



**Universidad Metropolitana para
la Educación y el Trabajo**

Técnicas Avanzadas de la Programación
“Spring Nro. 1”

Alumno: Leandro Pasteur

Profesora: Antonieta Kuz

Fecha: 01/04/2025

Índice:

- 1) Elicitación de requerimientos uso de entrevistas**PAG 3.**
- 2) Identificación de proyecto: Objetivos de la identificación de proyectos. Identificación de la necesidad/problema: necesidad, justificación de su abordaje. Análisis de la viabilidad del proyecto
PAG 3 – 5.
- 2) Enunciar los Objetivos del Proyecto. La formulación de proyectos: Alcances del proyecto básico y de detalle, análisis del tamaño..... **PAG 5 – 8.**
- 3) Analizar el contexto general y las necesidades en general y en particular. FODA**PAG 8 – 9.**
- 4) Análisis de factibilidad técnica de alternativas de solución y lenguaje seleccionado con justificación en cuanto a costos y técnicamente **PAG 9 – 10.**
- 5) Tablero de control del proyecto, de tipo scrum..... **PAG 10.**
- 6) Anexo: Entrevistas y cuestionarios (abiertos y cerrados)**PAG 11-15.**

1) Definir y realizar la elicitación de requerimientos.

Para saber cuáles son los requerimientos específicos de un sistema de gestión para un jardín maternal, utilice varias técnicas de elicitación. A continuación, las voy a detallar:

Entrevistas y cuestionarios (abiertos y cerrados):

Se realizaron entrevistas y cuestionarios con los actores más importantes del jardín maternal:

Directora del Jardín, para conocer las necesidades operativas y normativas dentro del jardín.

Maestra, para saber cómo administra el bienestar, alimentación, educación y descanso de los niños.

Padre, para conocer su expectativa en cuanto al bienestar, comunicación y seguimiento del desarrollo de sus hijos.

Personal administrativo, para analizar el proceso de inscripción, cobro de cuotas y gestión de asistencia.

En el apartado de anexo, dejo de forma resumida las partes de mayor importancia dentro de las entrevistas y los cuestionarios.

2) Identificación del proyecto**A) Objetivos de la identificación del proyecto**

El objetivo principal de la identificación del proyecto es reconocer las necesidades del jardín maternal en términos de gestión, comunicación y seguimiento del desarrollo infantil, con el fin de proponer una solución tecnológica eficiente y adaptada a las dinámicas propias de una institución que atiende a niños de 0 a 3 años.

Se busca desarrollar un sistema que facilite la digitalización y automatización de los procesos administrativos y pedagógicos, promoviendo una mejor comunicación entre padres y los principales actores dentro del jardín, así como garantizando un registro seguro y accesible de la información diaria de cada niño.

Este proyecto pretende mejorar la trazabilidad de las actividades, optimizar el tiempo de las maestras y ofrecerles a los padres una experiencia más cercana y confiable en relación con el bienestar de sus hijos.

B) Identificación de la necesidad/ problema.

Necesidad: El jardín maternal enfrenta dificultades con la organización y gestión diaria debido a la falta de un sistema digital que permita registrar y compartir información clave sobre los niños, como asistencia, alimentación, descanso y comunicación con los padres. La falta de digitalización genera retraso en la entrega de información, dificultad en la coordinación entre maestras y administrativos, y una comunicación limitada con la familia.

Justificación sobre el abordaje.

- La implementación de un sistema digital permitirá:
- Mejorar la comunicación entre maestras y padres mediante alertas en tiempo real y reportes diarios.
- Optimizar la gestión de la información diaria de cada niño, reduciendo errores en los registros.
- Agilizar la administración de asistencia, horarios y necesidades específicas de cada niño.
- Asegurar el cumplimiento de protocolos alimenticios y de salud, con alertas automáticas para restricciones dietéticas y medicaciones.
- Reducir la carga administrativa del personal, permitiendo que las maestras dediquen más tiempo al cuidado y educación de los niños.
- Ofrecer reportes de desempeño y desarrollo infantil que puedan ser utilizados para evaluar el progreso de los niños en sus primeros años de vida.
- Ofrecer un sistema digital con alertas y pagos en línea de los aranceles del jardín.

C) Análisis de la viabilidad del proyecto.

El proyecto es viable desde las siguientes tres perspectivas:

Viabilidad Técnica: Existen plataformas como **KinderPass**, **HiMama** y **Brightwheel**, que ya ofrecen soluciones para la gestión de jardines maternos. Estas plataformas permiten registrar la asistencia de los niños, enviar reportes en tiempo real a los padres y gestionar actividades diarias. Tomar como referencia estas plataformas para realizar un nuevo proyecto podría ser de gran ayuda.

Spring Nro. 1- Leandro Pasteur – Técnicas avanzadas de la programación

Viabilidad Económica: La inversión inicial en la implementación de software comercial o en el desarrollo de una aplicación propia puede justificarse con la optimización de tiempos administrativos y la reducción de costos operativos.

Viabilidad Operativa: El personal del jardín ha mostrado disposición a adoptar herramientas digitales. Por ejemplo, una correcta capacitación en el uso de la plataforma digital puede permitir que las maestras registren fácilmente información sobre cada niño y la compartan con los padres en tiempo real, mejorando la comunicación sin interrumpir su rutina de trabajo.

2. Identificación del Proyecto

Enunciar los Objetivos del Proyecto

El propósito principal del proyecto es desarrollar un Sistema de Gestión Integral para un jardín maternal que atiende a niños de 0 a 3 años. Este sistema tiene como objetivo optimizar y centralizar las operaciones administrativas y pedagógicas, mejorando la comunicación entre el personal educativo y las familias, y asegurando un entorno seguro y eficiente para el desarrollo infantil.

Objetivos Específicos

Implementar un módulo que permita registrar la asistencia diaria de los niños mediante identificación digital, como códigos QR o huellas dactilares, reduciendo errores y facilitando el seguimiento de la asistencia.

Desarrollar herramientas que permitan a las maestras registrar y monitorear los horarios y detalles de alimentación y descanso de cada niño, asegurando un seguimiento personalizado y preciso.

Crear una plataforma que facilite la comunicación en tiempo real entre el jardín maternal y los padres o tutores, proporcionando información diaria sobre las actividades, estado de salud y bienestar de los niños.

Integrar funcionalidades para la gestión eficiente de inscripciones, pagos electrónicos, horarios de ingreso y salida, y otros procesos administrativos, reduciendo la carga de trabajo manual y minimizando errores.

Implementar un módulo que permita la generación de informes detallados sobre la evolución y desarrollo de cada niño, incluyendo aspectos como hábitos de alimentación,

Spring Nro. 1- Leandro Pasteur – Técnicas avanzadas de la programación

patrones de sueño e hitos de desarrollo, facilitando el seguimiento por parte de maestras y familias.

Desarrollar un sistema de alertas que notifique de inmediato a los padres y al personal en caso de emergencias médicas o incidentes dentro del jardín, permitiendo una respuesta rápida y coordinada.

A) Formulación del Proyecto

Alcance del Proyecto: **“Alcance Básico”**

- Registro de asistencia de los niños mediante identificación digital (código QR).
- Registro de alimentación y descanso con detalles de horarios y cantidades consumidas.
- Comunicación básica con los padres a través de mensajes automatizados y notificaciones en una aplicación.
- Gestión de datos administrativos incluyendo inscripción, pagos electrónicos y organización de horarios de ingreso y salida.

“Alcance de Detalle”

- Implementación de una aplicación móvil con notificaciones en tiempo real sobre actividades diarias, estado de salud y comportamiento del niño.
- Generación de reportes detallados sobre la evolución de cada niño, incluyendo gráficos de desarrollo y hábitos de alimentación y sueño.
- Módulo de alertas para emergencias médicas o incidentes dentro del jardín, con opción de respuesta rápida de los padres.
- Integración con plataformas de pago en línea para automatizar la gestión de cobros, recordatorios y estados de cuenta.

B) Análisis del Tamaño del Proyecto

Al optar por desarrollar el sistema desde cero, es fundamental dimensionar el proyecto de forma integral para garantizar que la solución cumpla con las necesidades del jardín maternal y pueda escalar en función de su crecimiento. A continuación, se detallan los factores críticos a considerar.

I. Numero de registros niños en la plataforma: “Grande (más de 70-100 niños)”:

Requerimientos:

- Se necesita una infraestructura robusta que soporte altos volúmenes de transacciones y múltiples usuarios activos simultáneamente, incluyendo funciones avanzadas de reporte y análisis en tiempo real.

La arquitectura deberá estar diseñada para ser distribuida, posiblemente con microservicios, para manejar de manera eficiente la alta concurrencia y el gran volumen de datos.

II. Número de Educadores y Personal Administrativo

Cada educador y miembro del personal administrativo representa un usuario activo del sistema. Es fundamental que el software soporte múltiples accesos concurrentes sin degradar su rendimiento.

Es esencial considerar licencias escalables, sistemas de autenticación robustos y soporte para roles diferenciados (por ejemplo, permisos para edición, solo visualización, etc.).

III. Funcionalidades RequeridasNivel Básico (Mínimo Viable):

Registro digital de asistencia (con métodos de identificación como código QR o huella digital).

Registro de alimentación y descanso, con campos específicos para horarios, cantidades y observaciones.

Comunicación básica con los padres mediante mensajes y notificaciones automáticas.

Gestión de datos administrativos (inscripciones, cobros, horarios) en un módulo central.

Nivel Avanzado (Funcionalidades Detalladas):

Aplicación móvil con interfaz intuitiva, que permita notificaciones en tiempo real sobre eventos críticos (por ejemplo, alertas de emergencia o incidentes médicos).

Reportes detallados y gráficos de evolución del niño, generados a partir de la recopilación diaria de datos.

Spring Nro. 1- Leandro Pasteur – Técnicas avanzadas de la programación

Módulo de alertas personalizadas que notifique a los padres y al personal sobre incidencias (como alergias, cambios bruscos en el comportamiento, etc.).

Integración de un sistema de pagos en línea desarrollado a medida, que permita la automatización de cobros, recordatorios y gestión de facturación.

3) Análisis del Contexto y Necesidades

A. Contexto General

Muchos jardines maternos usan procesos manuales (papel, hojas de cálculo) para registros de asistencia, alimentación, descanso y comunicación, lo que genera ineficiencias, errores y falta de información en tiempo real.

Los padres exigen información instantánea sobre la jornada de sus hijos (alimentación, siestas, incidencias), requiriendo sistemas que faciliten una comunicación directa y actualizada.

La carga administrativa es alta por el manejo manual de inscripciones, pagos y registros, lo que puede optimizarse con la digitalización, liberando tiempo para la atención directa a los niños.

B. Necesidades en General y en Particular

Necesidades Generales:

Automatizar registros de asistencia, alimentación, descanso y actividades para reducir errores y aumentar la eficiencia.

Establecer canales y notificaciones directas para mantener informados a los padres sobre el estado y actividades de sus hijos.

Centralizar la administración de inscripciones, pagos y horarios para facilitar la coordinación interna.

Implementar medidas robustas que protejan los datos de niños, padres y personal, cumpliendo con la legislación vigente.

Necesidades Específicas:

Permitir a las maestras anotar información específica sobre alimentación (horarios, cantidades), descanso (duración, calidad) y comportamiento.

Spring Nro. 1- Leandro Pasteur – Técnicas avanzadas de la programación

Desarrollar un sistema de alertas para emergencias o incidentes que notifique de inmediato a padres y personal.

Diseñar una interfaz intuitiva y accesible para facilitar la adopción del sistema por educadores, administrativos y padres.

Generar reportes personalizados que muestren la evolución de cada niño, permitiendo evaluar tendencias e intervenir tempranamente.

Automatizar cobros e inscripciones con integración de plataformas de pago, facilitando recordatorios y seguimiento de pagos pendientes.

Análisis FODA del proyecto “Jardín Maternal”:



4. Análisis de Factibilidad Técnica de Alternativas de Solución y Lenguaje Seleccionado

Al desarrollar el sistema desde cero, se evaluaron varias alternativas técnicas y lenguajes de programación, considerando tanto los costos de desarrollo como los requisitos técnicos para garantizar un sistema robusto y escalable.

Arquitectura basada en Java con Spring Boot y ReactBackend:

Lenguaje y Framework: Java con Spring Boot.

Base de Datos: MySQL.

Justificación Técnica:

Robustez y Escalabilidad: Java es ampliamente reconocido por su rendimiento y estabilidad en aplicaciones. Spring Boot permite la creación de microservicios y facilita el desarrollo de APIs robustas.

Seguridad: Ofrece una amplia gama de bibliotecas y configuraciones para implementar medidas de seguridad (autenticación, autorización, cifrado) que son críticas al manejar datos sensibles de menores.

Costos: Licenciamiento: Tanto Java como Spring Boot son de código abierto, lo que reduce los costos iniciales.

Infraestructura: Se puede comenzar con un servidor de bajo costo y escalar en la nube (por ejemplo, AWS o Google Cloud) conforme crezca la demanda, optimizando los costos operativos.

Frontend:

Lenguaje y Framework: React

Justificación Técnica:

Interfaz Moderna y Responsiva: React permite desarrollar interfaces de usuario dinámicas y fáciles de usar, fundamentales para una experiencia intuitiva.

Comunidad y Soporte: Gran cantidad de librerías y recursos disponibles, lo que facilita el desarrollo y la integración con otras tecnologías.

Costos: Herramientas de desarrollo gratuitas y la posibilidad de reutilizar componentes reducen tiempos y costos.

5. Tablero de control del proyecto, de tipo scrum

<https://trello.com/b/tdLO7Qah/proyecto-sistema-jardin-maternal-scrum>

Anexo: Entrevistas y (cuestionarios abiertos y cerrados)

Entrevista con directora del jardín:

- ¿Cuáles son las principales dificultades en la organización diaria del jardín?

- Gestionar los horarios es de los principales desafíos, ya que cada niño tiene una rutina diaria de alimentación y descanso. También es complicado que cada maestra pueda garantizar un registro y comunicar concretamente lo que sucede con cada niño a lo largo del día.

A veces la cantidad de información que debemos manejar es demasiada y terminamos priorizando lo más urgente, dejando afuera algunos detalles importantes. Hay padres que requieren información más detallada sobre el bienestar de sus hijos, pero al final del día, el tiempo para dialogar con ellos es muy limitado.

- ¿Cómo gestionan actualmente la información de cada niño?

- Cada maestra anota los eventos del día en cuadernos individuales. Incluyendo que comieron, si durmieron o si hubo algún incidente. Aunque con respecto a las dificultades puedo notar que es un método un poco anticuado y poco eficiente porque a veces se presta a pérdidas de información o dificultades para acceder rápidamente a los datos cuando los padres hacen consultas.

Nos gustaría tener un sistema que nos permita registrar este tipo de información de una manera más estructurada y accesible.

- ¿Qué funcionalidades considera imprescindibles para un posible nuevo sistema digital de gestión?

- Principalmente nos gustaría un sistema que les permita a las maestras registrar rápidamente lo que sucede con cada niño a lo largo del día. Además, una funcionalidad que les permita a los padres acceder en tiempo real a esta información. Otro aspecto que tendría en cuenta es el control de asistencia, ya que algunos niños llegan y se retiran en diferentes horarios y hasta ahora eso se maneja de forma manual.

Por último, me gustaría incluir recordatorios de pagos y gestión administrativa para organizar mejor la parte financiera del jardín.

Entrevista con una maestra del jardín:

- ¿Cuáles son los registros más difíciles de llevar a cabo dentro de una jornada en el jardín?

- Los de la comida y el descanso porque algunos niños tienen necesidades dietéticas específicas y a veces recordar cada una de ellas es complicado si no tenes un registro bien organizado.

- ¿Cómo registra actualmente esos datos?

- Tenemos cuadernos donde anotamos lo que cada niño comió, si durmió y cualquier incidente importante. Todo esto a mano lo que a veces se vuelve lento y nos lleva a equivocarnos a veces.

- ¿Crees que un sistema digital de gestión podría ayudar a solucionar estos problemas?

- Seguramente. Si tuviéramos una aplicación sencilla donde podamos anotar todo de forma más detallada y concreta, nos ahorraríamos más tiempo y les brindaríamos una mejor atención a los niños y a los padres.

Entrevista con un padre:

- ¿Qué tipo de información considera esencial recibir diariamente sobre su hijo?

- Me gustaría saber que actividades realizo en el día, si comió bien, si durmió lo suficiente o si tuvo algún inconveniente. También sería genial recibir fotos o pequeños informes sobre sus actividades, para sentirme más conectado con su día a día.

- ¿Actualmente qué método utiliza el jardín para informarle sobre estos aspectos?

- Ahora mismo, la comunicación es bastante informal. Algunas veces me cuentan de manera rápida cuando voy a buscarlo, pero no siempre hay tiempo para recibir todos los detalles. A veces dejan notas en el cuaderno.

- ¿Le resultaría útil un sistema que permita recibir este tipo de información en tiempo real y tener una comunicación rápida con las maestras?

- Sí, definitivamente. Sería ideal poder enviar y recibir mensajes o incluso las maestras puedan reportar cualquier situación importante al momento. Así me sentiría más tranquilo y sabría que puedo responder o actuar rápidamente si es necesario.

Entrevista con un Miembro del Personal Administrativo:**- ¿Cuáles son los principales desafíos en la gestión administrativa del jardín?**

- El control de asistencia y los pagos. A veces los padres olvidan pagar a tiempo y tenemos que hacer múltiples recordatorios. También, la actualización de datos de los niños suele ser desordenada.

- ¿Cómo manejan actualmente la facturación y los cobros?

- Se hacen manualmente, enviamos recordatorios por WhatsApp o llamamos a los padres.

- ¿Cree que un sistema digital con alertas y pagos en línea mejoraría la administración?

- Sin duda. Si pudiéramos automatizar los recordatorios y permitir pagos digitales, ahorraríamos mucho tiempo y evitaríamos retrasos innecesarios.

Cuestionarios (Abiertos y Cerrados)

Se aplicaron cuestionarios para conocer la percepción de los padres y de los principales actores del jardín sobre la gestión de la información y la comunicación.

Cuestionario Cerrado - Directora**- ¿Cómo registran actualmente la asistencia de los niños? (Papel/Excel/Ningún registro formal)**

- Papel, en planillas.

- ¿Con qué frecuencia se presentan problemas de comunicación con los padres? (Nunca/Rara vez/A veces/Frecuentemente)

- A veces, especialmente cuando hay urgencias.

- ¿Tienen algún sistema digital para registrar la información de los niños? (Sí/No)

- No.

- ¿Considerarían útil un sistema digital para mejorar la gestión? (Sí/No)

- Sí, nos ayudaría a optimizar tiempos y organización.

Cuestionario Cerrado – Maestra

- **¿Cómo registran actualmente la alimentación y descanso de los niños?**

(Papel/Excel/Ningún registro formal)

- Papel.

- **¿Qué tan efectiva considera la comunicación actual con los padres? (Muy efectiva/Efectiva/Regular/Ineficiente)**

- Regular.

- **¿Cuántos minutos en promedio dedica a actualizar registros diarios de cada niño?**

(Menos de 5/5-10/10 o más)

- 5-10 minutos.

- **¿Considera útil un sistema digital para registrar información diaria de cada niño?**

(Sí/No)

- Sí, reduciría errores y mejoraría la comunicación.

Cuestionario Cerrado – Padres

- **¿Cómo le gustaría recibir información sobre el bienestar diario de su hijo? (Notas en papel/Mensajes en WhatsApp/Aplicación específica)**

- Aplicación específica.

- **¿Siente que recibe suficiente información sobre su hijo? (Sí/No)**

- No, podría ser más completa.

- **¿Le gustaría recibir alertas en tiempo real sobre la alimentación y descanso de su hijo? (Sí/No)**

- Sí.

- **¿Utilizaría una aplicación para comunicarse con los educadores del jardín? (Sí/No)**

- Sí.

Cuestionario Abierto – Directora

- **¿Qué mejoras cree que necesita la gestión administrativa del jardín?**

Spring Nro. 1- Leandro Pasteur – Técnicas avanzadas de la programación

- Necesitamos automatizar los pagos y mejorar el control de asistencia.
- **¿Cómo afecta la falta de un sistema digital a la organización diaria?**
- Perdemos mucho tiempo en tareas manuales y a veces hay errores en los registros.

Cuestionario Abierto – Maestra

- **¿Qué información cree que debería incluir un sistema de gestión?**
- Registro de alimentación, descanso y alertas de salud.
- **¿Cuáles es la mayor dificultad que enfrenta al registrar la información de los niños?**
- Falta de tiempo y acumulación de tareas.

Cuestionario Abierto – Padre

- **¿Qué aspectos de la gestión del jardín considera que podrían mejorar?**
- Mayor rapidez y claridad en la comunicación.
- **¿Cuáles son las principales preocupaciones al dejar a su hijo en el jardín?**
- No recibir información a tiempo sobre su estado y actividades.



**Universidad Metropolitana para
la Educación y el Trabajo**

Técnicas Avanzadas de la Programación
“Spring Nro. 2”

Alumno: Leandro Pasteur

Profesora: Antonieta Kuz

Fecha: 12/04/2025

Índice:

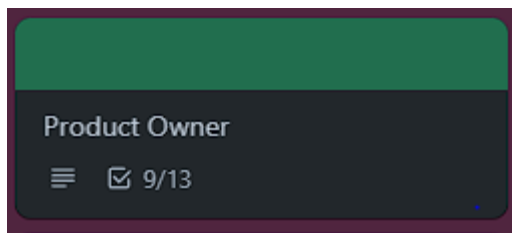
1) Tableros del proyecto (Scrum)	PAG 3 - 11
2) Definir los casos de uso completos	PAG 12 - 17
3) Análisis funcional (requisitos funcionales y no funcionales).....	PAG 18
4) Diagrama de Historias de usuario.....	PAG 19
5) Diagrama burn down	PAG 19
6) Repositorio GitHub	PAG 19

1) Tablero del proyecto “Jardín Maternal” Scrum:

A continuación, detallo todas las características del tablero de tipo Scrum, hecho con la aplicación “TRELLO”.



Roles: En este apartado se encuentran cada uno de los integrantes que participan en el desarrollo de este proyecto:



Ejemplo de como se visualizan los integrantes del proyecto en el Tablero de Trello

Product Owner (PO)

Función Principal: Representa al cliente (jardín maternal), define el valor del producto y prioriza las tareas que generan más impacto para los usuarios.

Analista Funcional (AF)

Función Principal: Traducir las necesidades del usuario en especificaciones técnicas comprensibles para el equipo de desarrollo.

Scrum Master (SM)

Función Principal: Facilita el proceso SCRUM, elimina obstáculos y asegura que el equipo trabaje con fluidez bajo la metodología ágil.

Documentador Técnico (D)

Función Principal: Redactar y mantener actualizada toda la documentación del proyecto para su correcta entrega y evaluación.

Diseñador UX/UI

Función Principal: Diseñar la interfaz del sistema centrada en el usuario, asegurando una experiencia intuitiva, estética y accesible.

Desarrollador Frontend (DF)

Función Principal: Implementar el diseño de interfaces en el navegador y conectar con el backend para mostrar los datos correctamente.

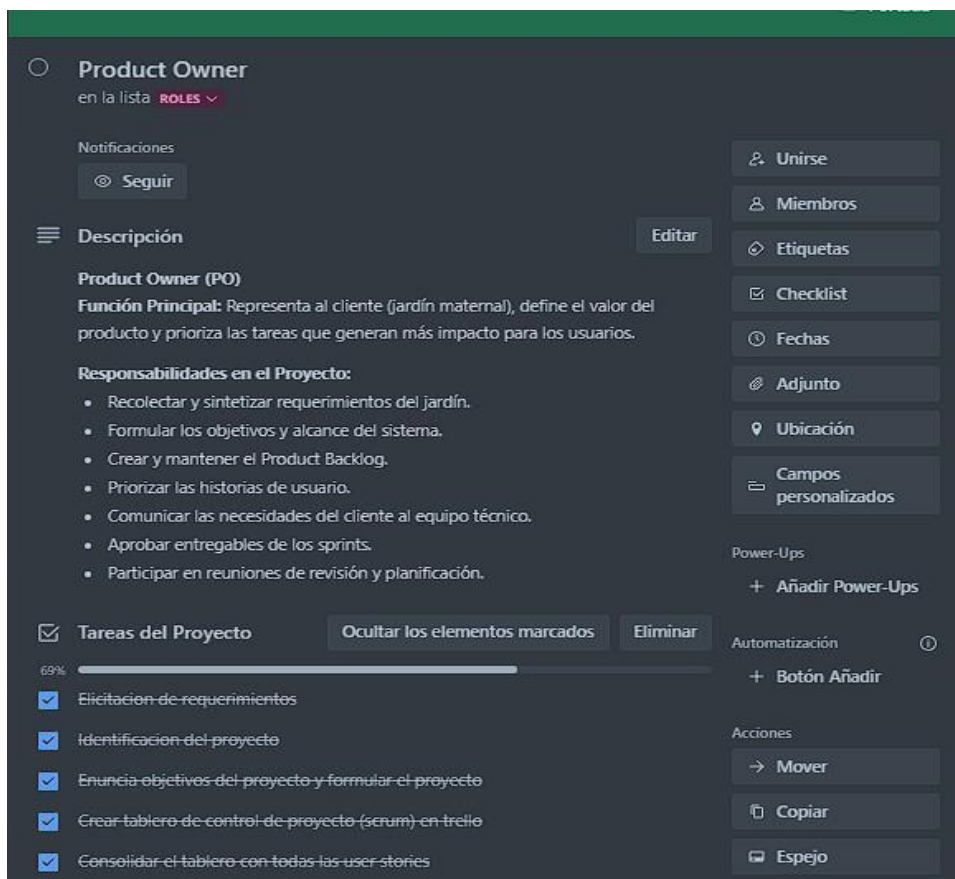
Desarrollador Backend (DB)

Función Principal: Programar la lógica del sistema, la base de datos, API y las funcionalidades principales del sistema.

Desarrollador Fullstack (FS)

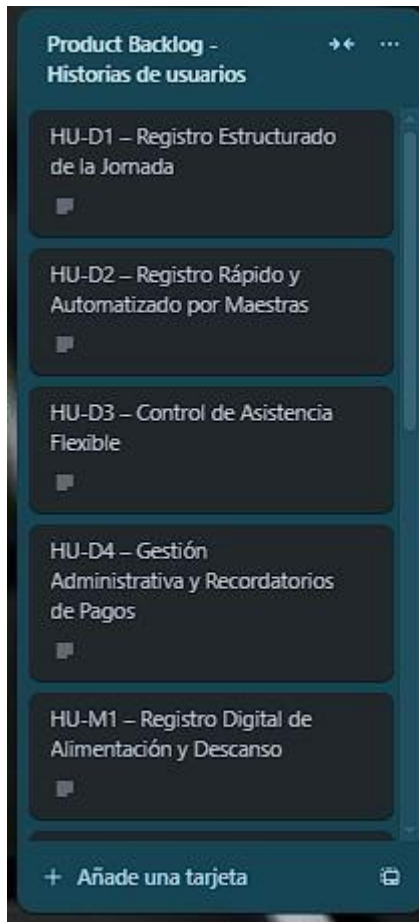
Función Principal: Es responsable de implementar tanto el frontend como el backend del sistema. Tiene una visión integral del desarrollo, lo que permite integrar y sincronizar correctamente la base de datos, la lógica del negocio y la interfaz de usuario.

Dentro de cada tarjeta de los integrantes del proyecto se visualiza una descripción, y un apartado de Checklist con cada una de las tareas que debe realizar. Cuando se termina las tareas se debe marcar como realizada.



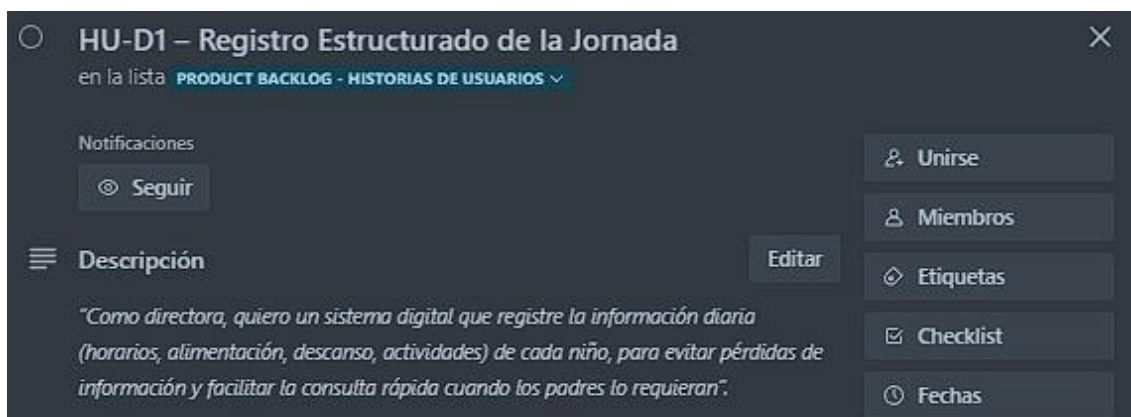
Apartado llamado Product Backlog – Historias de usuario.

Dentro de este apartado se encuentran tarjetas con cada una de las historias de usuarios:



Ejemplo de cómo se visualizan las historias de uso en el Tablero de Trello

Al ingresar se puede visualizar una descripción de cada una de las historias:



Ejemplo de cómo se visualizan las tarjetas de las historias de uso en el Tablero de Trello

Apartado llamado Product Backlog.

En este apartado se detallan todas las tareas que se van a desarrollar a lo largo del proyecto:

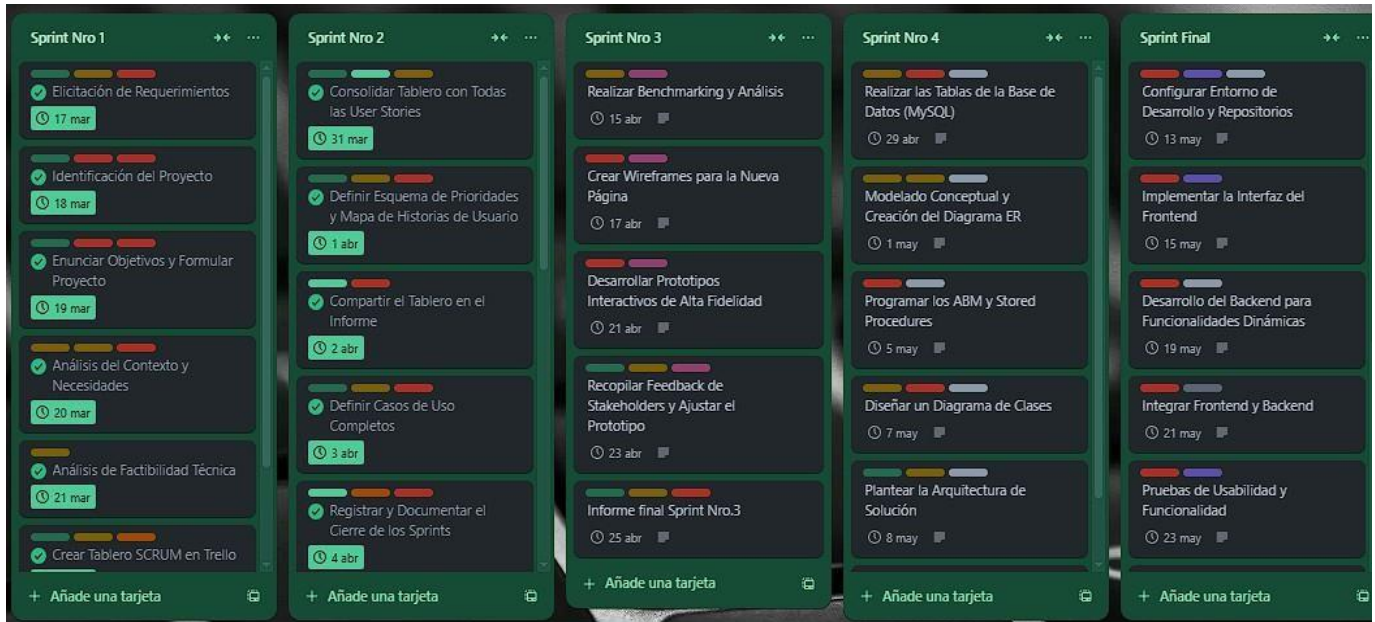


Dentro de las tarjetas se pueden visualizar etiquetas donde se detalla quien debe realizar la tarea y cuál es su prioridad (Alta, Media o Baja). Además, se realiza una descripción de la tarea a realizar.



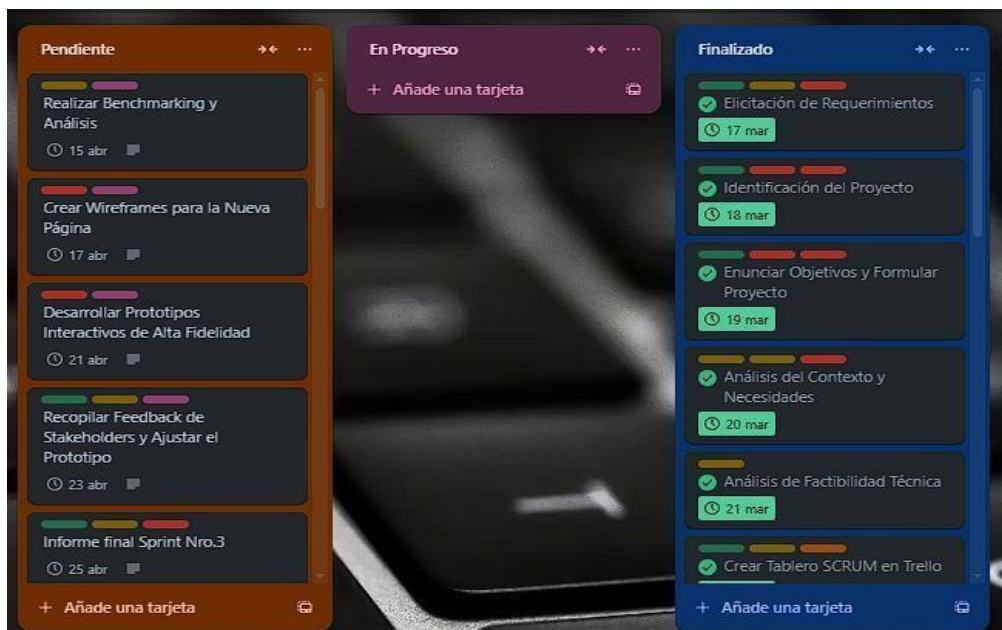
Apartados de Sprints

Armamos este apartado para dividir las tareas por número de sprint, detallando quien debe realizar la tarea, cuál es su prioridad y las fechas de finalización de cada tarea (marcadas con una tilde cuando están realizadas).



Apartados con estado de las tareas

En estos apartados se pueden ver los estados de las tareas (Pendiente, En proceso, Finalizado).



Link del Trello: <https://trello.com/b/tLO7Qah/proyecto-sistema-jardin-maternal-scrum>

Se divide en las siguientes columnas:

- ### Sprint 1:

Nro. de Sprint	Tarea	Quién	Estado	Día 1/ Lunes	Día 2/ Martes	Día 3/ Miercoles	Día 4/ Jueves	Día 5/ Viernes	Día 6/ Lunes	Día 7/ Martes	Día 8/ Miercoles	Día 9/ Jueves	Día 10/ Viernes	Horas Pendientes	Fecha de duración
1	Elicitación de Requerimientos	Product owner/ Analista	Completada	8	0	0	0	0	0	0	0	0	0	0 hrs	17/3/2025
1	Identificación del Proyecto	Product Owner	Completada	0	8	0	0	0	0	0	0	0	0	0 hrs	18/3/2025
1	Enunciar Obj. y Formular el Proyecto	Product owner/ Documentador	Completada	0	0	8	0	0	0	0	0	0	0	0 hrs	19/3/2025
1	Análisis del Contexto y Necesidades	Analista/ Documentador	Completada	0	0	0	8	0	0	0	0	0	0	0 hrs	20/3/2025
1	Análisis de Factibilidad Técnica	Analista	Completada	0	0	0	0	8	0	0	0	0	0	0 hrs	21/3/2025
1	Crear Tablero de Control del Proyecto (SCRUM) en	Product owner/ Scrum Master	Completada	0	0	0	0	0	8	0	0	0	0	0 hrs	24/3/2025
1	Informe Final del Sprint Nro.1	Product owner/ Documentador	Completada	0	0	0	0	0	0	8	0	0	0	0 hrs	25/3/2025
Final del Sprint Nro 1 - Desde el día 17/03/2025 al 27/03/2025															

2	Consolidar Tablero con Todas las User Stories	Product owner/ Analista	Completada	8	0	0	0	0	0	0	0	0	0	0 hrs	31/3/2025
2	Definir Esquema de Prioridades y Mapa de	Product owner/ Documentador	Completada	0	8	0	0	0	0	0	0	0	0	0 hrs	1/4/2025
2	Compartir el Tablero en el Informe	Documentador	Completada	0	0	8	0	0	0	0	0	0	0	0 hrs	2/4/2025
2	Definir Casos de Uso Completos	Product owner/ Analista	Completada	0	0	0	8	0	0	0	0	0	0	0 hrs	3/4/2025
2	Registrar y Documentar el Cierre de los Sprints	Scrum Master/ Documentador	Completada	0	0	0	0	8	0	0	0	0	0	0 hrs	4/4/2025
2	Realizar Análisis Funcional (RF y RNF)	Analista	Completada	0	0	0	0	0	8	0	0	0	0	0 hrs	7/4/2025
2	Agregar Diagrama de Casos de Uso/Mapa	Analista/ Documentador	Completada	0	0	0	0	0	0	8	0	0	0	0 hrs	8/4/2025
2	Elaborar Gráfico Burn Down	Scrum Master/ Documentador	Completada	0	0	0	0	0	0	0	8	0	0	0 hrs	9/4/2025
2	Crear Repositorio GitHub y Cargar Documentación	Documentador	Completada	0	0	0	0	0	0	0	0	8	0	0 hrs	10/4/2025
2	Informe final Sprint Nro.2	Product owner/ Documentador	Completada	0	0	0	0	0	0	0	0	0	8	0 hrs	11/4/2025
Final del Sprint Nro 2 - Desde el día 31/03/2025 al 11/04/2025															

[illegible]

4	Realizar las Tablas de la Base de Datos (MySQL)	Analista/ Desarrollador Backend	Pendiente	8	8	0	0	0	0	0	0	0	0	16 hs	28/04/2025 - 29/04/2025
4	Modelado Conceptual y Creación del Diagrama ER	Analista/ Desarrollador Backend	Pendiente	0	0	8	8	0	0	0	0	0	0	16 hs	30/04/2025 - 1/05/2025
4	Programar ABM y Stored Procedures	Desarrollador Backend	Pendiente	0	0	0	0	8	8	0	0	0	0	16 hs	2/05/2025 - 5/05/2025
4	Diseñar un Diagrama de Clases	Documentador/ Desarrollador Backend	Pendiente	0	0	0	0	0	0	8	8	0	0	16 hs	6/05/2025 - 7/05/2025
4	Plantear la Arquitectura de Solución	Product owner/ Desarrollador Backend	Pendiente	0	0	0	0	0	0	0	0	8	0	8 hs	8/5/2025
4	Informe final Sprint Nro.4	Product owner/ Documentador	Pendiente	0	0	0	0	0	0	0	0	0	8	8 hrs	9/5/2025
Final del Sprint Nro 4 - Desde el día 28/04/2025 al 09/05/2025															

Final	Configurar Entorno de Desarrollo y Repositorio	Desarrollador Backend/ Desarrollador Frontend	Pendiente	8	8	0	0	0	0	0	0	0	0	16 hs	12/05/2025 - 13/05/2025
Final	Implementar la Interfaz del Frontend	Desarrollador Frontend	Pendiente	0	0	8	8	0	0	0	0	0	0	16 hs	14/05/2025 - 15/05/2025
Final	Desarrollo del Backend para Funcionalidades	Desarrollador Backend	Pendiente	0	0	0	0	8	8	0	0	0	0	16 hs	16/05/2025 - 19/05/2025
Final	Integrar Frontend y Backend	Fullstack	Pendiente	0	0	0	0	0	0	8	8	0	0	16 hs	20/05/2025 - 21/05/2025
Final	Pruebas de Usabilidad y Funcionalidad	Desarrollador Frontend	Pendiente	0	0	0	0	0	0	0	0	8	8	16 hs	22/05/2025 - 23/05/2025
Final del Sprint Final Parte 1 - Desde el dia 12/05/2025 al 23/05/2025															
Final	Optimización y Ajustes de Rendimiento	Desarrollador Frontend	Pendiente	8	8	0	0	0	0	0	0	0	0	16 hs	26/05/2025 - 27/05/2025
Final	Deployment y Configuración del	Desarrollador Backend	Pendiente	0	0	8	0	0	0	0	0	0	0	8 hrs	28/5/2025
Final	Revisión Final, Correcciones y	Product Owner	Pendiente	0	0	0	8	0	0	0	0	0	0	8 hrs	29/5/2025
Final	Documentación Técnica y Manual de Usuario	Documentador	Pendiente	0	0	0	0	8	0	0	0	0	0	8 hrs	30/5/2025
Final del Sprint Final Parte 2 - Desde el dia 26/04/2025 al 30/05/2025															

Al final de la tabla se puede visualizar las horas trabajadas y las horas restantes

Horas Realizadas	136 Horas
Horas Restantes	280 Horas

- *Se adjuntará a la entrega el archivo Excel de la tabla. Además, se encuentra adjunto en el Trello (Product Backlog / “Crear Tablero de Control del Proyecto (SCRUM)”)*

2) Definir casos de uso: A continuación, definiremos los casos de uso dentro del proyecto.

CU01 Registrar Asistencia

Actores:

Directora / Maestra y Niño (identificado mediante QR o huella)

Precondiciones:

- El usuario debe estar autenticado.
- El dispositivo (lector de QR o sensor biométrico) debe estar operativo.

Flujo Principal:

- ✓ La maestra inicia sesión en el sistema.
- ✓ Se activa el módulo de registro.
- ✓ El dispositivo capta la identificación del niño mediante QR o huella.
- ✓ El sistema registra la asistencia con la hora exacta.
- ✓ Se muestra una confirmación visual y/o sonora.

Flujos Alternativos:

- Si la identificación falla, se solicita reintentar el escaneo o se activa el registro manual temporal.

Postcondiciones:

- El registro queda almacenado con fecha y hora.

Requerimientos Especiales:

- Respuesta rápida y de alta precisión.

CU02 Registrar Alimentación y Descanso

Actores: Maestra

Precondiciones:

- La maestra está autenticada.
- Disponibilidad de formularios digitales preconfigurados.

Flujo Principal:

- ✓ La maestra accede al módulo correspondiente.
- ✓ Selecciona al niño y completa el formulario de alimentación (tipo, cantidad, hora) y descanso (inicio, fin, observaciones).
- ✓ El sistema guarda la información en la base de datos.
- ✓ Se muestra una confirmación del registro.

Flujos Alternativos:

- Si falta información, el sistema solicita completarla antes de guardar.

Postcondiciones:

- Los datos se asocian al historial del niño.

Requerimientos Especiales:

- Formularios adaptables para necesidades específicas (alergias, dietas especiales).

CU03 Enviar Notificaciones en Tiempo Real

Actores: Sistema (con intervención de maestra).

Precondiciones:

- El sistema debe estar configurado para notificaciones.
- Los padres tienen dispositivos con notificaciones activadas.

Flujo Principal:

- ✓ La maestra registra un evento significativo (incidente, cambio de horario, etc.).
- ✓ El sistema procesa la información y genera una notificación.
- ✓ La notificación se envía a los dispositivos de los padres.
- ✓ El padre recibe el aviso y puede confirmar o responder.

Flujos Alternativos:

- Si el envío falla, se reintenta y se notifica a la maestra.

Postcondiciones:

- El evento se queda registrado en el historial de notificaciones.

Requerimientos Especiales:

- Baja latencia y alta fiabilidad.

CU04 Comunicación Directa (Chat)

Actores: Maestra y Padres

Precondiciones:

- Los usuarios deben estar autenticados y tener acceso al módulo de chat.

Flujo Principal:

- ✓ La maestra inicia una conversación a través del chat.
- ✓ Se envían mensajes (texto, imágenes o archivos).
- ✓ El sistema guarda y muestra el historial del chat.
- ✓ El padre puede responder y la conversación queda registrada.

Flujos Alternativos:

- Si falla el envío, se notifica al usuario y se sugiere reintentar.

Postcondiciones:

- Se registra el historial completo de la conversación.

Requerimientos Especiales:

- Interfaz intuitiva y soporte para multimedia.

CU05 Gestión Digital de Inscripciones y Pagos

Actores: Administrativo, Padre y Directora.

Precondiciones:

- El administrativo tiene acceso al módulo.
- Los padres ya están registrados en el sistema.
-

Flujo Principal:

- ✓ El administrativo ingresa nuevos registros o actualiza existentes.
- ✓ Se generan cuotas y se procesa el pago digitalmente.
- ✓ El sistema actualiza el estado del pago y notifica recordatorios en caso de mora.

Flujos Alternativos:

- En caso de error en la transacción, se notifica y se solicita reintentar.

Postcondiciones:

- Los registros de inscripciones y pagos se actualizan correctamente.

Requerimientos Especiales:

- Seguridad en la transacción y validación de datos.

CU06 Generar Reportes

Actores: Directora y Administrativo

Precondiciones:

- El sistema cuenta con datos completos (asistencia, alimentación, descansos, pagos).

Flujo Principal:

- ✓ El director solicita la generación del reporte.
- ✓ El sistema procesa la información y muestra gráficos y tablas con indicadores.
- ✓ El usuario puede filtrar y visualizar los datos según lo necesite.

Flujos Alternativos:

- Si faltan datos, el sistema notifica las entradas incompletas.

Postcondiciones:

- Se visualizan reportes claros y actualizados.

Requerimientos Especiales:

- Interfaz intuitiva y opciones de filtrado.

CU07 Sistema de Alertas para Emergencias

Actores: Sistema (con intervención del personal del jardín y padres)

Precondiciones:

- El sistema debe estar configurado para detectar eventos críticos.
- Los dispositivos de notificación deben estar activos.

Flujo Principal:

- ✓ Se detecta un evento de emergencia (por ejemplo, incidente médico).
- ✓ El sistema envía alertas inmediatas a padres y personal.
- ✓ La notificación contiene información básica y, de ser posible, ubicación o detalles del incidente.

Flujos Alternativos:

- Si no se envía, se reintenta y se notifica al personal.

Postcondiciones:

- La emergencia queda registrada y notificada en el historial.

Requerimientos Especiales:

- Baja latencia y alta robustez en el envío.

CU08 Informe Final y Actualización del Tablero SCRUM

Actores: Product Owner y Scrum Master

Precondiciones:

- Todos los sprints deben estar cerrados y documentados.

Flujo Principal:

- ✓ El Product Owner recoge la evidencia del tablero SCRUM y de cada sprint.
- ✓ Se elabora un informe final que consolida el progreso, diagramas y métricas.
- ✓ Se comparte el informe.

Postcondiciones:

- El informe final queda archivado y disponible para revisión.

Requerimientos Especiales:

- Documentación clara, con evidencia visual y datos actualizados.

3) Requisitos funcionales y no funcionales

Requisitos Funcionales	Requisitos No Funcionales
<ul style="list-style-type: none"> • El sistema debe permitir el registro automático de asistencia mediante QR o huella dactilar. • El sistema debe permitirles a las maestras registrar alimentación y descansos de cada niño. • El sistema debe enviar notificaciones en tiempo real a los padres sobre eventos importantes. • El sistema debe contar con un módulo de comunicación (chat) entre maestras y padres. • El sistema debe gestionar inscripciones de alumnos, incluyendo información personal y médica. • El sistema debe permitir realizar y gestionar pagos digitales de cuotas. • El sistema debe generar reportes de asistencia, alimentación y descanso por niño. • El sistema debe emitir alertas en caso de emergencias médicas o incidentes. • El sistema debe registrar el historial del niño (comidas, siestas, eventos, reportes). • El sistema debe registrar usuarios con diferentes roles: administrador, maestra o padre. • El sistema debe permitir a los docentes subir imágenes y observaciones del día. 	<ul style="list-style-type: none"> • El sistema debe ser accesible desde dispositivos móviles y de escritorio. • El sistema debe tener estar disponible durante el horario escolar (08:00 a 18:00). • El tiempo de respuesta ante una acción del usuario debe ser instantáneo. • El sistema debe proteger los datos personales de acuerdo con la normativa de protección de datos. • La autenticación de usuarios debe contar con encriptación y niveles de acceso diferenciados. • Las notificaciones deben enviarse instantáneamente desde el momento del evento. • El sistema debe permitir escalar en cantidad de usuarios sin perder rendimiento. • Las actualizaciones del sistema no deben interrumpir el servicio y deben ser programadas. • La interfaz debe ser intuitiva y usable incluso por personas con conocimientos básicos de informática. • El sistema debe ser capaz de integrarse con plataformas de pago seguras (MercadoPago, etc.).

4) **Historias de usuario**

El diagrama de historias de usuario fue realizado con la aplicación miro desde su sitio web. A continuación, dejo el link de acceso al Mapa:

<https://miro.com/app/board/uXjVIDygLlg=/>

5) **Diagrama Burn Down:**



En este grafico decidimos poner 28 puntos esperados, que se van restando a lo largo de los 26 días que duraron tanto el Sprint 1 sumando al Sprint 2. Los días no laborables que son los fines de semana, no se restaron puntos.

Se adjuntará a la entrega el archivo Excel

6) **Repositorio del proyecto en GITHUB**

<https://github.com/Leanpas98/Tecnicas-avanzadas-de-la-programacion---Proyecto-Jardin-Maternal->



**Universidad Metropolitana para
la Educación y el Trabajo**

Técnicas Avanzadas de la Programación
“Spring Nro. 3”

Alumno: Leandro Pasteur

Profesora: Antonieta Kuz

Fecha: 25/04/2025

Índice:

1) Perfiles representativo de usuario final (User persona)	Pág. 3 - 4 - 5
2) Definición y justificación de los colores seleccionados	Pág. 6 - 7
3) Logo del sistema (Versión color y monocromática)	Pág. 8
4) Bocetos de Wireframe de pantallas	Pág. 8
5) Esquemas visuales de estructura de navegación (SiteMap)	Pág. 9 - 10
6) Diseños responsivos	Pág. 11 – 12 – 13
7) Guía tipográfica	Pág. 14

Introducción

Para diseñar una experiencia de usuario (UX) adecuada y alineada a las necesidades reales del sistema que estamos desarrollando, decidimos identificar y representar a dos actores principales:


Mariana Pérez: Madre de un niño de 2 años que asiste al jardín maternal. Su perspectiva nos permite enfocar el diseño en las necesidades de los familiares responsables del seguimiento diario del niño, como la comunicación con el personal del jardín, la recepción de alertas en tiempo real y la gestión digital de trámites y pagos.

Romina Diaz: Maestra del jardín maternal. Representa al personal educativo. Ella es quien interactúa directamente con los niños, mantiene un contacto constante con las familias y gestiona tareas clave como la asistencia, el registro de actividades, el seguimiento del bienestar de cada niño y la comunicación con los padres.

Elegir a estos actores principales nos permite entender de forma más precisa las necesidades reales, las limitaciones diarias, y cómo nuestra plataforma podría optimizar sus tiempos, mejorar la comunicación y facilitar el seguimiento pedagógico.

1) User Persona – Mariana Pérez

User Persona - Mariana Pérez - Madre



Mariana Perez

Madre de niño que asiste a "Luz de Luna"

Contexto y Entorno

- Vive a 7 cuadras del jardín maternal (10 min caminando).
- Trabaja medio tiempo como asistente administrativa en empresa de servicios.
- Su pareja tiene turnos rotativos, por lo que ella deja y recoge a su hijo.
- Se desplaza en auto; cuando no, usa transporte público.

Datos Basicos

- Nombre: Mariana Pérez
- Edad: 34 años
- Género: Femenino
- Rol: Madre de familia / Tutora principal
- Educación: Licenciatura en Administración de Empresas
- Profesión: Asistente administrativa (Medio tiempo)
- Residencia: Zona urbana, barrio Flores

Tecnologias

- Smartphone: Samsung Galaxy S23 (Android)
- Laptop personal: Windows (oficina y hogar)
- Tablet familiar: iPad (uso esporádico)

Herramientas y metodos actuales

- Libro de asistencia físico: Firma de padres al llegar.
- Cuaderno de comunicación: Alimentaciones, siestas, incidencias escritas a mano.
- WhatsApp grupal: Avisos y envío de fotos.

Objetivos

- Saber asistencia, alimentación, siestas y estado de ánimo de su hijo.
- Tener un canal ágil para ausencias, consultas y novedades sin depender de llamadas ni papel.
- Recibir alertas inmediatas ante cualquier eventualidad.
- Gestionar pagos, autorizaciones y formularios desde un sitio web.

Frustraciones


- Retraso en información clave del día.
- Hacer filas o llenar papeles para trámites simples.
- No saber 'qué pasó hoy' hasta la entrega.
- Temor a no ser notificada ante urgencias.

Necesidades

- Consultar en tiempo real datos de su hijo.
- Alertas para malestares, accidentes o cambios de horario.
- Historial accesible para ver evolución en alimentación, sueño y desarrollo.
- Pagar cuotas, recibir facturas y comprobantes.
- Protección de datos sensibles de su hijo y la familia.

1) User Persona – Romina Diaz

User Persona - Romina Diaz - Mestra



Romina Diaz
Mestra de Jardín Maternal "Luz de Luna"

Datos Basicos

- Nombre: Romina Diaz
- Edad: 30 años
- Género: Femenino
- Rol: Maestra de sala (Niños de 2-3 años)
- Educación: Título terciario en Educación Inicial
- Profesión: Maestra en Jardín Maternal (jornada completa)
- Residencia: Zona urbana, barrio Flores

Objetivos

- Registrar información diaria de forma rápida y sin errores
- Mantener a las familias informadas con claridad y sin interrupciones
- Reducir la carga administrativa
- Documentar incidentes en el momento

Contexto y Entorno

- El jardín se encuentra a 15 cuadras de su casa; llega en 10 minutos.
- Comparte la sala con otra maestra; cada una atiende de 5 a 10 niños.
- Llega unos minutos antes de las 8:00 AM para organizar el aula.
- Mantiene actualizado el cuaderno de comunicaciones.
- No tiene computadora personal ni espacio privado en el aula.

Tecnologias

- iPhone 12 (iOS): uso para WhatsApp y fotos
- Laptop personal y PC del jardín (Windows)

Frustraciones

- Anotar todo a mano le quita tiempo y genera errores
- Falta de respuestas rápidas por parte de los padres
- Pérdida o daño de cuadernos
- No tiene una vía oficial para comunicar incidentes
- Poco tiempo para comunicar con padres al final del día

Herramientas y metodos actuales

- Libro de asistencia físico (firmado por padres)
- Cuaderno de registro diario (alimentación, siestas, incidentes)
- WhatsApp grupal para avisos y fotos

Necesidades

- Herramienta digital para registrar asistencia, comidas, siestas
- Plataforma institucional para comunicar con padres sin usar su celular
- Notificaciones rápidas para incidentes
- Acceso fácil a historial de cada niño

Las tarjetas de User Persona fueron realizadas con la aplicación Figma.

A continuación, dejamos el enlace de acceso

<https://www.figma.com/design/cpMl0WHWOi7MSfejJpVhMR/Proyecto-Jardin-Maternal?node-id=0-1&p=f&t=3epMVkugg51Q5W7D-0>

2) Paleta de colores:

<u>Uso</u>	<u>Color (Hex)</u>	<u>Nombre formal</u>
Fondo de sección principal	#FFE7D0	Durazno pálido
Barra de navegación superior	#A5DFF1	Turquesa pastel
Encabezados y títulos	#A93A2F	Coral profundo
Indicadores de estado positivo	#4EE07A	Verde esmeralda
Indicadores de advertencia	#DD6A22	Naranja ámbar
Indicadores de estado crítico	##DD2222	Rojo carmesí
Indicadores de información	#5853EA	Violeta grisáceo
Texto principal y enlaces	#000000	Negro puro
Superficies neutrales (tarjetas, pop-ups)	#FDF3F3	Rosado muy claro
Botones de llamada a la acción (CTA)	#F9D97B	Amarillo cálido claro

Justificación de los colores**Durazno pálido (#FFE7D0)**

Utilizado como fondo principal de secciones para ofrecer una sensación de calidez. Este tono suave reduce la fatiga visual en padres y maestras al revisar información.

Turquesa pastel (#A5DFF1)

Empleado en la barra de navegación superior para transmitir serenidad y confianza. Este color facilita la orientación dentro de la plataforma y crea un entorno tranquilo, acorde con el bienestar infantil.

Coral profundo (#A93A2F)

Reservado para encabezados y títulos, aporta peso visual sin recurrir al negro absoluto. Refuerza la identidad de marca con un matiz cálido, manteniendo la legibilidad y resaltando jerarquías de contenido.

Verde esmeralda (#4EE07A)

Utilizado en indicadores de “estado positivo” (completo, presente) para indicar éxito o aprobación. El verde, en su variante esmeralda, es fácilmente reconocible como señal de progreso.

Naranja ámbar (#DD6A22)

Asignado a indicadores de advertencia o estados intermedios (“moderado”, “próximo pago”). Su calidez capta la atención sin generar alarma, ayudando a los usuarios a priorizar acciones.

Rojo carmesí (#DD2222)

Destinado a estados críticos (“ausente”, “pendiente”). Su intensidad provoca una respuesta rápida del usuario ante incidencias o recordatorios urgentes, lo que es esencial para la seguridad y la gestión oportuna dentro del entorno escolar.

Violeta grisáceo (#5853EA)

Usado para indicadores de información secundaria. Al situarse entre lo cálido y lo frío, este violeta ofrece una opción distintiva para datos relevantes sin competir con los colores de alerta ni sobrecargar la interfaz.

Negro puro (#000000)

Color de texto principal y enlaces. Garantiza el máximo contraste sobre fondos claros, favorece la legibilidad continua y es legible para usuarios con visión reducida.

Rosado muy suave (#FDF3F3)

Aplicado en superficies neutrales como tarjetas y pop-ups. Este tono muy claro permite diferenciar bloques de contenido con sutileza, aportando sensación de limpieza, orden y amplitud al diseño.

Amarillo cálido claro (#F9D97B)

Elegido para botones de llamada a la acción. Su luminosidad evoca optimismo y estimula la interacción, sin resultar estridente, lo que lo hace ideal para destacar acciones clave (por ejemplo, “Descargar reporte semanal”).

3) Logo del sistema:**Justificación del logo utilizado para el jardín maternal “Luz de Luna”**

El logo seleccionado representa de manera simbólica y emocional la esencia del jardín maternal “Luz de Luna”. Se compone de una luna creciente sonriente acompañada de estrellas, junto al nombre del jardín en una tipografía clara y amigable.

Simbología y significado emocional:

La luna es un símbolo universal de protección, calma y ternura, atributos especialmente valorados en el cuidado infantil. Al tratarse de un jardín maternal, la luna transmite la sensación de un entorno seguro, acogedor y tranquilo, ideal para el desarrollo temprano.

Las estrellas refuerzan la idea de fantasía, sueños y creatividad, fomentando una atmósfera mágica que apela tanto a los niños como a los padres.

Tipografía y color:

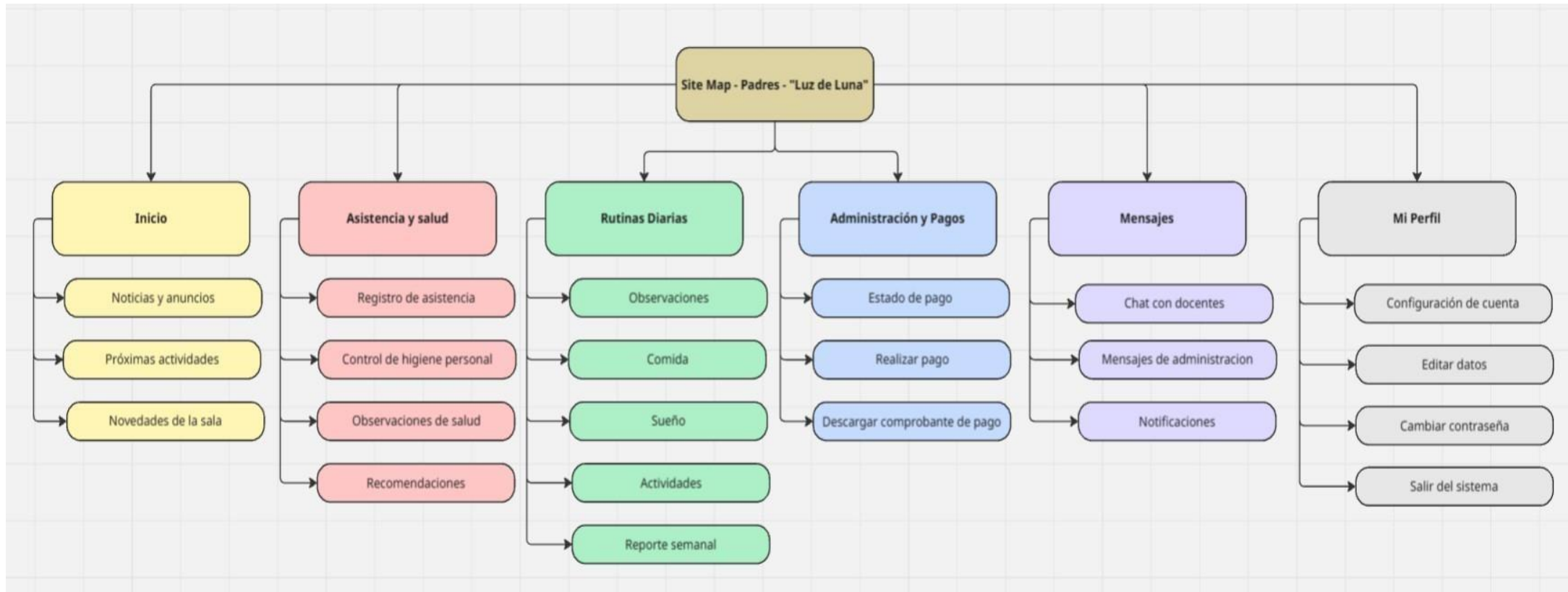
La tipografía redondeada refuerza la idea de cercanía y amabilidad. Además, el uso de un color azul oscuro genera contraste y profesionalismo, transmitiendo confianza, mientras que el amarillo claro aporta calidez y alegría, lo cual resulta ideal para un entorno infantil.

4) Wireframes de pantalla

Los cuatro bocetos de wireframes de pantalla fueron realizados con la aplicación **Figma** desde su sitio web. A continuación, dejamos el enlace de acceso:

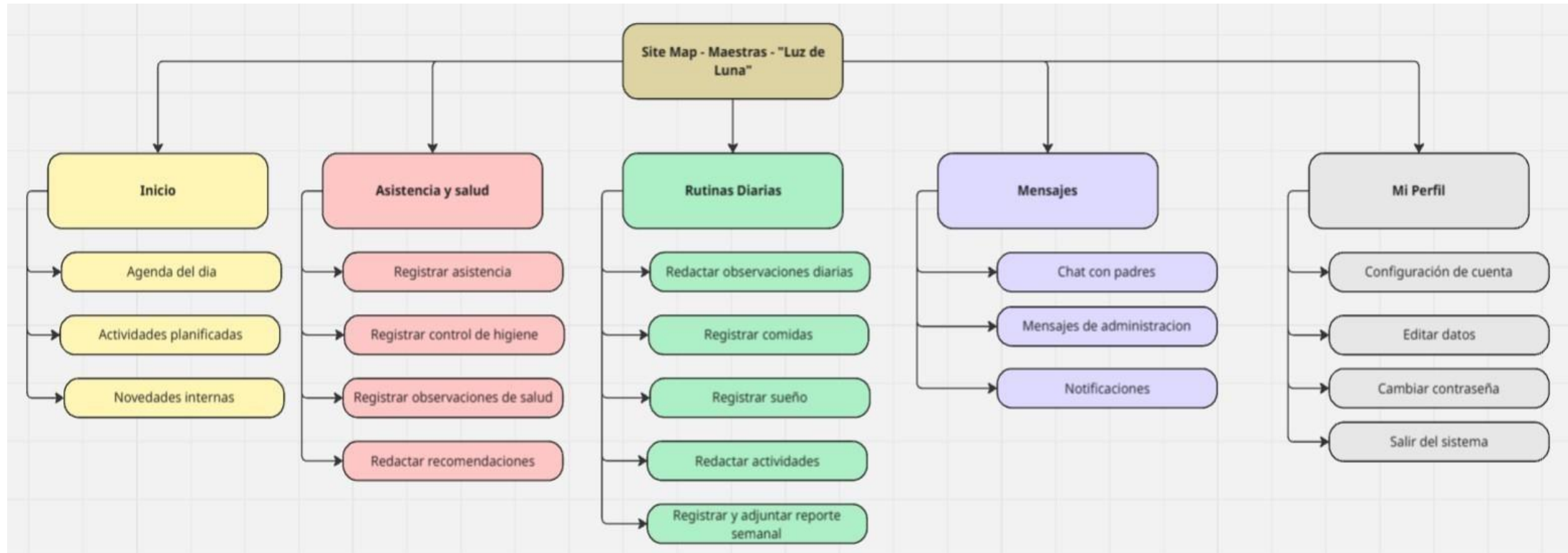
<https://www.figma.com/design/cpMI0WHWOi7MSfejPvHMR/Proyecto-Jardin-Maternal?node-id=71-2&t=QzLLj4zEEoC3tJzz-0>

5) Mapa de navegación (Site Map) – Padres



Mapa de navegación realizado con la plataforma Miro

5) Mapa de navegación (Site Map) – Maestras



Mapa de navegación realizado con la plataforma Miro

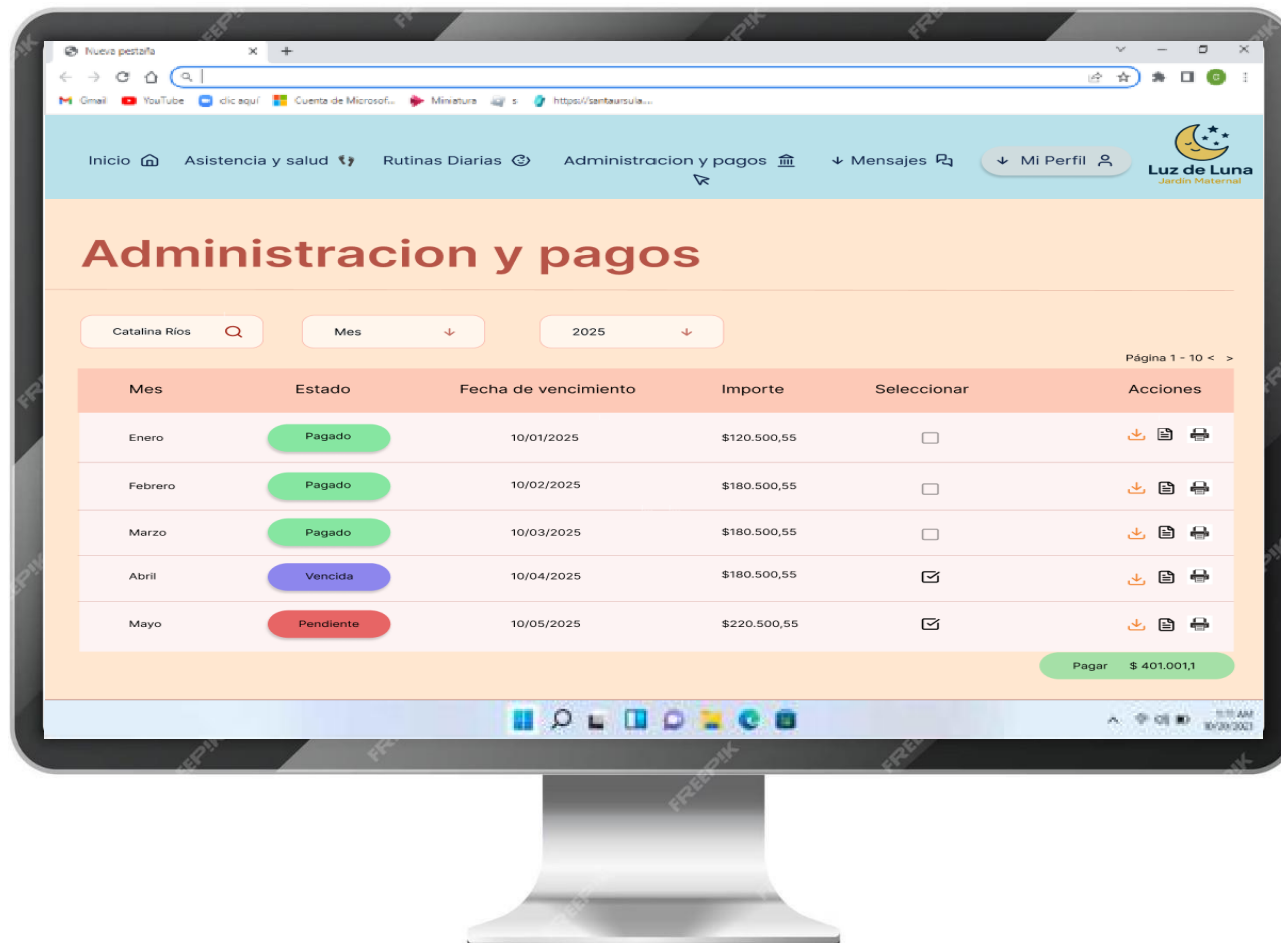
6) Diseño responsivo Celular:

Página web vista desde el celular (Android Médium; 700 x 840)



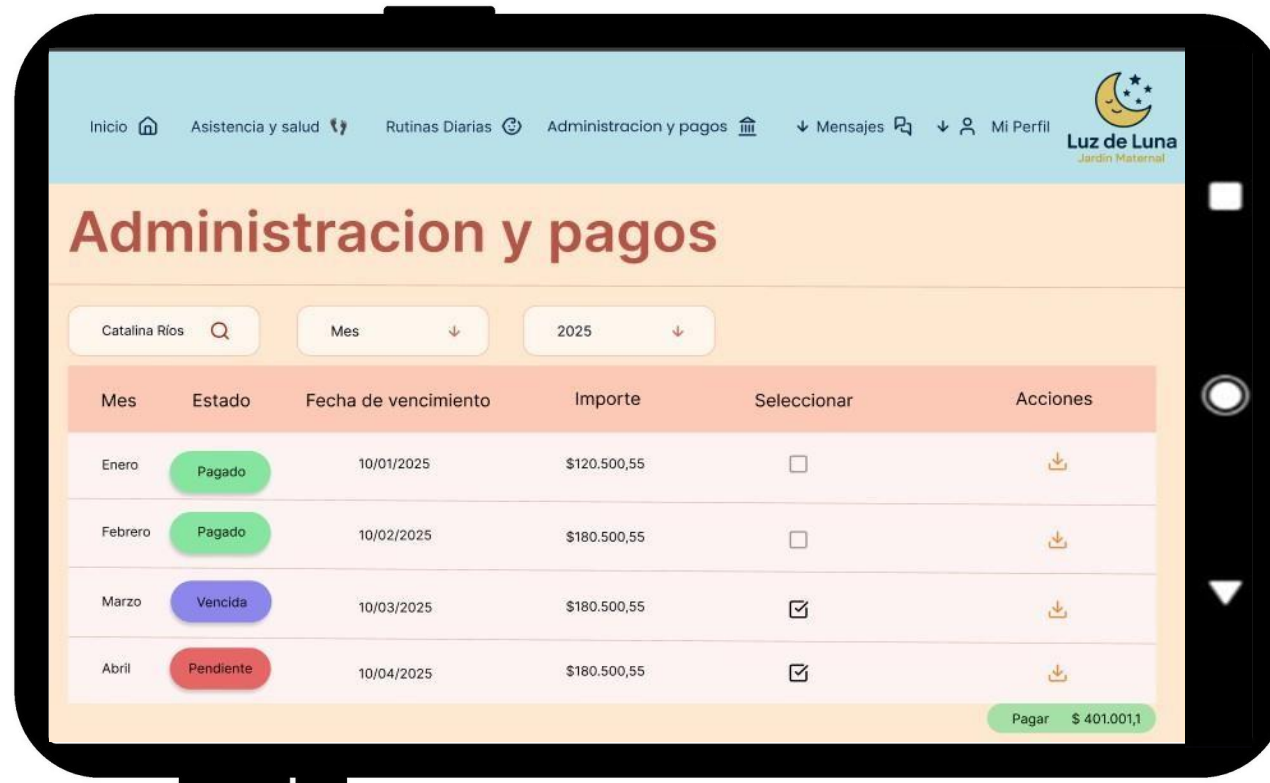
6. Diseño responsivo Desktop:

Página web vista desde una Desktop (1440 x 1024)



6) Diseño responsivo Tablet:

Página web vista desde una Tablet (1280 x 800)



Todos los modelos fueron diseñados con la aplicación Figma

<https://www.figma.com/design/cpMl0WHWOi7MSfejJpVhMR/Proyecto-Jardin-Maternal?node-id=51-32&t=Xp98harZuImE6nE9->

7) Tabla de guía tipográfica utilizada en los bocetos de la pagina

Familia tipográfica	Elemento	Tamaño	Color	Peso	Uso
Inter	Iconos de acceso (barra superior)	20	#10244A	Regular	Etiquetas de menú en la barra superior
Inter	Títulos de sección	64	#A93A2F	Bold	Encabezados principales
Inter	Texto general (párrafos y formularios)	15–20	#000000	Regular	Cuerpo de texto, y etiquetas
Mystery Quest	Título principal de bienvenida	120	#B20707	Regular	Encabezado grande en página de bienvenida
Mystery Quest	Subtítulo destacado de bienvenida	200	#072B3D	Regular	Lema central
Baloo 2	Títulos secundarios	40	#A93A2F	Regular	Títulos de tarjetas o secciones secundarias
Quicksand	Texto explicativo y subtítulos menores	27	#000000	Regular	Descripciones, subtítulos y leyendas informativas



**Universidad Metropolitana para
la Educación y el Trabajo**

Técnicas Avanzadas de la Programación

“Spring Nro. 4”

Alumno: Leandro Pasteur

Profesora: Antonieta Kuz

Fecha: 09/05/2025

Índice:

1) Diagrama de clases.....	Pag 3 - 6
2) Diagrama de arquitectura en capas	Pag 6 - 10
3) Flujo de datos del sistema.....	Pag 11 - 12

I. Diagrama de clases UML “Jardín Maternal”

El propósito de este diagrama es modelar las entidades clave del sistema web del jardín maternal “Luz de luna”, representando las clases, atributos, métodos y relaciones entre los actores del sistema.

El diagrama de Clases UML fue realizado con la aplicación Miro, a continuación, dejamos el link de acceso:

<https://miro.com/app/board/uXjVI8EG0ak=/>

El diagrama fue organizado en una disposición horizontal) para mejorar la legibilidad y evitar el cruce de flechas. Se definieron los siguientes grupos principales:

1) Clase Usuario:

Clase para autenticar y administrar usuarios del sistema.

Atributos	Métodos
-ID:int -nombreUsuario: string -contraseña:string -fechaRegistro:date -estadoCuenta:string	+iniciarSesion() +cerrarSesion() +recuperarContraseña() +actualizarContraseña()

Relaciones:

- **Herencia:** Es la clase base generalizada de Tutor, Maestra y Administrador, quienes heredan sus atributos y métodos básicos de autenticación.

2) Clase Tutor: (Padre/Madre) – Tipo: Entidad.

Atributos	Métodos
-ID:int -nombre:string -apellido:string -DNI:int -domicilio:string -telefono:int -mail:string	+consultarPagos() +consultarMensajes() +consultarRutina() +consultarAsistenciaYSalud() +descargarReporteSemanal() +pagarCuota() +editarPerfil()

Relaciones:

- **Herencia:** Hereda de Usuario.

- **Composición:** Compone a Niño (1..1) → Cada niño debe estar vinculado a un único tutor.

- **Agregación:** Agrega a Pago (0..*) → Un tutor puede tener múltiples pagos, que pueden subsistir sin él.

Spring Nro.4 – Técnicas avanzadas de la programación – Leandro Pasteur

- **Asociación:** Se asocia con Mensaje (0..*) → Puede enviar y recibir múltiples mensajes.

3) Clase Maestra: (Encargada de registrar actividades) – Tipo: Entidad.

Atributos	Métodos
-ID:int -nombre:string -apellido:string -DNI:int -domicilio:string -telefono:int -mail:string	+registrarAsistencia() +cargarRutinasDiarias() +registrarSalud() +verMensajes() +editarPerfil()

Relaciones:

-**Herencia:** Hereda de Usuario.

-Dependencia:

Depende de Inscripcion (0..1) → Puede estar involucrado en el proceso de inscripción.

Depende de RutinaDiaria y AsistenciaYSalud (1..1) → Es quien registra rutinas y controles diarios.

-**Asociación:** Se asocia con Mensaje (0..*).

4) Clase Administrador: Tipo: Entidad / Posible controlador.

Atributos	Métodos
-ID:int -nombre:string -mail:string -contraseña:string	+gestionarUsuarios() +administrarPagos() +administrarMensajes() +generarReportes()

Relaciones:

-**Herencia:** Hereda de Usuario.

-**Asociación:** Se asocia con Mensaje (0..*) → Participa en la comunicación institucional.

-**Dependencia:** Depende de Administrador (0..1) → Un administrador puede gestionar y supervisar los pagos realizados, sin ser parte directa del objeto

5) Clase Niño: Tipo: Entidad.

Atributos	Métodos
-ID:int -nombre:string -apellido:string -fechaNacimiento:date -grupoSala:string -tutorID:int	+obtenerHistorialSalud() +obtenerReporteDiario() +verHistorialAsistencias() +verPagosAsociados()

Relaciones:**-Composición:**

Compuesto por Tutor (1..1) → Su existencia depende de un adulto responsable.

Compone a Inscripcion, RutinaDiaria, AsistenciaYSalud (1..1 cada una) → Elementos del historial diario del niño.

6) Clase Inscripcion: Tipo: Entidad.

Atributos	Métodos
-IDInscripción:int -niñoID:int -fechaInscripción:int -estado:string -observaciones:string	+crearInscripción() +modificarInscripción() +eliminarInscripción()

Relaciones:

-Composición: Compone a Niño (1..1) → Cada inscripción está ligada a un niño específico.

-Dependencia: Depende de Maestra (0..1) → Puede ser iniciada por una maestra.

7) Clase Rutina Diaria: Tipo: Entidad.

Atributos	Métodos
-ID:int -fecha:date -niñoID:int -alimentación:string -sueño:string -higiene:string -actividades:string	+registrarRutina() +consultarRutina() +generarReporteSemanal()

Relaciones:

-Composición: Compone a Niño (1..1) → Cada rutina está registrada para un niño.

-Dependencia: Depende de Maestra (1..1) → Solo una maestra puede registrar la rutina.

8) Clase Asistencia y Salud: Tipo: Entidad.

Atributos	Métodos
-ID:int -niñoID:int -fecha:date -presente:string	+registrarAsistencia() +registrarSalud() +generarFichaSalud()

Relaciones:

-Composición: Compone a Niño (1..1) → La información se registra por niño.

-Dependencia: Depende de Maestra (1..1) → Es registrada por maestras a cargo.

9) Clase Pago: Tipo: Entidad.

Atributos	Métodos
-ID:int -tutorID:int -monto:double -fechaPago:date -fechaVencimiento:date	+generarPago() +consultarPago() +emitirComprobante() +pagar() +descargarRecibo()

Relaciones:

- **Agregación:** Agregado por Tutor (0..*) → Un tutor puede realizar múltiples pagos relacionados.

- **Dependencia:** Depende de Administrador (0..1) → Un administrador puede gestionar y supervisar los pagos realizados, sin ser parte directa del objeto

10) Clase Mensaje: Tipo: Entidad.

Atributos	Métodos
-ID:int -emisor:string -receptor:string -fecha:date -contenido:string	+enviarMensaje() +recibirMensaje() +marcarComoLeido()

Relaciones:

- **Asociación:** Se asocia con Tutor, Maestra, Administrador (0..*) → Todos los perfiles pueden enviarse mensajes entre sí.

II. El diagrama de Arquitectura fue realizado con la aplicacion Figma, a continuación, dejamos el link de acceso:

<https://www.figma.com/design/cpMI0WHWOi7MSfejPvHMR/Proyecto-Jardin-Maternal?node-id=0-1&p=f&t=b0dSUUfi9tzPL2IQ-0>

II. Descripción del Diagrama de Arquitectura en capas “Luz de Luna”

La estructura sigue un patrón de capas apiladas, donde cada capa ofrece servicios a la capa superior y consume servicios de la capa inferior, promoviendo una clara separación de preocupaciones y facilitando la modularidad y el mantenimiento.

El diagrama se estructura verticalmente, con las capas superiores representando niveles de abstracción más cercanos al usuario final y las capas inferiores gestionando aspectos de bajo nivel, como la persistencia y el acceso a los datos fundamentales del sistema.

Nivel de Interacción del Usuario:

En la parte superior del diagrama, se identifican los Usuarios o Actores del sistema. Estos corresponden a los distintos roles que interactuarán activamente con la aplicación para realizar tareas específicas dentro del jardín maternal: maestras, padres, directora y el personal administrativo.

Estos actores inician las interacciones con el sistema a través de un Navegador Web. El Navegador actúa como el cliente de la aplicación, ejecutando el código de la interfaz de usuario y sirviendo como el medio a través del cual el usuario accede y manipula el sistema. La comunicación entre el Usuario y la Capa de Presentación se realiza mediante el protocolo HTTP/S. Este protocolo, garantiza que las solicitudes y respuestas se transmitan de forma cifrada, protegiendo la confidencialidad e integridad de los datos intercambiados a través de la red.

Primera Capa: Capa de Presentación (Frontend)

Es la responsable primaria de la interfaz de usuario y de la gestión de la interacción directa con los actores. Su objetivo fundamental es capturar las entradas del usuario (acciones, datos) y enviar estas interacciones a la capa subsiguiente para su procesamiento.

Implementada utilizando React, una biblioteca de JavaScript para construir interfaces de usuario. React permite desarrollar componentes reutilizables, lo que acelera el desarrollo y mejora la consistencia de la UI.

Contiene las Interfaces de Usuario, que son las representaciones visuales de las distintas funcionalidades y flujos de trabajo del sistema. Esto incluye pantallas dedicadas para el Login y Registro de usuarios, el módulo de Inicio con información relevante según el rol, secciones específicas para la gestión de Asistencia y Salud de los niños, Rutinas Diarias, Administración y Pagos, el módulo de Mensajes (Chat) para la comunicación, y la gestión del Perfil de Usuario.

Comprende los Elementos de Interfaz, que son los controles interactivos básicos y componentes visuales. Esto abarca desde botones (para iniciar acciones), campos de texto (para la entrada de datos alfanuméricos), tablas (para la visualización estructurada de listados de datos como asistencias o pagos), iconos (para acciones rápidas o indicativos), barras de navegación y menús (para la orientación dentro del sistema), y otros componentes interactivos.

Incorpora el Diseño Adaptable. Este aspecto asegura que la interfaz se adapte dinámicamente a las características del dispositivo de visualización (tamaño de pantalla, orientación), en dispositivos (desktops, laptops, celulares, tablets)

Define el Estilo Visual, aplicando la paleta de colores, la guía tipográfica y otros elementos gráficos definidos en el proyecto.

La Capa de Presentación se comunica exclusivamente con la Capa de Aplicación. Esta comunicación se realiza mediante el envío de Solicitudes API (Application Programming Interface), estructuradas típicamente en formato JSON (JavaScript Object Notation). Estas solicitudes encapsulan las acciones del usuario (ej. registrar asistencia) o las peticiones de información (ej. obtener registros diarios de un niño). A continuación, recibe Respuestas API (también en JSON) que contienen los resultados de las operaciones ejecutadas en el backend o los datos requeridos para actualizar dinámicamente la interfaz de usuario.

Segunda Capa: Capa de Aplicación (Backend)

Situada directamente debajo de la Capa de Presentación, la Capa de Aplicación (Backend o Capa de Lógica de Negocio). Su función principal es procesar las solicitudes recibidas del Frontend, ejecutar la lógica de negocio, aplicar las reglas definidas por el jardín maternal y coordinar las interacciones con la capa de datos y otros componentes del sistema.

Compuesta por componentes internos que colaboran para procesar las solicitudes:

Controladores API: Reciben los datos enviados por el cliente, realizan validaciones preliminares (ej. formato de datos) y delegan la ejecución de la lógica de negocio compleja a los Servicios correspondientes.

Servicios (Lógica): Cada Servicio se especializa en una funcionalidad o área del negocio (Ej: Servicio de Asistencia gestiona todo lo relacionado con el registro de entradas y salidas; Servicio de Pagos maneja la generación, procesamiento y seguimiento de cuotas). Interactúan con los Repositorios para acceder a los datos y pueden coordinar acciones con otros Servicios o componentes transversales (como el envío de notificaciones).

Entidades (Modelos): Clases que representan los conceptos fundamentales del dominio del negocio del sistema (ej. Niño, Usuario, Pago). Actúan como la estructura de datos principal, encapsulando los atributos relevantes de cada concepto. Sirven como los objetos con los que interactúa la lógica de negocio (Servicios) y que son gestionados por los Repositorios para su persistencia en la Capa de Datos.

Repositorios (Spring Data JPA): Constituyen una capa de abstracción para el acceso a la Base de Datos. Su función es proporcionar una interfaz clara para realizar operaciones de persistencia (CRUD) sobre las Entidades. Utilizando Spring Data JPA, implementan la lógica necesaria para interactuar con la Capa de Datos (MySQL) mediante JPA/JDBC, desacoplando los Servicios de los detalles técnicos del acceso a datos de bajo nivel.

Manejo de Errores: Conjunto de mecanismos implementados para interceptar, procesar y responder de manera controlada a las excepciones y errores que pueden ocurrir durante la ejecución de la lógica en los Controladores y Servicios. Su función es proporcionar respuestas de error informativas y estandarizadas al Frontend, y registrar los incidentes para su posterior análisis y depuración.

La Capa de Aplicación recibe solicitudes JSON de la Capa de Presentación y, tras procesarlas, envía respuestas JSON de vuelta. Su interacción principal con la Capa de Datos se realiza a través de los Repositorios.

Tercera Capa: Capa de Datos

Responsable de la persistencia de toda la información del sistema. Su función principal es almacenar, organizar, gestionar y proporcionar acceso seguro y eficiente a los datos de manera duradera.

Utiliza MySQL como sistema de gestión de base de datos relacional que gestiona el almacenamiento y la recuperación de los datos.

Define el Esquema DB y las Tablas. El esquema de base de datos es la estructura lógica que organiza los datos. Las Tablas son las unidades fundamentales de almacenamiento en una base de datos relacional, donde se guardan los datos en filas y columnas. Cada tabla corresponde generalmente a una Entidad definida en la Capa de Aplicación (Ej: la tabla usuarios almacena los datos de los usuarios, la tabla registros_asistencia guarda los registros de asistencia). Las relaciones entre las tablas (mediante claves primarias y foráneas) aseguran la integridad y consistencia de los datos.

Spring Nro.4 – Técnicas avanzadas de la programación – Leandro Pasteur

Asegura que los datos guardados en la base de datos se mantengan disponibles de forma duradera a lo largo del tiempo, independientemente del estado de ejecución de la aplicación o de posibles reinicios del servidor.

La Capa de Datos es accedida exclusivamente por la Capa de Aplicación, específicamente por los Repositorios, para realizar operaciones de lectura y escritura de datos. Esta interacción se lleva a cabo utilizando protocolos y tecnologías de acceso a bases de datos como JPA / JDBC.

Componente Transversal: Seguridad

Representado lateralmente a la Capa de Aplicación, el componente de Seguridad, implementado típicamente con Spring Security en un entorno Spring Boot. Su rol es proteger el sistema y sus recursos de accesos no autorizados y garantizar que los usuarios solo puedan realizar las acciones para las que tienen permiso.

Se encarga de la Autenticación, el proceso de verificar la identidad de un usuario que intenta acceder al sistema (ej. validando credenciales como usuario y contraseña).

Contribuye activamente a la Protección de Datos Sensibles, asegurando que la información confidencial, especialmente la relacionada con los niños y sus registros, solo sea accesible por usuarios debidamente autorizados.

Interactúa con los Repositorios (en la Capa de Aplicación) para obtener la información de usuario, roles y permisos necesaria para realizar sus procesos de autenticación y autorización.

III. Flujo de datos

El flujo de datos fue realizado con la aplicación Miro. A continuación, dejamos el link de acceso:

https://miro.com/app/board/uXjVI3qWCa8=

Componentes del Diagrama

Usuario: Representa a cualquier persona que interactúa con el sistema (maestra, padre, directora, administrativo).

Se conecta al navegador para iniciar la interacción.

Navegador Web: Medio por el cual el usuario accede a la aplicación web.

Envía solicitudes HTTP/S al Frontend.

Frontend (Capa de Presentación): Interfaz hecha en React que captura la acción del usuario.

Genera una solicitud API (en JSON) hacia los controladores del backend.

Controladores API: Reciben solicitudes y las redirigen.

Seguridad (Spring Security): Intercepta las solicitudes y verifica si el usuario está autenticado y tiene permiso.

Servicios: Contienen la lógica de negocio (por ejemplo, registrar asistencia, enviar mensaje, etc.).

Repositorios: Se encargan de acceder a la base de datos mediante JPA o JDBC.

Base de Datos MySQL (Capa de Datos): Almacena la información persistente.

Recibe consultas y operaciones de guardado desde los repositorios.

Responde con datos (entidades) que se devuelven hacia atrás por el mismo flujo.

Uniones / Flujo de Datos

Estas son las conexiones clave:

Usuario → Navegador: El usuario accede al sistema.

Navegador → Frontend: El navegador envía una solicitud al frontend.

Frontend → Controladores API: Envía la solicitud de acción en formato JSON.

Controladores API → Seguridad: Verifica si el usuario puede realizar la acción.

Seguridad → Controladores API: Devuelve si está autorizado.

Controladores API → Servicios: Si todo está bien, se llama a la lógica de negocio.

Servicios → Repositorios: Si se necesita guardar o leer datos.

Repositorios → BD: Se ejecutan consultas o escrituras en la base de datos.

BD → Repositorios → Servicios: Los datos fluyen de regreso.

Servicios → Controladores API: Se genera una respuesta.

Controladores API → Frontend: Se envía respuesta JSON.

Frontend → Navegador: Se actualiza la interfaz del usuario.



**Universidad Metropolitana para
la Educación y el Trabajo**

Técnicas Avanzadas de la Programación

“Spring Nro. 5”

Alumno: Leandro Pasteur

Profesora: Antonieta Kuz

Fecha: 24/05/2025

Índice:

1) Diagrama Entidad-Relación.....	Pag 3
1) Diagrama Entidad-Relación: Descripción	Pag 4 - 11
2) Store procedures: (Base de datos “Jardín Maternal)	Pag 11 - 16
3) Vistas del sistema (Base de datos “Jardín maternal”)	Pag 17 - 18



1) Introducción al Diagrama de Base de Datos

Este informe tiene como objetivo principal proporcionar una descripción detallada del esquema de una base de datos relacional, visualizado a través de un diagrama Entidad-Relación (ER).

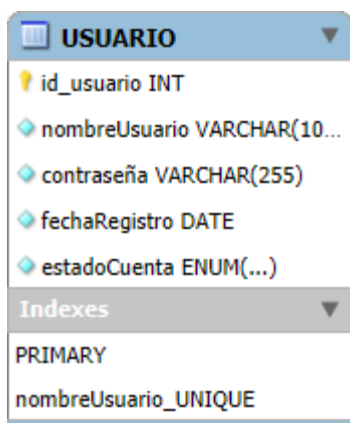
Este sistema busca centralizar y organizar la información relacionada con niños, sus tutores, el personal educativo (maestras y administradores), así como registros cruciales de asistencia, salud, rutinas diarias y aspectos financieros como los pagos.

El diagrama fue realizado con el sistema de gestión MySQL Workbench.

A continuación, la descripción detallada de cada tabla y sus relaciones:

A) Tabla USUARIO:

Propósito: Almacena la información general de todos los usuarios del sistema.



Campos:

id_usuario: Clave primaria, identificador único del usuario.

nombreUsuario: Nombre de usuario (VARCHAR).

contrasena: Contraseña (VARCHAR).

fechaRegistro: Fecha de registro (DATE).

estadoCuenta: (ENUM:'Activo', 'Inactivo', 'Bloqueado').

Índices: id_usuario (PRIMARY KEY), nombreUsuario (UNIQUE).

Relaciones:

1:1 con ADMINISTRADOR, TUTOR, NIÑO, MAESTRA: Cada uno de estos son tipos específicos de USUARIO (es decir, un administrador es un usuario, un tutor es un usuario, etc.).

1:N (en dos roles) con MENSAJE: Un usuario puede ser el emisor de muchos mensajes y el receptor de muchos mensajes.

B) Tabla ADMINISTRADOR:

Propósito: Almacena información específica de los usuarios con rol de administrador.

ADMINISTRADOR	
id_administrador	INT
id_usuario	INT
nombre	VARCHAR(100)
mail	VARCHAR(150)
contraseña	VARCHAR(255)

Indexes	
PRIMARY	
mail_UNIQUE	
fk_admin_usuario_idx	

Atributos:

id_administrador (INT): Clave primaria, identificador único del administrador.

id_usuario (INT): Clave foránea que referencia USUARIO.

nombre (VARCHAR): Nombre del administrador.

Índices: id_administrador (PRIMARY), id_usuario (FOREIGN), mail (UNIQUE)

Relación: 1:1 con USUARIO: Un usuario puede ser un administrador, y un administrador es un usuario.

C) Tabla TUTOR:

Propósito: Almacena la información detallada de los tutores.

TUTOR	
id_tutor	INT
id_usuario	INT
nombre	VARCHAR(100)
apellido	VARCHAR(100)
DNI	INT
domicilio	VARCHAR(150)
telefono	VARCHAR(20)
mail	VARCHAR(150)

Indexes	
PRIMARY	
DNI_UNIQUE	
mail_UNIQUE	
fk_tutor_usuario_idx	

Atributos:

id_tutor (INT): Clave primaria, identificador único del tutor.

id_usuario (INT): Clave foránea que referencia USUARIO.

nombre (VARCHAR): Nombre del tutor.

apellido (VARCHAR): Apellido del tutor.

DNI (INT): Documento Nacional de Identidad del tutor.

domicilio (VARCHAR): Dirección del tutor.

telefono (VARCHAR): Número de teléfono.

mail (VARCHAR): Correo electrónico.

Índices: id_tutor (PRIMARY), id_usuario (FOREIGN), DNI (UNIQUE), mail (UNIQUE).

Relaciones:

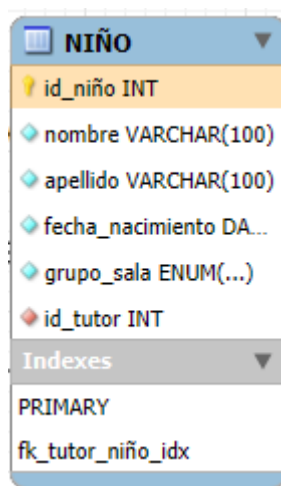
1:1 con USUARIO: Un usuario puede ser un tutor, y un tutor es un usuario.

1:N con PAGO: Un tutor puede realizar múltiples pagos registrados en la tabla PAGO.

N:M con NIÑO: mediante la tabla intermedia NIÑO_TUTOR, permite que un niño tenga más de un tutor (madre, padre, tutor legal), y que un tutor esté asociado a más de un niño.

D) Tabla NIÑO:

Propósito: Almacena la información de los niños inscriptos en el sistema.

**Atributos:**

id_niño (INT): Clave primaria, identificador único del niño.

nombre (VARCHAR): Nombre del niño.

apellido (VARCHAR): Apellido del niño.

fecha_nacimiento (DATE): Fecha de nacimiento del niño.

grupo_sala: (ENUM ('abejitas', 'sapitos', 'vaquitas'))

id_tutor (INT): Clave foránea que referencia TUTOR,

Índices: id_niño (PRIMARY), id_tutor (FOREIGN).

Relaciones:

N:M con TUTOR: A través de NIÑO_TUTOR, permite más de un tutor por niño.

1:N con RUTINA_DIARIA: Se registran varias rutinas por día para el niño.

1:N con INSCRIPCIÓN: Permite múltiples inscripciones a lo largo del tiempo.

1:N con ASISTENCIA_Y_SALUD: Registra varios controles de salud o asistencias diarias.

E) Tabla NIÑO_TUTOR:

Propósito: Representa la relación N:M (muchos a muchos) entre NIÑO y TUTOR, permitiendo que un niño tenga múltiples tutores y un tutor tenga múltiples niños.

NIÑO_TUTOR	
id_niño	INT
id_tutor	INT
relacion	VARCHAR(50)
Indexes	
PRIMARY	
niño_tutor_ibfk_2	
niño_tutor_ibfk_1	

Atributos:

id_nino (INT): Clave foránea que referencia NIÑO.

id_tutor (INT): Clave foránea que referencia TUTOR.

relación (VARCHAR): el tipo de relación entre niño y tutor.

Índices: id_nino (FOREIGN), id_tutor (FOREIGN).

F) Tabla PAGO:

Propósito: Registra los pagos realizados en el sistema.

PAGO	
id_pago	INT
id_tutor	INT
monto	DECIMAL(10,2)
fecha_pago	DATE
fecha_vencimiento	DATE
estado	ENUM(...)
Indexes	
PRIMARY	
fk_pago_tutor_idx	

Atributos:

id_pago (INT): Clave primaria.

id_tutor (INT): Clave foránea que referencia TUTOR, indicando quién realizó el pago.

monto (DECIMAL): Cantidad del pago.

fecha_pago (DATE): Fecha en que se realizó el pago.

fecha_vencimiento (DATE): Fecha de vencimiento del pago.

estado: (ENUM: 'pendiente', 'pagado', 'vencido')

Índices: id_pago (PRIMARY), id_tutor (FOREIGN)

Relación: N:1 con TUTOR (un tutor puede realizar muchos pagos).

G) Tabla ASISTENCIA Y SALUD:

Propósito: Registra la asistencia y aspectos de salud de los niños.

ASISTENCIA_Y_SALUD	
id_asistencia_y_salud	INT
id_niño	INT
fecha	DATE
presente	ENUM(...)
observacion_salud	TEXT
Indexes	
PRIMARY	
fk_asistencia_niño_idx	

Atributos:

id_asistencia_y_salud (INT): Clave primaria.

id_niño (INT): Clave foránea que referencia NIÑO.

fecha (DATE): Fecha del registro de asistencia/salud.

presente (ENUM:'si', 'no', 'justificado')

observación_salud (TEXT): observación relevante

Índices: id_asistencia_y_salud (PRIMARY), id_niño (FOREIGN)

Relación: N:1 con NIÑO (un niño puede tener muchos registros de asistencia/salud).

H) Tabla INSCRIPCION:

Propósito: Registra las inscripciones de los niños.

INSCRIPCION	
id_inscripcion	INT
id_niño	INT
fecha_inscripcion	DATE
estado	ENUM(...)
observacion	TEXT
Id_grupo_sala	INT
Indexes	
PRIMARY	
fk_inscripcion_niño_idx	
fk_inscripcion_grupo_sala_i...	

Atributos:

id_inscripcion (INT): Clave primaria.

id_niño (INT): Clave foránea que referencia NIÑO.

fecha_inscripcion (DATE): Fecha de la inscripción.

estado (ENUM: 'activa', 'completada', 'pendiente').

observacion (TEXT): Notas adicionales sobre la inscripción.

id_grupo_sala(INT): Clave foránea que referencia a GRUPO_SALA.

Índices: id_inscripcion (PRIMARY), id_niño (FOREIGN).

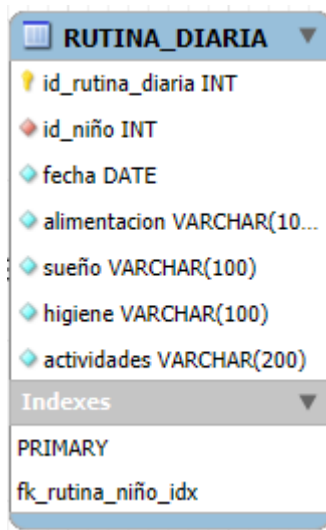
Relaciones:

N:1 con NIÑO (un niño puede tener varias inscripciones).

N:1 con GRUPO_SALA (una sala puede tener muchas inscripciones, pero cada inscripcion pertenece a una única sala).

I) RUTINA DIARIA (RUTINA DIARIA)

Propósito: Registra las actividades diarias de los niños.

**Atributos:**

id_rutina_diaria (INT): Clave primaria.

id_niño (INT): Clave foránea que referencia NIÑO.

fecha (DATE): Fecha de la rutina.

alimentación (VARCHAR): Detalle de la alimentación.

sueño (VARCHAR): Detalle del sueño.

higiene (VARCHAR): Detalle de la higiene.

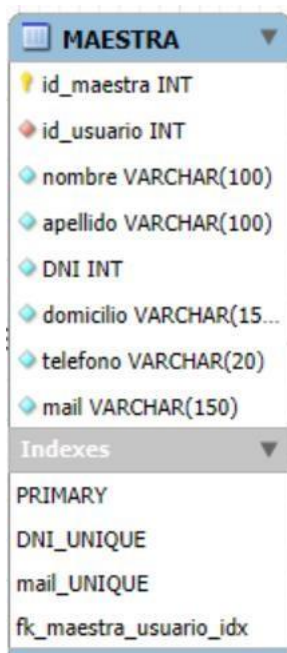
actividades (VARCHAR): Descripción de las actividades

Índices: id_rutina_diaria (PRIMARY), id_niño (FOREIGN).

Relación: N:1 con NIÑO (un niño puede tener muchos registros de rutina diaria).

J) Tabla MAESTRA

Propósito: Almacena información de las maestras del jardín.

**Atributos:**

id_maestra (INT): Clave primaria.

id_usuario (INT): Clave foránea que referencia USUARIO.

nombre (VARCHAR): Nombre de la maestra.

apellido (VARCHAR): Apellido de la maestra.

DNI (INT): Documento Nacional de Identidad.

domicilio (VARCHAR): Dirección.

telefono (VARCHAR): Número de teléfono.

mail (VARCHAR): Correo electrónico.

Índices: id_maestra (PRIMARY), id_usuario (FOREIGN), DNI (UNIQUE), mail (UNIQUE).

Relaciones:

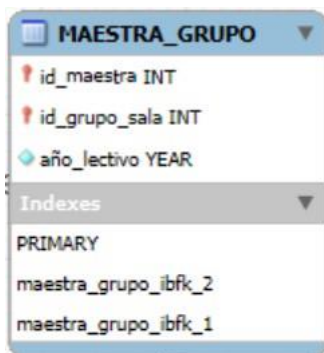
1:1 con USUARIO (una usuaria puede ser maestra, y una maestra es una usuaria).

1:N con MAESTRA_GRUPO: Una maestra puede estar asignada a varios grupos.

N:1 con NIÑO (un grupo puede tener muchos niños).

K) Tabla MAESTRA GRUPO

Propósito: Modela la relación N:M entre MAESTRA y GRUPO_SALA, permitiendo que una maestra esté a cargo de varios grupos y un grupo tenga varias maestras.

**Atributos:**

Id_grupo_sala (INT): Clave foránea que referencia GRUPO_SALA.

id_maestra (INT): Clave foránea que referencia MAESTRA.

Año_lectivo (YEAR): Año lectivo que se cursa.

Índices: id_grupo_sala (FOREIGN), _id_maestra (FOREIGN).

L) Tabla GRUPO SALA:

Propósito: Organiza a las maestras en grupos o salas.

**Atributos:**

id_grupo_sala (INT): Clave primaria.

sala: ENUM ('abejitas', 'sapitos', 'vaquitas')

Índices: id_grupo_sala (PRIMARY).

Relaciones:

1:N con MAESTRA_GRUPO (un grupo puede tener varias maestras).

N:1 con INSCRIPCION (una sala puede tener muchas inscripciones, pero cada inscripcion pertenece a una única sala).

M) Tabla MENSAJE:

Propósito: Almacena los mensajes enviados dentro del sistema.

**Atributos:**

id_mensaje (INT): Clave primaria.

fecha (DATETIME): Fecha y hora del mensaje.

id_receptor (INT): Clave foránea que referencia USUARIO (el destinatario del mensaje).

id_emisor (INT): Clave foránea que referencia USUARIO (el remitente del mensaje).

contenido (TEXT): Contenido del mensaje.

Índices: id_mensaje (PRIMARY), id_usuario (FOREIGN como emisor y receptor)

Relación: N:1 con USUARIO (tanto para el emisor como para el receptor, un usuario puede enviar/recibir muchos mensajes).

2) Store procedures: Creación de procedimiento (Base de datos “Jardin Maternal”)

A continuación, vamos a describir la implementación de procedimientos almacenados en nuestra base de datos. Decidimos utilizar el sistema de gestión de datos llamado MySQL.

Lo utilizaremos para gestionar información relacionada con niños, incluyendo sus rutinas diarias, asistencias, estado de salud, y el registro de pagos mensuales de sus tutores.

Describiremos como se detallan tres procedimientos almacenados específicos: sp_registrar_rutina_diaria, sp_registrar_asistencia_salud, y sp_registrar_pago. Además de la creación de estos procedimientos, se dará a conocer cómo insertar datos en las tablas correspondientes (RUTINA_DIARIA, ASISTENCIA_Y_SALUD, y PAGO) y cómo realizar consultas combinando estas tablas para obtener una visión más completa de la información.

A) Procedimiento que registra una rutina diaria por niño

Este procedimiento almacenado tiene como objetivo registrar la rutina diaria de un niño. Cuando se llama a este procedimiento, se le deben proporcionar los datos (ID del niño, fecha, alimentación, sueño, higiene y actividades) y el procedimiento se encargará de insertar esta información como un nuevo registro en la tabla.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_registrar_rutina_diaria`(
    IN p_id_niño INT,
    IN p_fecha DATE,
    IN p_alimentacion VARCHAR(100),
    IN p_sueño VARCHAR(100),
    IN p_higiene VARCHAR(100),
    IN p_actividades TEXT(200)
)
BEGIN
    INSERT INTO `RUTINA_DIARIA` (
        `id_niño`,
        `fecha`,
        `alimentacion`,
        `sueño`,
        `higiene`,
        `actividades`
    ) VALUES (
        p_id_niño,
        p_fecha,
        p_alimentacion,
        p_sueño,
        p_higiene,
        p_actividades
    );
END
```

A continuación, se deben insertar en la tabla de RUTINA_DIARIA los datos que deseamos guardar dentro de ella:

- `INSERT INTO RUTINA_DIARIA (id_niño, fecha, alimentacion, sueño, higiene, actividades) VALUES (1,'2025-05-01','Leche y tostadas','Durmió 1 hora','Cambio de pañal','Juegos con bloques'), (2,'2025-05-02','Yogur con frutas','Durmió 1h 30m','Lavado de manos','Pintura y dibujo'), (3,'2025-05-03','Papilla de verduras','Durmió 2 horas','Baño breve','Cuentos y canciones'),`

(En la imagen se muestran solo 3 rutinas insertadas, pero decidimos insertar 30)

Hemos decidido combinar dos tablas (RUTINA_DIARIA con NIÑO) para que la consulta sea más completa:

```
-- =====
-- Une tabla NIÑO con RUTINA_DIARIA
-- =====
SELECT
  RD.id_rutina_diaria AS ID_Rutina_Diaria,
  RD.id_niño AS ID_Niño,
  N.nombre AS nombre_niño,
  N.apellido AS apellido_niño,
  N.grupo_sala AS sala,
  RD.fecha,
  RD.alimentacion,
  RD.sueño,
  RD.higiene,
  RD.actividades
FROM
  RUTINA_DIARIA AS RD
INNER JOIN
  NIÑO AS N ON RD.id_niño = N.id_niño
```

Finalmente, al ejecutar esta sentencia podemos visualizar la consulta que realizamos:

	ID_Rutina_Diaria	ID_Niño	nombre_niño	apellido_niño	sala	fecha	alimentacion	sueño	higiene	actividades
▶	1	1	Agustin	Alvarez	abejitas	2025-05-01	Leche y tostadas	Durmió 1 hora	Cambio de pañal	Juegos con bloques
	2	2	Benjamin	Burgo	sapitos	2025-05-02	Yogur con frutas	Durmió 1h 30m	Lavado de manos	Pintura y dibujo
	3	3	Catalina	Cavallo	vaquitas	2025-05-03	Papilla de verduras	Durmió 2 horas	Baño breve	Cuentos y canciones
	4	4	Delfina	Dominguez	abejitas	2025-05-04	Puré de papas	Durmió 45 minutos	Cambio de ropa	Juego libre
	5	5	Emilio	Estevez	vaquitas	2025-05-05	Cereal con leche	No durmió	Lavado de manos	Rompecabezas
	6	6	Franco	Funes	sapitos	2025-05-06	Leche y tostadas	Durmió 1 hora	Cambio de pañal	Cuentos y canciones
	7	7	Gala	Garcia	abejitas	2025-05-07	Yogur con frutas	Durmió 1h 30m	Baño breve	Juegos con bloques
	8	8	Ignacio	Herrera	vaquitas	2025-05-08	Papilla de verduras	Durmió 2 horas	Lavado de manos	Pintura y dibujo
	9	9	Juana	Ibarra	sapitos	2025-05-09	Puré de papas	Durmió 45 minutos	Cambio de ropa	Cuentos y canciones
	10	10	Kevin	Juarez	abejitas	2025-05-10	Cereal con leche	No durmió	Cambio de pañal	Rompecabezas
	11	11	Lara	Klein	vaquitas	2025-05-11	Leche y tostadas	Durmió 2 horas	Lavado de manos	Juego libre
	12	12	Mateo	Lopez	sapitos	2025-05-12	Yogur con frutas	Durmió 1 hora	Baño breve	Pintura y dibujo
	13	13	Nadia	Mendoza	abejitas	2025-05-13	Papilla de verduras	No durmió	Cambio de ropa	Cuentos y canciones
	14	14	Oscar	Nieto	vaquitas	2025-05-14	Puré de papas	Durmió 1h 30m	Lavado de manos	Rompecabezas
	15	15	Pedro	Ortega	sapitos	2025-05-15	Cereal con leche	Durmió 45 minutos	Cambio de pañal	Juegos con bloques

B) Procedimiento que registra las asistencias y comentarios sobre la salud de los niños:

Este procedimiento almacenado tiene como objetivo registrar la asistencia y observaciones sobre la de un niño. Cuando se llama a este procedimiento, se le deben proporcionar los datos (ID del niño, fecha, presente, observación de salud) y el procedimiento se encargará de insertar esta información como un nuevo registro en la tabla


```
CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_registrar_asistencia_salud`(
  IN p_id_niño INT,
  IN p_fecha DATE,
  IN p_presente ENUM('Sí', 'No', 'Justificado'),
  IN p_observacion_salud TEXT(300)
)
BEGIN
  INSERT INTO `ASISTENCIA_Y_SALUD` (
    `id_niño`,
    `fecha`,
    `presente`,
    `observacion_salud`
  ) VALUES (
    p_id_niño,
    p_fecha,
    p_presente,
    p_observacion_salud
  );
END|
```

A continuación, se deben insertar en la tabla de ASISTENCIA_Y_SALUD los datos que deseamos guardar dentro de ella:

```
• INSERT INTO ASISTENCIA_Y_SALUD (id_niño, fecha, presente, observacion_salud) VALUES
(1, '2025-05-20', 'Sí', 'Sin novedad, muy participativo hoy. '),
(2, '2025-05-20', 'Sí', 'Leve rasguño en el brazo, curado. '),
(3, '2025-05-20', 'No', 'Ausente por enfermedad, notificado. ');
```

(En la imagen se muestran solo 3 rutinas insertadas, pero decidimos insertar 30)

Hemos decidido combinar dos tablas (ASISTENCIA_Y_SALUD con NIÑO) para que la consulta sea más completa:

```
-- =====
-- muestra todas las ASISTENCIA_Y_SALUD por niño
-- =====
SELECT
  A.id_asistencia_y_salud AS ID_Asistencia_Salud,
  A.id_niño AS ID_Niño,
  N.nombre AS Nombre_Niño,
  N.apellido AS Apellido_Niño,
  N.grupo_sala AS Sala,
  A.fecha,
  A.presente AS Estado_Asistencia,
  A.observacion_salud AS Observaciones_Salud
FROM
  ASISTENCIA_Y_SALUD AS A
INNER JOIN
  NIÑO AS N ON A.id_niño = N.id_niño
```

Finalmente, al ejecutar esta sentencia podemos visualizar la consulta que realizamos:

	ID_Asistencia_Salud	ID_Niño	Nombre_Niño	Apellido_Niño	Sala	fecha	Estado_Asistencia	Observaciones_Salud
►	1	1	Agustin	Alvarez	abejitas	2025-05-20	Sí	Sin novedad, muy participativo hoy.
	2	2	Benjamin	Burgo	sapitos	2025-05-20	Sí	Leve rasguño en el brazo, curado.
	3	3	Catalina	Cavallo	vaquitas	2025-05-20	No	Ausente por enfermedad, notificado.
	4	4	Delfina	Dominguez	abejitas	2025-05-20	Sí	Asistencia normal, buen apetito.
	5	5	Emilio	Estevez	vaquitas	2025-05-20	Sí	Se quejó de dolor de muelas leve, se avisó a los...
	6	6	Franco	Funes	sapitos	2025-05-20	Sí	Activo y contento durante todo el día.
	7	7	Gala	Garcia	abejitas	2025-05-20	Sí	Sin observaciones, se durmió temprano en la sie...
	8	8	Ignacio	Herrera	vaquitas	2025-05-20	Justificado	Cita con el pediatra.
	9	9	Juana	Ibarra	sapitos	2025-05-20	Sí	Un poco decaído al principio, mejoró luego.
	10	10	Kevin	Juarez	abejitas	2025-05-20	Sí	Puntual, sin novedad médica.
	11	11	Lara	Klein	vaquitas	2025-05-20	Sí	Presente y muy juguetón.
	12	12	Mateo	Lopez	sapitos	2025-05-20	Sí	Tuvo un pequeño tropiezo, sin consecuencias.
	13	13	Nadia	Mendoza	abejitas	2025-05-20	Sí	Sin quejas de salud, disfrutó de las actividades.
	14	14	Oscar	Nieto	vaquitas	2025-05-20	No	Ausente sin justificación.

C) Procedimiento que registra el estado de un pago mensual perteneciente a un tutor:

Este procedimiento almacenado tiene como objetivo registrar el estado de un pago mensual que pertenece a un tutor. Cuando se llama a este procedimiento, se le deben proporcionar los datos (ID del tutor, monto, fecha de pago, fecha de vencimiento, estado de pago) y el procedimiento se encargará de insertar esta información como un nuevo registro en la tabla

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_registrar_pago`(
    IN p_id_tutor INT,
    IN p_monto DECIMAL(10,2),
    IN p_fecha_pago DATE,
    IN p_fecha_vencimiento DATE,
    IN p_estado ENUM('Pendiente', 'Pagado', 'Vencido', 'Cancelado')
)
BEGIN
    INSERT INTO `PAGO` (
        `id_tutor`,
        `monto`,
        `fecha_pago`,
        `fecha_vencimiento`,
        `estado`
    ) VALUES (
        p_id_tutor,
        p_monto,
        p_fecha_pago,
        p_fecha_vencimiento,
        p_estado
    );
END;
```

A continuación, se deben insertar en la tabla de PAGO los datos que deseamos guardar dentro de ella:

```
• INSERT INTO PAGO (id_tutor, monto, fecha_pago, fecha_vencimiento, estado) VALUES
  (1, 150000.00, '2025-05-01', '2025-05-31', 'Pagado'),
  (2, 150000.00, '2025-05-31', '2025-05-31', 'Pendiente'),
  (3, 150000.00, '2025-05-03', '2025-05-31', 'Pagado'),
```

Hemos decidido combinar dos tablas (ASISTENCIA_Y_SALUD con NIÑO) para que la consulta sea más completa:

```
-- =====
-- Une tabla PAGO con TUTOR
-- =====
SELECT
  P.id_pago,
  P.monto,
  P.fecha_pago,
  P.fecha_vencimiento,
  P.estado,
  T.id_tutor,
  T.nombre AS nombre_tutor,
  T.apellido AS apellido_tutor,
  T.DNI,
  T.domicilio,
  T.telefono,
  T.mail
FROM
  PAGO AS P
JOIN
  TUTOR AS T ON P.id_tutor = T.id_tutor;
```

Finalmente, al ejecutar esta sentencia podemos visualizar la consulta que realizamos:

	id_pago	monto	fecha_pago	fecha_vencimiento	estado	id_tutor	nombre_tutor	apellido_tutor	DNI	domicilio	telefono	mail
	1	150000.00	2025-05-01	2025-05-31	Pagado	1	Laura	Perez	30011223	Av. Corrientes 1500, Palermo	1123412345	laura.perez@example.co
	2	150000.00	2025-05-31	2025-05-31	Pendiente	2	Juan	Gomez	20022334	Av. Santa Fe 500, Recoleta	1123123456	juan.gomez@example.co
	3	150000.00	2025-05-03	2025-05-31	Pagado	3	Maria	Rodriguez	25033445	Calle Perú 200, San Telmo	1133256789	maria.rodriguez@exampl
	4	150000.00	2025-05-04	2025-05-31	Pagado	4	Carlos	Martinez	31044556	Calle Florida 800, Belgrano	1143678901	carlos.martinez@exampl
	5	150000.00	2025-05-31	2025-05-31	Pendiente	5	Silvia	Garcia	27055667	Calle Defensa 1200, La Boca	1193456712	silvia.garcia@example.co
▶	6	150000.00	2025-05-31	2025-05-31	Vencida	6	Diego	Lopez	29066778	Av. Corrientes 1800, Palermo	1163456123	diego.lopez@example.co
	7	150000.00	2025-05-07	2025-05-31	Pagado	7	Veronica	Diaz	32077889	Av. Santa Fe 700, Recoleta	1173456234	veronica.diaz@example.co
	8	150000.00	2025-05-31	2025-05-31	Pendiente	8	Federico	Silva	33088990	Calle Perú 300, San Telmo	1183456345	federico.silva@example.co
	9	150000.00	2025-05-09	2025-05-31	Pagado	9	Camila	Fernandez	34099001	Calle Florida 900, Belgrano	1193456456	camila.fernandez@exampl
	10	150000.00	2025-05-31	2025-05-31	Pendiente	10	Leo	Suarez	35010112	Calle Defensa 1300, La Boca	1123456567	leo.suarez@example.co
	11	150000.00	2025-05-11	2025-05-31	Pagado	11	Sofia	Torres	36021223	Av. Corrientes 1700, Palermo	1123456678	sofia.torres@example.co
	12	150000.00	2025-05-12	2025-05-31	Pagado	12	Lucas	Ramirez	37032334	Av. Santa Fe 600, Recoleta	1123456789	lucas.ramirez@example.co
	13	150000.00	2025-05-31	2025-05-31	Pendiente	13	Valentina	Reyes	38043445	Calle Perú 400, San Telmo	1123456890	valentina.reyes@exampl
	14	150000.00	2025-05-14	2025-05-31	Pagado	14	Martin	Arias	39054556	Calle Florida 100, Belgrano	1123456901	martin.arias@example.co

3) Vistas del sistema (Base de datos “Jardin maternal”)

A continuación, exploraremos la creación de vistas del sistema para diversas tablas de nuestra base de datos. Estas vistas nos brindarán la capacidad de visualizar consultas complejas sobre los datos almacenados, simplificando el acceso y la comprensión de la información relevante.

A) Vista: vista_asistencia_salud_detallada

- Este código crea una vista que facilita la consulta de un informe combinado con los datos de asistencia y salud de cada niño, incluyendo su nombre completo y sala al que pertenece.

```
CREATE VIEW vista_asistencia_salud_detallada AS
SELECT
    A.id_asistencia_y_salud,
    A.id_niño,
    N.nombre AS nombre_niño,
    N.apellido AS apellido_niño,
    N.grupo_sala AS sala_niño,
    A.fecha,
    A.presente AS estado_asistencia,
    A.observacion_salud
FROM
    ASISTENCIA_Y_SALUD AS A
INNER JOIN
    NIÑO AS N ON A.id_niño = N.id_niño;
```

- Ejecutamos el siguiente comando para ver como quedo la vista

```
SELECT * FROM vista_asistencia_salud_detallada;
```

	id_asistencia_y_salud	id_niño	nombre_niño	apellido_niño	sala_niño	fecha	estado_asistencia	observacion_salud
►	1	1	Agustin	Alvarez	abejitas	2025-05-20	Sí	Sin novedad, muy participativo hoy.
	2	2	Benjamin	Burgo	sapitos	2025-05-20	Sí	Leve rasguño en el brazo, curado.
	3	3	Catalina	Cavallo	vaquitas	2025-05-20	No	Ausente por enfermedad, notificado.
	4	4	Delfina	Dominguez	abejitas	2025-05-20	Sí	Asistencia normal, buen apetito.
	5	5	Emilio	Estevez	vaquitas	2025-05-20	Sí	Se quejó de dolor de muelas leve, se avisó a los...
	6	6	Franco	Funes	sapitos	2025-05-20	Sí	Activo y contento durante todo el día.
	7	7	Gala	Garcia	abejitas	2025-05-20	Sí	Sin observaciones, se durmió temprano en la sie...
	8	8	Ignacio	Herrera	vaquitas	2025-05-20	Justificado	Cita con el pediatra.
	9	9	Juana	Ibarra	sapitos	2025-05-20	Sí	Un poco decaído al principio, mejoró luego.
	10	10	Kevin	Juarez	abejitas	2025-05-20	Sí	Puntual, sin novedad médica.

B) Vista: vista_rutinas_diarias_niños

- Este código crea una vista que permite consultar los detalles de las rutinas diarias de cada niño (alimentación, sueño, higiene, actividades) junto con su nombre, apellido y sala.

```
CREATE VIEW vista_rutinas_diarias_niños AS
SELECT
    RD.id_rutina_diaria,
    RD.id_niño,
    N.nombre AS nombre_niño,
    N.apellido AS apellido_niño,
    N.grupo_sala AS sala_niño,
    RD.fecha,
    RD.alimentacion,
    RD.sueño,
    RD.higiene,
    RD.actividades
FROM
    RUTINA_DIARIA AS RD
INNER JOIN
    NIÑO AS N ON RD.id_niño = N.id_niño;
```

- Ejecutamos el siguiente comando para ver como quedo la vista

```
SELECT * FROM vista_rutinas_diarias_niños;
```

	id_rutina_diaria	id_niño	nombre_niño	apellido_niño	sala_niño	fecha	alimentacion	sueño	higiene	actividades
►	1	1	Agustin	Alvarez	abejitas	2025-05-01	Leche y tostadas	Durmió 1 hora	Cambio de pañal	Juegos con bloques
	2	2	Benjamin	Burgo	sapitos	2025-05-02	Yogur con frutas	Durmió 1h 30m	Lavado de manos	Pintura y dibujo
	3	3	Catalina	Cavallo	vaquitas	2025-05-03	Papilla de verduras	Durmió 2 horas	Baño breve	Cuentos y canciones
	4	4	Delfina	Dominguez	abejitas	2025-05-04	Puré de papas	Durmió 45 minutos	Cambio de ropa	Juego libre
	5	5	Emilio	Estevez	vaquitas	2025-05-05	Cereal con leche	No durmió	Lavado de manos	Rompecabezas
	6	6	Franco	Funes	sapitos	2025-05-06	Leche y tostadas	Durmió 1 hora	Cambio de pañal	Cuentos y canciones
	7	7	Gala	Garcia	abejitas	2025-05-07	Yogur con frutas	Durmió 1h 30m	Baño breve	Juegos con bloques
	8	8	Ignacio	Herrera	vaquitas	2025-05-08	Papilla de verduras	Durmió 2 horas	Lavado de manos	Pintura y dibujo
	9	9	Juana	Ibarra	sapitos	2025-05-09	Puré de papas	Durmió 45 minutos	Cambio de ropa	Cuentos y canciones
	10	10	Kevin	Juarez	abejitas	2025-05-10	Cereal con leche	No durmió	Cambio de pañal	Rompecabezas

! El diagrama de Entidad-Relación, los procedimientos, las vistas y la base de datos, serán adjuntados junto con este informe en formato comprimido.