
6.2.b Corrección de errores	8	Andino, Daniel Agustín

6.3 Pruebas de la inteligencia artificial	20	
6.3.a Ejecución de pruebas de clasificación	8	Andino, Daniel Agustín
6.3.b Corrección de entrenamiento de modelo	12	Andino, Daniel Agustín
Hito 2: Se completa el desarrollo del módulo de triaje con la interfaz de usuario y las funcionalidades de backend. Se seleccionó el modelo clasificadorio en base a los requerimientos y realizó el modelo de inteligencia artificial, así como sus respectivas pruebas.	0	
7 Integración del módulo de atención médica	16	Andino, Daniel Agustín
8 Desarrollo del módulo de registro estadístico	36	
8.a Diseño de la interfaz de usuario	4	Andino, Daniel Agustín
8.b Elaborar informe de avance 3	6	Andino, Daniel Agustín
8.c Desarrollo de la interfaz de usuario	8	Andino, Daniel Agustín

8.d Diseño de modelo de datos	6	Andino, Daniel Agustín
8.e Desarrollo de funcionalidades de backend	12	Andino, Daniel Agustín
9 Pruebas del módulo de registro estadístico	16	
9.a Diseñar pruebas del modulo de estadistica	3	Andino, Daniel Agustín
9.b Ejecutar pruebas del modulo de estadistica	3	Andino, Daniel Agustín
9.c Elaborar informe de prueba	2	Andino, Daniel Agustín
9.d Corregir errores	8	Andino, Daniel Agustín
Hito 3: Se completa el desarrollo del módulo estadístico con la interfaz de usuario y las funcionalidades de backend, con sus respectivas pruebas.	0	
10 Integración del proyecto a producción y cierre del proyecto	16	
10.a Incorporar la aplicación realizada al sistema SaDER de producción	6	Andino, Daniel Agustín
10.b Elaborar informe final del proyecto.	10	Andino, Daniel Agustín
Hito 4: Se concluye el proyecto, logrando una integración en el sistema de producción que no altere el funcionamiento previo del sistema.	0	

Cronograma de actividades

A continuación se muestra el cronograma de actividades, junto con la duración de cada una y sus fechas de inicio y fin. Se destaca que todas las tareas serán realizadas por la única persona a cargo del proyecto, por ende se obvia la asignación de recursos. Los hitos del proyecto se muestran como rombos negros en el gráfico.



(Figura 2. Cronograma de actividades)

Puntos de seguimiento

Informe de Avance 1

Fecha de entrega: 05/09/2024

En este informe se incluye la información referida desde la fase 1 hasta la 2 mencionadas en la metodología. El informe contendrá la especificación de requerimientos de software (SRS), el diseño de modelo de datos, el diseño de interfaz de usuario, el informe de pruebas del módulo de admisión.

Informe de Avance 2

Fecha de entrega: 10/10/24

En este informe se incluyen los entregables de la fase 3 y 4. Contiene el informe de investigación de sistemas de triaje de libre acceso, el informe de modelo clasificadorio seleccionado y una referencia opcional a una base de datos de entrenamiento para el modelo clasificadorio.

Informe de Avance 3

Fecha de entrega: 19/11/2024.

En este informe se incluye los entregables de la fase 5 hasta la fase 7. El diseño de modelo de datos y el diseño de la interfaz de usuario para el módulo de clasificación de urgencias. También se plasma información sobre el desarrollo del software de la interfaz de usuario y las funcionalidades de backend de dicho módulo, así como información referida al desarrollo del modelo clasificadorio y su entrenamiento. Además se agrega el informe de pruebas de las funcionalidades y del modelo clasificadorio.

Informe Final

Fecha de entrega: 4/12/24

En este informe se incluye la información de la fase 8 hasta la fase 10. Se abarca el proceso de diseño del modelo de datos y la interfaz de usuario del módulo de estadísticas, así como del desarrollo de las funcionalidades de backend y de la interfaz de usuario. También se incluye el informe de los resultados de las pruebas de dicho módulo. Por último, el informe final estará complementado con el código fuente del proyecto.

Riesgos

Según el PMBOK, los objetivos de la gestión de los riesgos del proyecto son aumentar la probabilidad y/o el impacto de los riesgos positivos y disminuir la probabilidad y/o el impacto de los riesgos negativos, a fin de optimizar las

posibilidades de éxito del proyecto. Por ello, la gestión de riesgos es esencial para lograr el cumplimiento del objetivo en proyectos.

A continuación, se llevará a cabo la identificación y evaluación de los riesgos considerados para este proyecto. También, se incluirá la estrategia para cada riesgo y el plan de contingencia en aquellos casos que sean necesarios.

Los niveles de probabilidad de los riesgos que se consideran para este proyecto son los siguientes: Bajo, Medio, Alto, Muy Alto. En cuanto a los impactos se toman: Bajo, Medio y Alto. A partir de los cuáles se definen las estrategias para cada combinación posible de probabilidad e impacto:

Probabilidad \ Impacto	Bajo	Medio	Alto
Bajo (< 20 %)	A.P	A.A	Mitigar
Medio (> 20 % y <40 %)	A.P	A.A	Mitigar
Alto(> 40 % y <70 %)	A.A	Mitigar	Evitar
Muy Alto(>70 %)	A.A	Mitigar	Evitar

Las siglas A.A significan aceptar activamente y A.P aceptar pasivamente. Su diferencia radica en que la primera contempla un plan de contingencia, que es una serie de pasos o actividades que se deben realizar luego de que ocurra el riesgo, mientras que el otro no.

Los riesgos que se considerarán para este proyecto son los siguientes:

Número: 001	Nombre: Cambio en los requisitos.	
Descripción: Se proponen nuevas funcionalidades, no contempladas en el documento de requerimientos.		
Probabilidad: Medio	Impacto: Medio	Estrategia: Aceptar Activamente
Indicadores: Los interesados en el proyecto presentan poca o nula conformidad con los entregables en tiempo y sin realizar comentarios objetivos para realizar cambios o mejoras del entregable.		
Plan de contingencia:		

<ul style="list-style-type: none"> • Se relevará el requerimiento y se propondrá su implementación en futuras versiones del proyecto, 	
Número: 002	

<p>Descripción: El director del proyecto, dadas las circunstancias, abandona de manera parcial o permanente la dirección del proyecto.</p>		
Probabilidad: Bajo	Impacto: Alto	Estrategia: Mitigar
<p>Respuesta de mitigación:</p> <ul style="list-style-type: none"> • Establecer una comunicación con el director de proyecto con la frecuencia necesaria para anticipar las posibilidades de ocurrencia del riesgo. La comunicación no debe exceder más de dos semanas entre cada una y se puede realizar a través de correos electrónicos. • Proponer herramientas de comunicación que permitan una retroalimentación distinta a la de correos electrónicos, como Google Chat. 		
<p>Indicadores: El director muestra poca o nula comunicación con el responsable. Las devoluciones o evaluaciones de los informes se retrasan más de dos semanas.</p>		
<p>Plan de contingencia:</p> <ul style="list-style-type: none"> • Continuar con las actividades del proyecto al mismo tiempo que se busca un reemplazante como director de proyecto. 		
Número: 003	Nombre: Reubicación en las instalaciones donde se realizará el proyecto	
<p>Descripción: Las instalaciones donde se ejecutará el proyecto son reubicadas en otra infraestructura de manera parcial o permanente.</p>		
Probabilidad: Bajo	Impacto: Alto	Estrategia: Mitigar
<p>Respuesta de mitigación:</p> <ul style="list-style-type: none"> • Implementar medidas de trabajo remoto, tanto para la realización de las tareas como para las reuniones establecidas en el plan de comunicación, y planificar el aprovisionamiento de los recursos necesarios para el trabajo remoto. • Configurar los recursos necesarios para poder hacer uso del servidor, pueden ser: VPN (red privada virtual), credencial de acceso SSH, entre otros. 		
<p>Indicadores: El organismo reubica otras oficinas. Cambios en el presupuesto del organismo que permiten u obligan el cambio de lugar físico de las instalaciones.</p>		
<p>Plan de contingencia:</p> <ul style="list-style-type: none"> • Emplear las medidas de trabajo remoto propuestas. 		
Número: 004	Nombre: Cambio en la disponibilidad de los recursos y/o equipamientos utilizados para realizar el proyecto.	
<p>Descripción: Uno o más de los recursos y/o equipamiento proveídos no se encuentran disponibles para su uso.</p>		
Probabilidad: Bajo	Impacto: Alto	Estrategia: Mitigar

<p>Respuesta de mitigación:</p> <ul style="list-style-type: none"> • Informarse constantemente sobre el estado de los recursos y equipamientos, analizando posibles fallos o inestabilidad en un futuro. • Mantener una copia de la base de datos de pruebas por fuera del servidor. 		
<p>Indicadores: El equipamiento o recursos comienzan a presentar fallas físicas con frecuencia.</p>		
<p>Plan de contingencia:</p> <ul style="list-style-type: none"> • Analizar el recurso o equipamiento que no está disponible y la posibilidad de ser reemplazado en el momento. • Si el recurso o equipamiento no puede ser reemplazado en el momento, entonces se elevará el respectivo informe con su presupuesto. <p>Se considerará el caso especial, dónde el equipamiento no disponible sea el servidor de pruebas:</p> <ul style="list-style-type: none"> ○ Se llevarán a cabo las actividades de pruebas de manera local en la computadora de escritorio. <ul style="list-style-type: none"> • De ser necesario el recurso para la actividad que se desarrolla en el momento de la ocurrencia del riesgo, se modificará el cronograma y el presupuesto. 		
Número: 005	Nombre: Falta de disponibilidad de los recursos humanos del proyecto.	
<p>Descripción: Los recursos humanos no se encuentran en capacidad de realizar las tareas del cronograma.</p>		
Probabilidad: Bajo	Impacto: Medio	Estrategia: Aceptar Activamente.
<p>Indicadores: Imprevistos que condicionan la disponibilidad del recurso humano (enfermedades, viajes imprevistos, trámites laborales o judiciales).</p>		
<p>Plan de contingencia:</p> <ul style="list-style-type: none"> • Si el recurso no se encuentra disponible por un periodo menor a 3 días: <ul style="list-style-type: none"> a. El proyecto avanza conforme al cronograma de actividades o está adelantado, entonces se modificará el cronograma, aclarando el motivo. b. El proyecto está retrasado con respecto al cronograma de actividades, entonces se modificará el cronograma en la menor cantidad de tiempo posible. Si se presenta la situación de que el retraso provoque la entrega parcial o nula de un entregable o más, se comunicará con un día de antelación a la fecha pactada a las partes interesadas sobre el retraso. • Si el recurso no se encuentra disponible por un periodo mayor a 3 días se propondrá la implementación de medidas de trabajo remoto, intentando no modificar el cronograma. 		
Número: 006	Nombre: Retraso en el cronograma del proyecto.	
<p>Descripción: El cronograma establecido para el proyecto se retrasa a causa de no finalizar las actividades propuestas o no disponer del entregable para la fecha requerida.</p>		
Probabilidad: Medio	Impacto: Alto	Estrategia: Mitigar

Respuesta de mitigación:

- Mantener un control al comienzo de cada semana sobre el avance de las actividades, con el objetivo de prevenir adelanto o retraso de las mismas, y el estado general del entregable
- Agregar horas extras de trabajo con el objetivo de finalizar las actividades en las fechas

estimadas.

Indicadores: Las tareas llevan más tiempo del estimado para ser completadas. Incapacidad de avanzar en el proyecto ocurrido el riesgo 004.

Plan de contingencia:

- Se procederá a evaluar el impacto del retraso, identificando las tareas críticas afectadas y las posibles causas del retraso. También se ajustará el cronograma del proyecto.
- Se comunicará, con no menos de dos días de anticipación, a los interesados del proyecto acerca del estado del entregable. Se propondrá una entrega imparcial en el tiempo estipulado o una entrega tardía con todos los requerimientos.

Número: 007	Nombre: Baja por parte del patrocinador o despido del integrante del proyecto.
--------------------	---

Descripción: El patrocinador del proyecto decide no seguir apoyando el desarrollo del proyecto o el integrante es desvinculado de la organización.

Probabilidad: Medio	Impacto: Medio	Estrategia: Aceptar Activamente
----------------------------	-----------------------	--

Indicadores: Cambios en el presupuesto del organismo o leyes de contrato del estado.

Plan de contingencia:

- Analizar el estado de avance del proyecto y cambiar el alcance del proyecto a partir del análisis.
- Proponer al jefe del área la posibilidad de continuar concurriendo a las instalaciones de la organización hasta finalizar el proyecto, realizando el trabajo sin dependencia laboral.
 - En caso de que no se acepte la propuesta anterior, reunirse con el director del proyecto y proponer un nuevo alcance donde se cumpla el objetivo general, modificando los objetivos específicos que sean dependientes del sistema SaDER.

Número: 008	Nombre: Brotes epidémicos potenciales
--------------------	--

Descripción: Emergencias sanitarias donde se fuerce al aislamiento social parcial o total.

Probabilidad: Bajo	Impacto: Medio	Estrategia: Aceptar Activamente
---------------------------	-----------------------	--

Indicadores: Organismos gubernamentales imponen medidas de restricción.

Plan de contingencia:

- Proponer la implementación de medidas de trabajo remoto, tanto para la realización de las tareas como para las reuniones establecidas en el plan de comunicación, y planificar el aprovisionamiento de los recursos necesarios para el trabajo remoto.

Número: 009	Nombre: Falla de integración de las funcionalidades que ya posee el sistema.	
Descripción: Caso en que las funcionalidades que utiliza el sistema de producción produzcan conflictos con los módulos desarrollados en este proyecto.		
Probabilidad: Alto	Impacto: Alto	Estrategia: Evitar
<p>Respuesta de mitigación:</p> <ul style="list-style-type: none"> Investigar sobre posibles fallos que pueden generarse entre las librerías, lenguajes y tecnologías utilizados en el sistema actual y el sistema a desarrollar en este proyecto. A partir de la investigación, elaborar una lista de incompatibilidades o errores con el objetivo de desarrollar las funcionalidades de cada módulo evitando los conflictos mencionados en dicha lista. Realizar modelos de diseños de cada módulo cuya interacción con las funcionalidades del sistema actual sean las mínimas y necesarias para el funcionamiento de ambas, procurando que el impacto sobre el sistema actual sea el menor posible. 		
<p>Indicadores: Conflictos producidos frecuentemente por motores de bases de datos, tecnologías o librerías utilizadas en el sistema actual y en el propuesto por proyecto.</p>		
<p>Plan de contingencia:</p> <ul style="list-style-type: none"> Analizar la causa de las fallas de la integración. <ul style="list-style-type: none"> Si se encuentran errores de codificación, se cambiará el cronograma añadiendo una tarea de corrección de errores. Si no se encuentran errores, se cambiará el cronograma añadiendo una actividad de pruebas con el objetivo de encontrar las fallas y otra de corrección de errores. 		
Número: 010	Nombre: Falta de datos apropiados para realizar el entrenamiento del modelo clasificadorio.	
Descripción: No se poseen datos, la cantidad necesaria no es suficiente o éstos no están adaptados para realizar el entrenamiento del modelo de inteligencia artificial que clasificará el triaje.		
Probabilidad: Alto	Impacto: Medio	Estrategia: Mitigar
<p>Respuesta de mitigación:</p> <ul style="list-style-type: none"> Investigar en centros médicos locales sobre la disponibilidad de datos digitales o analógicos, con el objetivo de ser utilizados como base de datos ante la falta de la misma. Analizar la posibilidad de utilizar bases de datos de procedencia extranjera, adaptándolos para el entrenamiento específico del proyecto, de modo que se pueda disminuir o no el alcance. 		
<p>Indicadores: Los interesados en el proyecto no tienen conocimiento de la existencia o disponibilidad de los datos requeridos.</p>		
<p>Plan de contingencia:</p> <ul style="list-style-type: none"> Optar por un modelo clasificadorio que no requiera entrenamiento supervisado, como lógica borrosa o árboles de decisiones. 		

Recursos

En esta sección, se describirán los diversos recursos necesarios para llevar a cabo este proyecto de manera efectiva.

Recurso	Descripción	Cantidad
Recursos Humanos	Estudiante	1
	Director de proyecto	1
	Asesor profesional	2
Equipo	Servidor	1
	Computadora escritorio	1
	Monitor	1
	Mouse	1
	Teclado	1
	Escritorio	1
	Silla oficina	1
Servicios	Servicio de Internet	1
	Servicio de electricidad	1
	Servicio de limpieza	2

Presupuesto

En esta sección se presentará el presupuesto necesario para llevar a cabo la ejecución del proyecto. El presupuesto está conformado las amortizaciones de los bienes capitales y los distintos gastos específicos de cada rubro,

Según el cronograma la cantidad de horas trabajadas es de 340, contando un total de 86 días laborales. Por lo que, para el cálculo de los gastos se consideran aquellos pagos por hora, por día y por mes.

Los honorarios por hora fueron aproximados a los recomendados por el Consejo Profesional de Ciencias Informáticas de la Provincia de Córdoba (CPCIPC) a fecha de marzo del año 2024.

Bienes de capital

Para amortizar se utilizará la amortización lineal, dada por la siguiente fórmula:

$$Amortización = \frac{Valor\ Nuevo - Valor\ Residual}{Vida\ útil}$$

La vida útil de cada bien será medida en años. El valor residual para todos los bienes se estimará en un 10 % de su valor nuevo. Además al valor de amortización anual se lo multiplica por un coeficiente teniendo en cuenta sólo la cantidad de meses desde el inicio hasta la finalización del proyecto.

$$Amortización = \frac{Valor\ Nuevo - Valor\ Residual}{Vida\ útil} \cdot \frac{Duración\ Proyecto\ en\ Meses}{12\ meses}$$

A continuación se listarán los bienes de capital necesarios:

- Computadora Banghó Cross B02, CPU AMD Athlon 3000G 8 GB ram
 - Valor Nuevo = \$300 000
 - Valor Residuo = \$30 000
 - Vida Útil = 5 años
 - Amortización anual = \$60 000
 - Amortización Proyecto = \$18 000
- Monitor Banghó
 - Valor Nuevo = \$120 000
 - Valor Residuo = \$12 000
 - Vida Útil = 3,5 años
 - Amortización anual = \$34 285
 - Amortización Proyecto = \$10 285
- Mouse y teclado
 - Valor Nuevo = \$20 000
 - Valor Residuo = \$2 000
 - Vida Útil = 2 años
 - Amortización anual = \$10 000
 - Amortización Proyecto = \$3 000
- Servidor Lenovo ThinkSystem SR550 136gb ram 20 CPU Intel Xeon Silver 4114
 - Valor Nuevo = \$ 9 000 000
 - Valor Residuo = \$900 000
 - Vida Útil = 15 años
 - Amortización anual = \$600 000 por año

- Amortización proyecto = \$180 000
- Escritorio oficina
 - Valor Nuevo = \$100 000
 - Valor Residuo = \$10 000
 - Vida útil = 10 años
 - Amortización anual = \$10 000
 - Amortización proyecto = \$3 000
- Silla oficina
 - Valor Nuevo \$50 000
 - Valor Residuo = \$5 000
 - Vida útil = 5 años
 - Amortización anual = \$10 000
 - Amortización proyecto = \$3 000

Consultorías y Servicios

Dentro de las consultorías se considerarán los trabajos de aquellas personas que aportan al proyecto en su realización pero que no son responsables de realizarlas. Entre ellos se encuentran:

- Médico asesor: 2 horas de trabajo, se consultó su presupuesto de pago en usd\$35 por hora. Considerando el dólar oficial a la fecha como \$911 (novecientos once pesos argentinos por cada dólar estadounidense). Por lo que el gasto en el médico asesor asciende a \$63 770.
- Segundo asesor: se trata del jefe de área del organismo donde se realizará el proyecto. El total de horas corresponden a 28, siendo las reuniones establecidas en el plan de comunicaciones de 1 hora de duración cada una. Su pago por hora es de \$9 500, de modo que su pago será de \$266 000.

En cuanto a los servicios necesarios para el proyecto se tendrán en cuenta los días de diciembre como un mes completo, contabilizando 5 meses para las facturas mensuales:

- Servicio de internet: \$35 000 por mes
- Servicio de electricidad: \$50 000 por mes.

El valor total del servicio de internet por el tiempo de trabajo es de \$170 000 y el del servicio de electricidad es de \$250 000.

También se tendrá en cuenta el servicio de limpieza de las instalaciones, que cuenta con dos empleados. Su trabajo es de 1 hora semanal, siendo el pago de cada uno \$1 500 por hora. Por lo que, el gasto total en servicio de limpieza es de \$57 000.

Infraestructura

En cuanto a los gastos de infraestructura se considerará el alquiler del inmueble donde se realizará el proyecto, valorado en \$300 000 mensuales. Por lo que, el costo total durante el proyecto será de \$1 500 000.

Viajes y Viáticos

Para este rubro se tendrán en cuenta dos tipos de viajes: urbanos con límite en la ciudad de Paraná e interurbano con límite entre Paraná y Santa Fe. Para ambos se contabilizarán el duplicado de viajes totales, dado que se cuenta tanto el viaje de ida como el viaje de vuelta.

- Los primeros tendrán un valor de \$940 por día. Contando los 86 días del proyecto, el total es de \$161 680.
- Los viajes destinados a Santa Fe y Paraná tendrán un valor de \$1 240.

Contabilizando solo 3 ocasiones, el total del costo es de \$7 440.

Por lo que, el total de gastos en viajes y viáticos asciende a \$169 120.

Recursos Humanos

Para el pago de los recursos humanos se tendrá en cuenta la cantidad de horas trabajadas y se pagará por hora. Respecto a los recursos humanos se utilizará un programador y un director de proyecto.

Para el Programador su pago por hora es de \$5 500, teniendo en cuenta el total de horas de trabajo del proyecto, el costo total es de \$1 870 000.

El director de proyecto: siguiendo el plan de comunicación, se realizarán 4 reuniones con el director del proyecto de 1 hora y 30 minutos de duración cada una. Se considerarán 14 horas extras destinadas para sus tareas, sumando un total de 20 horas. Su pago por hora es de \$14 000, entonces el gasto en el director de proyecto será de \$280 000.

Relacionando todos los gastos enumerados anteriormente, se puede concluir que el presupuesto estimado del proyecto es de \$4 843 175 en pesos argentinos.

Plan de comunicaciones

En esta sección, se presentará el plan de comunicaciones del proyecto. El PMBOK resalta la importancia de este plan, dado que se asegura que las necesidades de información del proyecto y de sus interesados se satisfagan a través del desarrollo de objetos y de la implementación de actividades diseñadas para lograr un intercambio eficaz de información.

En la siguiente tabla se muestra la información con respecto a cada comunicación especificando la necesidad de realizarla, el canal o medio que se utilizará y las fechas establecidas.

Matriz de Comunicaciones

Proyecto: Desarrollo de módulo de guardia ambulatoria con triaje automatizado mediante inteligencia artificial en el sistema web Sader.							
ETAPA	Elemento de la EDT	¿Qué comunicamos?	Destinatarios (stakeholders)	Método de Comunicación	Responsables	Fecha de inicio	Frecuencia
Número y nombre de la etapa a la que corresponde la comunicación	Número y nombre del elemento en la EDT. Puede ser una actividad, entregable o hito.	Describe el asunto o tema que se quiere comunicar.	Nombre o rol de la persona al que va dirigido.	Describe la forma en que será comunicado.	Responsables de elaboración y envío.	Fecha en que debe comenzar el envío.	Indica la frecuencia del envío.
Etapa 1 hasta 10		Reunión de avances y logros semanales	Jefe de Área	Presencial	Andino, Daniel Agustín	7/8/2024	7 días
Etapa 1 Relevamiento de requerimientos de guardia	Especificación de requerimientos de software (SRS)	Requerimientos relevados para el proyecto	Jefe de Área	Presencial	Andino, Daniel Agustín	13/8/2024	
Etapa 2 Desarrollo del módulo de admisión	Reunión previa a la entrega del informe de avance 1. Informe de avance 1.	Comunicar logros y avances del proyecto realizados hasta la fecha de reunión	Director del proyecto	Presencial/Virtual	Andino, Daniel Agustín	29/8/2024	
Etapa 2 Desarrollo del módulo de admisión	Diseño de modelo de datos. Diseño de interfaz de usuario.	Diseño y código del módulo de admisión de pacientes	Jefe de Área	Presencial	Andino, Daniel Agustín	30/8/2024	
Etapa 3 Pruebas del módulo de admisión	Informe de pruebas. Hito 1	Resultados de pruebas del módulo de admisión	Jefe de Área	Presencial	Andino, Daniel Agustín	9/9/2024	
Etapa 4 Investigación de modelos de clasificación y diseño del modelo de clasificación	Informe de sistemas de triaje de libre acceso. Informe de modelo clasificatorio seleccionado.	Diseño del modelo de clasificación	Jefe de Área	Presencial	Andino, Daniel Agustín	18/9/2024	
Etapa 4 Investigación de modelos de clasificación y diseño del modelo de clasificación	Reunión previa a la entrega del informe de avance 2. Informe de avance 2.	Comunicar logros y avances del proyecto realizados hasta la fecha de reunión	Director del proyecto	Presencial/Virtual	Andino, Daniel Agustín	3/10/2024	
Etapa 5 Desarrollo del módulo de clasificación de triaje y entrenamiento del modelo de inteligencia artificial.	Diseño de modelo de datos. Diseño de interfaz de usuario. Modelo clasificatorio entrenado.	Diseño y código del módulo de clasificación. Modelo clasificatorio entrenado.	Jefe de Área	Presencial	Andino, Daniel Agustín	21/10/2024	
Etapa 6 Pruebas del módulo de clasificación de triaje	Informe de pruebas. Hito 2	Resultados de pruebas del módulo de admisión	Jefe de Área	Presencial	Andino, Daniel Agustín	5/11/2024	
Etapa 7 Integración del módulo de atención médica	Software del módulo integrado.	Código de la integración del módulo de atención médica al proyecto	Jefe de Área	Presencial	Andino, Daniel Agustín	11/11/2024	
Etapa 7 Integración del módulo de atención médica	Reunión previa a la entrega del informe de avance 3. Informe de avance 3.	Comunicar logros y avances del proyecto realizados hasta la fecha de reunión	Director del proyecto	Presencial/Virtual	Andino, Daniel Agustín	12/11/2024	
Etapa 8 Desarrollo del módulo de estadísticas	Diseño de interfaz de usuario.	Diseño y código del módulo de estadísticas.	Jefe de Área	Presencial	Andino, Daniel Agustín	22/11/2024	
Etapa 9 Pruebas del módulo de estadísticas.	Informe de pruebas. Hito 3	Resultados de pruebas del módulo de estadísticas	Jefe de Área	Presencial	Andino, Daniel Agustín	28/11/2024	
Etapa 9 Pruebas del módulo de estadísticas.	Reunión previa a la entrega del informe final	Comunicar logros y avances del proyecto realizados hasta la fecha de reunión	Director del proyecto	Presencial/Virtual	Andino, Daniel Agustín	3/12/2024	
Etapa 10 Integración del proyecto a producción y cierre del proyecto	Software del sistema completo desarrollado. Hito 4	Código de la implementación del sistema completo en producción.	Jefe de Área	Presencial	Andino, Daniel Agustín	4/12/2024	
Etapa 10 Integración del proyecto a producción y cierre del proyecto	Informe final del proyecto. Hito 4.	Confirmar la entrega del informe final del proyecto.	Director del proyecto	Presencial/Virtual	Andino, Daniel Agustín	4/12/2024	

(Figura 3. Plan de Comunicación)

Bibliografía

- Project Management Institute. (2017). *A guide to the project management body of knowledge (PMBOK® guide)* (6th ed.). Project Management Institute.
- Manual para la implementación de un sistema de triaje para los cuartos de urgencias. (2010). Washington, DC.
- Busso, L., Bellavita, A., Heim, S., & Maidana, C. (2020). *Implementación del sistema triaje en los hospitales públicos de la provincia de Buenos Aires*.
- Castello, D., Bollo, C., Gauna, C., Montes, I., & Rocha Vargas, C. (2010). *Políticas de software libre en el Estado Argentino*. Universidad Nacional de Córdoba.
- Gómez Jiménez, J. (2003). *Clasificación de pacientes en los servicios de urgencias y emergencias: Hacia un modelo de triaje estructurado de urgencias y emergencias*.
- Kipourgos, G., Tzenalis, A., Diamantidou, V., Koutsojannis, C., & Hatzilygeroudis, I. (2022). *An artificial intelligence-based application for triage nurses in emergency department, using the emergency severity index protocol*.
- Farahmand, S., Shabestari, O., Pakrah, M., Hossein-Nejad, H., Arbab, M., & Bagheri-Hariri, S. (2017). *Artificial intelligence-based triage for patients with acute abdominal pain in emergency department; a diagnostic accuracy study*.
- Honorable Congreso de la Nación Argentina. (2023). *Ley 27706: Programa Federal Único de Informatización y Digitalización de Historias Clínicas de la República Argentina*.

Referencias

- [1] INDEC(2022). Censo Nacional de Población, Hogares y Viviendas 2022. Cuadro 1.8. Provincia de Entre Ríos. Población total en viviendas particulares. Cobertura de salud según departamento.
- [2] Ministerio de Salud de la Nación. Total de beneficiarios inscriptos al Programa Sumar con Cobertura Efectiva Básica (CEB). <https://datos.gov.ar>
- [3] ¿Qué es el Programa Sumar? <https://www.argentina.gob.ar/salud/sumar>
- [4] Busso, L., Bellavita, A., Heim, S., Maidana, C. (2019). Implementación del sistema triaje en los hospitales públicos de la provincia de Buenos Aires.
- [5] Manual para la implementación de un sistema de triaje para los cuartos de urgencias (2010). Washington, DC
- [6] “Todo es demasiado estresante”. La fuga de médicos de guardia en hospitales profundiza la crisis sanitaria
<https://www.lanacion.com.ar/sociedad/todo-es-demasiado-estresante-la-fuga-de-medicos-de-guardia-en-hospitales-profundiza-la-crisis-nid17022024/>
- [7] “Está en riesgo la seguridad del paciente” Éxodo de médicos y servicios que bajan su nivel en un sistema sanitario que cruce.
<https://www.lanacion.com.ar/sociedad/esta-en-riesgo-la-seguridad-del-paciente-exodo-de-medicos-y-servicios-de-menor-nivel-en-un-sistema-nid18052023/>
- [8] Vítolo, F. (2014). SATURACIÓN DE LAS GUARDIAS DE EMERGENCIA Parte II. Posibles Soluciones. Noble Compañía de Seguros
- [9] Historia de Salud Integrada <https://hsi.pladema.net/>
- [10] Castello, Bollo, Gauna, Montes, Rocha Vargas (2010). Políticas de software libre en el estado argentino. Universidad Nacional de Córdoba, pág 3.
- [11] IT Colmena Group S.R.L <https://www.itcolmena.com/Castellano/index.html>
- [12] Geclisa <https://geclisa.com/>
- [13] Gómez Jiménez, J. (2003). Clasificación de pacientes en los servicios de urgencias y emergencias: Hacia un modelo de triaje estructurado de urgencias y emergencias.
- [14] Ventajas y desventajas de la Inteligencia Artificial en Empresas <https://nexusintegra.io/es/ventajas-y-desventajas-de-la-inteligencia-artificial/>
- [15] Razones para incorporar la inteligencia artificial a tu empresa <https://neuroons.com/es/razones-incorporar-inteligencia-artificial-empresa/>
- [16] Más demanda en hospitales públicos del conurbano
<https://www.pagina12.com.ar/727339-mas-demanda-en-hospitales-publicos-del-conurbano>
- [17] Hospitales: Aumentan las consultas por subas en prepagas
<https://www.unoentrerios.com.ar/la-provincia/hospitales-aumentan-consultas-subas-prepagas-n10111205.html>
- [18] Informe sobre el impacto en los aumentos de las prepagas de la Resolución N° 1/2024
<https://observatoriociudad.org/informe-sobre-el-impacto-en-los-aumentos-de-las-prepagas-de-la-resolucion-n-1-2024/>