



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

**MONITOREO Y GESTIÓN DE INDICACIONES MÉDICAS EN PACIENTES
HOSPITALIZADOS EN EL HOSPITAL DR. EXEQUIEL GONZÁLEZ CORTÉS**

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

LESTER ANTONIO MUÑOZ TOLEDO

PROFESOR GUÍA:
SANDRA DE LA FUENTE GONZÁLEZ

MIEMBROS DE LA COMISIÓN:
CLAUDIO GUTIÉRREZ GALLARDO
AIDAN HOGAN

SANTIAGO, CHILE
2020

RESUMEN

El Hospital Dr. Exequiel González Cortés es un establecimiento de alta complejidad, autogestionado y dependiente del Servicio de Salud Metropolitano Sur; en su servicio de hospitalización tiene disponibles aproximadamente 120 camas, y cuenta con un sistema de información hospitalario llamado coloquialmente “ficha” que es usado durante las atenciones.

En la ficha se registra la información de las atenciones y evaluaciones realizadas durante la atención; estas evaluaciones pueden tener indicaciones que deben ser ejecutadas por el personal clínico, y estas indicaciones a su vez generan tareas que deben ser realizadas oportunamente.

Dependiendo del tipo de indicación la realización de éstas se programa de forma externa a la ficha o en algunos casos simplemente no existe una programación y debido a esto no queda un registro electrónico de la programación ni una forma de comprobar que se esté cumpliendo de manera correcta con la programación.

En este trabajo se detalla el diseño e implementación de las funcionalidades necesarias en la ficha para poder programar tareas generadas de las indicaciones, haciendo un seguimiento de estas tareas desde el momento en que son programadas hasta que son realizadas o canceladas; para esto se realizaron modificaciones en el modelo de datos, se realizaron cambios en la forma en que se guardan las ejecuciones de la indicación, se modificó la API y se crearon nuevas vistas en la ficha para programar indicaciones y poder hacer seguimiento de las tareas generadas. Todo esto siguiendo los estándares de seguridad y usabilidad existentes en la ficha.

Adicionalmente se crea un monitor de tareas que permite tener un resumen de todas las programaciones que se realizan en un área en particular, incluyendo alertas en caso de que existan atrasos en las tareas pendientes, y en caso de que existan indicaciones sin programación; este monitor es construido pensando en una arquitectura basada en eventos para mostrar información en tiempo real.

Se cumple con los objetivos específicos propuestos y se logró un sistema funcional. Se pasó el sistema a los servidores de prueba del hospital donde se realizaron pruebas para validar el correcto funcionamiento de todas las nuevas funcionalidades implementadas.

*A mi familia.
Por su apoyo incondicional.*

Tabla de Contenido

1. Introducción	1
1.1. Contexto	1
1.2. Problema	2
1.3. Objetivos	4
1.4. Descripción general de la solución	4
1.4.1. Back-end	5
1.4.2. Front-end	5
1.4.3. Monitor de tareas	6
2. Marco Teórico	7
2.1. Tecnología usada	7
2.1.1. Back-end (API)	8
2.1.2. Front-end (ficha)	8
2.1.3. Monitor de Tareas	8
2.2. Metodología de trabajo	9
3. Problema	10
4. Solución	13
4.1. Back-end	15
4.1.1. Bases de datos	15
4.1.1.1. Tabla schedule	18
4.1.1.2. Tabla program_type	19
4.1.1.3. Tabla program	19
4.1.1.4. Tabla program_types	20
4.1.1.5. Modelo actualizado	21
4.1.2. Kafka	22
4.1.3. API	23
4.2. Front-end	25
4.2.1. Plan de la Atención	25
4.2.2. Tratamientos y cuidados	30
4.2.3. Exámenes y procedimientos	32
4.3. Monitor de Tareas	34
5. Validación	36
5.1. Encuesta	38
5.1.1. Preguntas sobre el plan de la atención en la ficha	38

5.1.2. Preguntas sobre el monitor de tareas.	38
6. Conclusiones	40
Glosario	42
Acrónimos	43
Bibliografía	44
Anexo A. Fotos	45
A.1. Fotos Flamenco	45
A.2. UPC	49

Índice de Ilustraciones

1.1.	Resumen de Indicaciones en evaluación clínica	3
4.1.	Modelo de datos previo a las modificaciones	17
4.2.	Modelo de datos final incluyendo las modificaciones y las tablas nuevas	21
4.3.	Sección del plan de la atención	25
4.4.	Pop-up que muestra el historial de programaciones de la indicación.	27
4.5.	Vista del Plan de la Atención en la pestaña Cuidados	28
4.6.	Pop-up para editar un programa.	29
4.7.	Pop-up para crear un programa.	30
4.8.	Vista de medicamentos activos.	30
4.9.	pop-ups de administraciones y de historial.	31
4.10.	Vista de cuidados activos.	32
4.11.	Vista de otras indicaciones activas.	32
4.12.	Vista de indicaciones de tipo estudio diagnóstico.	33
4.13.	Vista de indicaciones de exámenes de laboratorio y UMT.	33
4.14.	Monitor de tareas.	35
5.1.	Ejemplo de un issue en el Gitlab.	37
A.1.	Tarjeta de programación de medicamentos por horarios	45
A.2.	Tarjeta de programación de medicamentos SOS	46
A.3.	Tarjeta de programación de tratamiento endovenoso SOS	46
A.4.	Tarjeta de programación de tratamiento endovenoso por horarios	47
A.5.	Tarjeta de programación de medicamentos por horarios	47
A.6.	Hoja con listado de indicaciones de paciente oncológico.	48
A.7.	Tarjeta con listado de medicamentos(horarios y SOS) paciente UPC.	49

Capítulo 1

Introducción

Este capítulo contiene un resumen del trabajo realizado y este está dividido en 4 secciones, la primera sección corresponde al contexto donde se introduce el contexto y la relevancia de este trabajo, le sigue la sección de problema donde se explica en mayor detalle el problema que se busca solucionar, continuando con la sección de objetivos donde se define el objetivo general junto con los objetivos específicos y finalmente la última sección contiene una descripción general de la solución donde se describe brevemente las partes que componen el trabajo realizado.

1.1. Contexto

El Hospital Dr. Exequiel González Cortés (HEGC) es un establecimiento de alta complejidad, autogestionado y dependiente del Servicio de Salud Metropolitano Sur. Se encuentra ubicado en la comuna de San Miguel y atiende a una población asignada de 300.000 niños y adolescentes de 11 comunas de la zona sur de Santiago [4].

El HEGC cuenta con equipos multidisciplinarios comprometidos con el más alto nivel de excelencia profesional y tecnológico. En consecuencia, en el 2007 obtiene la calidad de Hospital Autogestionado en Red, y en el año 2011 se acredita en Calidad y Seguridad del Paciente, siendo el primer hospital y la segunda organización pública en obtenerla. Dicha acreditación ha sido renovada en los años 2015 y 2019, convirtiéndose en el primer hospital en acreditarse tres veces consecutivas [5].

El Hospital cuenta con las siguientes atenciones:

- Unidad de Emergencia 24 horas
- Unidad de Cuidados Intensivos (UCI)
- Unidad de Tratamiento Intermedio (UTI)
- Unidad de hospitalización Médico-Quirúrgica (MQ)
- Consultas médicas, con 20 especialidades

Anualmente, el hospital realiza aproximadamente 92.000 atenciones de urgencia, 86.000 consultas de especialidad, 4.000 cirugías electivas, y 7.000 atenciones de hospitalización. Para esto último, cuenta con 120 camas distribuidas en las unidades UCI, UTI y MQ.

En octubre del año 2017 el Hospital se traslada a un nuevo edificio, con lo cual se decide construir e implementar Ámbar, un Sistema de Información Hospitalario (HIS), donde se registra la información de las distintas atenciones y se coordinan los servicios de apoyo clínico (laboratorio, imágenes, etc) para la atención de todos los pacientes.

En particular, en Ámbar se realizan las indicaciones médicas para los pacientes hospitalizados, que posteriormente son programados por las enfermeras y ejecutados por el personal clínico.

En la actualidad, la programación de la realización de las indicaciones médicas se debe realizar manualmente en hojas de enfermería destinadas para ello. Para el equipo clínico esto implica ingresar a la ficha clínica de cada paciente y revisar el listado de indicaciones que posee cada uno por separado para luego poder realizar las acciones correspondientes. Esto significa que el equipo clínico debe revisar constantemente el listado de indicaciones médicas de cada uno de los pacientes hospitalizados, para ejecutar todas las tareas relacionadas a las indicaciones cuando correspondan y con la frecuencia oportuna.

Considerando un escenario con ocupación total de camas, además que cada paciente hospitalizado tiene en promedio 15 indicaciones médicas (medicamentos, cuidados, exámenes, interconsultas, etc.), y que cada indicación médica genera entre 1 a 5 tareas (por ejemplo, un medicamento puede estar indicado para administrar a las 15:00, 23:00 y 07:00 hrs), entonces el personal de enfermería debe generar, programar y controlar la ejecución de un orden de 5.000 tareas al día.

Con este trabajo de espera poder reducir la posibilidad de que ocurran errores médicos, en particular evitar que no se realicen tareas asociadas a indicaciones por olvido o confusión. Debido a que a no realización de alguna de estas tareas puede causar serios daños en la salud o el tratamiento del paciente, incluso la muerte.

Un estudio estima que los errores médicos son la tercera mayor causa de muerte en Estados Unidos [7] y usando la Clasificación internacional de enfermedades (CIE-10), no existe el error médico como causa de muerte [7], por lo que es difícil estimar la cantidad de muertes ocurridas por errores médicos. Debido a esto es de suma importancia contar con sistemas que puedan ayudar a reducir la posibilidad de que estos errores ocurran.

1.2. Problema

Durante la atención de los pacientes se pueden realizar distintas evaluaciones, éstas pueden incluir diversas indicaciones.

La Figura 1.1 muestra la sección de indicaciones dentro de una evaluación clínica; en la parte superior se muestra el menú usado para realizar nuevas indicaciones estas se dividen en 3 categorías y cada categoría incluye varios tipos de indicaciones; bajo esta lista existe un resumen con todas las indicaciones activas (de evaluaciones anteriores) y las indicaciones nuevas que forman parte de la evaluación actual, una vez terminada la evaluación clínica todas las nuevas indicaciones son guardadas y quedan activas para ser ejecutadas [6].

EVALUACIÓN CLÍNICA																													
INDICACIONES																													
TRATAMIENTOS Y CUIDADOS			EXÁMENES Y PROCEDIMIENTOS				OTROS																						
MEDICAMENTOS	CUIDADOS	INTERVENCIÓN EN PABELLÓN	LABORATORIO / UMT	IMAGENOLÓGIA	ESTUDIO DIAGNÓSTICO	PROCEDIMIENTOS	INTERCONSULTAS	OTRAS INDICACIONES																					
<div style="display: flex; justify-content: space-between;"> <div style="width: 15%;"> <p>Todas 18</p> <p>Medicamentos 1</p> <p>Cuidados 15</p> <p>Intervención Quirúrgica 2</p> </div> <div style="width: 85%;"> <p>MEDICAMENTOS</p> <ul style="list-style-type: none"> Paracetamol (Oral) 1mg 2. Vía: Oral NUEVO <p>CUIDADOS</p> <table border="0" style="width: 100%;"> <tr> <td style="width: 50%;">Colchón antiescaras (04-07-19)</td> <td style="width: 5%; text-align: center;">✘</td> <td style="width: 45%;">Cambios de posición (04-07-19)</td> <td style="width: 5%; text-align: center;">✘</td> </tr> <tr> <td>Protección de zonas de apoyo (04-07-19)</td> <td style="text-align: center;">✘</td> <td>Lubricación de la piel (04-07-19)</td> <td style="text-align: center;">✘</td> </tr> <tr> <td>Educación a los padres (04-07-19)</td> <td style="text-align: center;">✘</td> <td>Uso de contención mecánica (04-07-19)</td> <td style="text-align: center;">✘</td> </tr> <tr> <td>Mantener cama o cuna frenada (04-07-19)</td> <td style="text-align: center;">✘</td> <td>Mantener vigilancia (04-07-19)</td> <td style="text-align: center;">✘</td> </tr> <tr> <td>Mantener barandas en alto (04-07-19)</td> <td style="text-align: center;">✘</td> <td>Aspiración de secreciones (04-07-19)</td> <td style="text-align: center;">✘</td> </tr> </table> </div> </div>										Colchón antiescaras (04-07-19)	✘	Cambios de posición (04-07-19)	✘	Protección de zonas de apoyo (04-07-19)	✘	Lubricación de la piel (04-07-19)	✘	Educación a los padres (04-07-19)	✘	Uso de contención mecánica (04-07-19)	✘	Mantener cama o cuna frenada (04-07-19)	✘	Mantener vigilancia (04-07-19)	✘	Mantener barandas en alto (04-07-19)	✘	Aspiración de secreciones (04-07-19)	✘
Colchón antiescaras (04-07-19)	✘	Cambios de posición (04-07-19)	✘																										
Protección de zonas de apoyo (04-07-19)	✘	Lubricación de la piel (04-07-19)	✘																										
Educación a los padres (04-07-19)	✘	Uso de contención mecánica (04-07-19)	✘																										
Mantener cama o cuna frenada (04-07-19)	✘	Mantener vigilancia (04-07-19)	✘																										
Mantener barandas en alto (04-07-19)	✘	Aspiración de secreciones (04-07-19)	✘																										

Figura 1.1: Resumen de Indicaciones en evaluación clínica

En el sistema actual de ficha clínica las indicaciones se diferencian en 11 categorías; cada una de ellas incluye varios tipos de indicaciones, por ejemplo, la categoría de medicamentos incluye indicaciones de tipo fármaco, Bombas de infusión continua (BIC), fleboclisis, entre otras. Cada una de ellas puede generar distintos tipos de tareas, por ejemplo, un fármaco puede estar indicado con una administración por horarios y también estar indicado como SOS¹ (es necesario administrarlo al cumplir una condición SOS como fiebre sobre 40° o dolor), una indicación del tipo fleboclisis puede ser constante, una de aseo bucal puede estar indicada de forma post-prandial (después de la comida).

Para poder ejecutar las indicaciones el equipo clínico debe ingresar a la ficha electrónica y revisar el listado de estas en cada paciente por separado, ya que no existe una forma de ver un resumen agrupado por área o especialidad; luego en caso de que la indicación necesite programación (ej. medicamento c/8hrs) debe registrar en algún lugar externo a la ficha clínica esta información y en base a esto se programa para realizar las tareas necesarias en el momento oportuno.

En la ficha clínica solo se puede registrar un tipo de acción relacionado con las indicaciones que es “administrar” para el caso de las indicaciones en la categoría de medicamentos y “realizar” para las otros tipos de indicaciones (cuidados, exámenes, procedimientos y otras indicaciones).

Debido a esto para diferenciar entre los distintos tipos de tareas es necesario que la persona que ejecute la indicación escriba en el campo de comentario de la administración o realización el tipo de tarea al que corresponde esta acción escribiendo en el comentario por ejemplo “administración de medicamento SOS” o “administración correspondiente a las 20:00”, pero sólo queda registro de donde se marco explícitamente la realización o no realización de la indicación y por esto no existe una forma de corroborar si se registraron todas las tareas correspondientes a una indicación.

Para la administración de medicamentos el equipo clínico usa tarjetas de papel en la que escriben el nombre del paciente, el medicamento, dosis y a la hora en que debe ser administrado. Luego todas las tarjetas son ordenadas por hora, y antes del horario programado,

¹ Del latín “si opus sit” que significa “si es necesario”

una persona prepara los medicamentos. Finalmente a la hora programada estos son administrados a los pacientes hospitalizados y se registra en la ficha clínica de cada paciente que el medicamento fue administrado.

Toda la programación de indicaciones se realiza de forma manual, si bien funciona en la práctica, es necesario mejorar su operatividad debido a que no hay un control riguroso, así como tampoco los medios de verificación cabal de las actividades realizadas, lo que puede generar diversos problemas en caso de que alguna de estas tarjetas de programación se pierda o se traspapelen. Un error en el cumplimiento de las indicaciones médicas puede ser dañino para la salud de los pacientes.

1.3. Objetivos

Objetivo General

Optimizar el cumplimiento y ejecución oportuna de las indicaciones médicas de los pacientes hospitalizados, a través de un sistema de programación y monitoreo de las tareas derivadas de las indicaciones.

Objetivos Específicos

1. Determinar los tipos de indicaciones médicas que se generan y cómo cada uno de estos debe ser presentado y manejado por el sistema.
2. Definir un modelo de datos que soporte las funcionalidades esperadas del sistema.
3. Obtener las indicaciones médicas en el momento que son indicadas.
4. Generar un plan de atención del paciente donde se programen las tareas derivadas de las indicaciones médicas.
5. Crear las vistas necesarias con las tareas que debe realizar el equipo clínico en el día diferenciando por área o especialidad según las necesidades de los usuarios.
6. Generar procedimientos que alerten sobre el incumplimiento en la realización de las tareas programadas de las indicaciones médicas.

1.4. Descripción general de la solución

A partir de la situación actual, se requiere mejorar los procedimientos para el monitoreo y la gestión de las indicaciones médicas con la finalidad de poder programar tareas asociadas a estas.

1.4.1. Back-end

Para la solución del problema se modificó el modelo de datos, agregando las tablas necesarias para guardar programaciones y que estas queden asociadas a indicaciones.

Para guardar las tareas se usó una tabla existente que permite guardar tareas, esta tabla se usaba para guardar las programaciones de las intervenciones en pabellón (el único tipo de indicación programable previo a este trabajo y existen en un sistema de pabellón distinto de la ficha).

Para la arquitectura basada en eventos fue necesario instalar y configurar un cluster de servidores de Apache Kafka, una plataforma de transmisión de datos que funciona de manera distribuida y que permite comunicar eventos en tiempo real.

Se modificaron los end-points de la API para poder enviar los eventos necesarios a Kafka cuando ocurrieran cambios en las indicaciones y tareas, adicionalmente se crearon los end-points necesarios para crear programaciones.

1.4.2. Front-end

Para cumplir con los objetivos fue necesario crear una nueva sección en la ficha que permitiera la creación y realizar seguimiento a las tareas generadas a partir de las indicaciones; para esto se creó una nueva sección en la ficha clínica llamada “Plan de la atención”; esta sección tiene 5 pestañas que se detallan a continuación:

La primera pestaña es una vista general de todas las indicaciones programadas y además incluye una línea de tiempo que muestra la cantidad de tareas programadas para las próximas 24 horas en el paciente.

Las otras 4 pestañas son de programación, donde se puede programar las indicaciones agrupadas en 4 categorías que son: medicamentos, cuidados, exámenes y otras indicaciones.

En estas pestañas de programación se muestran 2 listas correspondientes a las indicaciones programadas y a las indicaciones sin programar del tipo seleccionado.

En la sección de tratamientos y cuidados se modificó la pestaña de administrar medicamentos y realizar indicaciones de cuidados y otras indicaciones; en estas pestañas, en el listado de indicaciones, se puede ver la programación que tiene cada indicación junto con el horario de la próxima administración en caso que corresponda y en caso de existir se muestra la tarea con mayor atraso.

En caso que la indicación sea programada con horarios, al momento de hacer una realización (o administración) es necesario seleccionar el horario programado correspondiente al registro que se está realizando.

Finalmente para hacer seguimiento de todas las tareas asociadas a una indicación se modificó el pop-up de las historial de la indicación para mostrar el tipo de programación

elegido al momento de la realización y el horario elegido en caso que corresponda.

Para la programación de los exámenes se modificó la pestaña de exámenes y procedimientos, agregando a la lista de exámenes pendientes la programación asociada.

1.4.3. Monitor de tareas

Se creó un monitor de tareas en un nuevo proyecto que permite tener una visión general de la cantidad de tareas agrupadas por área del hospital; este monitor muestra una lista de todos los pacientes por área junto con la cantidad de tareas programadas para las próximas 2 horas y tiene indicadores que permiten ver de manera fácil pacientes que tienen indicaciones sin programar o que tienen tareas atrasadas.

Capítulo 2

Marco Teórico

Este capítulo se divide en dos secciones, la primera sección corresponde a las tecnologías usadas donde se incluyen las tecnologías relevantes junto con una breve descripción de su uso; y la segunda sección corresponde a la metodología de trabajo.

2.1. Tecnología usada

Este trabajo se divide en 3 partes, las modificaciones a la API (back-end), las modificaciones a la ficha (front-end), y el monitor de tareas (front-end y back-end).

En el hospital existe una arquitectura de microservicios, donde se manejan muchas aplicaciones relativamente pequeñas con objetivos específicos. La comunicación entre aplicaciones ocurre generalmente mediante distintas APIs REST, por lo que es posible que distintas aplicaciones sean escritas en distintos lenguajes de programación según los requerimientos específicos de estas.

En este trabajo también se propone una arquitectura basada en eventos[8] como alternativa al uso de API REST para la comunicación de distintas aplicaciones, donde las aplicaciones emitan eventos cuando ocurran ciertas acciones que puedan ser relevantes para otras aplicaciones, y estos eventos son recibidos de manera asíncrona por las aplicaciones que esperan por estos eventos.

Con una arquitectura basada en eventos se puede reducir la complejidad de las aplicaciones dado que estas solo se encargan de emitir eventos relevantes sin necesidad de conocer la API de las aplicaciones que leen estos eventos. Otra ventaja es que la aplicación que emite eventos no necesita esperar que el mensaje sea entregado simplemente debe asegurarse que el mensaje haya quedado publicado.

Una desventaja de una arquitectura basada en eventos es que el emisor de eventos no tiene forma de corroborar que estos eventos han sido recibidos de manera correcta por el consumidor; pero en el caso de uso propuesto para el monitor de tareas, esto no es un problema, ya que la API debe seguir funcionando incluso si el monitor de tareas no está actualizando su información correctamente.

2.1.1. Back-end (API)

La API está escrita en php[2] un lenguaje de programación de propósito general, interpretado y muy usado en el desarrollo web; esta API usa el framework SLIM[3] un micro framework para php diseñado para hacer aplicaciones webs y APIs. Para el almacenamiento de los datos la API se conecta a una base de datos relacional SQL MariaDB[1] que es una popular base de datos de código abierto.

2.1.2. Front-end (ficha)

El front-end de la ficha está escrito en HTML, CSS y Javascript, en esta parte es donde conviven la mayor cantidad de frameworks y librerías distintos como jquery, handlebars y vue-js. La nueva sección de la ficha correspondiente al plan de la atención fue escrita usando el framework vue-js, más algunas librerías como moment.js y Axios para el formato de fechas y llamadas AJAX respectivamente.

Para las modificaciones dentro de la ficha se mantuvo las librerías que usaban las componentes, como handlebars en el caso de la pestaña de exámenes y vue-js en la pestaña de tratamientos y cuidados.

2.1.3. Monitor de Tareas

Se decidió usar Javascript para escribir el monitor de tareas usando Node.js, de esta forma se puede usar el mismo lenguaje tanto para el front-end como para el back-end, como librería para el front-end se usa vue-js al igual que en la ficha. En el servidor se usa express.js como servidor web junto con Socket.IO para la comunicación bidireccional del monitor de tareas.

Para poder usar una arquitectura de eventos es necesario usar un tipo de cola de eventos para esto se evaluaron alternativas como RabbitMQ y Apache Kafka[10], usando estas herramientas como colas de mensajes.

RabbitMQ tiene las funcionalidades necesarias para recibir los eventos y crear colas para que estos sean leídos por los consumidores, pero tiene algunas desventajas si se pretende tener una arquitectura basada en eventos más genérica, como el hecho que los mensajes una vez consumidos son olvidados y esto obligaría a crear una cola distinta por cada aplicación que quiera consumir estos eventos; esto significaría que en caso de que el monitor de tarea funcione como un microservicio y se tenga una instancia por cada área del hospital, se necesitarían tantas colas en RabbitMQ como áreas tenga el hospital.

Apache Kafka en cambio deja que cada consumidor maneje el orden en que lee los mensajes[9], permitiendo que muchos consumidores lean los mismos mensajes compartiendo los mismos tópicos; también Kafka escala de manera horizontal, esto es al agregar más servidores la carga de estos se reparte incluso los mensajes de un mismo tópico y esto permite tener mejor rendimiento y redundancia de los mensajes.

Finalmente para que el monitor de tareas pueda leer los mensajes de Kafka se decidió

usar la librería `node-rdkafka`¹.

2.2. Metodología de trabajo

Este trabajo fue realizado en el contexto de una metodología ágil, por lo que constantemente se mantenían reuniones para mostrar el estado del trabajo y definir los cambios necesarios o las nuevas funcionalidades a implementar. De la misma forma se tenían reuniones frecuentes con los desarrolladores de la API para validar los cambios propuestos en las bases de datos.

Se usó el sistema de issues de Gitlab para mantener un registro de todas las funcionalidades requeridas y los bugs encontrados.

Para las nuevas funcionalidades definidas en las reuniones de requerimientos o en las revisiones, estas se dividían en múltiples issues que representan tareas o funcionalidades específicas requeridas, luego son evaluados en cuanto a su complejidad, para estimar un tiempo de desarrollo y en base a esto se priorizan para su desarrollo.

Una vez priorizados los issues, estos se desarrollan comenzando con los de mayor prioridad y al ser resueltos se marcan con un tag para su posterior revisión. Para cerrar el issue es necesario que este sea revisado por alguien distinto a quien realizó su solución, y una vez que se comprueba el issue está correctamente resuelto este se cierra.

En caso que se encuentren nuevos bugs o errores durante las revisiones, estos también son agregados como issues para luego ser priorizados.

Para las vistas se contó con la ayuda de la diseñadora de la ficha. Para la creación de cada vista primero se crearon las componentes agregando las reglas de estilo mínimas para poder estructurarlas y se agregan clases para que la diseñadora pueda modificar el estilo de las nuevas componentes de forma que sigan el mismo estilo gráfico que el resto de las componentes de la ficha. La coordinación con la diseñadora fue a través de los issues de gitlab, donde se le asignaban los issues en los que se necesitara ayuda con el estilo y donde ella podía comentar en caso de necesitar algún cambio en la estructura de las componentes o nuevas clases en los elementos HTML.

¹ <https://github.com/Blizzard/node-rdkafka>

Capítulo 3

Problema

En este capítulo se detalla el problema, y se exponen los principales requisitos, incluyendo requisitos de seguridad y usabilidad.

Ámbar es el Sistema de Información Hospitalario (HIS) usado en el HEGC, dentro de este sistema está la ficha electrónica de los pacientes, esta ficha se usa por el personal clínico durante las atenciones para registrar la información del paciente y se utiliza para las atenciones ambulatorias, atenciones de pacientes hospitalizados así como también en las atenciones de urgencia.

Durante las atenciones se realizan distintas evaluaciones por parte del personal clínico y éstas se registran en la ficha, las que pueden ser de distintos tipos, como por ejemplo: evaluación clínica, de cuidados, nutricional, funcional, etc. Algunas de estas evaluaciones solo permiten escribir texto como es el caso en la evaluación de salud mental y la evaluación familiar y social, mientras que otros tipos de evaluaciones pueden generar indicaciones como es el caso de las evaluaciones clínicas y las evaluaciones de cuidados.

En caso que se realice una evaluación que incluya indicaciones, éstas se registran en el sistema, actualmente existen 11 categorías de indicaciones.

1. Medicamentos
2. Cuidados
3. Intervenciones en pabellón
4. Exámenes de laboratorio
5. Exámenes de imagenología
6. Estudios diagnóstico
7. Procedimientos
8. Interconsultas
9. Indicaciones al alta

10. Controles

11. Otras indicaciones

Luego estas 11 categorías de indicaciones se subdividen en tipos de indicación, por ejemplo la categoría de medicamentos se subdivide según el tipo de administración en fármaco, Bombas de infusión continua (BIC), fleboclisis, etc. Y el responsable del cumplimiento de cada indicación dependerá del tipo de ésta.

Algunas de estas indicaciones pueden generar tareas a realizar por el personal clínico es necesario poder llevar un registro de estas, actualmente en la ficha existen secciones donde se lleva un registro de las acciones en las indicaciones, en particular existe una pestaña de tratamientos y cuidados donde se puede registrar la administración o no administración de un medicamento y también se pueden registrar acciones en indicaciones de cuidados y otras indicaciones. También existe una pestaña de exámenes y procedimientos donde se puede registrar la ejecución de estos.

Pero en la ficha solo se puede registrar las ejecuciones de las indicaciones y no existe una forma de programar las ejecuciones o comprobar que se estén realizando de manera correcta y oportuna.

Para algunas indicaciones como es el caso de las indicaciones de medicamentos el personal clínico usa tarjetas de papel donde llevan registro de los distintos medicamentos que deben administrar en cada paciente con los horarios correspondientes de administración. Pero esta programación solo se realiza en las tarjetas de papel no queda ningún tipo de registro electrónico sobre la programación de las administraciones de los medicamentos y por esto es imposible comprobar de una forma fácil si se están administrando todos los medicamentos indicados con la frecuencia correspondiente.

También hay indicaciones donde es necesario registrar sus ejecuciones pero donde no existe una programación en papel y solo se registra en la ficha su ejecución como por ejemplo una indicación de baño diario o muda frecuente, para este tipo de indicaciones también sería útil poder realizar una programación para que el personal clínico pueda organizar su tiempo de mejor manera sabiendo la cantidad de tareas que tiene que hacer en un horario específico y tener alertas en caso de que una tarea programada esté atrasada.

Es por esto que es necesario en la ficha clínica una sección que permita programar ejecuciones (generar tareas) para ciertos tipos de indicaciones y luego poder registrar el correcto y oportuno cumplimiento de las tareas programadas al ejecutar la indicación (administrar medicamentos, realizar cuidado, etc.).

Debido a que es necesario que este trabajo este incluido dentro de la ficha clínica como una nueva sección, es necesario que haga uso de los sistemas existentes de autenticación y de permisos de usuarios, y de esta forma poder cumplir con los estándares de seguridad y privacidad de los datos generados. De la misma forma las nuevas vistas deben tener un diseño acorde al diseño gráfico de la ficha y las nuevas funcionalidades deben ser de fácil uso de forma que no generen un obstáculo para el trabajo del personal clínico.

Para poder hacer un seguimiento de las tareas programadas y tener una visión general

de éstas es necesario también tener un monitor de tareas donde se pueda ver la cantidad de tareas programadas por paciente, pero con mayor importancia, que muestre alertas en caso de que el paciente tenga indicaciones sin programar y en el caso que el paciente tenga tareas programadas atrasadas. De esta forma se podrá tener una visión general de la carga de trabajo que existe a cada hora y también poder hacer un seguimiento del cumplimiento de las tareas programadas y de esta forma poder hacer un mejor seguimiento al cumplimiento de las indicaciones. Adicionalmente es deseado que el monitor de tareas muestre la información lo más actualizada posible y que refleje los cambios en las tareas a medida que estos ocurren.

Capítulo 4

Solución

En este capítulo se muestra la solución en detalle, comenzando por una descripción de los elementos generales de la solución, luego se separa la solución en 3 principales secciones: el back-end que incluye cambios en la base de datos, envíos de mensajes a Kafka y cambios en la API; el front-end que corresponde a las nuevas secciones en la ficha y las modificaciones realizadas; y finalmente el monitor de tareas.

Como primer paso fue necesario definir los tipos de indicaciones programables y los tipos de programación que puede tener cada indicación según su tipo; en primera instancia se pensó en permitir poder programar todas las indicaciones que se pueden ejecutar en la pestaña de tratamientos y cuidados; estas serían las categorías de *medicamentos*, *cuidados* y *otras indicaciones*, pero existen indicaciones de *cuidados* que se ejecutan de una forma especial como son las indicaciones de *régimen*, *reposo*, *oxigenoterapia* y *avisar si*, este tipo de indicaciones no se muestran de la misma forma que las otras indicaciones en la tabla por lo que se decidió que este tipo de indicaciones no permitirá programación.

Con ayuda de una médico pediatra del área de MQ se definieron como programables los siguientes tipos de indicaciones agrupados por tipo:

- **Medicamentos:** Farmaco, BIC y fleboclisis.
- **Cuidados:** Hemoglucotest, otros cuidados, baño diario y cambio de ropa de cama, aseo bucal, aseo nasal, aseo ocular, muda y/o aseo genital, aspiración de secreciones, mantener vigilancia, mantener barandas en alto, mantener cama o cuna frenada, uso de contención mecánica, educación a los padres, lubricación de la piel, protección de las zonas de apoyo, colchón antiescaras y cambios de posición.
- **Exámenes y procedimientos:** Exámenes de laboratorio, exámenes de imagenología, estudio diagnóstico y procedimientos.

Estos son los tipos de indicaciones donde existe un registro de sus realizaciones y que se realizan en la ficha en las pestañas de tratamientos y cuidados o en la pestaña de exámenes y procedimientos, pero esto no incluye todas las indicaciones que se hacen con horarios y a una frecuencia determinada; un ejemplo de esto son las indicaciones de toma de signos vitales

que generalmente se indica con una frecuencia específica (ej. $c/4 \text{ hrs}^1$) pero el registro de éstos se realiza en una pestaña distinta dentro de la ficha, y por esto la realización de este tipo de indicaciones se registra de manera diferente al resto de las indicaciones y no existe la necesidad que sean programables.

Luego se definieron todos los tipos de programaciones posibles para el plan de la atención y se llegó a la siguiente lista: Horarios, SOS, Constante, Sólo una vez, A diario, Una vez por turno, Pre-prandial, Post-prandial y Post baño o muda. Se definió que estos tipos de programaciones son válidos para todos los tipos de indicaciones y que la aplicación no restringiría los tipos de programa según el tipo de indicación y que quedaría a criterio del personal clínico encargado de hacer la programación de asegurarse que se elija un tipo de programación que tenga sentido para la indicación.

Se definió que los programas de indicaciones de exámenes, procedimientos y medicamentos solo se deben tener un tipo de programación, mientras que los otros tipos de indicaciones como cuidados y otras indicaciones deben permitir una programación con tipos de horarios y SOS simultáneamente, esto es porque en caso de que sea necesario realizar un examen, procedimiento o administrar un medicamento con horarios predeterminados y de forma SOS, se deben realizar 2 indicaciones distintas; una correspondiente a cada tipo de programación.

Como restricción adicional en las indicaciones de exámenes si se programa con horarios se debe limitar la cantidad de tareas generadas a 1 tarea, ya que esta tarea corresponde generalmente a la toma de muestra de los exámenes y esta se realiza una sola vez por indicación, cabe mencionar que en caso que se necesite repetir exámenes se debe crear nuevas indicaciones de exámenes y al ser indicaciones distintas se pueden programar de manera independiente.

Una vez terminadas estas definiciones el proyecto se separó en 3 partes: el back-end que corresponde a todos los cambios en el modelo de datos y la API para soportar las nuevas funcionalidades; el front-end que corresponde a los cambios en la ficha para poder realizar las programaciones y hacer un correcto seguimiento de las tareas, esto incluye todas las vistas con las que los usuarios interactúan dentro de la ficha; y finalmente el monitor de tareas que permite hacer un seguimiento de las tareas programadas.

¹ Control de signos vitales cada 4 horas

4.1. Back-end

4.1.1. Bases de datos

El trabajo realizado en el back-end se divide en 3 partes, primero los cambios al modelo de datos, luego el uso de Kafka y finalmente los cambios en la API.

Para poder soportar las nuevas funcionalidades requeridas para el plan de la atención fue necesario modificar el modelo de datos, agregando nuevas tablas a la base de datos y modificando otra, en la Figura 4.1 se pueden ver las tablas de la base de datos relevantes para este proyecto, y las relaciones entre ellas. A continuación se agrega una corta descripción de cada una de ellas.

- **patient:** Cada fila de esta tabla corresponde a un paciente y se guardan los datos propios del paciente como nombres, fecha de nacimiento, RUT, etc.
- **care_manager:** Esta tabla se usa para agrupar atenciones, cada fila de esta tabla está asociado a un paciente y en general corresponde a una visita al hospital.
- **location:** Esta tabla corresponde a una lista de las posibles ubicaciones en las que puede ocurrir una atención; cada fila corresponde a una ubicación distinta. En el caso de los pacientes hospitalizados, cada cama corresponde a una ubicación. Otros ejemplos de ubicaciones son salas de espera, box de atención ambulatoria, pabellones, etc.
- **patient_service:** Esta tabla corresponde a una atención y cada fila está asociada a un care_manager. Pueden haber varias atenciones asociadas al mismo care_manager, por ejemplo si un paciente se atiende por urgencia y luego es hospitalizado, sus atenciones de hospitalización y de urgencia serían distintas pero asociadas al mismo care_manager.
- **clinician:** Esta tabla contiene la lista de los usuarios que pueden usar la ficha. Cada fila en esta tabla corresponde a un usuario y contiene información adicional como la especialidad.
- **evaluation:** Corresponde a una evaluación del paciente realizada por el personal clínico; puede ser de distintos tipos como evaluación clínica, evaluación de cuidados, triage de urgencia, alta, etc.
- **indication:** Esta tabla contiene las indicaciones que se generan en las evaluaciones. Cada fila corresponde a una indicación, y se guarda una referencia a la evaluación a la que esta indicación pertenece y también existe una columna que indica el tipo de la indicación.
- **indication_category:** Esta tabla contiene una lista de las categorías de indicaciones que existen; se usa para agrupar indicaciones de distinto tipo que pertenecen a la misma categoría
- **indication_type:** Esta tabla tiene un listado de todos los tipos de indicaciones que se pueden crear. Tiene una columna donde se indica la categoría del tipo; por ejemplo aseo nasal y aseo ocular son dos tipos de indicaciones distintos, pero ambos pertenecen a la categoría de cuidados.

- **schedule**: Esta tabla corresponde a una lista de tareas; se ocupa para guardar las horas de las atenciones ambulatorias y también para las tareas de las indicaciones de pabellón.
- **schedule_state**: Esta tabla contiene un listado de todos los estados que puede tener una tarea como por ejemplo agendada, en curso, finalizado, cancelado, etc.

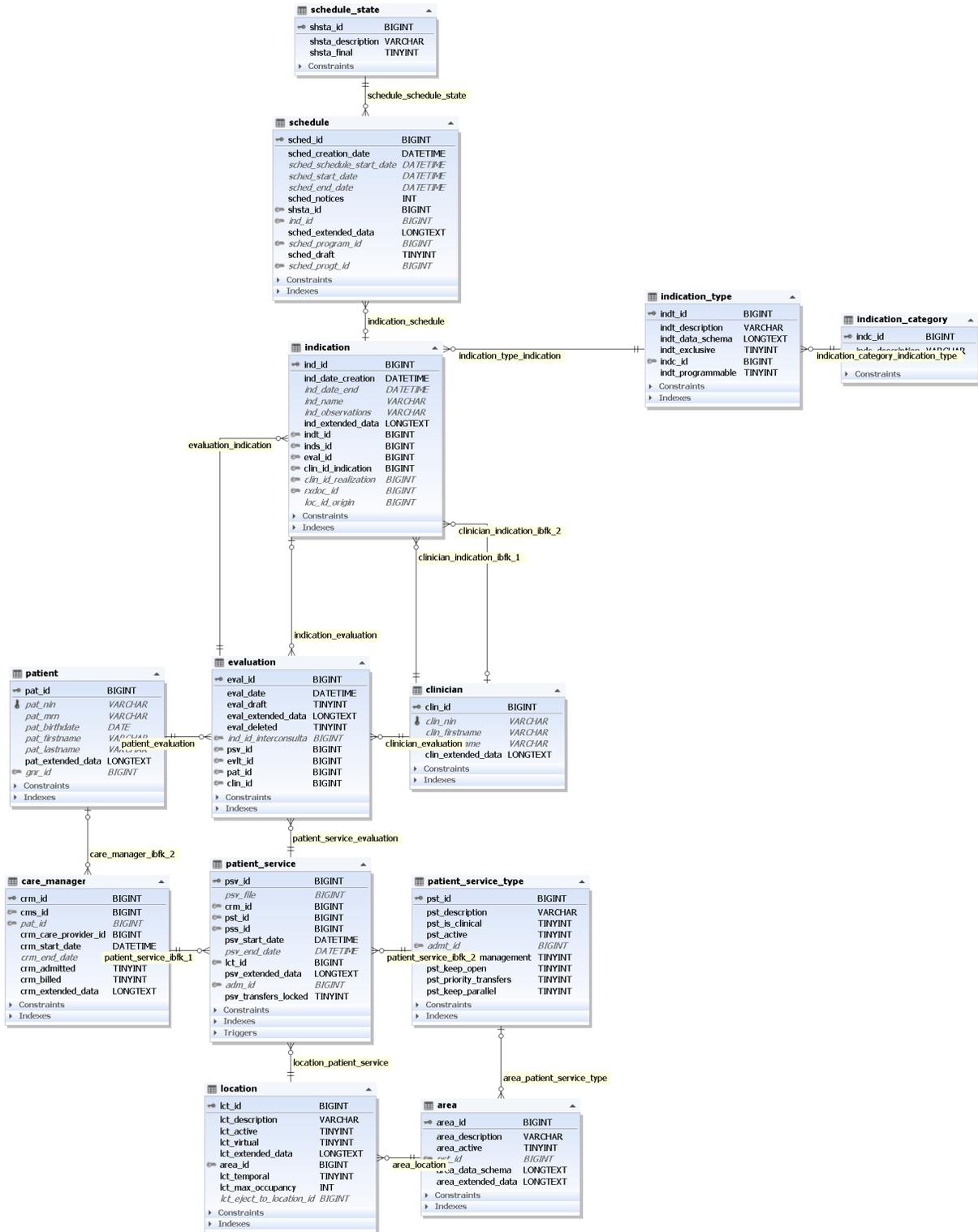


Figura 4.1: Modelo de datos previo a las modificaciones

Se modificó el modelo de datos agregando las tablas necesarias para poder guardar programaciones y que éstas queden asociadas a indicaciones. Para guardar las tareas se usó una tabla existente que permite guardar tareas; esta tabla se usaba para guardar las programaciones de las intervenciones en pabellón (el único tipo de indicación programable previo a

este trabajo, usadas en un sistema de pabellón distinto de la ficha).

Para poder guardar las programaciones y asociarlas a las tareas fue necesario crear 3 tablas nuevas: una tabla que contiene la información de la programación, una tabla que contiene los tipos de programaciones posibles (horarios, SOS, etc.) y una última tabla que asocia las programaciones a los tipos de indicaciones, esto debido a que la relación entre programa y tipo de programación es de N a N ya que una programación puede tener más de un tipo (horarios y SOS).

Usando las tablas descritas anteriormente es posible guardar en la base de datos toda la información necesaria para las programaciones de indicaciones; a continuación se describen las columnas de las tablas creadas o modificadas y su uso.

4.1.1.1. Tabla schedule

Cada fila de esta tabla corresponde a una tarea; antes de este proyecto se usaba exclusivamente para guardar las tareas de intervención en pabellón y agendas de atenciones ambulatorias. Las intervenciones en pabellón corresponden a un tipo de indicación por lo que tiene sentido usar la misma tabla para guardar todas las programaciones de tareas asociadas a indicaciones, y para el caso de las horas de atenciones ambulatorias aunque en la mayoría de los casos éstas no están asociadas a una indicación, existe indicación de tipo interconsulta que puede generar una tarea de atención ambulatoria por lo que una atención ambulatoria también puede quedar relacionada a una indicación.

Como esta tabla ya era una tabla existente en el modelo de datos actual a continuación se muestra el listado de las columnas que contiene.

- **sched_id**: Id única de cada tarea.
- **sched_creation_date**: Fecha de creación de la tarea calculada automáticamente usando la hora del servidor.
- **sched_schedule_start_date**: Fecha en la que esta tarea está programada para comenzar.
- **sched_start_date**: Fecha en la que se comenzó a realizar la tarea.
- **sched_end_date**: Fecha en la que se terminó de realizar la tarea.
- **sched_notices**: Número de veces que se ha llamado al paciente; se usa para las tareas de atención ambulatoria donde es necesario llamar a los pacientes antes de realizar la atención.
- **schsta_id**: Llave foránea a la tabla schedule_state que contiene la lista de los estados en los que puede estar una tarea (agendado, en curso, finalizado, cancelado, en espera, eliminado).
- **sched_extended_data**: JSON que puede contener información adicional que no se puede guardar en las otras columnas.

- **sched_program_id**: Id que hace referencia al programa dentro del cual esta tarea fue creada; esta es la única columna nueva que se agrega a esta tabla.
- **sched_draft**: Indica si esta tarea es un borrador; se usa para las tareas de atenciones que aún no han sido confirmadas.

A continuación se muestra el código de usado para modificar la tabla:

Código 4.1: Modificaciones a la tabla schedule.

```

1 ALTER TABLE schedule
2   ADD `sched_program_id` BIGINT(20) NULL DEFAULT NULL;
3
4 ALTER TABLE schedule
5   ADD CONSTRAINT `FK_schedule_program` FOREIGN KEY (`sched_program_id`)
6   ↪ REFERENCES `program` (`prog_id`);

```

4.1.1.2. Tabla program_type

Esta tabla contiene una lista de todos los tipos de programación válidos y contiene 2 columnas; se usa la columna de id como llave primaria y además contiene la restricción que el nombre del tipo de programación debe ser único.

- **progt_id**: Id única de cada tipo de indicación
- **progt_name**: Nombre del tipo de indicación

A continuación se muestra el código de creación de la tabla:

Código 4.2: Código CREATE tabla program_type.

```

1 CREATE TABLE `program_type` (
2   `progt_id` BIGINT(20) NOT NULL AUTO_INCREMENT,
3   `progt_name` VARCHAR(255) NOT NULL DEFAULT '0',
4   PRIMARY KEY (`progt_id`),
5   UNIQUE INDEX `progt_name` (`progt_name`)
6 )
7 COLLATE='utf8_general_ci'
8

```

4.1.1.3. Tabla program

En esta tabla se guarda toda la información de las programaciones; tiene llaves foráneas a las tablas de *clinician* e *indication* que permiten relacionar esta programación con el usuario que la crea y con la indicación que se está programando respectivamente; esta tabla contiene las siguientes columnas:

- **prog_id**: Id única de cada tipo de programación.

- **prog_creation_date**: Fecha en la que se creó esta programación.
- **prog_indication_id**: Id de la indicaciones a la que esta programación pertenece.
- **prog_description**: Observaciones que el usuario puede escribir al momento de crear una programación.
- **prog_extended_data**: JSON guardado como string que puede ser usado para guardar información adicional de la programación que no se guarde en otra columna.
- **prog_creator_id**: Id del usuario que creó esta programación.

A continuación se muestra el código de creación de la tabla:

Código 4.3: Código CREATE tabla program.

```

1 CREATE TABLE `program` (
2   `prog_id` BIGINT(20) NOT NULL AUTO_INCREMENT,
3   `progt_id` BIGINT(20) NULL DEFAULT NULL,
4   `prog_creation_date` DATETIME NULL DEFAULT current_timestamp(),
5   `prog_indication_id` BIGINT(20) NULL DEFAULT NULL,
6   `prog_description` TEXT NOT NULL DEFAULT "",
7   `prog_extended_data` LONGTEXT NOT NULL DEFAULT '{}',
8   `prog_creator_id` BIGINT(20) NULL DEFAULT NULL,
9   INDEX `prog_id` (`prog_id`),
10  INDEX `FK_program_program_type` (`progt_id`),
11  INDEX `aaa` (`prog_indication_id`, `prog_creation_date`),
12  INDEX `FK_program_clinician` (`prog_creator_id`),
13  CONSTRAINT `FK_program_clinician` FOREIGN KEY (`prog_creator_id`)
14    ↪ REFERENCES `clinician` (`clin_id`),
15  CONSTRAINT `FK_program_indication` FOREIGN KEY (`prog_indication_id`)
16    ↪ REFERENCES `indication` (`ind_id`),
17  CONSTRAINT `FK_program_program_type` FOREIGN KEY (`progt_id`)
    ↪ REFERENCES `program_type` (`progt_id`)
)

```

4.1.1.4. Tabla program_types

Esta tabla asocia los programas con los tipos de programación existente, por lo que solo tiene 2 columnas que son llaves foráneas a las tablas *program* y *program_type*; esto permite que una programación pueda tener más de un tipo de programación. La llave primaria de esta tabla no es una columna sino la combinación de ambas columnas.

- **prog_id**: Id del programa.
- **progt_id**: Id del tipo de programa.

A continuación se muestra el código de creación de la tabla:

4.1.2. Kafka

Para poder tener un monitor de tareas es necesario poder mostrar el estado de las tareas en tiempo real; para esto es necesario crear una interfaz de usuario que muestre el estado de todas las tareas pendientes asociadas a las indicaciones y que se actualice a medida que éstas se realicen, pero como objetivo adicional se decidió no hacer un polling constante a la API para obtener el estado de las tareas ya que esto podría sobrecargar la base de datos y la API con demasiadas llamadas para actualizar los datos.

Se propuso una arquitectura basada en eventos donde se crean eventos cada vez que exista un cambio en las indicaciones o en las tareas, donde posteriormente estos mensajes son consumidos por el monitor de tareas en tiempo real y actualice la información mostrada en las pantallas de monitoreo.

Para esta arquitectura basada en eventos se decidió ocupar Kafka como base por lo que se instaló y configuró un cluster de Kafka. Se instaló Kafka en 3 servidores con la misma configuración y los mismos tópicos para de esta forma tener redundancia.

Se definieron los tópicos de Kafka que se usarán para los mensajes y el contenido de estos mensajes; los mensajes se pensaron con la idea de que sean lo suficientemente genéricos para que luego puedan ser consumidos por distintas aplicaciones que también requieran información de los cambios de ubicación o del estado de las indicaciones de los pacientes.

Los tópicos creados fueron los siguientes:

- **indicaciones:** Este tópico tendrá todos los mensajes de los cambios que se hagan sobre las indicaciones y las tareas asociadas a éstas; es decir habrá mensajes de creación de indicaciones, de cambios de estado de las indicaciones como también mensajes cuando se creen o cambien de estado las tareas asociadas a las indicaciones.
- **cambio-ubicacion:** Este tópico tendrá los mensajes relacionados con los cambios de ubicación de los pacientes, es decir cuando el paciente se hospitaliza y se le asigna una cama, cuando cambia de cama, cuando va a pabellón y cuando se da de alta.

Los mensajes enviados a los tópicos mencionados anteriormente se enviarán en formato JSON y se envía la menor cantidad de información posible; esto es solo las ID relevantes para el mensaje, por ejemplo en el caso de mensajes de indicaciones, se envía la id del paciente, la id del care-manager, la id del patient-service, la id de la indicación y un String que describe el tipo evento que ocurrió en la indicación.

Código 4.5: Ejemplo de mensaje en el tópico indicaciones

```
1 {  
2   "patientId": 504151,  
3   "careManagerId": 1209231,  
4   "patientServiceId": 1837411,  
5   "indicationId": 7562761,  
6   "type": "ejecucionRealizada"  
7 }
```

Cabe destacar que aunque este trabajo está pensado para los pacientes hospitalizados, los mensajes en Kafka se generarán para todos los cambios de ubicación y en las indicaciones, independiente si el paciente está en urgencia, hospitalizado o en una atención ambulatoria. De este modo cualquier otra aplicación que necesite información sobre los cambios de ubicación o en las indicaciones pueda consumir estos mismos tópicos de Kafka.

4.1.3. API

Para poder enviar y recibir datos desde el servidor de Kafka fue necesario instalar una nueva librería en PHP, para esto se decidió usar el conector “php-rdkafka” por ser uno de los conectores en PHP más activos y con más estrellas en github², adicionalmente la API de esta librería es similar a la API de *librdkafka* una librería escrita en C/C++ que es la base para muchas librerías en otros lenguajes incluyendo *node-rdkafka* en Node.js que es la librería usada en el monitor para consumir los mensajes.

La API agrupa la lógica de sus componentes en distintas clases de PHP donde en general cada clase corresponde a una entidad en el modelo de datos; a continuación se describen los cambios en las distintas clases dentro de la API y las nuevas clases que se crearon.

- **Clase Program:** Nueva clase que corresponde a los programas; esta clase tiene la funcionalidad de crear programas y al mismo tiempo crear el listado de tareas asociadas al programa en caso que el programa sea de tipo horarios. Si el programa no es del tipo horarios, las tareas son creadas al momento de su ejecución. Esta clase también contiene las funciones para realizar consultas a la base de datos, filtrando por distintos valores como id de indicación.
- **Clase ProgramType:** Nueva clase que no permite crear nuevos tipos de programas, y solo tiene funciones que permiten consultar la lista de tipos disponibles u obtener todas la información de un tipo usando una búsqueda con la id del tipo.
- **Clase ProgramTypes:** Nueva clase que permite asociar los programas con los tipos; tiene un método donde recibe la id del programa y del tipo que se le quiere agregar e inserta el registro en la base de datos. También en el método de consulta tiene un filtro para poder buscar todos los tipos de un programa usando la id del programa.
- **Clase Schedule:** Se modificó la clase Schedule permitiendo agregar una id de programa y el tipo de programa cada vez que se crea una tarea. Se agregaron los filtros para poder buscar tareas asociadas a una indicación o a un programa y también se agregó la funcionalidad de que se envíen los mensajes a Kafka cada vez que ocurra un cambio en las tareas;, esto es tanto cuando se crean como cuando cambian de estado.
- **Clase Indication:** Se agregó la funcionalidad de que cuando se realicen ejecuciones de las indicaciones, se guarde una tarea; en caso de que se esté realizando una tarea programada solo se cambia el estado de la tarea asociada y en caso de que no sea una tarea programada se crea una nueva tarea con estado finalizada asociada a la indicación; adicionalmente se agregó la funcionalidad de que se envíen mensajes a Kafka cuando

² <https://github.com/arnaud-lb/php-rdkafka>

se crea una indicación, se realicen ejecuciones de la indicación y cuando ésta cambia de estado, esto se hizo para todos los cambios de estado como creación de una indicación, indicación eliminada, indicación expirada³.

- **Clase PatientService:** Se agregó la funcionalidad de enviar los mensajes correspondientes a Kafka cuando ocurre un traslado de pacientes y cuando se crea una nueva atención.
- **Clase Location:** Se agregó la funcionalidad de enviar mensajes a Kafka cada vez que ocurre un cambio de ubicación

Se crearon nuevos Endpoints para poder usar las nuevas funcionalidades; a continuación se listan los nuevos Endpoints REST agregados a la API.

- **postProgram:** Endpoint usado para crear un nuevo programa. Recibe en el cuerpo un JSON con toda la información necesaria para crear el programa; esto incluye la lista de los horarios en caso de que la programación sea del tipo horarios.
- **getProgram:** Endpoint usado para obtener la lista de programas; es necesario agregar como query string el id de la indicación para la cual se quiere obtener los programas.

³ Solo las indicaciones de medicamentos se crean con fecha de expiración el resto de las indicaciones se mantienen hasta que alguien las elimine o el paciente sea dado de alta

4.2. Front-end

Se creó una nueva sección en la ficha clínica llamada “Plan de la atención”, y se modificó la sección de “Tratamientos y cuidados” junto con la sección “Exámenes y procedimientos”.

4.2.1. Plan de la Atención

The screenshot shows the 'Plan de la Atención' section of a medical system. At the top, there is a patient profile card with a name, age (0 AÑOS 10 MESES 12 DÍAS), and hospital information (CESFAM CAROL URZUA DE SAN BERNARDO, FONASA B). Below this, there are several tabs: 'MI FAMILIA Y ENTORNO SOCIAL', 'MIS ANTECEDENTES CLÍNICOS', 'MI EDUCACIÓN', 'MI VIDA COTIDIANA', and 'MI HISTORIAL'. The main content area is divided into a timeline and a table of programmed instructions. The timeline shows tasks scheduled for today (HOY) from 4:00 to 3:00. The table has columns for 'INDICACIONES PROGRAMADAS', 'PROGRAMACIÓN', 'OBSERVACIONES', 'RESPONSABLE PROB', and 'ACCIONES'. A red circle highlights the 'FILTRAR' button in the table header. The table contains several rows of instructions, such as 'Paracetamol (Oral): 5 mg 3. Por: 1 día(s). Via: Oral.', 'Aspiración de secreciones: sos', and 'Muda y/o aseo genital: c/4 h o sos'. The 'ACCIONES' column shows icons for 'NUEVO', 'REVISAR', and 'COMPLETADO'.

Figura 4.3: Sección del plan de la atención

La sección “plan de la atención” tiene 5 pestañas.

La primera pestaña es una vista general de todas las indicaciones programadas y se puede ver en la Figura 4.3, en la parte superior de esta sección, una línea de tiempo muestra la cantidad de tareas que hay programadas dentro de las próximas 24 horas; esta línea de tiempo está separada en 4 filas que agrupa las tareas según el tipo de la indicación al que estas tareas pertenecen y tiene 24 columnas correspondientes a las próximas 24 horas; las horas en la línea de tiempo avanzan automáticamente al cambiar de hora, pero la cantidad de tareas pendiente no se actualiza de manera automática a medida que las tareas son realizadas.

Bajo la línea de tiempo está la tabla que contiene la lista de todas las indicaciones activas y programadas del paciente; en esta tabla se muestran las indicaciones de todos los tipos pero existe una separación entre los tipos usando una línea separadora y usando una línea de color en el borde izquierdo de la tabla.

En la primera columna de la tabla se muestra el nombre de la indicación junto con su descripción y las observaciones que pueda haber escrito el médico al momento de indicarla; en caso de que sea una programación nueva (se considera nueva una indicación creada hace menos de 1 minuto) se muestra un recuadro que dice “NUEVO” en la misma columna.

La segunda columna tiene la información sobre la programación; en caso de que sea por horarios se muestra el horario de la próxima tarea y de la tarea con mayor atraso en caso de

que existan. Si el horario es del mismo día no se muestra la fecha y solo se ve la hora. Si todas las tareas se han realizado se muestra un mensaje que dice “Todas las tareas programadas han sido realizadas.”. Para los programas horarios se agrega el botón “ver prog.” como se puede ver en el círculo número 4 de la Figura 4.3; este botón abre un modal similar al de programar pero no permite editar los valores. En caso que la programación sea por horarios y SOS, se agrega SOS luego de los horarios y en caso que la programación tenga solo 1 tipo y no sea horarios solo se escribe el nombre del tipo de la indicación y se le agrega el ícono correspondiente a la izquierda. En esta columna también existe un filtro en la parte superior, como se puede ver en el círculo número 3 de la Figura 4.3. Este filtro permite filtrar las indicaciones que se muestran en la tabla según el tipo de programación que tienen, por ejemplo se puede usar para ver solo las indicaciones con programación SOS o para ocultar las indicaciones con programación constante.

La tercera columna contiene las observaciones que se escribieron al momento de guardar la programación.

La cuarta columna contiene el nombre de la persona que creó la programación y la fecha de la última modificación a esta programación.

La quinta columna contiene las acciones. Existen 2 acciones que se pueden realizar sobre las indicaciones programadas. La primera acción se ve resaltada en el círculo número 1 de la Figura 4.3, es la de editar la programación y abre un nuevo pop-up donde se puede modificar la programación; este modal es el mismo que se usa para crear nuevos programas y se puede ver en la Figura 4.6. La segunda acción que se puede ver resaltada en el círculo 2 de la Figura 4.3 abre un pop-up con el historial de las programaciones de la indicación, como se puede ver en la Figura 4.4, este pop-up tiene una tabla donde cada fila representa una programación, donde en la primera columna se muestra el nombre del usuario que realizó la programación o la modificación y en la segunda columna se ve la información de la programación como los tipos que tiene, las observaciones escritas y el listado de tareas generadas si es de tipo horarios.

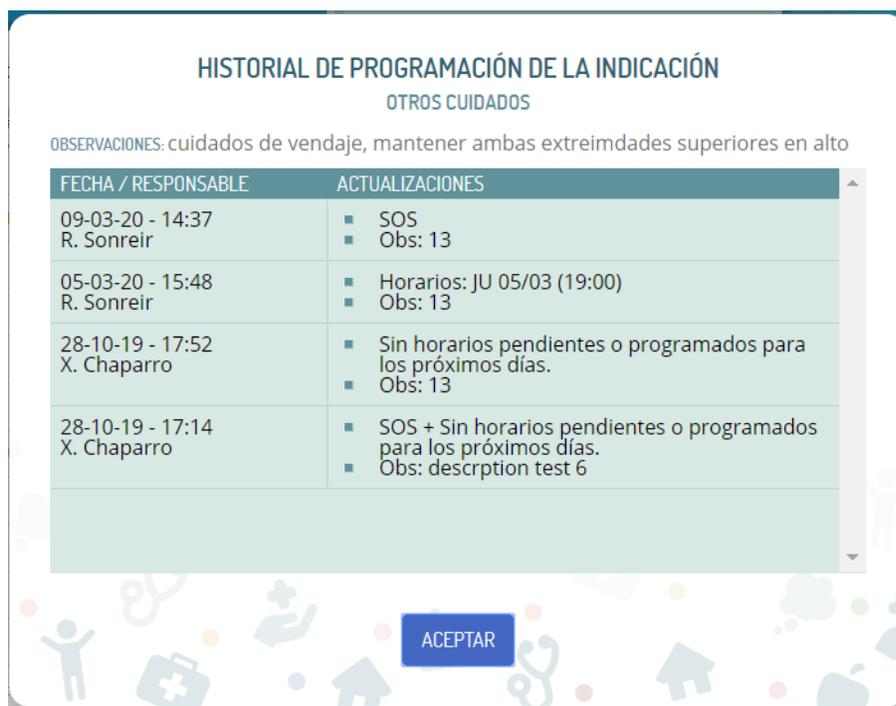


Figura 4.4: Pop-up que muestra el historial de programaciones de la indicación.

Las otras 4 pestañas son las pestañas de programación, diferenciadas en 4 categorías que son *medicamentos*, *cuidados*, *exámenes* y *otras indicaciones*; estas pestañas solo se diferencian en las indicaciones que se muestran. Si existen indicaciones sin programar del tipo correspondiente se puede ver un punto rojo en la esquina superior izquierda del botón de la pestaña como se puede ver en la Figura 4.5.

En las pestañas de programación, si existen indicaciones del tipo seleccionado, se muestra la línea de tiempo en la parte superior pero oscureciendo los colores de las tareas de indicaciones de tipos distinto al seleccionado. Bajo la línea de tiempo se muestra la tabla con la lista de las indicaciones programadas que es similar a la de la pestaña principal con la única diferencia es que no tiene la línea de color en el borde izquierdo, y bajo esta tabla se encuentra la tabla de programación en caso de que existan indicaciones pendientes de programación.

En la tabla de programación cada fila representa una indicación pendiente de programación. La primera columna contiene al información de la indicación y a la izquierda tiene un ícono que es un punto rojo si la programación seleccionada es invalida y una marca de chequeo en caso de que la programación sea valida.

La segunda columna es un checkbox para indicar que la programación es del tipo SOS, la tercera columna es un radio button para marcar que la programación es constante, la cuarta columna es para marcar las indicaciones de tipo horarios con un radio button para indicar que es de tipo horario y un botón con un reloj que abre el pop-up de programación que se puede ver en la Figura 4.7; si solo se selecciona el radio button la programación no es válida como se puede ver en el caso de la indicación “Obs fiebre, vómitos, convulsiones” en el ejemplo. Una vez seleccionado el horario se muestra a la derecha del ícono con el reloj. La quinta

columna permite seleccionar los otros tipos de programa; tiene un input del tipo dropdown donde se puede seleccionar uno de los otros tipos de programación. La sexta columna contiene las observaciones donde se puede escribir una pequeña observación al momento de crear la programación que se verá al momento de realizar las ejecuciones de las indicaciones; si no se ha seleccionado un tipo de programación el input de las observaciones permanece bloqueado. La última columna contiene un ícono de una escoba que elimina toda la información de la programación.

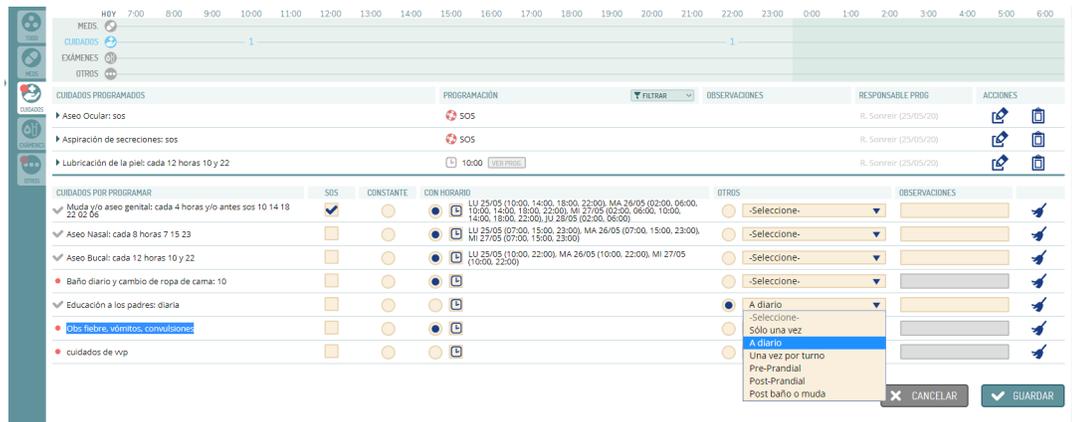


Figura 4.5: Vista del Plan de la Atención en la pestaña Cuidados

Al presionar el botón con el reloj en la tabla de indicaciones por programar se abre un pop-up como el que se puede ver en la Figura 4.7. En la parte superior de este pop-up se puede ver un resumen de la información del paciente en atención; en la siguiente fila se puede ver el nombre de la indicación junto con su descripción y a la derecha se pueden seleccionar los tipos de programación; estos son los mismos tipos que se pueden seleccionar en la tabla y se actualizan simultáneamente. En caso de que se marque la programación como tipo horarios en la parte central de pop-up se pueden ver los elementos para seleccionar horarios. En la parte izquierda se ve un recuadro para sugerir programa donde se puede seleccionar una fecha y hora de la primera tarea, la frecuencia de repetición y la cantidad de días por los que se programará, y al hacer click en el botón visualizar, se seleccionan todos los horarios correspondientes a la programación seleccionada. Bajo el recuadro de sugerencia se puede ver la leyenda que indica lo que significa el ícono sobre cada hora, por ser este el mismo pop-up que se usa para modificar programaciones. En caso de que existan tareas anteriores realizadas, canceladas o pendientes se muestran los íconos correspondientes en las horas correspondientes.

Los horarios se ven en un cuadro que contiene una lista con los días donde existen 24 checkbox con las 24 horas del día, donde se puede seleccionar si se quiere programar una tarea para esa hora. En caso de que sea una hora anterior a la actual el checkbox está deshabilitado y se ve de color gris; dentro del cuadro de cada día se puede ver en la parte superior 2 íconos, uno que es la escoba que borra todos los checkbox seleccionados del día y el otro es una X que borra el día de la lista. También existe un botón de limpiar todo sobre los cuadros de los días que limpia todos los checkbox seleccionados. A la derecha de los cuadros con los días hay un input de fecha que al seleccionarlo permite elegir una nueva fecha para agregar al programa, y a la derecha un botón con un ícono + para agregar la fecha seleccionada. Para

el caso de las indicaciones de medicamentos que se crean con una fecha de expiración en vez de la opción de agregar una fecha se muestra la fecha de caducidad de la indicación y no se permite agregar nuevas fechas; es por esto que para los medicamentos se carga el pop-up mostrando todos los días de validez de la indicación y no se permite eliminar días con la el botón de X.

Finalmente en la parte inferior del pop-up de la Figura 4.6 se puede ver información sobre la última ejecución de la indicación y existe el input de texto para escribir las observaciones de la programación.

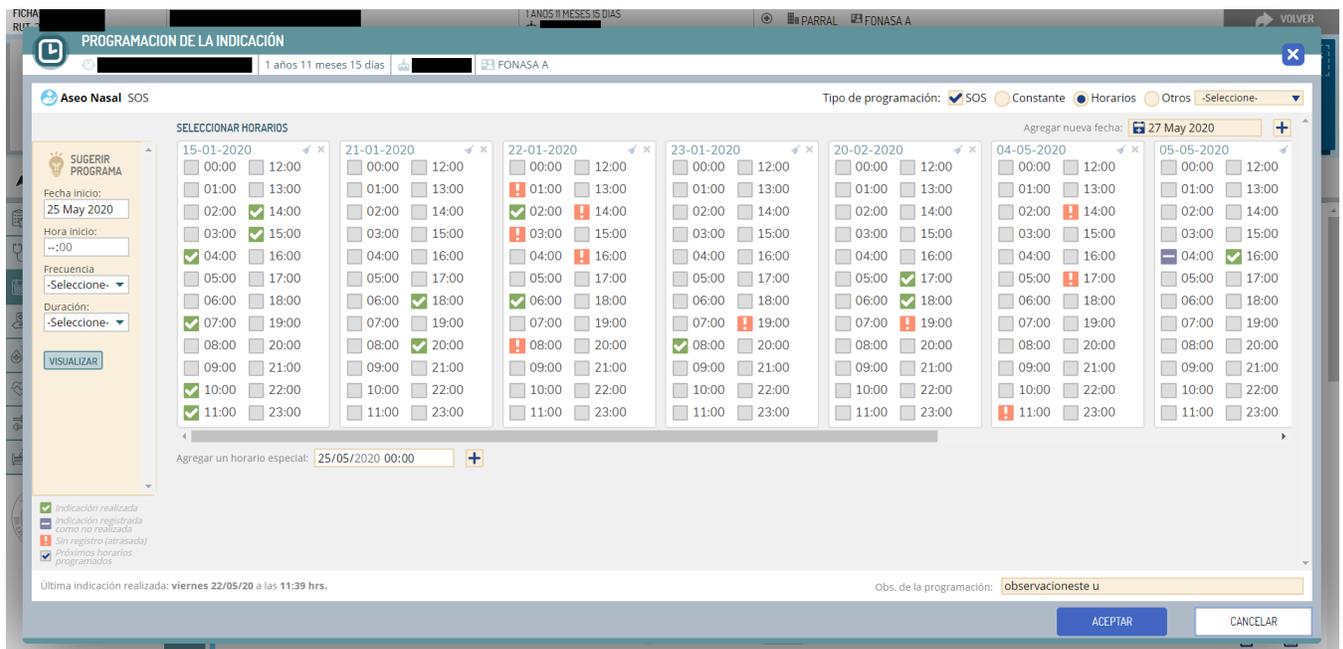


Figura 4.6: Pop-up para editar un programa.

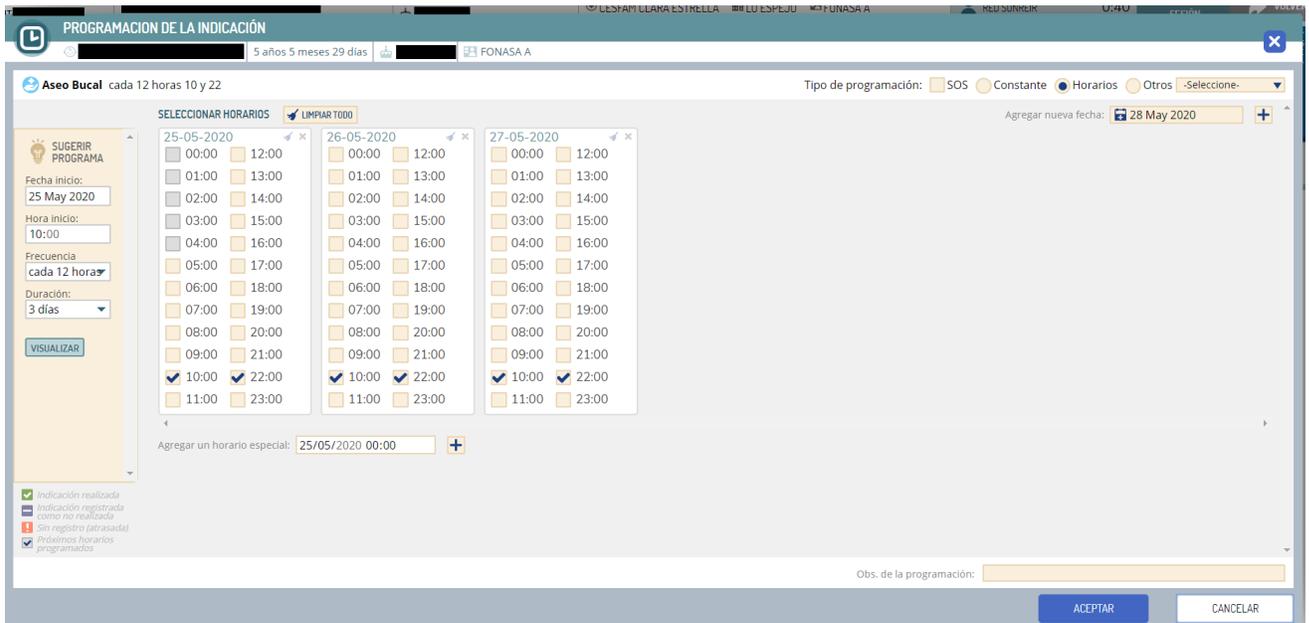


Figura 4.7: Pop-up para crear un programa.

4.2.2. Tratamientos y cuidados

En la pestaña de tratamientos y cuidados se modificó la vista de medicamentos, cuidados y otras indicaciones.

En la Figura 4.8 se ve el listado de medicamentos activos; en esta tabla se agregó la columna de programación que contiene el tipo de programación y las observaciones en caso de que exista una programación que no sea de tipo horarios.

MEDICAMENTOS	CUIDADOS	INTERVENCIÓN EN PABELLÓN	OTRAS INDICACIONES
MEDICAMENTOS INDICADOS			
<p>▶ Cloruro de sodio 1 g 2. Por: 1 día(s). Vía: Endovenosa. <small>Ult. dosis no administrada por: R. Sonreir 19-05-20 - 12:23. VER TIPO</small></p>	R. Sonreir (09-03-20 - 12:15)	10-03-20 - 18:15	<p>IMPRIMIR FOLIO A-16</p> <p>PROGRAMACIÓN Atrasado: 16:00 - 09/03/20</p> <p>ACCIONES </p>
<p>▶ Cloruro de sodio 1 g 2. Por: 1 día(s). Vía: Endovenosa. <small>Ult. dosis no administrada por: R. Sonreir 26-03-20 - 14:51. VER TIPO</small></p>	R. Sonreir (05-03-20 - 11:38)	06-03-20 - 16:38	<p>IMPRIMIR FOLIO A-15</p> <p>PROGRAMACIÓN SOS Obs: SOS si fiebre >40</p> <p>ACCIONES </p>
<p>▶ Paracetamol (Oral) 1 mg 1. Por: 1 día(s). Vía: Oral. <small>Ult. dosis no administrada por: R. Sonreir 26-03-20 - 14:51. VER TIPO</small></p>	R. Sonreir (09-03-20 - 12:15)	10-03-20 - 18:15	<p>IMPRIMIR FOLIO A-16</p> <p>PROGRAMACIÓN Atrasado: 15:00 - 09/03/20</p> <p>ACCIONES </p>
<p>▶ Paracetamol (Oral) 1 mg 1. Por: 1 día(s). Vía: Oral. <small>Ult. dosis administrada por: R. Sonreir 09-03-20 - 15:41. VER TIPO</small></p>	R. Sonreir (05-03-20 - 11:38)	06-03-20 - 16:38	<p>IMPRIMIR FOLIO A-15</p> <p>PROGRAMACIÓN Constante</p> <p>ACCIONES </p>
<p>▶ Paracetamol (Oral) 11 mg 2. Por: 1 día(s). Vía: Oral. Obs: aa <small>Ult. dosis administrada por: R. Sonreir 09-03-20 - 15:40. VER TIPO</small></p>	R. Sonreir (03-03-20 - 17:53)	04-03-20 - 22:53	<p>IMPRIMIR FOLIO A-14</p> <p>PROGRAMACIÓN Constante</p> <p>ACCIONES </p>
<p>CARGAR MEDICAMENTOS ANTERIORES</p>			

Figura 4.8: Vista de medicamentos activos.

Si la programación es de tipo horarios se muestra la hora de la próxima administración programada en menos de 13 horas, y la hora de la tarea con mayor atraso en caso de que existan; Si no hay tareas atrasadas o pendientes en las próximas 13 se muestra un mensaje que dice “No hay tareas programadas para las próximas 13 horas”. Si existen tareas posteriores a las 13 horas siguientes. Se muestra un mensaje que dice “Todas las tareas programadas han sido realizadas” en caso de que no existan tareas pendientes.

En la última columna de la tabla de medicamentos existen 2 íconos; una para marcar una dosis como administrada y otro para marcar la dosis como no administrada; ambos botones abren un pop-up que se ve en la Figura 4.9 a y b respectivamente. En este pop-up se muestra la información de la indicación y adicionalmente se muestra la programación. En caso que sea una programación por horarios y tenga tareas pendientes atrasadas o dentro de las próximas 13 horas, éstas se mostrarán como un listado de radio buttons, donde el usuario tiene que seleccionar un horario para poder realizar la administración, y se impide guardar el registro en caso de que no se seleccione ninguna opción o en caso que se quiera no administrar una dosis sin escribir una razón.

Finalmente para poder hacer seguimiento de todas las tareas asociadas a una indicación se modificó el pop-up de historial de la indicación donde se agregó una columna que contiene la hora seleccionada al momento de administrar la dosis; en caso que no sea una tarea programada por horarios en la columna se muestra el tipo del ultimo programa.



(a) pop-up administrar dosis.

(b) pop-up no administrar dosis.

(c) pop-up historial administraciones medicamentos.



(d) pop-up realizar cuidado.

(e) pop-up realizar otra indicación.

(f) pop-up historial de otras indicaciones.

Figura 4.9: pop-ups de administraciones y de historial.

En la pestaña de cuidados, como se puede ver en la Figura 4.10, tiene 4 recuadros en la parte superior que muestran información sobre las indicaciones de *régimen*, *reposo*, *oxigenoterapia* y *avisar si*, estas indicaciones no son programables por lo que no se realizó ningún cambio en su funcionamiento. Bajo estos recuadros se muestran 4 tablas con el resto de las indicaciones de cuidados agrupadas en 4 categorías: *cuidados generales*, *prevención de LPP*, *prevención de caídas* y *otros cuidados*; todas estas indicaciones si son programables por lo que en cada tabla se agregó una columna que incluye la información del ultimo programa como en el caso de medicamentos; la única diferencia en cuidados es que las indicaciones pueden estar programadas por horarios y SOS al mismo tiempo.

Para ejecutar las indicaciones de cuidados existen 2 botones en la última columna de la tabla al igual que en medicamentos, donde se abre un pop-up similar al de administrar medicamentos, existiendo una diferencia solo en el caso de que la indicación se programe por tipo horarios y SOS como es el caso de en la Figura 4.9, donde se debe seleccionar al momento de realizar la tarea si es una tarea programada con horarios o es una tarea SOS; en esa misma figura se puede ver el ícono de alerta para las tareas atrasadas y como se agrega la fecha debajo de la hora en caso de que la tarea sea de días anteriores, para las tareas siguientes nunca se muestra la fecha, ya que solo se muestran tareas pendientes dentro de las próximas 13 horas. El pop-up de historial también fue modificado al igual que en medicamentos para agregar el tipo de programación en cada tarea.

CUIDADOS GENERALES	PROGRAMACIÓN	ACCIONES	PREVENCIÓN DE LPP	PROGRAMACIÓN	ACCIONES
Aspiración de secreciones sos (27/08/19)	SOS	[Iconos]	Lubricación de la piel siempre 15 (27/08/19)	-	[Iconos]
Muda y/o aseo genital c/4 h o sos (27/08/19)	Atrasado: 06:00 - Ayer, Próx: 02:00 / SOS	[Iconos]	Educación a los padres (27/08/19)	Constante	[Iconos]
Aseo Nasal cada 4 hrs o sos (27/08/19)	Atrasado: 05:00 - Ayer, Próx: 05:00	[Iconos]	Mantener cama o cuna frenada siempre (27/08/19)	Constante	[Iconos]
Aseo Bucal después de alimentación (27/08/19)	Post-prandial	[Iconos]	Mantener barandas en alto 00-06-12-18 (27/08/19)	Atrasado: 06:00 - Ayer, Próx: 06:00 Obs: revisar	[Iconos]
Baño diario y cambio de ropa de cama c/24 h 15h (27/08/19)	Atrasado: 15:00 - Ayer, Próx: 15:00	[Iconos]	Mantener vigilancia 00-06-12-18 (27/08/19)	Atrasado: 06:00 - Ayer, Próx: 06:00 Obs: revisar	[Iconos]
OTROS CUIDADOS	PROGRAMACIÓN	ACCIONES			
Mantener oxigenoterapia: 0.5 litros por naricera, obs sitios de apoyo. (27/08/19)	-	[Iconos]			
Posición fowler. (27/08/19)	-	[Iconos]			
CSV cada 4 horas: 10-14-18-22-02-06 (27/08/19)	-	[Iconos]			

Figura 4.10: Vista de cuidados activos.

Finalmente se modificó la pestaña de otras indicaciones que se puede ver en la Figura 4.11, en esta pestaña el listado de indicaciones se separa en 2 tablas, y al igual que en cuidados se agregó la columna de programación y se modificaron los pop-up de realizar indicación y de historial de actualizaciones, en este caso los cambios son idénticos a los que se realizaron en cuidados ya que se usan las mismas componentes y se puede ver un ejemplo de estos en la Figura 4.9.

INDICACIONES ACTUALES	PROGRAMACIÓN	ACCIONES	INDICACIONES ACTUALES	PROGRAMACIÓN	ACCIONES
Inmovilizar dedo índice izquierdo (26/05/20)	Constante	[Iconos]	frio local (26/05/20)	A diario	[Iconos]

Figura 4.11: Vista de otras indicaciones activas.

4.2.3. Exámenes y procedimientos

Para la programación de los exámenes se modificó la pestaña de *exámenes y procedimientos*, esta pestaña se divide en 4 secciones: *exámenes de laboratorio y UMT*, *exámenes de imagenología*, *exámenes de estudio diagnóstico* y *procedimientos*. En cada una de estas vistas existe una tabla con la lista de los indicaciones pendientes. En cada una de las tablas

se agregó una columna que muestra la información de la programación; cabe destacar que las programaciones de las indicaciones de esta pestaña tienen la restricción donde no tener más de un tipo (como en medicamentos) y adicionalmente, en caso de que sea programada por horarios la cantidad de tareas se restringe a 1 y solo 1, por lo que en esta pestaña no se modifica el pop-up de toma de exámenes o realización de procedimientos ya que se asume siempre que si se ejecuta una indicación con programación por horarios que se está realizando la única tarea programada.

EXÁMENES DE LABORATORIO Y UMT		EXÁMENES DE IMAGENOLÓGIA		EXÁMENES DE ESTUDIO DIAGNÓSTICO		PROCEDIMIENTOS	
EXÁMENES DE ESTUDIO DIAGNÓSTICO INDICADOS							
Examen	Obs. Indicación	Solicitado por	Imprimir	Programación	Acciones		
Uroflujometría		RED SONREIR (2020-03-16 14:59:05)	<input type="checkbox"/>	-	🔄 🗑️ 📄		
Holter de arritmia		RED SONREIR (2020-03-16 14:59:05)	<input type="checkbox"/>	-	🔄 🗑️ 📄		
Estudio Urodinámico		RED SONREIR (2020-02-17 14:56:34)	<input type="checkbox"/>	SOS	🔄 🗑️ 📄		
<input type="checkbox"/> 📄 <input type="checkbox"/> 🗑️							
EXÁMENES DE ESTUDIO DIAGNÓSTICO EN PROCESO Y ANTERIORES							
Procedimiento	Obs. Indicación	Solicitado por	Obs. Realización	Realizado por	Estado		
Perfil Uretral (proc. Aut)		RED SONREIR (2020-02-17 14:56:33)		RED SONREIR (2020-02-25 12:26:53)	Finalizada		
Curva Relac. Flujo/Vol Basal		RED SONREIR (2020-02-17 14:56:33)		RED SONREIR (2020-02-25 12:54:30)	Finalizada		
Electrocardiograma		RED SONREIR (2020-02-17 14:56:33)		RED SONREIR (2020-02-25 12:51:37)	Finalizada		

Figura 4.12: Vista de indicaciones de tipo estudio diagnóstico.

EXÁMENES DE LABORATORIO Y UMT		EXÁMENES DE IMAGENOLÓGIA		EXÁMENES DE ESTUDIO DIAGNÓSTICO		PROCEDIMIENTOS	
EXÁMENES DE LABORATORIO INDICADOS							
Examen	Obs. Indicación	Solicitado por	Programación	Tomar Muestra	Acciones		
• Hemocultivo central		RED SONREIR (2020-02-17 12:51:55)	SOS	<input type="checkbox"/>	🔄 🗑️ 📄		
• Ácido Úrico en orina aislada		RED SONREIR (2020-02-17 12:51:55)	Atrasado: 16:00 - 27/04/20	<input type="checkbox"/>	🔄 🗑️ 📄		
• Creatinina		RED SONREIR (2020-02-17 12:51:55)	Atrasado: 18:00 - 17/03/20 Obs: observación de prueba	<input type="checkbox"/>	🔄 🗑️ 📄		
• Calcio		RED SONREIR (2020-02-17 12:51:55)	Atrasado: 17:00 - 19/02/20	<input type="checkbox"/>	🔄 🗑️ 📄		
• Directo al fresco		RED SONREIR (2020-02-17 12:51:55)	Atrasado: 18:00 - 04/03/20	<input type="checkbox"/>	🔄 🗑️ 📄		
• Hemograma		RED SONREIR (2020-03-16 19:59:20)	SOS Obs: 11	<input type="checkbox"/>	🔄 🗑️ 📄		
• Hemograma		RED SONREIR (2020-03-16 19:59:20)	Atrasado: 18:00 - 23/03/20	<input type="checkbox"/>	🔄 🗑️ 📄		
<input type="checkbox"/> 📄							
EXÁMENES DE LABORATORIO EN PROCESO Y ANTERIORES							
Examen	Obs. Indicación	Solicitado por	Obs. Toma de Muestra	Muestra Tomada por	Estado	Acciones	
• Hemoglobina		ERWIN DAGOBERTO ARIAS LABRIN (2019-11-19 17:59:35)		RED SONREIR	🟡		
• Hemoglobina		ERWIN DAGOBERTO ARIAS LABRIN (2019-11-19 17:58:41)		RED SONREIR	🟡		

Figura 4.13: Vista de indicaciones de exámenes de laboratorio y UMT.

4.3. Monitor de Tareas

Se creó un monitor de tareas en un nuevo proyecto cuyo objetivo es tener una visión general de la cantidad de tareas pendientes agrupadas por área del hospital; en la Figura 4.14 se puede ver un ejemplo de cómo se vería el monitor en una área de pacientes hospitalizados como es el área de cóndor.

Este monitor de tareas está escrito en Node.js, se configura con una lista de las áreas de las que el monitor de encargara de mantener el estado de las tareas, luego por cada una de estas áreas se carga la lista de pacientes hospitalizados, y por cada paciente, se carga la lista de todas las indicaciones activas junto con la programación en caso de existir y la lista de tareas. Una vez cargado todo el estado inicial el monitor comienza a consumir mensajes de Kafka sobre los cambios en las indicaciones o cambios de ubicaciones de pacientes, y cuando consume mensajes de cambios actualiza la información del paciente.

Este monitor guarda el estado de todos los pacientes y de todas las indicaciones activas en la memoria principal en un objeto de Javascript, y es el monitor el encargado de calcular las tareas atrasadas y agrupar las tareas por horas y tipo. Por cada actualización de la información se envía a los clientes un objeto con la información actualizada a través de un web-socket, por lo que los clientes no hacen llamadas a la API y solo se comunican con el monitor, y el monitor solo hace las llamadas a la API cuando recibe mensajes de Kafka de que han ocurrido cambios.

El monitor corre un servidor web para servir el HTML y CSS junto con el Javascript de la vista, y un servidor de web-socket para actualizar la información.

En la parte superior del monitor se muestra la fecha junto con la hora actual y en la parte izquierda se muestra un ícono correspondiente al área de hospitalizados; más abajo está la tabla principal que contiene la información relevante de las indicaciones y tareas, y en la parte inferior se muestra un círculo que indica la página; en caso de que la cantidad de pacientes supere la cantidad de filas que se pueden mostrar (13), la lista de pacientes se pagina y se cambia la lista de pacientes que se muestran cada 30 segundos.

La tabla cuenta con 4 columnas. La primera columna contiene el número de cama, el nombre del paciente y en la parte derecha existe un ícono de un calendario que indica que el paciente tiene indicaciones sin programar. La segunda columna contiene un listado de íconos que indican que existen tareas atrasadas de los tipos correspondientes; en el monitor las indicaciones se separan de forma distinta a cómo se separan en la ficha, y existe un ícono especial para exámenes de laboratorio y para exámenes de imagenología, los exámenes de otros tipos y los procedimientos se agrupan junto con las otras indicaciones en ícono de otros. Una tarea se concederá atrasada cuando ha pasado 1 hora desde la hora en que estaba programada, por ejemplo una tarea programada para las 15:00 no se considera atrasada hasta las 16:00. La tercera y la cuarta columna muestran la cantidad de tareas pendiente para la hora actual y la hora siguiente respectivamente agrupadas por los tipos de indicaciones, a medida que las tareas se van realizando este número disminuye y cuando se han realizado todas las tareas se muestra una marca de chequeado que indica que ya se han realizado todas las tareas pendientes.

TIPANIE HOSPITALIZADOS		MONITOR DE TAREAS PROGRAMADAS					LUNES, 24 DE FEBRERO DEL 2020					07:32				
UBIC.	PERSONA	ATRASADAS	10:00					11:00								
401-1	C. ALONSO							1			2		1			
401-2	C. PÉREZ										2	3			3	
401-3	J. IBARRA								3						2	
402-1	E. CORTÉZ							1		3				4		
402-2	D. MUÑOZ							4					1			3
402-3	A. VALENCIA									3		4			1	
403-1	M. ORELLANA							2	3						1	
403-2	S. HENRÍQUEZ							1	4		1		5	2		
403-3	C. ALONSO								2			3		3		2
404-1	J. IBARRA							4						2		
404-2	C. PÉREZ															
404-3	A. VALENCIA							3	1			3	2			1
405-1	E. CORTÉZ									1		2			2	
405-2	S. HENRÍQUEZ											1				3
405-3	M. ORELLANA								4	1				5	1	
402-3	A. VALENCIA									3		4			1	

Figura 4.14: Monitor de tareas.

Capítulo 5

Validación

Este capítulo se divide en 2 partes, la primera explica como se fue validando la solución a medida que esta se desarrollaba y la segunda contiene una encuesta a un médico que ayudó en el desarrollo y probó el sistema.

Como el desarrollo se realizó usando metodologías ágiles, el correcto funcionamiento de las nuevas funcionalidades se fue probando de manera continua. Durante la primera parte del trabajo se buscó tener un producto mínimo viable que se pudiera pasar a los servidores de prueba usados en el hospital, para luego iterar sobre este.

Primero se construyó una versión mínima del back-end con los cambios definidos en el modelo de datos y algunos end-point en la API REST básicos para probar el funcionamiento, de creación de programas y tareas, esta versión del back-end sólo se validó usando Postman¹ para comprobar el correcto funcionamiento.

Luego de tener una primera versión del back-end se creó una primera versión del plan de la atención en la ficha; esta versión se pasó a los servidores de prueba donde se mostró su funcionalidad y se realizaron pruebas, luego de esto se creó un listado de las nuevas funcionalidades requeridas y los problemas a corregir. Este listado de funcionalidades se subieron al servidor de Gitlab de la empresa como un listado de issues para luego ser priorizados para su desarrollo.

De esta forma se fue iterando semana a semana. Cada vez que se tuviera un conjunto suficientes de funcionalidades implementadas para probar en los servidores de prueba, se subía la última versión a estos servidores para generar un nuevo listado de funcionalidades requeridas y validar que las funcionalidades implementadas y los cambios realizados cumplan con las expectativas de los usuarios. De esta forma se obtuvo una validación constante a medida que se implementaban las nuevas funcionalidades.

Los issues sólo se cierran cuando se comprueba su correcto funcionamiento por una persona distinta al que programó la solución.

En la Figura 5.1 se puede ver el ejemplo de un issue desde que es creado, luego priorizado

¹ <https://www.postman.com/>

y una vez implementado se marca para su revisión y después de que la solución es validada por una segunda persona el issue es finalmente cerrado.

The screenshot shows a Gitlab issue page with the following details:

- Path: `pertikos_apps > ficha > Issues > #1539`
- Status: **Closed** (Opened 4 months ago by user D)
- Buttons: Reopen issue, New issue
- Title: **Agregar descripción sobre los íconos Editar, Historial, Limpiar**
- Description: Debe aparecer la descripción (**estilo tooltips**) sobre los íconos Editar, Historial, Limpiar al posar el mouse sobre la figura.
- Edited 4 months ago by Rosario Cerecera
- Reactions: 0 thumbs up, 0 thumbs down, 0 neutral
- Sort: Oldest first, Show all activity
- Discussion: 0, Designs
- Activity Log:
 - Timeline icon
 - D. (@d) changed milestone to %Programa atención 4 months ago
 - D. (@d) added **Ficha Hosp** **Pendiente** labels 4 months ago
 - Rr (@rr) added **Prioridad baja** **To Do** labels 4 months ago
 - Rr (@rr) assigned to @r 4 months ago
 - Rr (@rr) changed title from **Agregar descripción sobre íconos** to **Agregar descripción sobre los íconos Editar, Historial, Limpiar** 4 months ago
 - R (@r) changed the description 4 months ago
 - R (@r) assigned to @lester.munoz and unassigned @i 4 months ago
 - R (@r) added **Prioridad media** label 4 months ago
 - R (@r) removed **Prioridad baja** label 4 months ago
 - Lester Muñoz (@lester.munoz) added **Revisar** label 4 months ago
 - R (@r) closed 3 months ago

Figura 5.1: Ejemplo de un issue en el Gitlab.

El monitor de tareas al igual que la ficha fue validado continuamente durante su desarrollo usando el sistema de issues de Gitlab, y usando los servidores de prueba en el hospital.

Desde el comienzo de este trabajo uno de los objetivos fue realizar pruebas en condiciones reales, pero surgieron complicaciones externas que lo impidieron, como fueron el desarrollo de movilizaciones y paros de funciones del personal de salud, una auditoría al recinto hospitalario y finalmente la pandemia del coronavirus, COVID-19.

5.1. Encuesta

Como paso final de la validación se creó una encuesta para el personal clínico que participó en el proceso de desarrollo para obtener sus opiniones, esta encuesta contiene preguntas tanto sobre el plan de atención en la ficha como del monitor de tareas.

A continuación se agrega la respuestas de Daniela Mardones, médico pediatra y supervisora de una unidad de hospitalización médico quirúrgica.

5.1.1. Preguntas sobre el plan de la atención en la ficha

- a. **¿Cree que el plan de la atención cumple con todas las funcionalidades necesarias para programar tareas y hacer seguimiento de estas para los distintos tipos de indicaciones como medicamentos, cuidados y exámenes?**

Cumple con el objetivo de programar y ejecutar las tareas

- b. **¿Cómo evaluaría la facilidad de uso del plan de la atención?**

La facilidad de uso dependerá de la capacitación previa

- c. **¿Existe algún problema que dificulte o impida el uso del plan de la atención en su estado actual?**

Me parece que funcionaría bien en el estado actual

- d. **¿Existe alguna sección del plan de la atención que crea que necesite mejoras?**

No

- e. **¿Cómo compararía el plan de la atención con la forma como se programan las indicaciones actualmente?**

Muy similar a lo actual, pero este programa permite tener toda la información reunida en un lugar

- f. **¿Cómo el plan de la atención afecta la forma de trabajo del personal clínico?**

Permite organizar el trabajo y disponer de toda la información junta lo facilita

- g. **¿Algún comentario adicional que quiera agregar sobre el plan de la atención?**

Haría énfasis en que la capacitación en el uso del programa es lo más importante para su buen funcionamiento

5.1.2. Preguntas sobre el monitor de tareas.

- a. **¿Qué tan complejo es entender e interpretar la información que muestra el monitor de tareas?**

Para mi que conozco el monitor de antes, es fácil de entender, pero requiere que sea explicado previamente

- b. Sobre la información que se muestra en el monitor de tareas, ¿es relevante?, ¿es suficiente?**

Me parece que la información que se muestra es suficiente. Si se necesita info más detallada, se ingresa a la ficha.

- c. ¿Algún comentario adicional que quiera agregar sobre el monitor de tareas?**

Insisto en la capacitación previa.

Capítulo 6

Conclusiones

En este capítulo se presentan las conclusiones finales sobre este trabajo de memoria, desde las metodologías de trabajo elegidas, hasta el cumplimiento de los objetivos; finalmente se comenta sobre posibles trabajos futuros.

Se logró crear una nueva sección en la ficha con todas las funcionalidades necesarias para la programación de tareas asociadas a indicaciones; y un monitor de tareas para tener una visión general de las tareas pendientes. Con esto se cumplieron todos los objetivos específicos, y también se cumplió con los requisitos adicionales de seguridad y usabilidad.

Debido a que este software no será usado por un usuario común, sino por un personal clínico especializado, se trabajó muy de cerca con los usuarios finales durante el diseño y desarrollo de la solución. Para esto fue de mucha utilidad seguir una metodología ágil, ya que permitió responder de manera rápida a los requerimientos de los usuarios y tener una pronta retroalimentación con opiniones de los usuarios. Con esto se pudo conseguir una solución mucho más ajustada a las necesidades de los usuarios finales que si se hubieran fijado requisitos de usuario y de software con el entendimiento del problema que se tenía en las primeras semanas de desarrollo.

Se puso un particular énfasis en las interfaces de usuarios, siendo la gran mayoría de los issues relacionados a las interfaces, se contó con la ayuda de una diseñadora que trabajó en los archivos de CSS y diseño todos los íconos, para esto fue de particular utilidad contar con Gitlab donde se coordinó el desarrollo de los issues y el trabajo simultáneo en distintas ramas.

Se creo un monitor de tareas que muestra el estado de las tareas programadas agrupadas por área del hospital, para este monitor se usó una nueva forma de comunicación entre microservicios, a través de mensajes y una arquitectura basada en eventos. Gracias a esta arquitectura el monitor actualiza la información sobre el estado de las tareas en tiempo real y entrega esta información a los usuarios a través de un websocket evitando que los usuarios tengan que iniciar peticiones. Esto permite que el monitor sea escalable, pudiendo mantener actualizada la información de cientos de pacientes en decenas de usuarios conectados.

Una de las funcionalidades del monitor de tareas es mostrar alertas en caso que existan tareas atrasadas o indicaciones sin programación. De esta forma se reduce la posibilidad de

cometer errores por olvido.

Desde el comienzo de este trabajo se tuvo como objetivo poder implantarlo, pero existieron complicaciones que dificultaron el desarrollo y que finalmente significaron que no se pudiera realizar un paso a producción de este trabajo, como fueron los paros de los profesionales de la salud durante el 2019 y luego la pandemia del coronavirus durante el 2020 que cambió las prioridades del hospital y las nuevas funcionalidades que se esperaban de la ficha.

Dadas las prácticas ágiles que se tomaron en el desarrollo de este trabajo, la validación de la solución se fue realizando de manera continua por las personas encargadas de hacer pruebas, también se realizó una encuesta a un médico pediatra que participo en el diseño y desarrollo de este trabajo.

A pesar de haber logrado todos los objetivos específicos. Aún existen mejoras y trabajo que se puede hacer en el futuro.

- La realización de indicaciones se deben registrar en la ficha de una a la vez, para pacientes con muchas indicaciones registrar la correcta realización de cada una de ellas puede llegar a ser mucho trabajo, por lo que se podría buscar una nueva forma de registrar la realización de múltiples indicaciones al mismo tiempo, ya sea en la misma ficha o en alguna aplicación creada para ese propósito.
- Debido a que la cantidad de información presente en la ficha relacionada con las indicaciones aumentó, los tiempos de carga también aumentaron. Se puede buscar una forma de reducir los tiempos de carga, quizás usando la información del monitor de tareas y de esta forma evitar realizar demasiadas llamadas a la API.

Glosario

- AJAX** Conjunto de funcionalidades que permiten comunicación asincrónica entre un navegador web y un servidor. 8
- API** En ingles “Application Programming Interface”, conjunto de funcionalidades que permite la comunicación entre distintos software. i, 5, 7–9, 14, 15, 22–24, 34, 36, 41
- CSS** En ingles “Cascading Style Sheets”, es un lenguaje de diseño gráfico para definir y crear la presentación de un documentos HTML. 8, 34, 40
- fleboclisis** Administración de fluidos de diferentes composiciones en el torrente sanguíneo a través de un acceso venoso. 3, 11, 13
- HTML** En ingles “Hypertext markup language”, es el lenguaje estándar para generar documentos diseñados para ser vistos en un navegador web. 8, 9, 34
- Javascript** Lenguaje de programación muy usado en la WEB. 8, 34
- JSON** Formato de texto, para guardar datos, y muy usado en la web. 18, 20, 22, 24
- personal clínico** Trabajadores del área de la salud, como: médicos, enfermeros, TENS, tecnólogos médicos, etc. i, 2, 10, 11, 14, 15, 38, 40
- PHP** Es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web. 23
- REST** En inglés “representational state transfer”, es un estilo de arquitectura software para crear servicios web. 7, 24, 36
- SQL** Lenguaje de dominio específico usando en bases de datos relacionales. 8

Acrónimos

BIC Bombas de infusión continua. 3, 11, 13

CIE-10 Clasificación internacional de enfermedades. 2

HEGC Hospital Dr. Exequiel González Cortés. 1, 10

HIS Sistema de Información Hospitalario. 2, 10

LPP Lesiones por presión. 31

MQ Unidad de hospitalización Médico-Quirúrgica. 1, 13

TENS Técnicos en enfermería de nivel superior. 42

UCI Unidad de Cuidados Intensivos. 1

UMT Unidad de medicina transfusional. v, 32, 33

UTI Unidad de Tratamiento Intermedio. 1

Bibliografía

- [1] Mariadb foundation - mariadb.org. <https://mariadb.org/>, 2020. Accessed on 2020-06-28.
- [2] Php: Hypertext preprocessor. <https://www.php.net/>, 2020. Accessed on 2020-06-28.
- [3] Slim framework. <https://www.slimframework.com/>, 2020. Accessed on 2020-06-28.
- [4] HEGC. Cuenta pública 2018. <http://hegc.cl/wp-content/uploads/2019/04/Cuenta-P%C3%BAblica-2018.pdf>, Abril 2019. Accessed on 2019-06-15.
- [5] HEGC. Historia hegc. <http://hegc.cl/quienes-somos/historia-2/>, 2019. Accessed on 2019-06-15.
- [6] Cristian Julio. *Manual de usuario Lahuen*. 2019.
- [7] Martin A Makary and Michael Daniel. Medical error-the third leading cause of death in the us. *British Medical Journal*, 353, May 2016. ISSN 0959-8146. doi: 10.1136/bmj.i2139.
- [8] Brenda M. Michelson. Event-driven architecture overview. http://elementallinks.com/el-reports/EventDrivenArchitectureOverview_ElementalLinks_Feb2011.pdf, 2011. Accessed on 2019-06-15.
- [9] N. Narkhede, G. Shapira, and T. Palino. *Kafka: The Definitive Guide: Real-Time Data and Stream Processing at Scale*. O'Reilly Media, 2017. ISBN 9781491936115. URL <https://books.google.cl/books?id=dXwzDwAAQBAJ>.
- [10] Ben Stopford. Using apache kafka as a scalable, event-driven backbone for service architectures. <https://www.confluent.io/blog/apache-kafka-for-service-architectures/>, JULY 2017. Accessed on 2019-06-15.

ENFERMERIA
IMPRESA

NOMBRE: [REDACTED] SALA: 515 CAMA: [REDACTED]

Tratamiento	Dosis	Horario	Via	Observacion
Nifedipino	0.3mg	SOS	VO	Si PA > 102/57 en 4 oportunidades máx q/6hrs

Figura A.2: Tarjeta de programación de medicamentos SOS

TRATAMIENTO ENDOVENOSO ENFERMERA

NOMBRE: [REDACTED] SALA: 517 CAMA: [REDACTED]

TRATAMIENTO	DOSIS Y DILUCION	VELOCIDAD INFUSION	OBSERVACIONES	FECHA 23/07	FECHA 24/07	FECHA	FECHA	FECHA	FECHA	FECHA	FECHA
Cloxacilina I: 20/07	1,1gr	0.50ml		05 17	05 17						
Ceftioaxona I: 20/07	2gr		q/24hrs	07 20	16.						
Ondasentron	4mg	SOS	q/6hrs								
Omeprazol	20mg		q/24hrs		07						
Ketorolaco	10mg	SOS	q/8hrs								

Figura A.3: Tarjeta de programación de tratamiento endovenoso SOS

TRATAMIENTO ENDOVENOSO ENFERMERA

NOMBRE: [REDACTED] SALA: 516 CAMA: _____

Inicio Hidratación: 13:10

TRATAMIENTO	DOSES Y DILUCION	VELOCIDAD INFUSION	OBSERVACIONES	FECHA 23/07	FECHA 24/07	FECHA	FECHA	FECHA	FECHA	FECHA	FECHA
Furosemida	10mg	Dia 1	6 y 12 hrs	19	09						
Ciclofosfamida	840mg	en 2 hrs	Dial	17	/						
Misna	380mg	0-3-6-9 hrs mañan	Dial ciclo	17 20 23	02						
Vincristina	1mg	en bolo	Dial	19	/						
Dactinomicina	0.95mg		Dial	19 ³⁰	/						
Ondansetron	2mg			17	09 09						

Figura A.4: Tarjeta de programación de tratamiento endovenoso por horarios

NOMBRE: [REDACTED] SALA: 521 CAMA: _____

Inicio Hidratación: 21:00h

TRATAMIENTO	DOSES Y DILUCION	VELOCIDAD INFUSION	OBSERVACIONES	FECHA 23/07	FECHA 24/07	FECHA 25/07	FECHA	FECHA	FECHA	FECHA	FECHA
Furosemida	5mg		6 y 12 hrs de NTX	19	22						
Metotrexato	100 mg	30 min		19	20						
Metotrexato	900 mg	23 1/2 h		19	20						
Leucovorina	8mg		42-48 y 54 de NTX	19	20	4 10 16					
Ondansetron	2mg			19	07 15 23						
* Metotrexato	10mg	IT		*							
Purinetal	12.5mg (1/4 tab)	diario		21	21	21	21				

Figura A.5: Tarjeta de programación de medicamentos por horarios

folio [redacted]
\$ Urología +
LEUCEMIA LINFOBLASTICA
PROTOCOLO M

Nombre: [redacted] Fecha: [redacted]

Hto: 36 Leucocitos: 5370 Ran: 1718 Plaquetas 291.000

Creatininemia: 0,29 Transaminasas: 15 Na: 141,7 K: 4,77

Peso: 11600 g Talla: _____ Sup Corp: 0,5 Ciclo N°: 20

1.- Hidratación Volumen 3000 cc/m2 Volumen total 1500 cc

Dias 0 al 3 Iniciar hidratación a las 21 horas del día 0 Fechas 22/7 al 25/7

S.Glucosado 5 % 1000 cc Repetir por 1,5 veces

NaHCO 2/3 M 100 cc

KCl 10 % 20 cc Pasar a 63 cc/h

2.- Furosemda 0.5-1 mg/kg 5 mg. ev. a las 6 y 12 hrs. de iniciado Metotrexato

3.- Metotrexato 2000 mg/m2 1000 mg EV Día: 1 Iniciar a las 8 horas Fecha 23/7/19

1/10 dosis 100 mgs, diluido en 50 cc de S Glucosado 5% pasar en 30 minutos

9/10 dosis 900 mgs, diluido en 200 cc de S.Gluc 5 % a pasar en 23 1/2 hrs proteger de la luz

Mantener Ph orina > 7, si es = o < 7, colocar NaHCO3 8.4% 2 meq/kg _____ cc.

en agua destilada 2 cc/kg 20 cc a pasar en 1 hr.

4.- Metotrexato 40 mgs, Intratecal. Día 1 Fechas 23/7/19

5.- Leucovorin 15mg/m2 8 mgr EV a las 42.- 48 y 54 hrs.de iniciado Mtx.

6.- Purinetol 25mg/m2 12,5 mgs (1/4 tab.) VO cada 24 horas Fechas todos los días

7.- Ondasentron 0,15 mg/ kg 2 mg EV colocar 30' antes de QMT y repetir cada 8 horas

Dias 1 y 2

Nombre y código Médico oncólogo V. Ayar

Fechas 23 y 24/7

Fecha 22/7/19

Nombre y código Enfermera: [redacted]

Firma

Fecha

Nombre y código Q. farmacéutico [redacted]

Firma

Fecha

Figura A.6: Hoja con listado de indicaciones de paciente oncológico.

A.2. UPC

4 Bk C12u ✓
 5 Precaució contacto ✓
 6 Positivitat serològica ✓ Precaució ✓
 7 - Km ✓ Cuidats especials ✓
 8 - VM → 6 paràmetres Signe Pos. 21/7 Fa ↓ 20u ✓
 9 Per. 454 para sal 2 P3.1 ✓
 - Km 241 ✓

Nº	MEDICAMENTOS	VIA ADMIN	HORARIO	USO FARMACIA
1	33 Ketony 1500mg amantel → 15u litr		ptc.	(30)
2	(206) Fluoxetina 50mg 150u 5072 → 2u litr		ptc.	(45)
3	(05) Diclofenac sódico 400mg 150u SF → 717u		ptc.	
4	Hydrocortisona 250mg 4u sus		13-17-21-25-29-09	
5	Ceftriaxona 750mg 48m er		15-19-07	
6	Amoxicilina 250mg 424m er		15-19-07	
7	Clonidina 500mg 48u er		15-19-07	
8	Alendronato 25mg 424m er		15	
9	Paracetamol 500mg 412u er Signe Pos. 34 SOS.			
10	Espiridolona 6,25mg 424u sus		15-19-07	
11	Salsbutamol 4puft 44m er		15-19-07	
12	Bambusipropina 4puft 46m er		15-19-07	
13	Fluticasona 125mcg 2puft 12u ml		19-07	
14	Per 330 15p 424u sus			
15	Paracetamol 220mg 46u sus			

CÓDIGO MÉDICO [REDACTED] [REDACTED] [REDACTED]

Médico Intensivista UPC
 Hospital Externi González Torres-UPC

Fig. ODSur 525964 AAN-2015

Figura A.7: Tarjeta con listado de medicamentos (horarios y SOS) paciente UPC.