

Sistema de Administración de la Guardería Burbujas

Diego Rosas Cruz, Jose Angel Robles Otero, Felipe Sanchez Martinez
Escuela Superior de Cómputo, IPN

Entregable 3 de julio de 2023



II	Diseño	1
1.	Introducción	3
1.1.	Presentación	3
1.2.	Nomenclatura.	4
1.2.1.	Diseño Arquitectónico	4
1.2.2.	Modelado Estático.	4
1.2.3.	Modelado Dinámico.	5
1.2.4.	Modelos de Manejo de la Información	5
2.	Diseño Arquitectónico	7
2.1.	Propiedades del Software.	8
2.2.	Plataforma	9
2.3.	Arquitectura.	10
3.	Modelo Estático	13
3.1.	Diseño de Subsistemas	13
3.1.1.	Subsistema de Gestión de personal	15
3.1.2.	Subsistema de Gestión padres de familia	19
3.1.3.	Subsistema de Gestión infantes	20
3.1.4.	Subsistema de Gestión días hábiles	23
3.1.5.	Subsistema de Gestión clases	25
3.1.6.	Subsistema Salud	30
3.2.	Diseño de clases	33
3.2.1.	Empleado	34
3.2.2.	Nutriólogo	35
3.2.3.	Médico	36
3.2.4.	Profesor	37
3.2.5.	Capital Humano	39
3.2.6.	Director	40

3.2.7. Recursos humanos	41
3.2.8. Padre de familia	43
3.2.9. Infante	44
3.2.10. Reporte	46
3.2.11. Menu y comidas	48
3.2.12. Calendario y Eventos escolares.	49
3.2.13. Salas.	50
4. Modelo dinámico	51
4.1. Diagrama de secuencia del CU1:Registrar profesor	52
4.2. Diagrama de secuencia del CU4:Enlazar niño con padre	52
4.3. Diagrama de secuencia del CU5:Registrar profesor	53
4.4. Diagrama de secuencia del CU7:Enlazar niño con padre	53
4.5. Diagrama de secuencia del CU9:Registrar ingesta	54
4.6. Diagrama de secuencia del CU12:Registrar tarea	54
4.7. Diagrama de secuencia del CU15:Registrar incidencia medica	55
4.8. Diagrama de secuencia del CU19:Tomar asistencia	55
5. Modelo de manejo de la información.	57
5.1. Consultas para el Caso de Uso CU1	61
5.2. Consultas para el Caso de Uso CU2	61
5.3. Consultas para el Caso de Uso CU3	61
5.4. Consultas para el Caso de Uso CU4	61
5.5. Consultas para el Caso de Uso CU5	62
5.6. Consultas para el Caso de Uso CU6	62
5.7. Consultas para el Caso de Uso CU7	62
5.8. Consultas para el Caso de Uso CU8	63
5.9. Consultas para el Caso de Uso CU9	63
5.10. Consultas para el Caso de Uso CU11	63
5.11. Consultas para el Caso de Uso CU12	63
5.12. Consultas para el Caso de Uso CU14	64
5.13. Consultas para el Caso de Uso CU15	64
5.14. Consultas para el Caso de Uso CU17	64
5.15. Consultas para el Caso de Uso CU19	65
5.16. Consultas para el Caso de Uso CU22	65

Project Charter

Proyecto:	Sistema de Administración Interna de la Guardería Burbujas.		
Responsable:	IPN, Los bubulusuaves		
Autoriza:			
Background/Contexto:	La guarderia burbujas requiere un sistema confiable capaz de administrar todos sus procesos y tareas del personal y con sus clientes..		
Beneficios esperados:	Principales beneficios al término del proyecto.		
Costo estimado:	\$ 7,770,700.00 ± 13 %		
Fecha de inicio:	1 de mayo de 2023	Fecha de término:	3 de julio de 2023
Objetivo:	Crear un sistema para la administracion completa de una guardería.		
Entregables Principales			
C2-EP2	La documentación completa del sistema creado para la guarderia burbujas di-vertidas		
C2-EP3	El sistema elaborado y con las condiciones necesarias para su operabilidad		
Alcance del proyecto			
Incluye:	<ul style="list-style-type: none">● Subsistema de capital humano.● Subsistema de gestión de padres.● Subsistema de seguridad y control de accesos.● Subsistema de calendario y programación.● Subsistema de recursos humanos.● Subsistema de nutrición y calendario de alimentos.● Subsistema de notificación de incidencias.● Subsistema de gestión de salud y servicios médicos.		
Excluye:	<ul style="list-style-type: none">● Subsistema de administración de nomina.● Subsistema de evaluación y seguimiento del desarrollo de los niños.● Subsistema de comunicación interna.● Subsistema de gestión de inventario.● Subsistema de gestión financiera.		
Criterio de éxito:	Tener dos meses de funcionamiento dentro de la guardería.		
Metodología:	Se escogió una metodlogía hibrida, seguimiento de dos metodologías clásicas y una ágil. <ul style="list-style-type: none">● XP● Scrum● UP		

Cuadro 1: Resumen del proyecto

Project Charter

Datos de contacto	
Project Manager:	Ulises Velez, ESCOM, +5257296000.
Project owner:	<ul style="list-style-type: none"> • Diego Rosas, ESCOM, +525539561200. • Felipe Sánchez, ESCOM, +5255293011. • Ángel Robles, ESCOM, +5256296290.
Riesgos y peligros:	<ul style="list-style-type: none"> • Desconocimiento de las tecnologías utilizadas. • Retraso en los tiempos de entrega. • Falta de personal para el desarrollo del proyecto. • Cambios por parte del cliente sobre el sistema.
Supuestos:	<ul style="list-style-type: none"> • Disponibilidad de equipo de cómputo o dispositivo móvil, por parte de los clientes, con acceso a internet. • Disponibilidad de servicio de correo electrónico por parte de padres.
Restricciones y dependencias:	<ul style="list-style-type: none"> • Dependencias sobre un correcto manejo de usuarios y del sistema por parte de la guardería. • Dependencia de tener acceso a internet, para acceder al sistema.

Cuadro 2: Resumen del proyecto

Parte II

Diseño



Este documento contiene la Especificación del diseño sobre el proyecto “*Guardería Burbujas*” correspondiente al trabajo realizado en el 2023/1 para la materia de Análisis y diseño de sistemas en el grupo 4BM1 por el equipo *Los bubulusuaves*.

1.1. Presentación

Este documento contiene la especificación de los requerimientos del usuario y del sistema para el diseño y desarrollo del sistema de la guardería. Su objetivo principal es establecer una base clara y precisa de los elementos necesarios para construir un sistema que cumpla con las necesidades y expectativas de los usuarios finales, así como con los requisitos técnicos y funcionales.

- Los requerimientos del usuario se centran en comprender y documentar las necesidades específicas de los usuarios finales de la guardería. Esto implica identificar las funcionalidades y características que el sistema debe ofrecer para satisfacer estas necesidades, como la gestión de la información de los infantes, el control de acceso de los padres, la generación de reportes, entre otros.
- Los requerimientos del sistema se enfocan en definir las características técnicas, funcionales y de rendimiento que el sistema de la guardería debe cumplir. Esto incluye aspectos como la arquitectura del sistema, las interfaces de usuario, la seguridad de los datos, la escalabilidad para manejar un crecimiento futuro, la integración con otros sistemas, entre otros aspectos técnicos relevantes.

Al establecer estos requerimientos, se busca garantizar la calidad y el éxito del sistema de la guardería, asegurando su alineación con los objetivos del proyecto y la satisfacción de los usuarios finales. Además, estos requerimientos servirán como referencia durante todo el ciclo de vida del proyecto, facilitando la toma de decisiones, el seguimiento del progreso y la comunicación efectiva entre todos los involucrados en el desarrollo del sistema.

En resumen, este documento de especificación de requerimientos tiene como objetivo principal sentar las bases para el diseño y desarrollo exitoso del sistema de la guardería, asegurando una comprensión compartida de las necesidades y expectativas, tanto de los usuarios finales como del sistema en sí mismo

1.2. Nomenclatura.

La información del presente documento se encuentra estructurada mediante diagramas, tablas y secciones con nomenclaturas y estándares específicos. Este capítulo tiene como finalidad indicar la forma en que se deben leer estos elementos para un mejor entendimiento.

1.2.1. Diseño Arquitectónico


El diseño arquitectónico se enfoca en definir la estructura general del sistema, incluyendo la distribución de los componentes, los patrones de interacción y las decisiones clave relacionadas con la arquitectura del sistema. Este diseño proporciona una visión de alto nivel de cómo se organiza y se relaciona cada componente del sistema, y cómo se cumplen los requerimientos funcionales y no funcionales. Al crear un diseño arquitectónico sólido, se busca maximizar la escalabilidad, la modularidad, la seguridad y el rendimiento del sistema.

1.2.2. Modelado Estático.

Se enfoca en representar y describir la estructura y relaciones estáticas del sistema en desarrollo. Este modelado permite comprender los elementos del sistema y cómo se relacionan entre sí, sin tener en cuenta su comportamiento dinámico. El modelado estático incluye diagramas de clases, diagramas de componentes, diagramas de despliegue y otras técnicas que ayudan a visualizar y analizar la arquitectura del sistema desde una perspectiva estática mediante módulos.

El diseño de módulos se centra en la división del sistema en componentes más pequeños y cohesivos, llamados módulos. Estos módulos representan unidades funcionales independientes que pueden ser desarrolladas, probadas y mantenidas de manera individual. El diseño de módulos busca maximizar la cohesión dentro de cada módulo, lo que significa que las funcionalidades relacionadas se agrupan juntas, y minimizar la dependencia entre módulos, lo que permite cambios más fáciles y flexibilidad en el desarrollo del sistema.

Los diagramas de casos de uso son una herramienta usada para representar las transacciones entre un actor y el sistema, las cuales siempre tendrán un valor agregado o un propósito para que el actor las realice. En estos diagramas se podrán observar los siguientes elementos:

- Representa al sistema mediante un óvalo. ●
- Representa al actor que va a interactuar con el sistema. 
- Relación «extends». Indica que un caso de uso puede ejecutarse a partir de otro.
- Relación «include». Indica que un caso de uso debe ejecutarse a partir de otro.

La conexión entre un actor y un caso de uso se realiza mediante una línea como se muestra en la Figura 1.1.



Figura 1.1: Interacción del actor con el caso de uso

Los casos de uso se encontrarán dentro de paquetes (representados por carpetas) indicando así que pertenecen a un mismo módulo, como se muestra en la Figura 1.2.

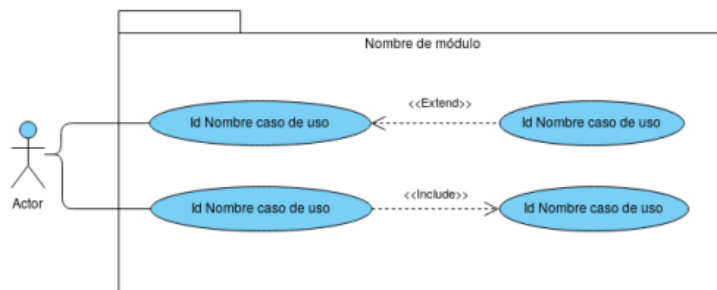


Figura 1.2: Un Actor con varios casos de uso dentro de un módulo

1.2.3. Modelado Dinámico.

Se centra en capturar y representar el comportamiento y las interacciones del sistema a lo largo del tiempo. Se utiliza para describir cómo los diferentes componentes del sistema interactúan entre sí y cómo se llevan a cabo las acciones y procesos. El modelado dinámico se realiza mediante diagramas de secuencia, diagramas de actividad, diagramas de estado y otras técnicas que permiten visualizar y analizar el flujo de eventos y el comportamiento dinámico del sistema.

1.2.4. Modelos de Manejo de la Información

Los modelos de manejo de la información se refieren a la representación y estructuración de los datos que serán almacenados y manipulados por el sistema. Estos modelos describen las entidades, atributos y relaciones entre los datos, y pueden incluir diagramas de entidad-relación, diagramas de clase y otros diagramas que ayudan a visualizar y comprender la estructura de los datos. El diseño de los modelos de manejo de la información busca garantizar la integridad, consistencia y eficiencia en la gestión de los datos, así como la adaptación a las necesidades específicas del sistema de la guardería.

Al abordar estos temas en el diseño del sistema de la guardería, se busca establecer una arquitectura sólida, modular y eficiente, así como una gestión efectiva de la información. Estos elementos son fundamentales para garantizar el rendimiento, la escalabilidad y la usabilidad del sistema, y para satisfacer las necesidades y expectativas de los usuarios finales.

Diseño Arquitectónico

Es una etapa clave en el desarrollo del sistema, donde se establecen las bases para su construcción y funcionamiento. En esta sección, se abordarán diversos aspectos relacionados con la arquitectura del sistema, como las propiedades del software, la plataforma utilizada, el costo y la arquitectura misma.

En cuanto a las propiedades del software, se considerarán aspectos como la modularidad, la escalabilidad, la flexibilidad y la fiabilidad. Estas propiedades son fundamentales para garantizar que el sistema pueda adaptarse a cambios futuros, crecer en función de las necesidades del usuario y mantener un rendimiento óptimo en diferentes situaciones.

La elección de la plataforma del sistema es otro aspecto importante a considerar. Esto incluye seleccionar el entorno de desarrollo, el lenguaje de programación, las herramientas y los frameworks que serán utilizados para implementar el sistema. La elección adecuada de la plataforma puede tener un impacto significativo en la eficiencia del desarrollo y en el rendimiento del sistema final.

El costo es otro factor crítico a tener en cuenta durante el diseño arquitectónico. Se deben considerar los recursos financieros disponibles, así como los costos asociados con la adquisición de hardware, licencias de software, mantenimiento y capacitación. El diseño arquitectónico debe buscar un equilibrio entre las funcionalidades y características deseadas y los recursos disponibles.

En cuanto a la arquitectura del sistema, se definirán los componentes principales, sus interacciones y la estructura global del sistema. Esto implica decidir si se utilizará una arquitectura monolítica, cliente-servidor, basada en microservicios u otra opción. La elección de la arquitectura adecuada dependerá de factores como los requisitos funcionales y no funcionales del sistema, la escalabilidad y el rendimiento esperado.

En resumen, el capítulo de "Diseño Arquitectónico" abarca diversos aspectos fundamentales para el desarrollo del sistema. Se consideran las propiedades del software, la elección de la plataforma, el costo y la definición de la arquitectura. Estos elementos sientan las bases para la construcción de un sistema robusto, eficiente y que cumpla con los requisitos establecidos.

2.1. Propiedades del Software.

En el diseño arquitectónico de un sistema, es importante tener en cuenta diversas propiedades del software que pueden influir en su calidad, rendimiento y mantenibilidad. Estas propiedades abarcan diferentes aspectos del sistema y su comportamiento, y se consideran como criterios clave para evaluar su eficacia y adecuación a los requisitos establecidos.

Algunas propiedades del software relevantes para el diseño arquitectónico del sistema son:

- **Escalabilidad de datos:** La escalabilidad de datos se refiere a la capacidad de un sistema para manejar y gestionar grandes volúmenes de datos a medida que crece. Implica diseñar y construir una infraestructura de datos que pueda adaptarse y mantener un rendimiento óptimo a medida que la cantidad de datos aumenta, debido al fácil aumento de reportes por día para cada infante. Dentro del sistema se buscó hacer escalable gracias a la base de datos **Cosmos DB**, que ofrece características y funcionalidades que permiten escalar datos de manera efectiva, ya sea a través de la escalabilidad horizontal, la replicación global, los índices eficientes y la integración con servicios en la nube.

Cosmos DB es una base de datos multimodelo y globalmente distribuida ofrecida por Azure. Permite almacenar y consultar datos de manera escalable y de alto rendimiento, con soporte para múltiples modelos de datos, como documentos, grafos, clave-valor y columnas. Al aprovechar la escalabilidad horizontal, Cosmos DB puede manejar grandes volúmenes de datos y crecer a medida que aumenta la carga de trabajo.

Además, Cosmos DB ofrece replicación global, lo que garantiza que los datos estén disponibles en múltiples regiones geográficas, mejorando la disponibilidad y la latencia de acceso a los datos en todo el mundo. Los índices eficientes permiten acelerar las consultas y mejorar el rendimiento de las operaciones, lo que es crucial para gestionar grandes volúmenes de datos en un tiempo razonable.

- **Flexibilidad:** Capacidad del sistema para adaptarse a cambios y requerimientos futuros sin necesidad de realizar modificaciones mayores en la arquitectura. Un diseño arquitectónico flexible permite incorporar nuevas funcionalidades, integrar tecnologías emergentes o adaptarse a diferentes entornos sin comprometer la estabilidad del sistema.
- **Seguridad:** La seguridad del software es esencial para proteger los datos, las funcionalidades y los usuarios del sistema. Un diseño arquitectónico seguro debe incluir mecanismos y controles adecuados para prevenir ataques, garantizar la confidencialidad, la integridad y la disponibilidad de la información, y mitigar posibles vulnerabilidades. Para ello es que se han agregado medidas de autenticación a través del uso de middlewares en Express.js, se pueden implementar mecanismos de autenticación, como JSON Web Tokens (JWT), para verificar la identidad de los usuarios y controlar el acceso a las funcionalidades del sistema. Además de la gestión de permisos y roles dentro del sistema.
- **Rendimiento:** El rendimiento se refiere a la capacidad del sistema para responder de manera eficiente a las solicitudes de los usuarios y procesar grandes volúmenes de datos en un tiempo razonable. Un diseño arquitectónico que optimice el rendimiento debe considerar aspectos como la optimización de algoritmos, el uso eficiente de recursos y la minimización de cuellos de botella.

Para lograr un alto rendimiento en el sistema, se busca implementar una infraestructura en la nube utilizando Azure. Azure es una plataforma de servicios en la nube que ofrece diversas ventajas en términos de rendimiento, escalabilidad y disponibilidad. Algunas de las razones por las cuales se elige Azure son:
- **Mantenibilidad:** La mantenibilidad se refiere a la facilidad con la que se puede realizar el mantenimiento y la evolución del sistema a lo largo del tiempo. Un diseño arquitectónico mantenible facilita la identificación y corrección de errores, la incorporación de mejoras y la adaptación a cambios en los requisitos o tecnologías subyacentes. Esto se logró dividiendo cada subsistema en una careta distinta en la REST API, la carpeta controladores,

contiene todos los controladores de cada entidad, y llevan toda la acciones y logica de los usuarios, la carpeta `modelos` contiene a los objetos creados dentro de la base de datos, la carpeta `rutas` definen las rutas y la gestión de las solicitudes HTTP de cada entidad.

Estas propiedades del software son solo algunas de las muchas consideraciones que deben tenerse en cuenta durante el diseño arquitectónico. La elección adecuada de estas propiedades dependerá de los requisitos y objetivos específicos del sistema, así como de las restricciones y contextos en los que se desarrolla. Al optimizar y equilibrar estas propiedades, se puede lograr un diseño arquitectónico sólido y efectivo que cumpla con las necesidades de la guardería.

2.2. Plataforma

La plataforma se refiere al entorno o conjunto de recursos tecnológicos utilizados para alojar y ejecutar la aplicación o sistema. En otras palabras, es la infraestructura sobre la cual se implementa y se ejecuta el sistema.

- **Cosmos DB:** Es una base de datos multimodelo y globalmente distribuida ofrecida por Azure como un servicio de base de datos como servicio (DBaaS). Cosmos DB permite almacenar y consultar datos de manera escalable y de alto rendimiento. Ofrece soporte para múltiples modelos de datos, como documentos, grafos, clave-valor y columnas, lo que brinda flexibilidad en el diseño de la base de datos según las necesidades del sistema. Además, Cosmos DB proporciona características como la replicación global, la latencia baja y la alta disponibilidad, lo que garantiza un acceso rápido y confiable a los datos en todo el mundo.
- **Express.js:** Es un framework de desarrollo web para Node.js. Express.js se encarga de construir la parte del servidor de la aplicación web, permitiendo definir rutas, manejar peticiones HTTP y establecer la lógica de negocio del sistema.
- **Vue.js:** Es un framework de JavaScript utilizado para construir la interfaz de usuario y la parte del cliente de la aplicación web. Vue.js facilita la creación de componentes reutilizables y la gestión del estado de la aplicación, lo que mejora la experiencia del usuario y la interactividad del sistema.
- **Node.js:** Es un entorno de ejecución de JavaScript basado en el motor V8 de Google Chrome. Node.js permite ejecutar código JavaScript en el lado del servidor, lo que proporciona un entorno coherente tanto en el cliente como en el servidor. Node.js ofrece una amplia gama de bibliotecas y módulos que facilitan el desarrollo de aplicaciones web robustas y escalables.
- **Azure PaaS:** Azure Platform as a Service (PaaS) es un conjunto de servicios en la nube ofrecidos por Azure que permite a los desarrolladores construir, desplegar y administrar aplicaciones sin tener que preocuparse por la infraestructura subyacente. Azure PaaS ofrece servicios específicos para el desarrollo y despliegue de aplicaciones web, como Azure App Service, que permite desplegar aplicaciones web basadas en Node.js, Express.js y Vue.js de manera rápida y sencilla.
- **Azure DBaaS:** Azure Database as a Service (DBaaS) es un servicio ofrecido por Azure que permite a los desarrolladores utilizar bases de datos en la nube sin tener que administrar la infraestructura subyacente. Azure DBaaS incluye servicios como Cosmos DB, que es una base de datos multimodelo globalmente distribuida. Estos servicios ofrecen escalabilidad, disponibilidad y rendimiento optimizados, lo que facilita el almacenamiento y acceso a datos de manera eficiente en el sistema.
- **Azure:** Se utiliza la plataforma en la nube Azure para alojar y desplegar el sistema. Azure proporciona una infraestructura escalable y altamente disponible, lo que garantiza un rendimiento óptimo y una disponibilidad continua.

del sistema. Además, Azure ofrece una amplia gama de servicios y herramientas que pueden utilizarse para optimizar y mejorar el rendimiento del sistema, como la escalabilidad automática, el equilibrio de carga y la gestión de recursos.

La elección de la MEVN stack como arquitectura del sistema, junto con el uso de servicios como Cosmos DB y Azure PaaS, ofrece numerosos beneficios, como la flexibilidad en el manejo de datos, la eficiencia en el manejo de peticiones, la capacidad de crear interfaces de usuario interactivas y una infraestructura escalable y confiable. Esto permite construir un sistema web moderno, escalable y de alto rendimiento.

En la Figura 2.1 se muestra una representación visual de la plataforma mencionada en operación.

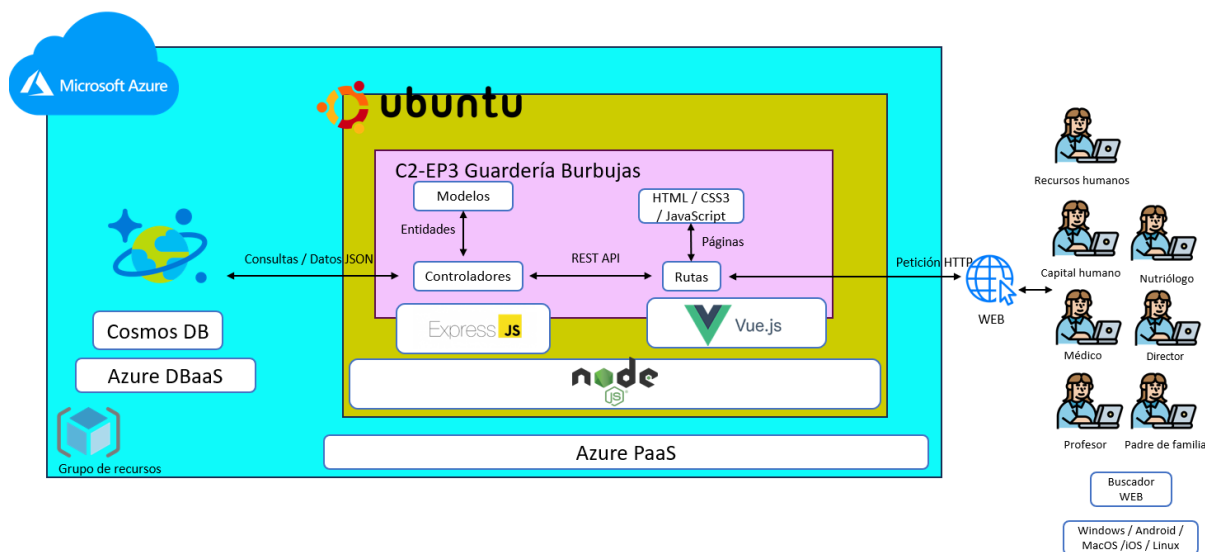


Figura 2.1: Plataforma del sistema.

2.3. Arquitectura.

Se puede considerar una arquitectura de tres capas que esta basado en la arquitectura del modelo vista controlador (MVC), que consiste en dividir la aplicación en capas lógicas distintas, cada una con su propósito y responsabilidades específicas. Estas capas son:

1. Capa de presentación o interfaz de usuario: Esta capa es responsable de la interacción con los usuarios finales y la presentación de la información de manera visualmente atractiva y fácil de usar. En este caso, Vue.js se encarga de construir la interfaz de usuario y proporcionar una experiencia interactiva al usuario. Esta capa también puede incluir la gestión de eventos del lado del cliente y la comunicación con el backend a través de API.
2. Capa de lógica de negocio: Esta capa contiene la lógica y reglas de negocio de la aplicación. Aquí es donde se procesan las solicitudes de los usuarios, se realizan operaciones en la base de datos y se aplican las reglas y validaciones necesarias. En la arquitectura MEVN stack, Express.js se encarga de construir el backend y manejar las solicitudes HTTP, las rutas y la lógica de negocio. En esta capa también se puede implementar la seguridad, la validación de datos y otras funcionalidades relacionadas con la lógica de negocio.

3. Capa de persistencia de datos: Esta capa se encarga del almacenamiento y acceso a los datos. Aquí es donde se interactúa con la base de datos para realizar operaciones de lectura y escritura. En el caso de este sistema, se utiliza Cosmos DB como la base de datos principal. Cosmos DB ofrece una integración nativa con Node.js y permite almacenar y consultar datos de manera eficiente y escalable. En esta capa también se pueden implementar estrategias de caché, gestión de transacciones y otros mecanismos relacionados con la persistencia de datos.

La arquitectura de tres capas proporciona varios beneficios, como la separación de responsabilidades, la modularidad, la reutilización de código y la facilidad de mantenimiento. Cada capa puede desarrollarse y escalarse de forma independiente, lo que permite un desarrollo ágil y una mayor flexibilidad a medida que el sistema evoluciona.

En la Figura 2.2 se muestra una representación visual de la arquitectura de tres capas dentro del sistema.

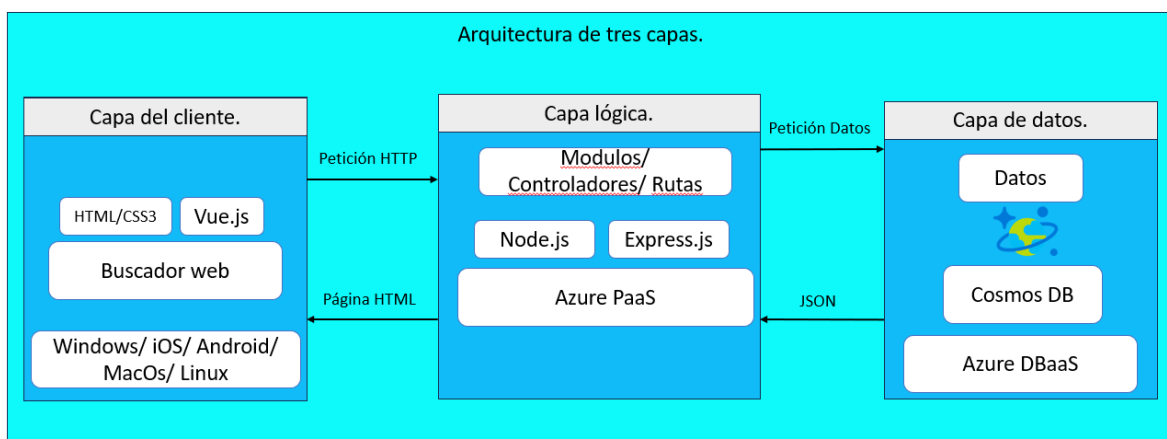


Figura 2.2: Arquitectura de tres capas en el sistema.

La utilización de la arquitectura MEVN stack junto con la arquitectura de tres capas permite construir un sistema web robusto, escalable y modular. La combinación de estas tecnologías y enfoques arquitectónicos proporciona una base sólida para desarrollar y desplegar aplicaciones web modernas y eficientes en la nube de Azure.

En el diseño de sistemas de software, el enfoque modular es esencial para crear aplicaciones eficientes, mantenibles y escalables. El diseño de módulos se basa en la división del sistema en componentes más pequeños y autónomos, conocidos como módulos, que encapsulan funcionalidades específicas y se comunican entre sí de manera coherente. En el contexto del sistema de la guardería "Burbujas", el diseño de módulos desempeña un papel crucial en la organización y la implementación de las diversas características y procesos que son necesarios para su funcionamiento.

El objetivo del diseño de módulos en el sistema de la "Guardería burbujas" es garantizar una estructura clara y modular, donde cada módulo se encargue de una funcionalidad específica y pueda ser desarrollado, probado y mantenido de forma independiente. Esto permite una mayor flexibilidad en el desarrollo y facilita la colaboración entre el equipo de desarrollo. Además, el diseño de módulos promueve la reutilización de código, ya que los módulos pueden ser utilizados en diferentes partes del sistema o en proyectos futuros.

Al diseñar los módulos del sistema de la "Guardería burbujas", se deben tener en cuenta diferentes aspectos, como la cohesión, la modularidad y la interoperabilidad. La cohesión se refiere a la medida en que los elementos dentro de un módulo están relacionados y trabajan juntos para cumplir una funcionalidad específica. Una alta cohesión indica que un módulo tiene una única responsabilidad y se centra en una tarea concreta. Por otro lado, la modularidad se refiere a la capacidad de un módulo de ser independiente y reemplazable sin afectar el funcionamiento de otros módulos. La interoperabilidad implica que los módulos deben poder comunicarse entre sí de manera eficiente y consistente, a través de interfaces bien definidas y protocolos de interacción.

3.1. Diseño de Subsistemas

En el diseño de sistemas complejos, como es el caso del sistema de la guardería "Guardería Burbujas", se utiliza el concepto de subsistemas para organizar y estructurar las diferentes partes o componentes del sistema. Un subsistema es una unidad funcionalmente independiente dentro de un sistema más grande, que tiene la capacidad de realizar tareas específicas y contribuir al funcionamiento global del sistema.

Los subsistemas se utilizan para dividir un sistema en componentes más pequeños y manejables, lo cual facilita el diseño, la implementación y el mantenimiento del sistema en su conjunto. Cada subsistema tiene su propia funcionalidad y responsabilidades, y puede interactuar con otros subsistemas para lograr los objetivos del sistema en su conjunto.

El diseño de subsistemas implica identificar las diferentes partes o módulos del sistema que pueden funcionar de manera independiente pero también interactuar entre sí. Cada subsistema puede tener su propia arquitectura interna, sus interfaces con otros subsistemas y sus propias reglas y lógica de funcionamiento.

El diseño de subsistemas se basa en el principio de modularidad, que consiste en dividir un sistema en componentes más pequeños y cohesivos. Cada subsistema se enfoca en una funcionalidad específica y puede ser desarrollado y probado de manera independiente. Esto permite un desarrollo paralelo y una mayor reutilización de componentes en futuros proyectos.

En el contexto del sistema de la guardería "Guardería Burbujas", se pueden identificar varios subsistemas, como el subsistema de Gestión de personal, el subsistema de Gestión de padres de familia, el subsistema de Gestión de días hábiles, el subsistema de Gestión de personal, el subsistema de Gestión de clases y el subsistema de Salud.

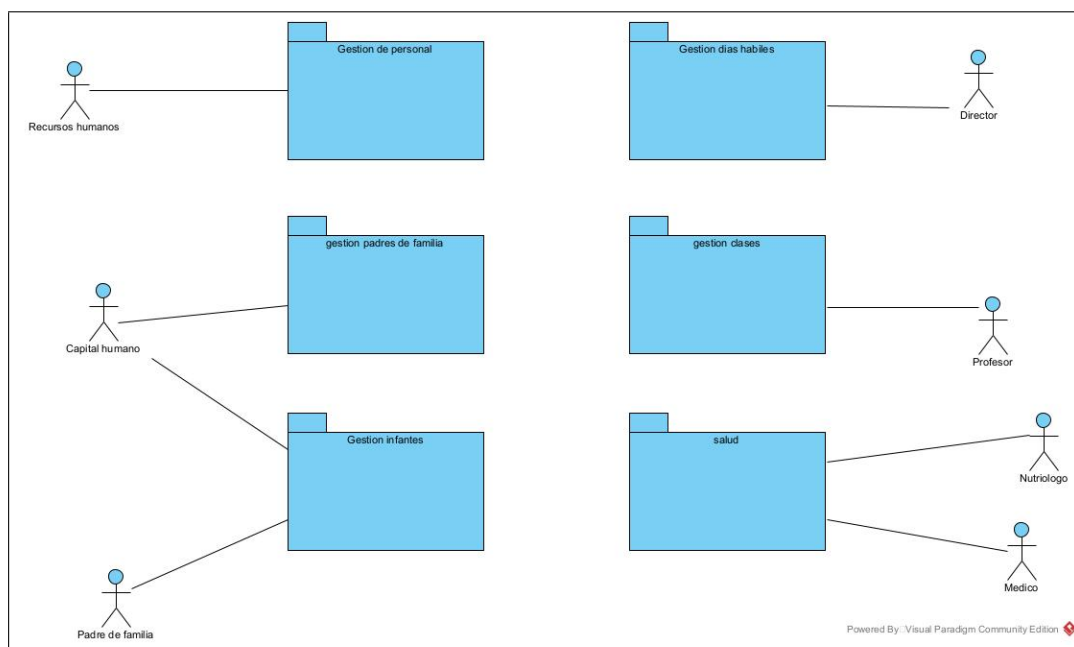


Figura 3.1: Diagrama de casos de uso del sistema.

Que son los subsistemas para los actores Recursos Humanos, Capital humano, Padre de familia, Médico, Nutriólogo, Director y Profesor.

3.1.1. Subsistema de Gestión de personal

El personal de la guardería es un elemento clave para brindar un ambiente seguro, cuidadoso y estimulante para los niños. El subsistema de Gestión de Personal tiene como objetivo garantizar que se cuente con un equipo de profesionales capacitados y comprometidos, que cumplan con los requisitos de calificación necesarios y estén disponibles en los momentos adecuados para brindar atención y cuidado a los infantes.

Este subsistema involucra diferentes actores, como el Director de la guardería, los Recursos Humanos y los propios miembros del personal. Cada actor desempeña un papel específico en la gestión del personal, desde la contratación y el proceso de selección hasta la asignación de tareas y la supervisión del rendimiento.

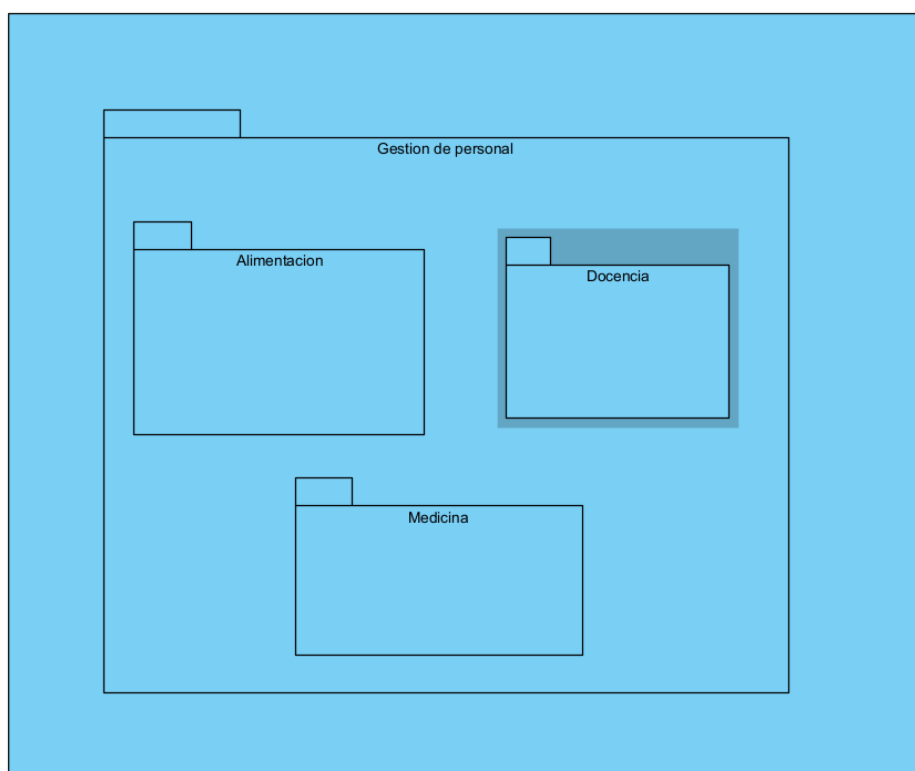


Figura 3.2: Subsistema Gestion de personal.

Submódulo de Docencia

El modulo "Docencia" es una parte integral del sistema de la guardería "Guardería Burbujas" se encarga de gestionar todos los aspectos relacionados con el personal docente. Este módulo permite llevar un control eficiente del cuerpo docente, garantizando una educación de calidad y un entorno de aprendizaje óptimo para los niños.

El módulo de Docencia ofrece las siguientes funcionalidades:

- * Registro de profesor: Esta funcionalidad permite ingresar y almacenar la información relevante de los profesores que formarán parte del equipo docente de la guardería. Se recopilan datos como el nombre, la experiencia educativa, las especialidades y cualquier otra información necesaria para la asignación adecuada de los profesores a los grupos de niños.

- * **Dar de baja a profesor:** En caso de que sea necesario, esta funcionalidad permite dar de baja a un profesor del sistema. Esto puede ocurrir por diversos motivos, como cambios en el personal, finalización de contrato o cualquier otra circunstancia que requiera la eliminación de un profesor del registro del sistema. Al dar de baja a un profesor, se actualiza la información de disponibilidad del personal docente.
- * **Actualización de datos del profesor:** Esta funcionalidad permite realizar modificaciones y actualizaciones en la información de los profesores registrados. Puede incluir cambios en la información de contacto, actualización de experiencia educativa, incorporación de nuevas especialidades o cualquier otro dato relevante para mantener los registros del profesorado al día.

El módulo de Docencia es esencial para garantizar que el personal docente esté adecuadamente registrado, actualizado y asignado a los grupos de niños correspondientes. Esto facilita la planificación y organización de las clases, asegurando una distribución equitativa de los profesores y una atención de calidad en el proceso educativo de los infantes.

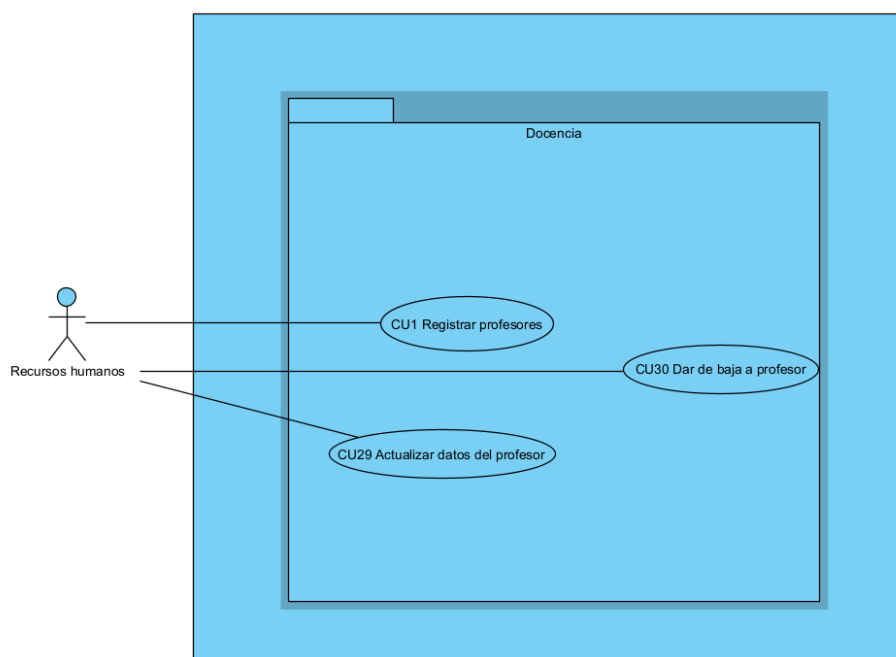


Figura 3.3: Módulo de Docencia del subsistema Gestión de personal.

Submódulo de Alimentación

El módulo .Alimentacion. es una parte integral del sistema de la guardería "Guardería Burbujasz se encarga de gestionar todos los aspectos relacionados con el personal de nutrición y alimentación. Este módulo permite llevar un control eficiente de los nutriólogos, asegurando una alimentación adecuada y equilibrada para los niños.

El módulo de Nutriólogo ofrece las siguientes funcionalidades:

- * **Registro de nutriólogo:** Esta funcionalidad permite ingresar y almacenar la información relevante de los nutriólogos que formarán parte del equipo encargado de la alimentación en la guardería. Se recopilan datos como el nombre,

la experiencia en nutrición, las especialidades y cualquier otra información necesaria para la asignación adecuada de los nutriólogos a los grupos de niños.

- * Dar de baja a nutriólogo: En caso de que sea necesario, esta funcionalidad permite dar de baja a un nutriólogo del sistema. Esto puede ocurrir por diversos motivos, como cambios en el personal, finalización de contrato o cualquier otra circunstancia que requiera la eliminación de un nutriólogo del registro del sistema. Al dar de baja a un nutriólogo, se actualiza la información de disponibilidad del personal de nutrición.
- * Actualización de datos del nutriólogo: Esta funcionalidad permite realizar modificaciones y actualizaciones en la información de los nutriólogos registrados. Puede incluir cambios en la información de contacto, actualización de experiencia en nutrición, incorporación de nuevas especialidades o cualquier otro dato relevante para mantener los registros del personal de nutrición al día.

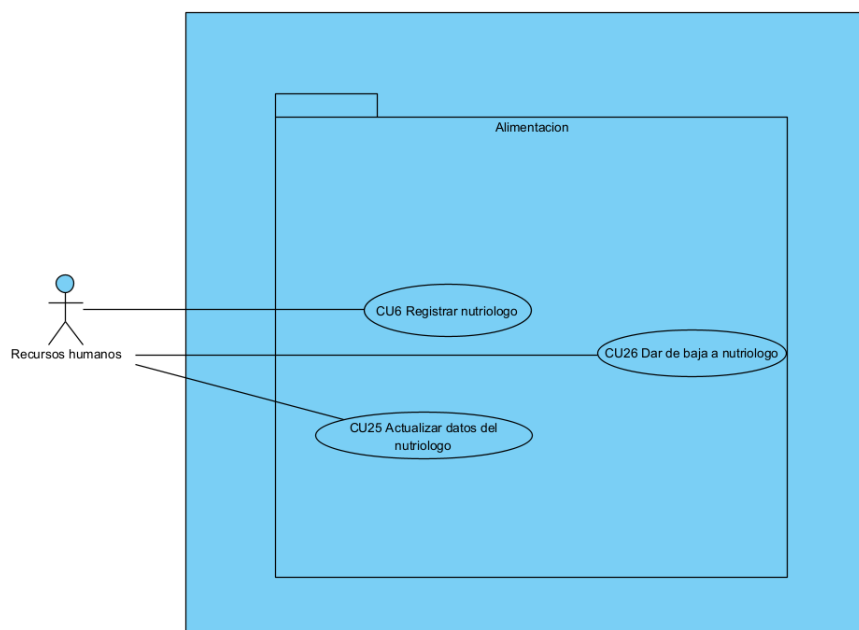


Figura 3.4: Módulo de Alimentación del subsistema Gestión de personal.

Submódulo de Medicina

El módulo "Medicina" es una parte integral del sistema de la guardería "Guardería Burbujasz" se encarga de gestionar todos los aspectos relacionados con el personal médico. Este módulo permite llevar un control eficiente de los médicos, asegurando la salud y el bienestar de los niños.

El módulo de Medicina ofrece las siguientes funcionalidades:

- * Registro de médico: Esta funcionalidad permite ingresar y almacenar la información relevante de los médicos que formarán parte del equipo médico de la guardería. Se recopilan datos como el nombre, la especialidad médica, la experiencia y cualquier otra información necesaria para la asignación adecuada de los médicos en el cuidado de los niños.

- * Dar de baja a médico: En caso de que sea necesario, esta funcionalidad permite dar de baja a un médico del sistema. Esto puede ocurrir por diversos motivos, como cambios en el personal, finalización de contrato o cualquier otra circunstancia que requiera la eliminación de un médico del registro del sistema. Al dar de baja a un médico, se actualiza la información de disponibilidad del personal médico.
- * Actualización de datos del médico: Esta funcionalidad permite realizar modificaciones y actualizaciones en la información de los médicos registrados. Puede incluir cambios en la información de contacto, actualización de especialidades médicas, incorporación de nuevas certificaciones o cualquier otro dato relevante para mantener los registros del personal médico al día.

El módulo de Medicina es esencial para garantizar que el personal médico esté adecuadamente registrado, actualizado y asignado a las tareas correspondientes. Esto facilita la atención médica de los niños, asegurando que reciban la atención adecuada en caso de enfermedades, lesiones u otras situaciones médicas que puedan surgir.

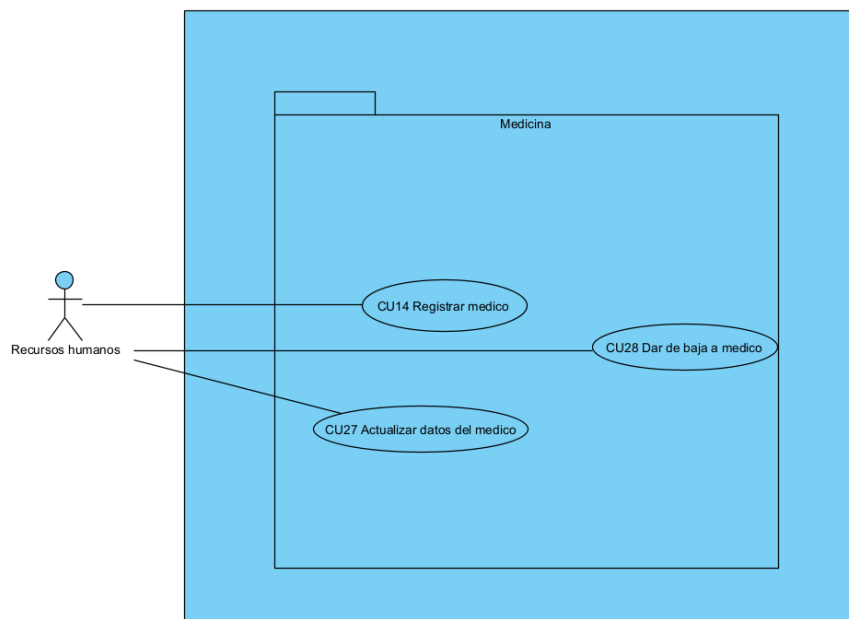


Figura 3.5: Modulo de Medicina del subsistema Gestion de personal.

3.1.2. Subsistema de Gestión padres de familia

El subsistema de "Gestión de padres de familia" es una parte esencial del sistema de la guardería "Guardería Burbujas". Este subsistema se encarga de administrar y mantener actualizada la información de los padres de familia de los infantes que asisten a la guardería. Su objetivo principal es tener la información de todos los padres de familia registrados para brindarles información relevante sobre sus hijos y garantizar una participación activa en el cuidado y desarrollo de los niños.

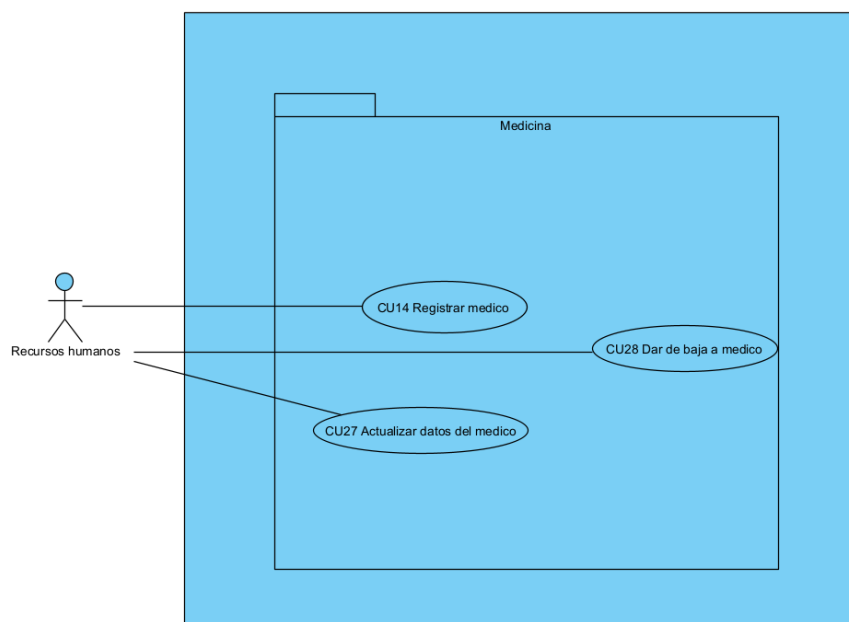


Figura 3.6: Subsistema Gestion padres de familia.

3.1.3. Subsistema de Gestión infantes

El subsistema de "Gestión de infantes" es una parte fundamental del sistema de la guardería "Guardería Burbujas". Este subsistema se encarga de administrar y mantener actualizada la información de los infantes que asisten a la guardería, así como de supervisar su bienestar, progreso y desarrollo durante su estancia en el centro.

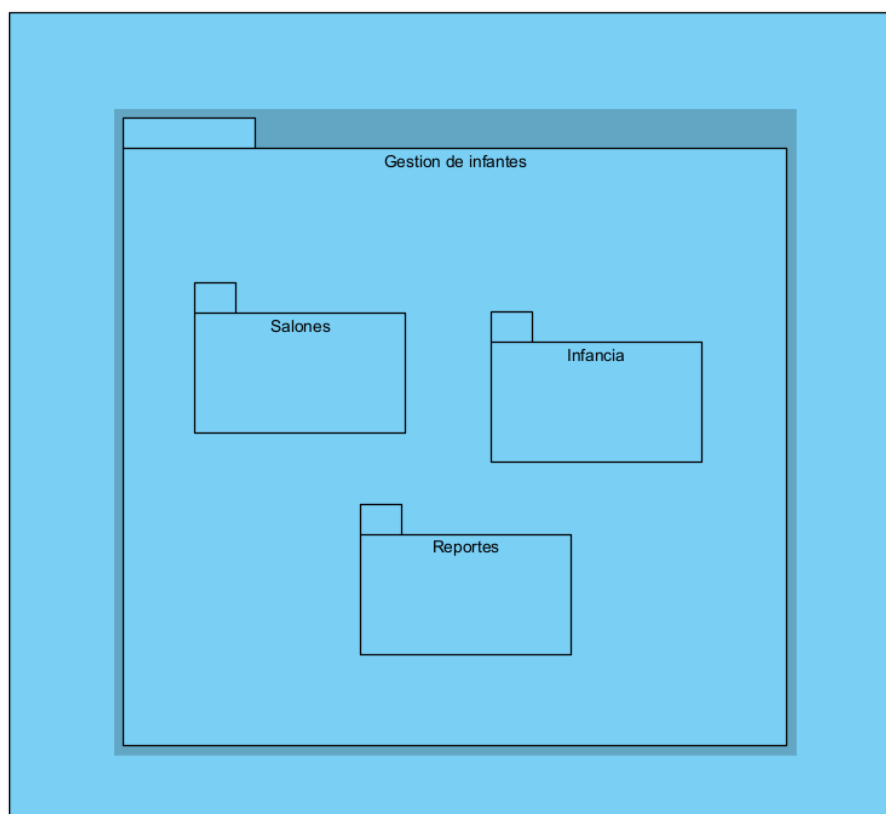


Figura 3.7: Subsistema Gestion infantes.

El subsistema de Gestión de infantes se divide en varios submódulos, que son los siguientes:

Submódulo de salones

El submódulo de salones tiene como objetivo principal organizar y administrar los diferentes salones de la guardería. En este submódulo se gestionan aspectos como la asignación de infantes a los salones correspondientes, la distribución equitativa de niños por grupo de edad y la supervisión de la capacidad máxima de cada salón. Además, se registran y actualizan datos relevantes de cada salón, como el nombre, la ubicación, los horarios y las actividades programadas.

Submódulo de infancia

El submódulo de infancia se encarga de recopilar y mantener actualizada la información personal y médica de cada infante. En este submódulo se registran datos como el nombre, la fecha de nacimiento, los contactos de emergencia, las alergias, las enfermedades crónicas y cualquier otra información relevante para brindar una atención adecuada y

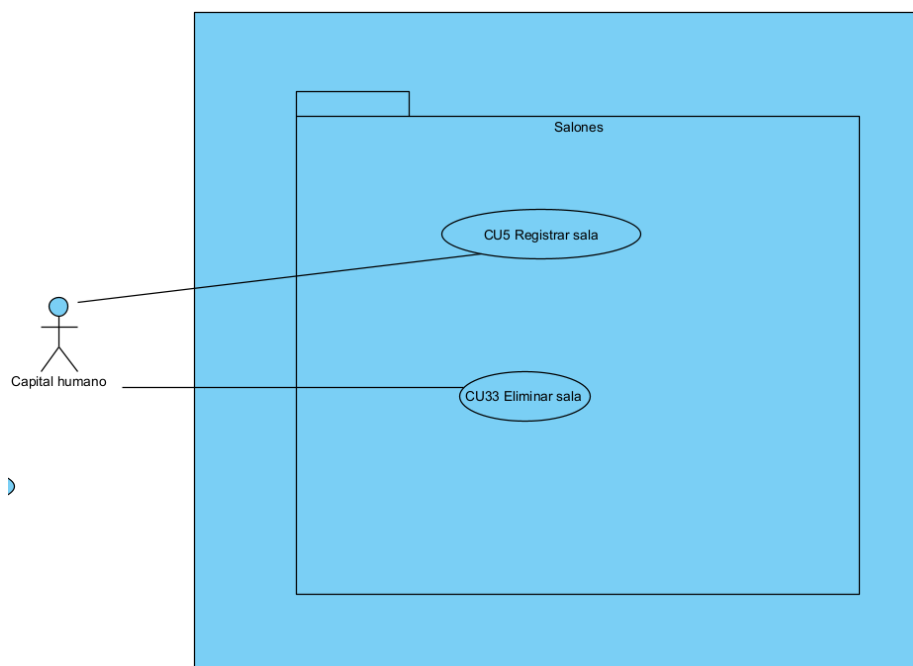


Figura 3.8: Diagrama del Submodulo de salones.

personalizada a cada niño. Además, se realiza un seguimiento del desarrollo y los hitos alcanzados por cada infante, lo que permite detectar y abordar oportunamente cualquier necesidad o preocupación.

Submódulo de reportes

El submódulo de reportes tiene como finalidad generar y proporcionar informes periódicos sobre el progreso y el bienestar de los infantes en la guardería. Estos informes incluyen aspectos como la asistencia, el desarrollo físico, emocional y cognitivo, las actividades realizadas, los logros alcanzados y cualquier otra información relevante. Estos reportes se comparten con los padres de familia y también pueden ser utilizados por el personal docente y administrativo para evaluar el desempeño de los infantes y realizar ajustes en los planes de atención y enseñanza.

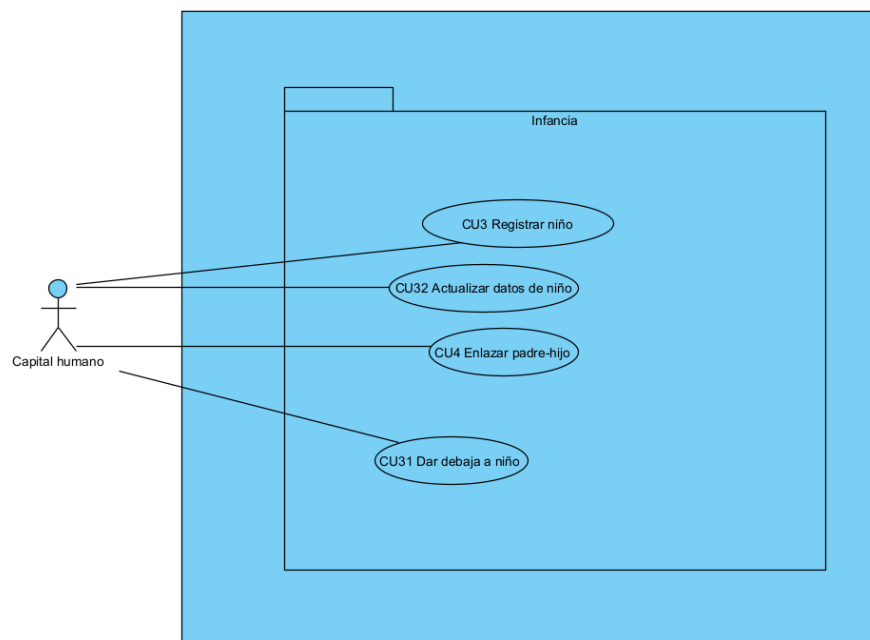


Figura 3.9: Diagrama del Submodulo de infancia.

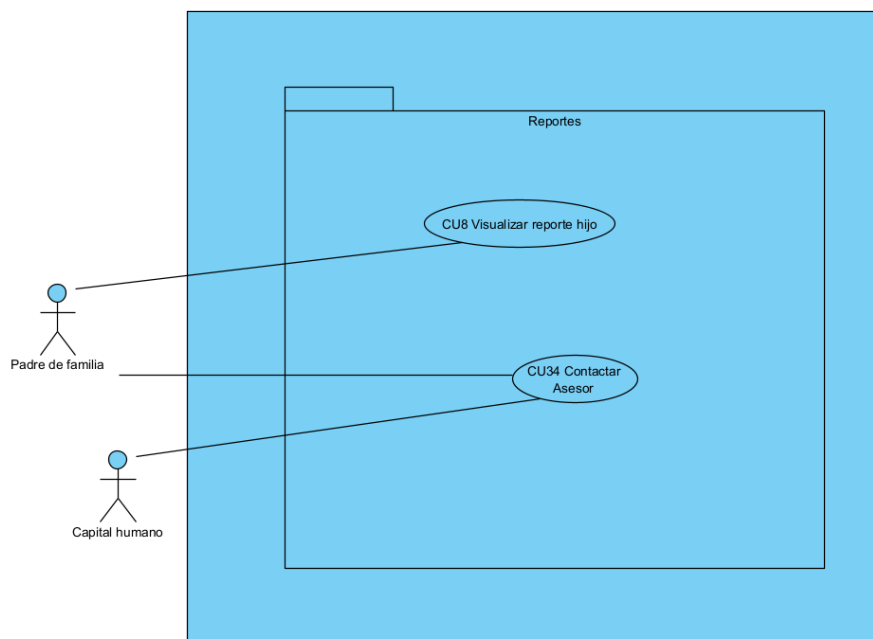


Figura 3.10: Diagrama del Submodulo de reportes.

3.1.4. Subsistema de Gestión días hábiles

El subsistema de "Gestión de días hábiles" se encarga de administrar y gestionar los días hábiles, eventos escolares y días inhábiles en la guardería, con el objetivo de asegurar una planificación eficiente y un funcionamiento adecuado del centro.

El subsistema de Gestión de días hábiles consta de dos casos de uso directos, que son los siguientes:

Registrar evento escolar

Este caso de uso permite registrar y programar eventos escolares en el calendario de la guardería. Estos eventos pueden incluir actividades especiales, celebraciones, salidas educativas, visitas de invitados, entre otros. Al registrar un evento escolar, se asigna una fecha y se proporciona información relevante como la descripción, los horarios y las personas involucradas. Esto facilita la planificación y organización de las actividades, permitiendo que tanto el personal docente como los padres de familia estén informados y preparados para el evento.

Registrar día inhábil

Este caso de uso permite registrar y marcar como día inhábil aquellos días en los que la guardería no estará operativa. Estos días pueden corresponder a feriados, días de descanso programados, mantenimiento o cualquier otra circunstancia que requiera que el centro esté cerrado. Al registrar un día inhábil, se bloquea esa fecha en el calendario de la guardería, evitando la asignación de actividades y la presencia de personal y niños en el centro. Esto permite una gestión adecuada de los recursos y una planificación acorde a los días disponibles de funcionamiento.

El subsistema de Gestión de días hábiles desempeña un papel crucial en la organización y planificación de la guardería "Guardería Burbujas". A través de los casos de uso de registrar eventos escolares y registrar días inhábiles, se establece un calendario que ayuda a mantener la coherencia y la eficiencia en el funcionamiento diario del centro. Esto garantiza que tanto el personal docente como los padres de familia estén al tanto de las actividades programadas y de los días en los que la guardería estará cerrada. En última instancia, este subsistema contribuye a crear un entorno estable y predecible para el desarrollo y cuidado de los infantes en la guardería.

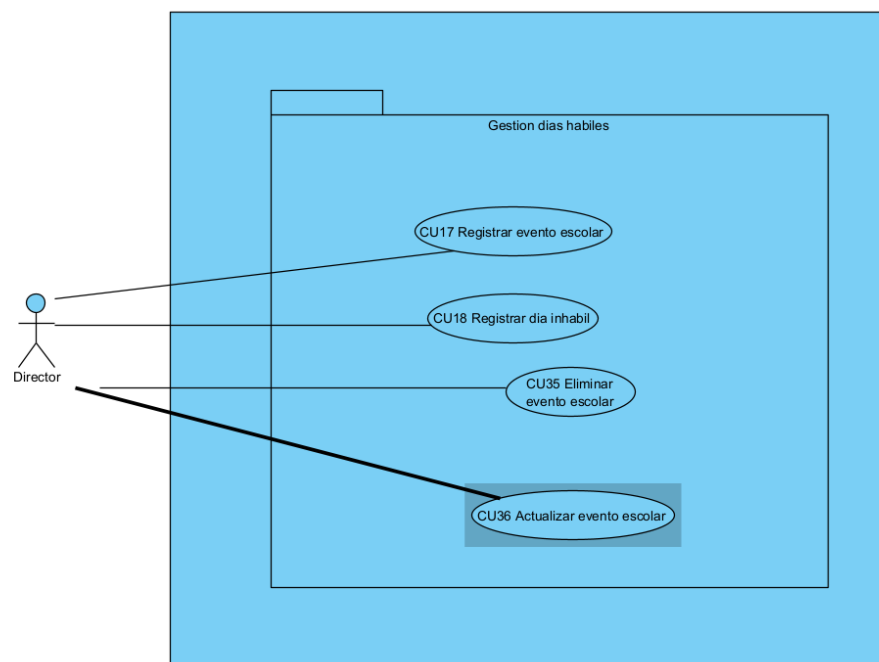


Figura 3.11: Diagrama del Subsistema de Gestion dias habiles.

3.1.5. Subsistema de Gestión clases

El subsistema de "Gestión de clases" es una parte fundamental del sistema de la guardería "Guardería Burbujas". Este subsistema se encarga de la administración y control de las actividades relacionadas con las clases impartidas a los niños en la guardería. El objetivo principal de este subsistema es garantizar un ambiente educativo adecuado y facilitar el proceso de enseñanza-aprendizaje.

El subsistema de Gestión de clases consta de los siguientes submódulos:

- **Asistencia:** Este módulo permite realizar el registro y seguimiento de la asistencia de los niños a las clases. El profesor puede marcar la asistencia de cada estudiante y mantener un registro actualizado de su asistencia diaria. Esto facilita el control de la asistencia y proporciona información valiosa para evaluar la participación y el progreso de los niños en las actividades escolares.
- **Reporte:** El módulo de Reporte permite generar informes y reportes sobre el desempeño y el progreso de los niños en las clases. El profesor puede ingresar datos relevantes, como calificaciones, observaciones y comentarios sobre el rendimiento de cada estudiante. Estos informes son útiles para evaluar el desarrollo de los niños, comunicarse con los padres y tomar decisiones educativas informadas.
- **Avisos:** El módulo de Avisos es una herramienta de comunicación entre el profesor y los padres. Permite enviar notificaciones, recordatorios y mensajes importantes relacionados con las clases y el progreso de los niños. Los profesores pueden compartir información relevante, como fechas de eventos, tareas o cualquier otra comunicación necesaria para mantener una buena comunicación con los padres y garantizar su participación activa en la educación de sus hijos.

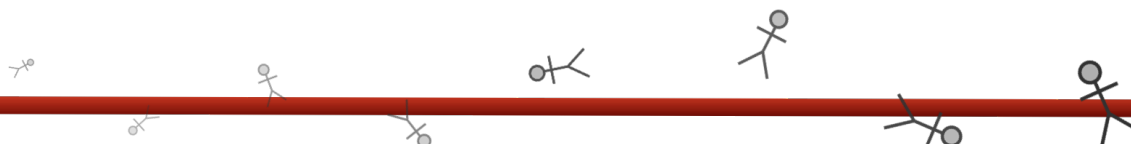
El actor principal que interactúa con el subsistema de Gestión de clases es el profesor. El profesor utiliza los diferentes módulos para tomar el control de las clases, registrar la asistencia de los niños, evaluar su desempeño y comunicarse de manera efectiva con los padres. A través de este subsistema, se busca promover un entorno educativo dinámico y propicio para el crecimiento y desarrollo de los niños en la guardería "Guardería Burbujas".

Submódulo de Asistencia

Este submódulo se encarga de realizar el registro y seguimiento de la asistencia de los niños a las clases, así como de registrar su salida al finalizar la jornada escolar. Su objetivo principal es mantener un control preciso y actualizado de la asistencia de los niños, lo que facilita la organización de las clases y proporciona información valiosa sobre su participación en las actividades escolares.

El submódulo de Asistencia ofrece las siguientes funcionalidades:

- **Tomar asistencia:** Esta funcionalidad permite al profesor tomar la asistencia de los niños al comienzo de cada clase. El profesor puede registrar la presencia o ausencia de cada estudiante, lo cual se refleja en el sistema. Esta información se utiliza para mantener un registro actualizado de la asistencia y generar informes precisos sobre la participación de los niños en las clases.
- **Registrar salida:** Al finalizar la jornada escolar, el profesor puede registrar la salida de los niños en el submódulo de Asistencia. Esto asegura que se tenga un registro completo de las horas de asistencia de cada niño y brinda a los padres la tranquilidad de saber que sus hijos han sido debidamente registrados al ingresar y salir de la guardería.



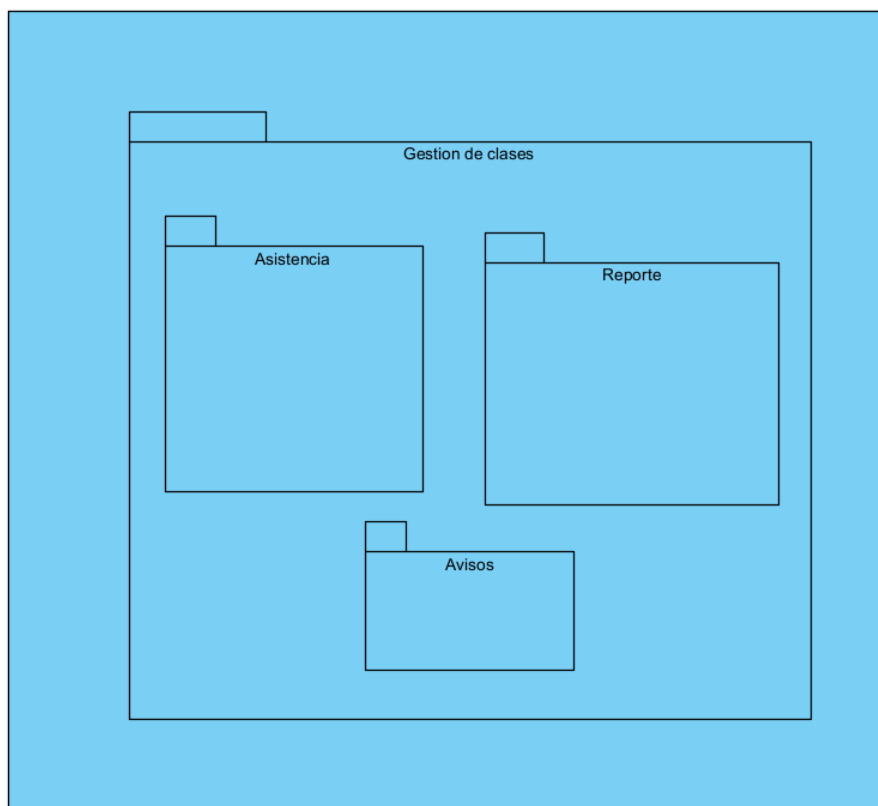


Figura 3.12: Diagrama del Subsistema de Gestion de Clases.

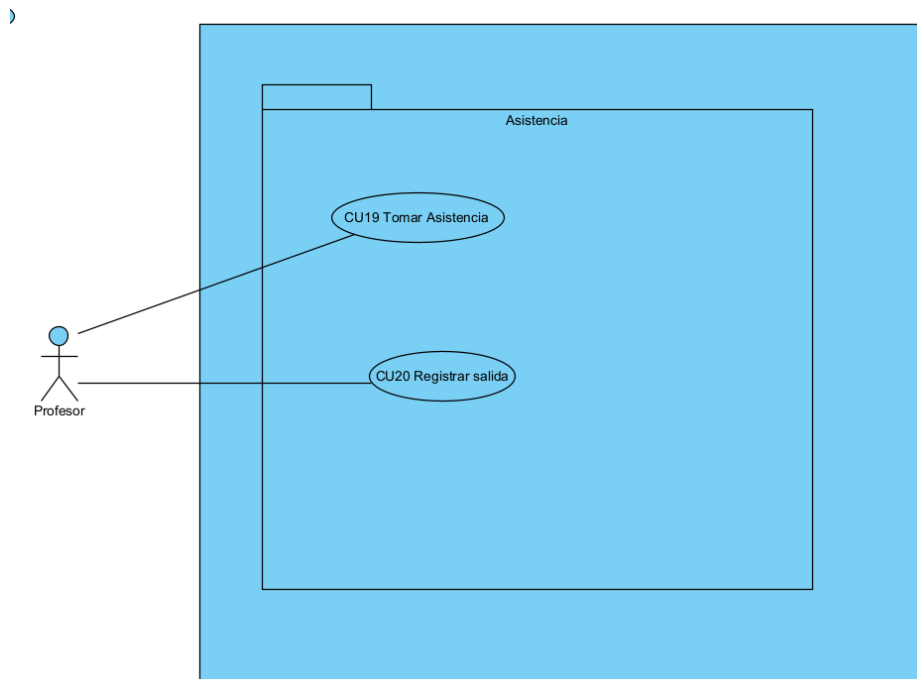


Figura 3.13: Diagrama del Submodulo de Asistencia.

Submódulo de Avisos

tiene como objetivo facilitar la comunicación entre el profesor y los padres de familia, brindando un canal eficiente para enviar y recibir avisos relevantes sobre actividades, eventos o cualquier otra información importante relacionada con la educación y el bienestar de los niños.

El submódulo de Avisos ofrece las siguientes funcionalidades:

- **Solicitar material:** Mediante esta funcionalidad, el profesor puede enviar avisos a los padres de familia solicitando materiales específicos para actividades escolares. Por ejemplo, si se va a realizar una manualidad, el profesor puede solicitar a los padres que envíen ciertos materiales como papel, pegamento, tijeras, entre otros. Esta funcionalidad agiliza el proceso de comunicación y permite al profesor asegurarse de contar con los recursos necesarios para llevar a cabo las actividades planificadas.
- **Contactar padre de familia:** Esta funcionalidad permite al profesor establecer comunicación directa con los padres de familia para tratar asuntos específicos relacionados con los niños. Puede ser utilizada para compartir información personalizada, responder preguntas o resolver cualquier inquietud que pueda surgir. La capacidad de contactar directamente a los padres de familia a través del submódulo de Avisos fomenta una comunicación fluida y eficiente entre ambas partes, promoviendo una colaboración activa en la educación y cuidado de los niños.

Submódulo de Reportes

El submódulo de Reportes es una parte esencial del subsistema de Gestión de clases en la guardería "Guardería Burbujas" se encarga de recopilar y registrar información relevante sobre diferentes aspectos relacionados con el cuidado y el desarrollo de los niños durante su estancia en la guardería.

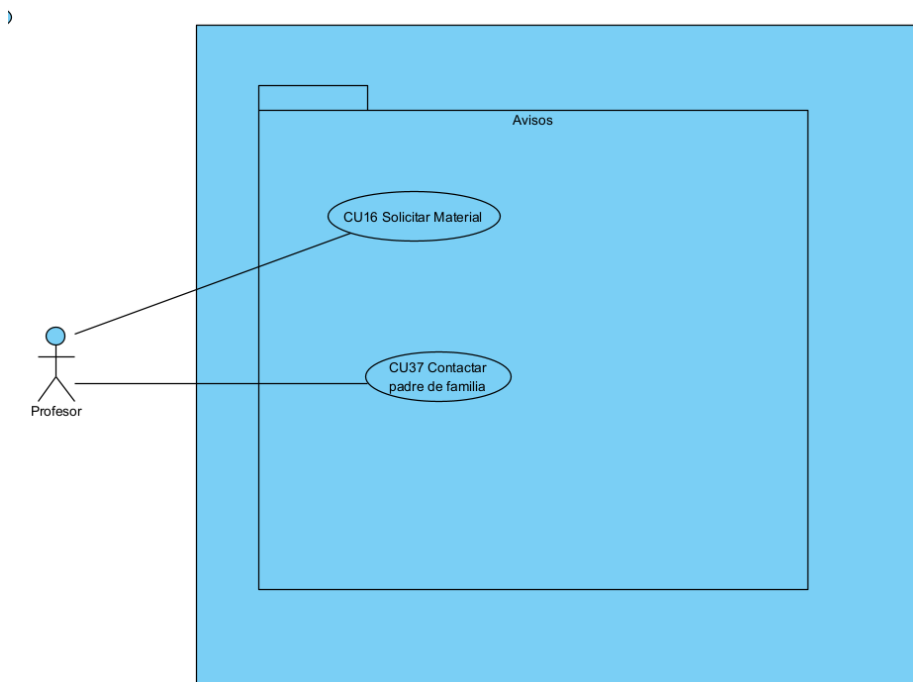


Figura 3.14: Diagrama del Submodulo de Avisos.

El submódulo de Reportes ofrece las siguientes funcionalidades:

- **Registrar evacuaciones:** Esta funcionalidad permite al profesor registrar y documentar las evacuaciones realizadas por cada niño. Se registra información como la fecha, hora y características de la evacuación, lo cual es importante para llevar un control adecuado de la salud y bienestar de los infantes.
- **Registrar ingesta:** Mediante esta funcionalidad, el profesor puede registrar la información sobre la ingesta de alimentos y bebidas de cada niño durante su estancia en la guardería. Se registra el tipo de alimento, la cantidad y cualquier otra observación relevante relacionada con la alimentación de los niños.
- **Registrar cambio de ropa:** Esta funcionalidad permite al profesor registrar cualquier cambio de ropa que deba realizarse a un niño durante el día. Se registra la fecha, hora y motivo del cambio de ropa, lo cual es útil para mantener un seguimiento adecuado de las necesidades individuales de los niños.
- **Registrar información adicional:** Esta funcionalidad permite al profesor registrar cualquier información adicional relevante sobre los niños. Puede incluir observaciones específicas, comportamientos destacados, eventos especiales o cualquier otra información que sea importante para comprender y atender las necesidades de los niños de manera individualizada.
- **Generar reporte:** Una vez registrada la información en los submódulos anteriores, el profesor puede utilizar esta funcionalidad para generar un reporte consolidado. El reporte contiene un resumen de la información registrada en cuanto a evacuaciones, ingesta, cambios de ropa, información adicional, tareas y actividades realizadas por los niños. Estos reportes pueden ser utilizados para compartir información con los padres de familia, evaluar el progreso de los niños y realizar un seguimiento adecuado de su desarrollo.

- **Registrar tareas y actividades:** Esta funcionalidad permite al profesor registrar las tareas y actividades que se realizan con los niños durante su estancia en la guardería. Se registra información como el tipo de tarea o actividad, la fecha, la duración y cualquier observación relevante. Esto facilita la planificación y organización de las actividades educativas y recreativas para el beneficio de los niños.

El submódulo de Reportes del subsistema de Gestión de clases juega un papel fundamental en la recopilación, registro y generación de información relevante sobre el cuidado y desarrollo de los niños en la guardería. Al utilizar este submódulo, se garantiza una documentación adecuada de las evacuaciones, la ingesta de alimentos, los cambios de ropa, la información adicional, las tareas y las actividades realizadas por los niños. Esto contribuye a la evaluación integral de su progreso, el seguimiento de sus necesidades individuales y la comunicación

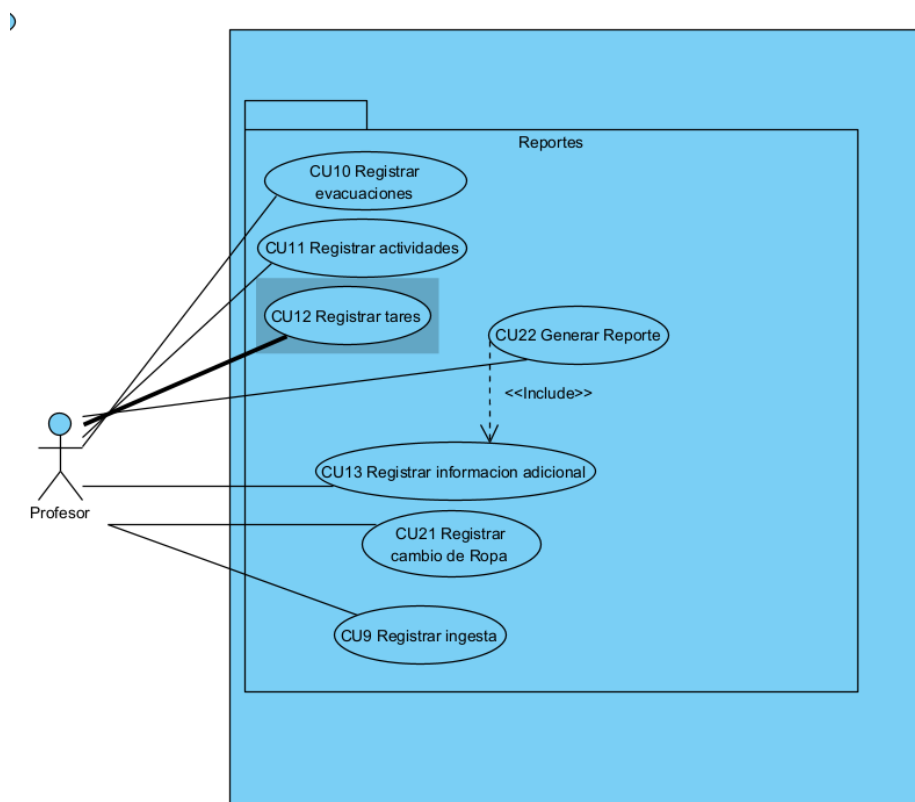


Figura 3.15: Diagrama del Submodulo de Reportes.

3.1.6. Subsistema Salud

El subsistema de Salud es una parte fundamental del sistema de la guardería "Guardería Burbujas". Este subsistema se encarga de garantizar el bienestar y la salud de los niños que asisten a la guardería. En este subsistema, se involucran dos actores principales: el Médico y el Nutriólogo.

El Médico es responsable de llevar a cabo acciones relacionadas con la salud de los niños. Esto incluye generar consultas médicas, evaluar el estado de salud de los infantes, realizar diagnósticos, prescribir medicamentos y llevar un registro de las incidencias médicas de cada niño. El Médico juega un papel crucial en el cuidado y la atención médica de los niños, asegurándose de que estén sanos y respondiendo de manera adecuada a cualquier problema de salud que puedan presentar.

Por otro lado, el Nutriólogo desempeña un papel importante en el aspecto nutricional de los niños. Su tarea principal es crear los menús balanceados y adecuados para cada niño, teniendo en cuenta sus necesidades dietéticas individuales, alergias, preferencias y restricciones alimentarias. El Nutriólogo se asegura de proporcionar una alimentación saludable y equilibrada, promoviendo hábitos alimentarios saludables y fomentando el bienestar nutricional de los niños.

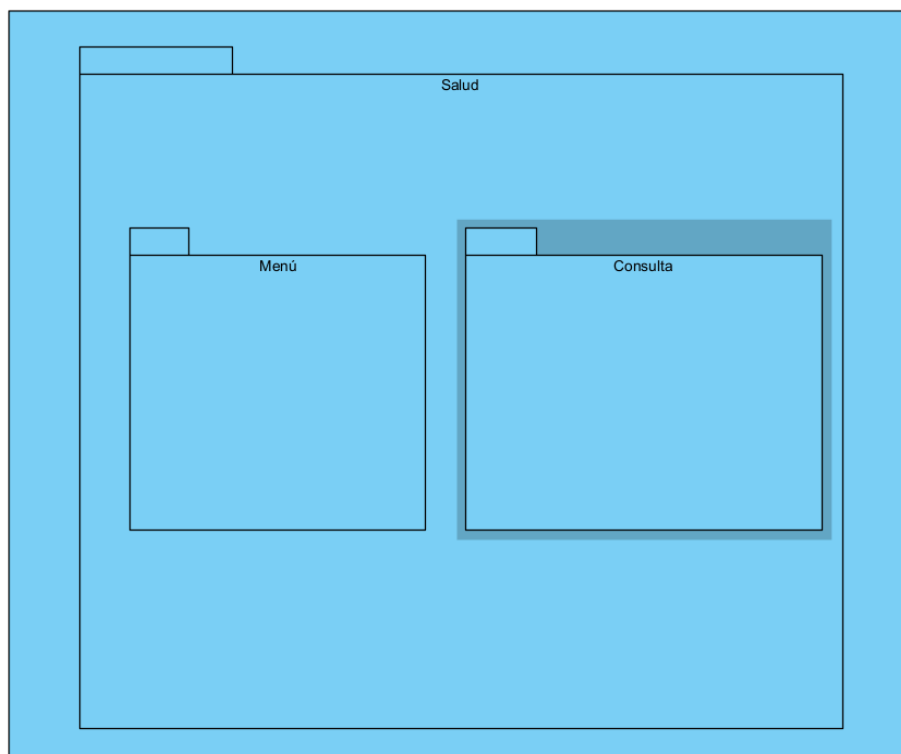


Figura 3.16: Subsistema Salud.

Submódulo de Menu

La subsección "Menú" dentro del subsistema de Salud se encarga de gestionar las acciones relacionadas con la creación y registro de menús para los niños en la guardería "Guardería Burbujas". Este módulo es fundamental para

garantizar una alimentación adecuada y equilibrada para los infantes, asegurando que se cumplan sus necesidades nutricionales.

El módulo del Menú permite realizar las siguientes acciones:

- * Registro de menú: En esta funcionalidad, el nutriólogo encargado puede registrar los menús diarios o semanales para los diferentes grupos de niños. El nutriólogo puede seleccionar los alimentos adecuados y planificar las comidas de acuerdo con las necesidades nutricionales y restricciones dietéticas de cada niño.
- * Creación de menú especial: En algunos casos, es posible que se requiera crear menús especiales para niños con necesidades dietéticas específicas, como alergias o intolerancias alimentarias. El nutriólogo puede diseñar menús especiales adaptados a las necesidades individuales de cada niño, garantizando su seguridad y bienestar.
- * Visualización de menú: El nutriologo pueden acceder a la visualización de los menús registrados. Esta funcionalidad les permite planear de antemano las comidas que se servirán a los niños en la guardería, lo que les brinda tranquilidad y les permite tomar decisiones informadas sobre la alimentación de los infantes

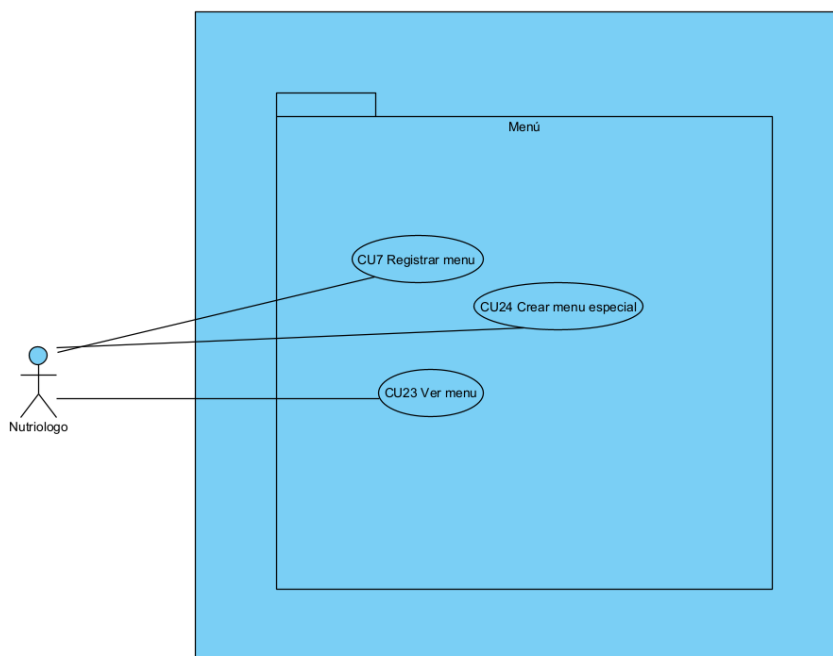


Figura 3.17: Modulo del Menu del subsistema Salud.

Submódulo de Consulta

La subsección Consulta dentro del subsistema de Salud se encarga de gestionar las acciones relacionadas con la atención médica y el registro de incidencias médicas en la guardería "Guardería Burbujas". Este módulo es fundamental para asegurar el cuidado y bienestar de los niños, proporcionando un seguimiento adecuado de su estado de salud.

El módulo de Consulta permite realizar las siguientes acciones:

- * Registro de incidencia médica: En esta funcionalidad, el médico de la guardería puede registrar y documentar cualquier incidencia o situación médica que ocurra con un niño. Esto incluye enfermedades, lesiones, alergias u otras condiciones de salud relevantes. El médico puede ingresar detalles sobre el diagnóstico, tratamiento, medicamentos recetados y cualquier recomendación adicional para el cuidado del niño.

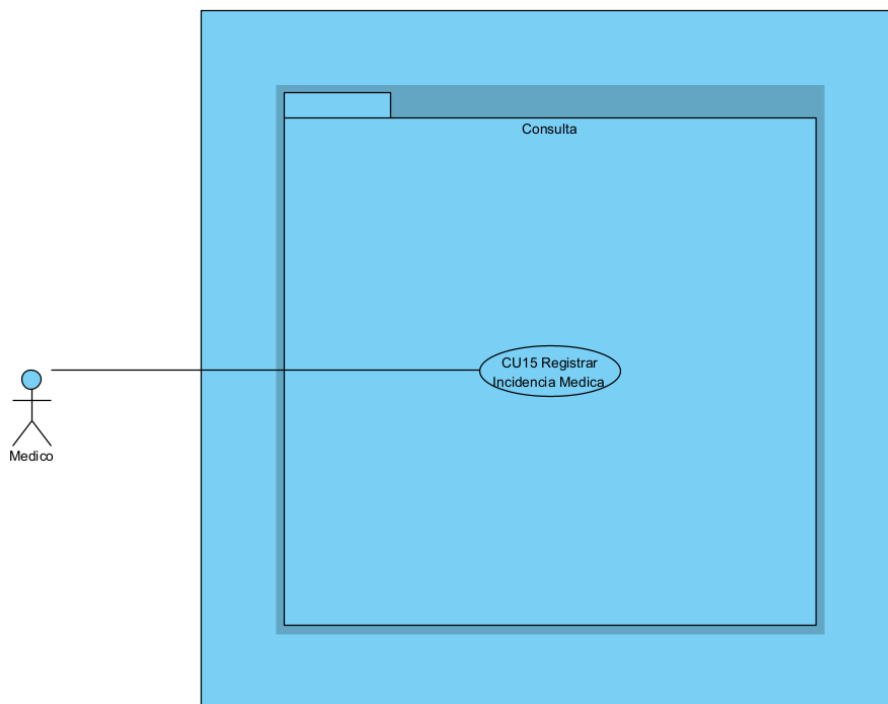


Figura 3.18: Modulo de Consulta del subsistema Salud.

3.2. Diseño de clases

El diseño de clases es una etapa esencial en el desarrollo de software basado en programación orientada a objetos (POO). En esta sección, nos centraremos en el diseño de clases relacionadas con diferentes entidades que forman parte de un sistema. Estas entidades incluyen a capital humano, los recursos humanos, los reportes, el menú de comidas, los padres de familia, los infantes, el director, el médico, el nutriólogo y el profesor.

En esta sección, abordaremos el diseño de clases para cada una de estas entidades, definiendo sus atributos, métodos y relaciones con otras clases. Un diseño de clases bien estructurado y coherente es fundamental para construir un sistema robusto y escalable, que cumpla con los requisitos y funcionalidades deseadas.

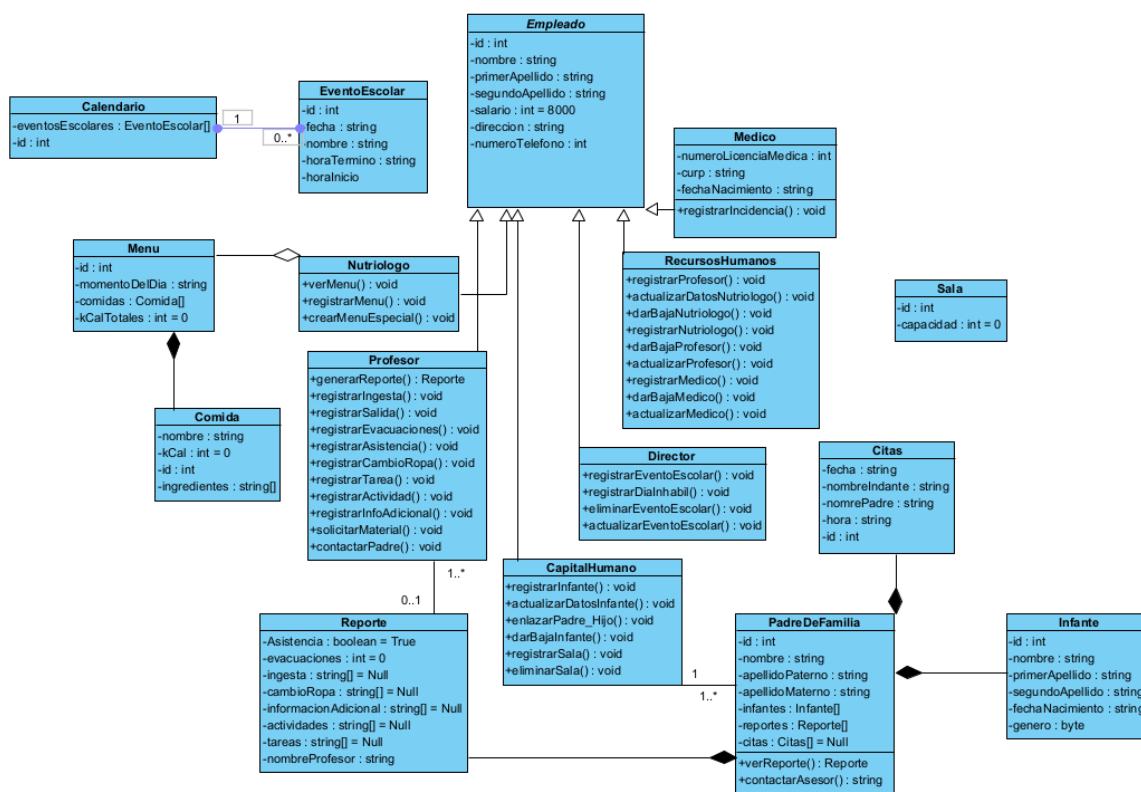


Figura 3.19: Diagrama de entidades del sistema.

3.2.1. Empleado

En esta subsección, se aborda el diseño de la clase abstracta que representa a un empleado dentro del sistema. Un empleado es una entidad fundamental en cualquier organización y puede tener diferentes roles y responsabilidades. A continuación, se definen los atributos principales de la clase abstracta *Empleado*.

La clase abstracta *Empleado* contiene los siguientes atributos:

- **nombre:** Representa el nombre del empleado, que puede incluir su primer nombre o nombres.
- **apellidoPaterno:** Indica el apellido paterno del empleado, que forma parte de su nombre completo.
- **apellidoMaterno:** Representa el apellido materno del empleado, que también forma parte de su nombre completo.
- **salario:** Indica la remuneración económica o sueldo que recibe el empleado por su trabajo.
- **direccion:** Representa la ubicación física o geográfica donde el empleado desempeña sus funciones dentro de la organización.
- **numeroTelefono:** Es el número de contacto telefónico del empleado, utilizado para comunicaciones y coordinación.
- **id:** Es un identificador único asignado al empleado dentro del sistema, que se utiliza para realizar operaciones y realizar seguimiento de su información.

La clase abstracta *Empleado* se utiliza como base para definir clases más específicas que representan diferentes tipos de empleados en el sistema, como *Director*, *Medico*, *Nutriologo* y otros. Al ser una clase abstracta, no se puede instanciar directamente, pero proporciona una estructura común y atributos compartidos para las clases derivadas.

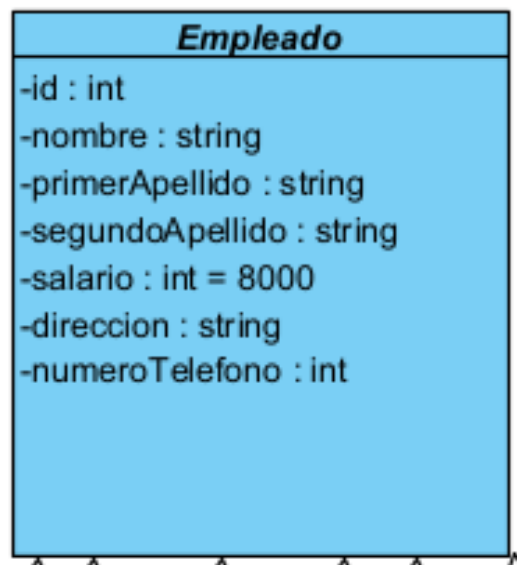


Figura 3.20: Diagrama de la clase abstracta empleado.

3.2.2. Nutriólogo

La entidad Nutriólogo, que es una subclase de la clase abstracta Empleado. El nutriólogo es un tipo específico de empleado que se especializa en la planificación y supervisión de la alimentación y nutrición de los infantes en la guardería "Guardería Burbujas".

La clase Nutriólogo hereda los atributos de la clase Empleado, que incluyen:

- **nombre:** el nombre del nutriólogo.
- **apellido paterno:** el apellido paterno del nutriólogo.
- **apellido materno:** el apellido materno del nutriólogo.
- **salario:** el salario del nutriólogo.
- **ubicación:** la ubicación del nutriólogo en la guardería.
- **número de teléfono:** el número de teléfono del nutriólogo.
- **id:** el identificador único del nutriólogo en el sistema.

La clase Nutriólogo también tiene los siguientes métodos:

- **verMenú():** este método permite al nutriólogo ver el menú registrado para los infantes en la guardería. Proporciona al nutriólogo una visión general de las comidas planificadas y les permite realizar ajustes si es necesario.
- **crearMenúEspecial():** este método permite al nutriólogo crear un menú especial para un infante con necesidades dietéticas específicas, como alergias o intolerancias alimentarias. El nutriólogo puede diseñar un menú adaptado a las necesidades individuales del infante, asegurando una alimentación segura y saludable.
- **registrarMenú():** este método permite al nutriólogo registrar el menú diario o semanal para los infantes en la guardería. El nutriólogo puede seleccionar los alimentos adecuados y planificar las comidas de acuerdo con las necesidades nutricionales y restricciones dietéticas de cada infante.

La clase Nutriólogo desempeña un papel fundamental en el subsistema de gestión de personal de la guardería "Guardería Burbujas". Con su experiencia y conocimientos en nutrición, contribuye a garantizar una alimentación adecuada y equilibrada para los infantes, promoviendo su bienestar y desarrollo saludable.

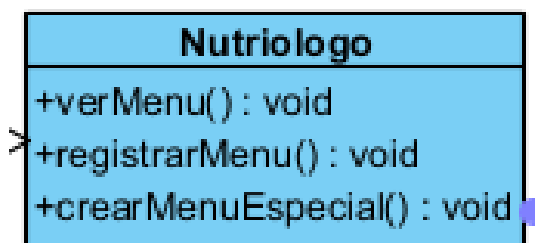


Figura 3.21: Diagrama de la clase anutriologo.

3.2.3. Medico

La entidad Médico, que es una subclase de la clase abstracta Empleado. El médico es un tipo específico de empleado que brinda atención médica y cuidado de la salud a los infantes en la guardería "Guardería Burbujas".

La clase Médico hereda los atributos de la clase Empleado, que incluyen:

- **nombre:** el nombre del médico.
- **apellido paterno:** el apellido paterno del médico.
- **apellido materno:** el apellido materno del médico.
- **salario:** el salario del médico.
- **direccion:** la ubicación del médico en la guardería.
- **número de teléfono:** el número de teléfono del médico.
- **id :** el identificador único del médico en el sistema.

Además, la clase Médico tiene el atributo adicional:

- **numeroLicenciaMedica:** el número de licencia médica del médico, que identifica su autorización legal para ejercer la medicina.

La clase Médico también tiene el siguiente método:

- **registrarIncidencia():** este método permite al médico registrar incidencias o eventos médicos relacionados con los infantes en la guardería. Puede incluir registros de enfermedades, lesiones o cualquier otra situación médica relevante. El médico puede proporcionar detalles y seguimiento adecuado para garantizar la salud y el bienestar de los infantes.

La clase Médico desempeña un papel crucial en el cuidado de la salud de los infantes en la guardería "Guardería Burbujas". Su experiencia médica y conocimientos permiten una atención adecuada en caso de enfermedad, lesiones o situaciones médicas imprevistas. Trabaja en estrecha colaboración con el personal y los padres de familia para garantizar un entorno seguro y saludable para los infantes.+

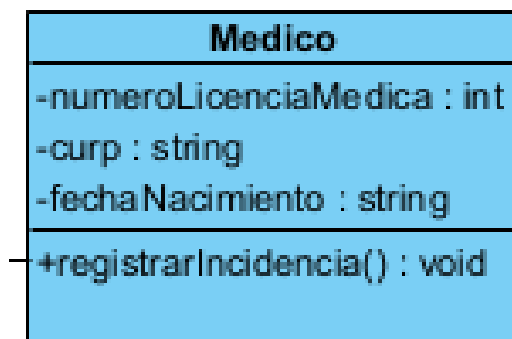


Figura 3.22: Diagrama de la clase del Medico.

3.2.4. Profesor

La entidad Profesor, que es una subclase de la clase abstracta Empleado. El profesor es un tipo específico de empleado que se encarga del cuidado y la educación de los infantes en la guardería "Guardería Burbujas".

La clase Profesor hereda los atributos de la clase Empleado, que incluyen:

- **nombre:** el nombre del profesor.
- **apellido paterno:** el apellido paterno del profesor.
- **apellido materno:** el apellido materno del profesor.
- **salario:** el salario del profesor.
- **direccion:** la ubicación del profesor en la guardería.
- **número de teléfono:** el número de teléfono del profesor.
- **id:** el identificador único del profesor en el sistema.

Además, la clase Profesor tiene los siguientes atributos adicionales:

- **registrarReporte:** este atributo representa la capacidad del profesor para registrar reportes sobre el progreso y el comportamiento de los infantes.
- **registrarIngesta:** este atributo indica la capacidad del profesor para registrar la ingesta de alimentos de los infantes durante las comidas.
- **registrarSalida:** este atributo permite al profesor registrar la hora de salida de los infantes al final del día.
- **registrarEvacuaciones:** este atributo representa la capacidad del profesor para registrar las evacuaciones o visitas al baño de los infantes.
- **registrarAsistencia:** este atributo indica la capacidad del profesor para registrar la asistencia de los infantes en la guardería.
- **registrarCambioRopa:** este atributo permite al profesor registrar los cambios de ropa de los infantes, por ejemplo, después de actividades o en caso de accidentes.
- **registrarTarea:** este atributo representa la capacidad del profesor para registrar las tareas asignadas a los infantes, como actividades de aprendizaje o tareas creativas.
- **registrarActividad:** este atributo indica la capacidad del profesor para registrar las actividades realizadas por los infantes, como juegos, ejercicios o proyectos.
- **registrarInfoAdicional:** este atributo permite al profesor registrar información adicional relevante sobre los infantes, como notas de comportamiento, preferencias o necesidades especiales.
- **solicitarMaterial:** este atributo representa la capacidad del profesor para solicitar material o recursos necesarios para las actividades y el cuidado de los infantes.
- **contactarPadre:** este atributo indica la capacidad del profesor para contactar a los padres de familia en caso de situaciones importantes, actualizaciones o consultas relacionadas con los infantes.

La clase Profesor desempeña un papel vital en el cuidado y la educación de los infantes en la guardería "Guardería Burbujas". Su responsabilidad incluye la supervisión, el apoyo y la instrucción de los infantes durante su estancia en la guardería. El profesor trabaja en estrecha colaboración con el personal y los padres de familia para garantizar el bienestar, el desarrollo y el aprendizaje de los infantes en un entorno seguro y estimulante.

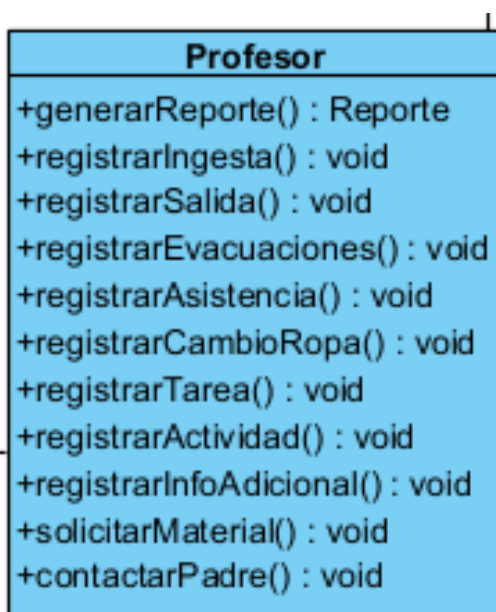


Figura 3.23: Diagrama de la clase del profesor.

3.2.5. Capital Humano

CapitalHumano, que es una subclase de la clase abstracta Empleado. La clase CapitalHumano se encarga de la gestión y administración de los recursos humanos relacionados con el cuidado de los infantes en la guardería "Guardería Burbujas".

La clase CapitalHumano hereda los atributos y métodos de la clase Empleado, y también tiene los siguientes métodos adicionales:

- **registrarInfante:** Este método permite registrar a un nuevo infante en el sistema. Se recopilan los datos necesarios del infante, como nombre, apellido, CURP, ubicación, edad, número de teléfono y ID en el sistema. Estos datos se utilizan para crear una nueva instancia de la clase Infante y agregarla al registro de infantes.
- **actualizarDatosInfante:** Este método permite actualizar los datos de un infante registrado en el sistema. Se identifica al infante por su ID en el sistema y se actualizan los atributos correspondientes, como nombre, apellido, CURP, ubicación, edad y número de teléfono.
- **enlazarPadre_Hijo:** Este método permite establecer una relación entre un padre de familia y su hijo en el sistema. Se recopilan los datos del padre de familia y del infante, y se realiza el enlace correspondiente en el registro de padres e infantes.
- **darBajaInfante:** Este método permite dar de baja a un infante del sistema. Se realiza mediante la identificación del infante por su ID en el sistema y eliminando su instancia correspondiente del registro de infantes.
- **registrarSala:** Este método permite registrar una nueva sala en el sistema. Se recopilan los datos necesarios de la sala, como nombre, capacidad y ubicación, y se crea una nueva instancia de la clase Sala para agregarla al registro de salas.
- **eliminarSala:** Este método permite eliminar una sala del sistema. Se realiza mediante la identificación de la sala por su ID en el sistema y eliminando su instancia correspondiente del registro de salas.

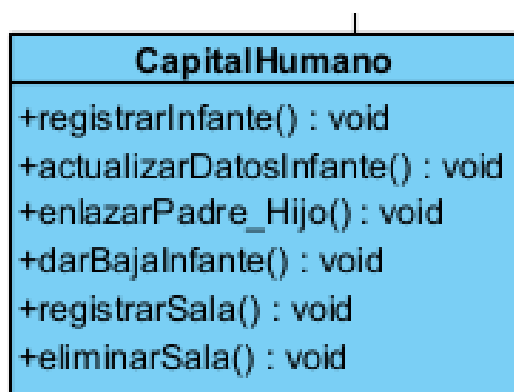


Figura 3.24: Diagrama de la clase del Capital humano.

3.2.6. Director

Director, que es una subclase de la clase abstracta Empleado. La clase Director representa al director de la guardería "Guardería Burbujasz se encarga de la gestión y supervisión general de la institución.

La clase Director hereda los atributos y métodos de la clase Empleado, y también tiene los siguientes métodos adicionales:

- **registrarEventoEscolar:** Este método permite registrar un evento escolar en el sistema. Se recopilan los datos del evento, como el nombre, la fecha, la descripción y cualquier otra información relevante. El evento se agrega al registro de eventos escolares y se asocia con los infantes y el personal correspondiente.
- **registrarDiaInhabil:** Este método permite registrar un día inhabil en el sistema. Se especifica la fecha y se agrega al registro de días inhabiles. Esto puede ser útil para planificar días en los que la guardería estará cerrada o no se ofrecerán servicios regulares.
- **eliminarEventoEscolar:** Este método permite eliminar un evento escolar del sistema. Se realiza mediante la identificación del evento por su ID en el sistema y eliminando su instancia correspondiente del registro de eventos escolares.
- **actualizarEventoEscolar:** Este método permite actualizar los datos de un evento escolar registrado en el sistema. Se identifica el evento por su ID en el sistema y se actualizan los atributos correspondientes, como el nombre, la fecha, la descripción y cualquier otra información relevante.

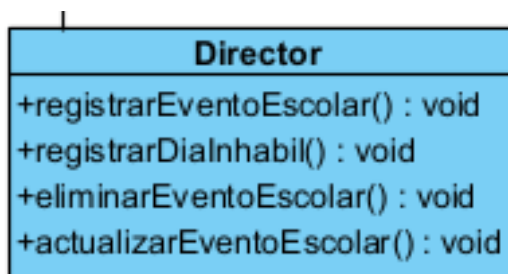


Figura 3.25: Diagrama de la clase director.

3.2.7. Recursos humanos

RecursosHumanos, que es una subclase de la clase abstracta Empleado. La clase RecursosHumanos se encarga de la gestión y administración del personal en la guardería "Guardería Burbujas".

La clase RecursosHumanos hereda los atributos y métodos de la clase Empleado, y también tiene los siguientes métodos adicionales:

- **registrarProfesor:** Este método permite registrar a un nuevo profesor en el sistema. Se recopilan los datos necesarios del profesor, como nombre, apellido, CURP, salario, ubicación, edad, número de teléfono y ID en el sistema. Estos datos se utilizan para crear una nueva instancia de la clase Profesor y agregarla al registro de profesores.
- **darBajaProfesor:** Este método permite dar de baja a un profesor del sistema. Se realiza mediante la identificación del profesor por su ID en el sistema y eliminando su instancia correspondiente del registro de profesores.
- **actualizarDatosProfesor:** Este método permite actualizar los datos de un profesor registrado en el sistema. Se identifica al profesor por su ID en el sistema y se actualizan los atributos correspondientes, como nombre, apellido, CURP, salario, ubicación, edad y número de teléfono.
- **actualizarDatosNutriologo:** Este método permite actualizar los datos de un nutriólogo registrado en el sistema. Se identifica al nutriólogo por su ID en el sistema y se actualizan los atributos correspondientes, como nombre, apellido, CURP, salario, ubicación, edad y número de teléfono.
- **registrarNutriologo:** Este método permite registrar a un nuevo nutriólogo en el sistema. Se recopilan los datos necesarios del nutriólogo, como nombre, apellido, CURP, salario, ubicación, edad, número de teléfono y ID en el sistema. Estos datos se utilizan para crear una nueva instancia de la clase Nutriologo y agregarla al registro de nutriólogos.
- **darBajaNutriologo:** Este método permite dar de baja a un nutriólogo del sistema. Se realiza mediante la identificación del nutriólogo por su ID en el sistema y eliminando su instancia correspondiente del registro de nutriólogos.
- **actualizarDatosMedico:** Este método permite actualizar los datos de un médico registrado en el sistema. Se identifica al médico por su ID en el sistema y se actualizan los atributos correspondientes, como nombre, apellido, CURP, salario, ubicación, edad y número de teléfono.
- **registrarMedico:** Este método permite registrar a un nuevo médico en el sistema. Se recopilan los datos necesarios del médico, como nombre, apellido, CURP, salario, ubicación, edad, número de teléfono y ID en el sistema. Estos datos se utilizan para crear una nueva instancia de la clase Medico y agregarla al registro de médicos.
- **darBajaMedico:** Este método permite dar de baja a un médico del sistema. Se realiza mediante la identificación del médico por su ID en el sistema y eliminando su instancia correspondiente del registro de médicos.

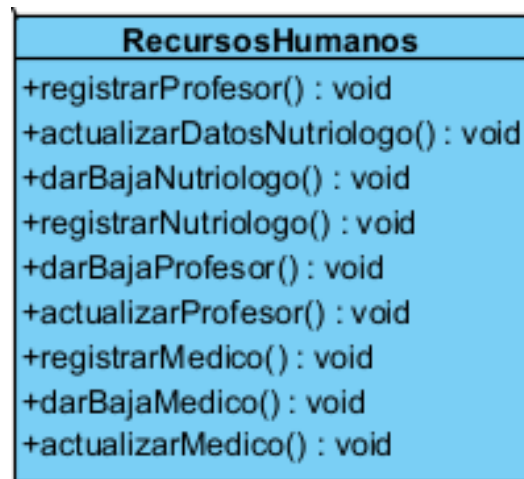


Figura 3.26: Diagrama de la clase recursos humanos.

3.2.8. Padre de familia

Padre de familia, que representa a los padres o tutores de los infantes inscritos en la guardería "Guardería Burbujas". Los padres de familia son parte integral del sistema y desempeñan un papel importante en la comunicación y el seguimiento del progreso de sus hijos.

La entidad Padre de familia tiene los siguientes atributos:

- **id**: Identificador único del padre de familia.
- **nombre, apellidos**: El nombre y apellidos del padre de familia.
- **infantes**: Una lista que contiene los infantes asociados a este padre de familia. Cada infante está representado por un objeto de la clase Infante.

Además, la entidad Padre de familia tiene los siguientes métodos:

- **verReporte**: Este método permite al padre de familia ver el reporte del progreso y desarrollo de sus infantes. El reporte puede incluir información sobre la asistencia, la alimentación, las actividades realizadas y cualquier otra observación relevante sobre el infante.
- **contactarAsesor**: Este método permite al padre de familia contactar al asesor o profesor encargado del cuidado y educación del infante. El padre de familia puede utilizar este método para realizar consultas, programar reuniones o comunicar cualquier inquietud o necesidad relacionada con su hijo.

Estos métodos facilitan la interacción y comunicación entre los padres de familia y el personal de la guardería, promoviendo una colaboración efectiva en el cuidado y desarrollo de los infantes. Los atributos de la entidad Padre de familia permiten mantener un registro y una asociación adecuada entre los padres y sus hijos inscritos en la guardería.

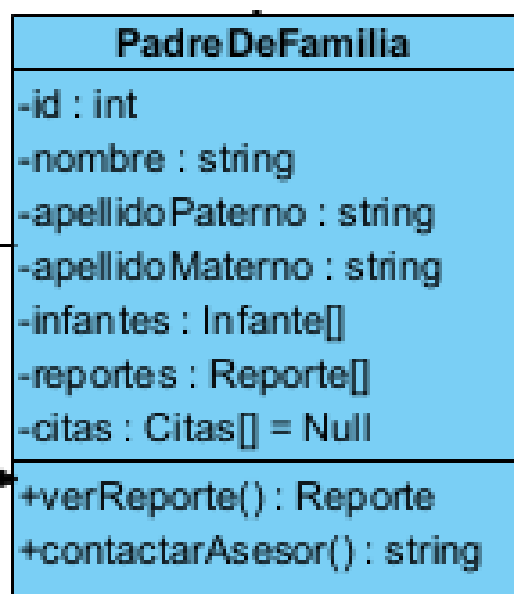


Figura 3.27: Diagrama de la clase padre de familia.

3.2.9. Infante

Infante, que representa a los niños inscritos en la guardería "Guardería Burbujas". Los infantes son el enfoque principal del sistema, ya que se centra en su cuidado, educación y bienestar.

La entidad Infante tiene los siguientes atributos:

- id: Identificador único del infante.
- nombre, apellidos: El nombre y apellidos del infante.
- fechaNacimiento: La fecha de nacimiento del infante.
- género: El género del infante, que puede ser masculino, femenino o cualquier otra categoría aplicable.

Además de estos atributos, la entidad Infante puede tener otros atributos adicionales según las necesidades específicas del sistema, como información médica relevante, alergias, requerimientos especiales, entre otros.

No se especifican métodos específicos para la entidad Infante en esta descripción, ya que los métodos y funcionalidades relacionadas con el cuidado y la gestión de los infantes pueden ser implementados en las clases que heredan de esta entidad, como el Profesor o el Padre de familia.

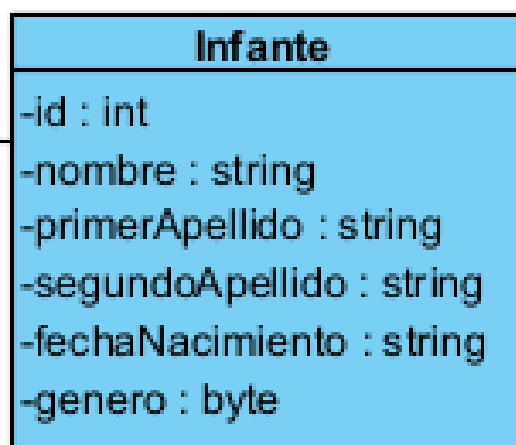


Figura 3.28: Diagrama de la clase infante.

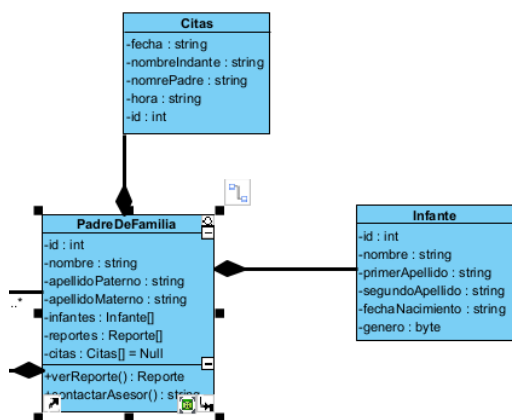


Figura 3.29: Diagrama de la relacion del padre de familia con infantes.

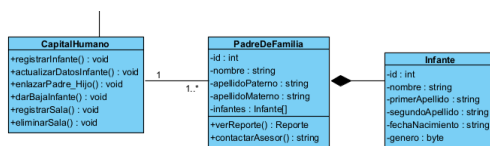


Figura 3.30: Diagrama de la relacion de capital humano con padre e infantes.

3.2.10. Reporte

Reporte, que representa un informe o registro de actividades y eventos relacionados con un infante en la guardería "Guardería Burbujas". Los reportes se utilizan para documentar diferentes aspectos del cuidado y el progreso del infante durante su estancia en la guardería.

La entidad Reporte tiene los siguientes atributos:

- asistencia: Registra la asistencia del infante a la guardería en un día específico.
- evacuaciones: Registra las evacuaciones realizadas por el infante durante su estancia en la guardería, incluyendo información relevante como la frecuencia y características.
- ingesta: Registra la información sobre la alimentación del infante, incluyendo la cantidad y tipo de alimentos consumidos.
- cambioRopa: Registra los cambios de ropa realizados al infante, por ejemplo, en caso de accidentes o necesidades de higiene.
- informacionAdicional: Permite registrar información adicional relevante sobre el infante, como observaciones especiales, necesidades particulares o eventos destacados.
- actividades: Registra las actividades en las que el infante ha participado durante su jornada en la guardería, como juegos, ejercicios o tareas educativas.
- tareas: Registra las tareas asignadas al infante para realizar en la guardería, como actividades de aprendizaje o responsabilidades específicas.
- nombreProfesor: Almacena el nombre del profesor o encargado responsable de registrar el reporte del infante.

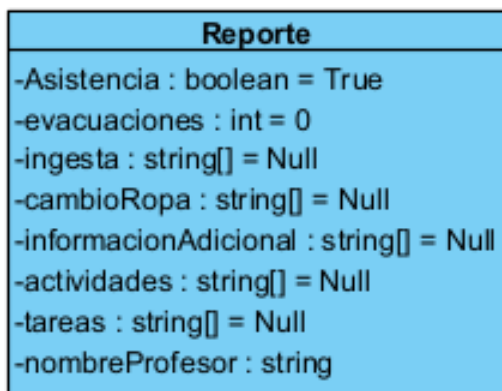


Figura 3.31: Diagrama de la clase reporte.

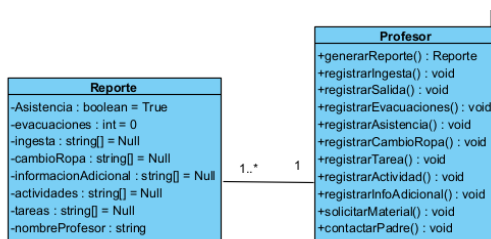


Figura 3.32: Diagrama de la relacion del profesor y los reportes.

3.2.11. Menu y comidas

Esta entidad Menú y se compone de la entidad Comida es importante para planificar y organizar las comidas ofrecidas a los infantes.

La entidad Menú tiene los siguientes atributos:

- id: Identificador único del menú.
- comidas: Una lista de comidas que conforman el menú. Cada comida está representada por un objeto de la entidad Comida.
- momentoDelDia: Indica el momento del día al que corresponde el menú, como desayuno, almuerzo o merienda.
- kCalTotales: El total de calorías estimadas para todas las comidas del menú.

La entidad Comida representa una comida específica que forma parte del menú. Tiene los siguientes atributos:

- id: Identificador único de la comida.
- nombre: Nombre de la comida.
- ingredientes: Lista de ingredientes utilizados en la preparación de la comida.
- kCal: Cantidad de calorías estimadas para la comida.

La entidad Menú se compone de varias instancias de la entidad Comida, lo que permite definir y organizar las opciones de alimentación para los infantes en la guardería. Cada comida tiene su propio conjunto de atributos, como nombre, ingredientes y calorías, lo que facilita la planificación de las comidas de acuerdo con los requisitos nutricionales y las preferencias alimenticias.

La entidad Menú y Comidas es esencial en el contexto de la guardería, ya que garantiza que se ofrezcan opciones de alimentación equilibradas y adecuadas para los infantes, promoviendo su salud y bienestar. Además, permite mantener un registro de los alimentos servidos y proporcionar información relevante a los padres de familia y al personal encargado de la alimentación.

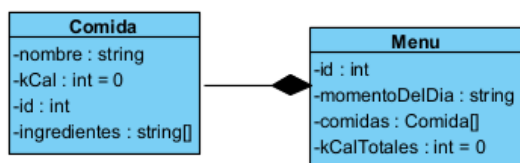


Figura 3.33: Diagrama de la clase de Menu y de clase Comidas.

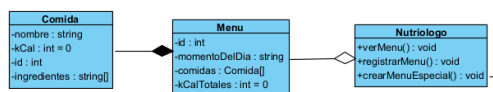


Figura 3.34: Diagrama de la relacion del menu con el nutriologo.

3.2.12. Calendario y Eventos escolares.

Para llevar un correcto manejo de los eventos escolares es que se usa un calendario, el cual se divide en dos entidades: La entidad "Eventos Escolares" tiene los siguientes atributos:

- **ID:** Identificador único del evento escolar.
- **Fecha:** Fecha en la que se llevará a cabo el evento escolar.
- **Hora de Inicio:** Hora de inicio del evento escolar.
- **Hora de Término:** Hora de término del evento escolar.
- **Nombre:** Nombre o descripción del evento escolar.

La entidad "Calendario" tiene los siguientes atributos:

- **ID:** Identificador único del calendario.
- **Eventos Escolares:** Un arreglo que almacena los eventos escolares programados en el calendario.

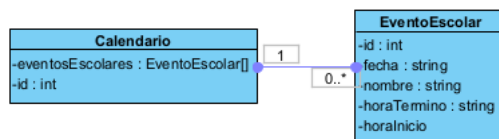


Figura 3.35: Diagrama de la relacion Eventos Escolares con calendario

3.2.13. Salas.

Las Salas son nuestras Salas en la guarderia, es por eso que es indispensable llevar un control de estas con su identificador y numero de capacidad. La entidad "Sala" tiene los siguientes atributos:

- **ID:** Identificador único del evento escolar.
- **Capacidad:** La capacidad maxima de esa sala

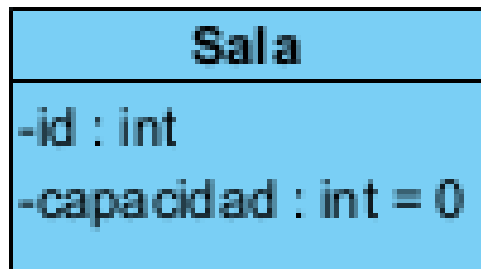


Figura 3.36: Diagrama de la entidad sala

El modelo dinámico es una parte fundamental en el diseño de sistemas de software, ya que se centra en capturar y representar el comportamiento y las interacciones del sistema a lo largo del tiempo. Este modelo nos permite comprender cómo los diferentes componentes del sistema interactúan entre sí y cómo se llevan a cabo las acciones y procesos. En el contexto del sistema de la "Guardería burbujas", el modelo dinámico nos ayudará a visualizar y analizar el flujo de eventos y el comportamiento en tiempo real del sistema.

El objetivo principal del modelo dinámico en el sistema de la "Guardería burbujas" es describir las secuencias de acciones y eventos que ocurren durante la ejecución del sistema. Esto implica identificar las interacciones entre los diferentes actores, como los usuarios y el sistema, y cómo se desarrollan las acciones a lo largo del tiempo. El modelo dinámico nos permite comprender cómo se realizan las transiciones entre los diferentes estados del sistema y cómo se gestionan los flujos de información y control.

Al diseñar el modelo dinámico del sistema de la "Guardería burbujas", se utilizan diferentes técnicas y diagramas para representar las secuencias de acciones y eventos. Uno de los diagramas más utilizados es el diagrama de secuencia, que muestra la interacción entre los diferentes objetos del sistema en función del tiempo. Este diagrama permite visualizar el orden y la duración de las acciones, así como las relaciones entre los objetos involucrados.

El modelo dinámico en el sistema es esencial para comprender y visualizar el comportamiento del sistema en tiempo real. Permite identificar los flujos de acciones y eventos, así como los procesos que se llevan a cabo durante la ejecución del sistema. Un diseño adecuado del modelo dinámico garantiza un sistema eficiente, confiable y orientado a las necesidades de los usuarios.

4.1. Diagrama de secuencia del CU1:Registrar profesor

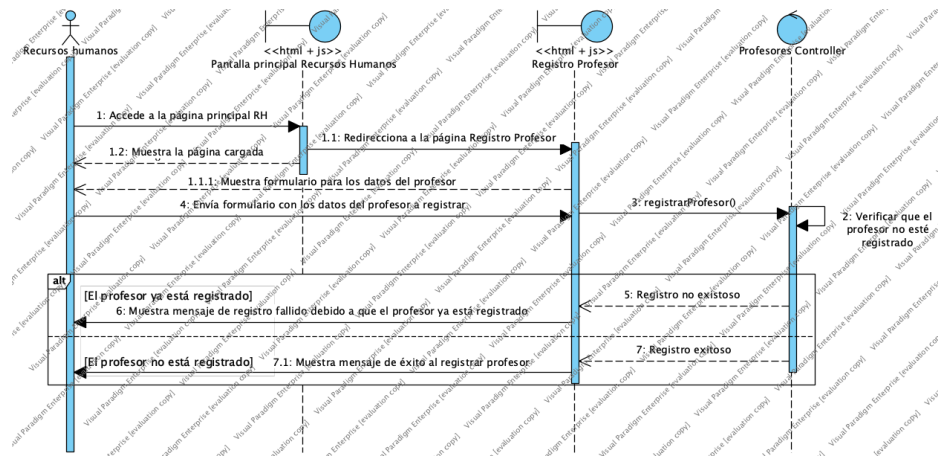


Figura 4.1: Diagrama de secuencia del CU1.

4.2. Diagrama de secuencia del CU4:Enlazar niño con padre

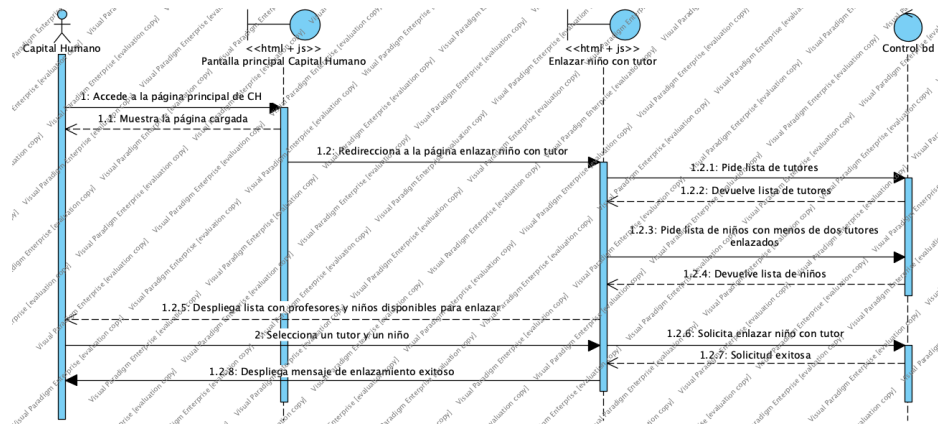


Figura 4.2: Diagrama de secuencia del CU4.

4.3. Diagrama de secuencia del CU5:Registrar profesor

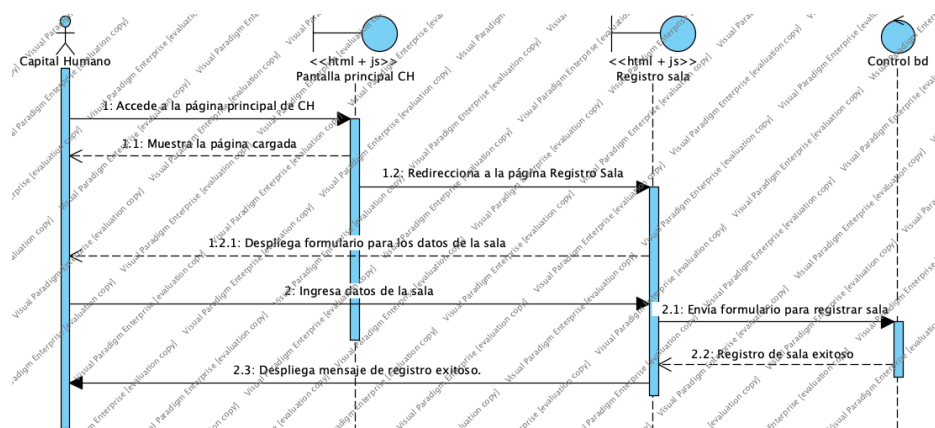


Figura 4.3: Diagrama de secuencia del CU5.

4.4. Diagrama de secuencia del CU7:Enlazar niño con padre

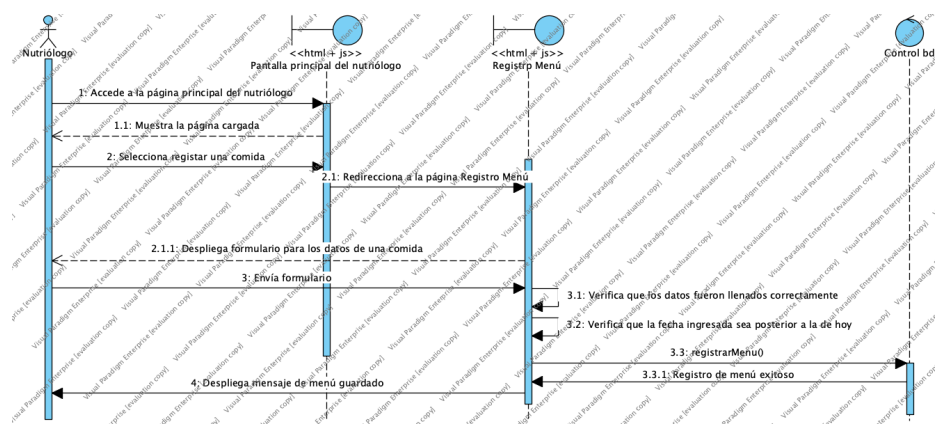


Figura 4.4: Diagrama de secuencia del CU7.

4.5. Diagrama de secuencia del CU9:Registrar ingesta

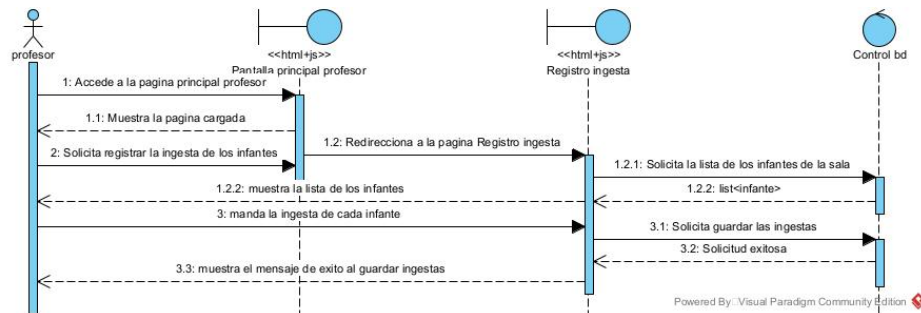


Figura 4.5: Diagrama de secuencia del CU9.

4.6. Diagrama de secuencia del CU12:Registrar tarea

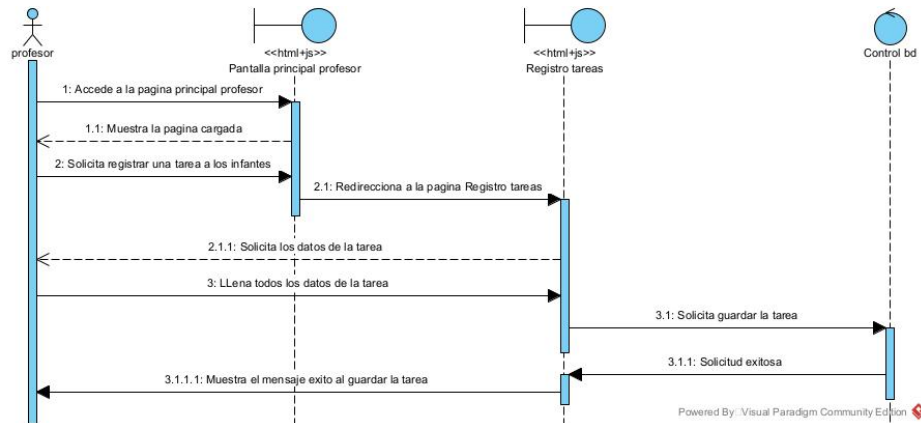


Figura 4.6: Diagrama de secuencia del CU12.

4.7. Diagrama de secuencia del CU15:Registrar incidencia medica

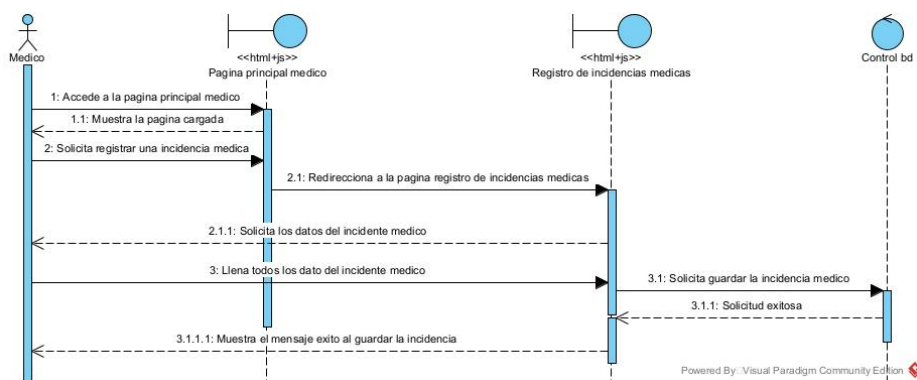


Figura 4.7: Diagrama de secuencia del CU15.

4.8. Diagrama de secuencia del CU19:Tomar asistencia

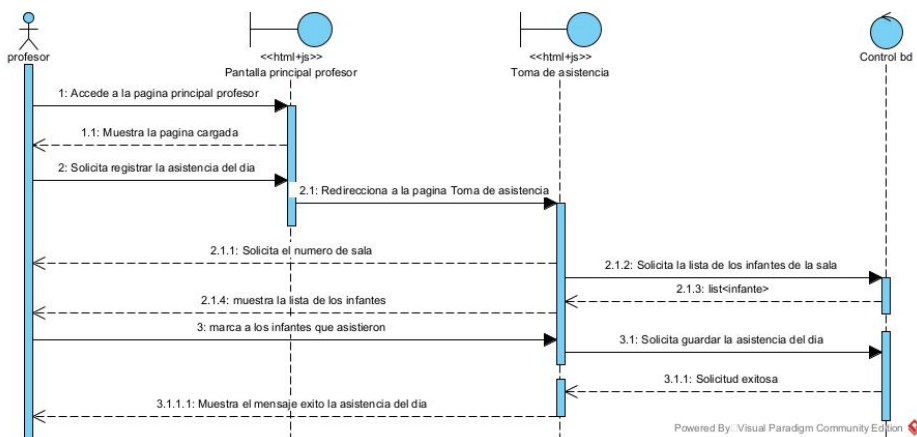


Figura 4.8: Diagrama de secuencia del CU19.

Modelo de manejo de la información.

El modelo de manejo de la información es un aspecto crucial en el diseño del sistema de la guardería "Guardería Burbujas". Este modelo se enfoca en cómo se gestiona y accede a la información dentro del sistema, especialmente a través de consultas a la base de datos. El manejo efectivo de la información es fundamental para garantizar un funcionamiento óptimo del sistema y satisfacer las necesidades de los usuarios.

En el contexto de la guardería "Guardería Burbujas", el modelo de manejo de la información se centra en cómo se almacenan y recuperan los datos relacionados con los niños, los padres, las actividades, los horarios y otros aspectos relevantes para la gestión de la guardería. Una base de datos es utilizada para almacenar esta información de manera estructurada y eficiente.

El modelo de manejo de la información se basa en el uso de consultas para acceder y manipular los datos almacenados en la base de datos. Las consultas son instrucciones o comandos que se envían a la base de datos para recuperar información específica de acuerdo a ciertos criterios o realizar operaciones de actualización en los datos.

En el sistema de la guardería "Guardería Burbujas", se utilizan consultas para diversas funcionalidades, como:

- Obtener la lista de niños y sus datos personales.
- Buscar los horarios de las actividades para un día determinado.
- Registrar la asistencia de los niños a las actividades.

Para realizar estas consultas, se utilizan lenguajes de consultas se usan funciones de **Mongoose** para la base de datos MongoDB que pueden ser usadas tambien en **CosmosDB**

El diseño adecuado del modelo de manejo de la información es esencial para garantizar un acceso eficiente y preciso a los datos en el sistema de la guardería "Guardería Burbujas". Se deben considerar aspectos como la estructura de la base de datos, la indexación de los datos, la optimización de consultas y la seguridad de la información.

En este capítulo, se presentarán las técnicas y consideraciones clave en el diseño del modelo de manejo de la información en el sistema de la guardería "Guardería Burbujas". Se analizará la estructura de la base de datos que son collecciones en base a las entidades previamente echas.

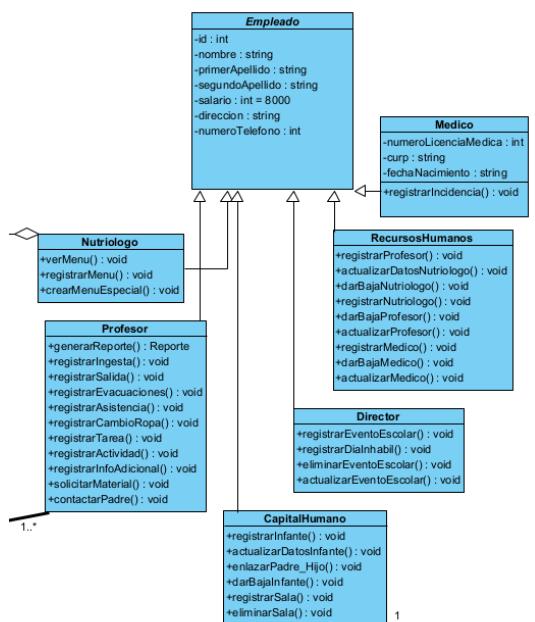


Figura 5.1: Diagrama de la coleccion de las entidades de los empleados

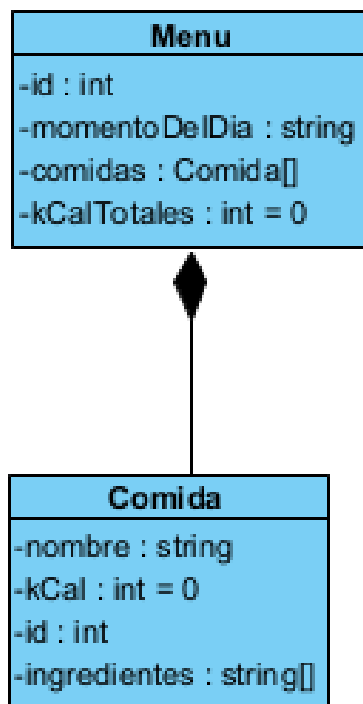


Figura 5.2: Diagrama de la coleccion del menu y las comidas

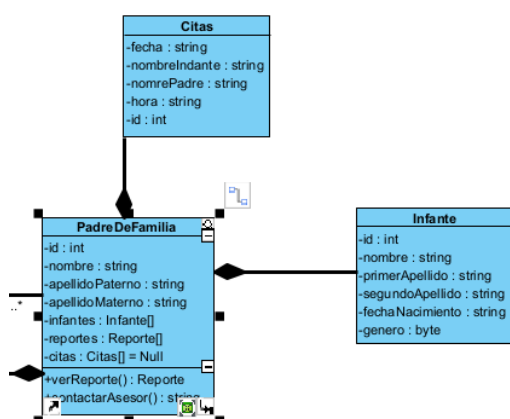


Figura 5.3: Diagrama de la coleccion deL padre de familia

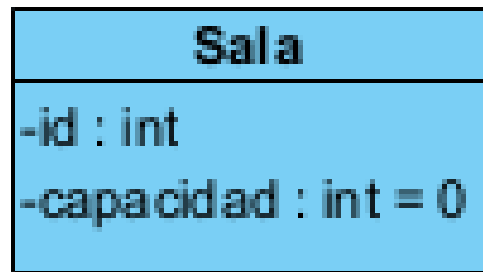


Figura 5.4: Diagrama de la coleccion de una sala

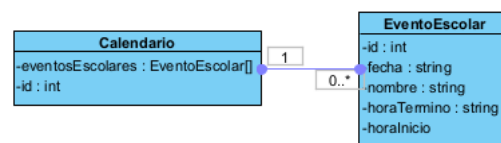


Figura 5.5: Diagrama de la coleccion deL calendario con Eventos Escolares

5.1. Consultas para el Caso de Uso CU1

- **Consulta001** usada para registrar un profesor

```
const nuevoProfesor = new Profesor({
  nombre: <nombreProfesor>,
  direccion: <direccionProfesor>,
  numeroTelefono: <numeroTelefonoProfesor>
});

nuevoProfesor.save();
```

5.2. Consultas para el Caso de Uso CU2

- **Consulta002** usada para registrar un padre de familia.

```
const nuevoPadre = new PadredeFamilia({
  nombre: <nombrePadre>,
  direccion: <direccionPadre>,
  numeroTelefono: <numeroTelefonoPadre>
});

nuevoPadre.save();
```

5.3. Consultas para el Caso de Uso CU3

- **Consulta003** usada para registrar un infante en el sistema.

```
const nuevoInfante = new Infante({
  nombre: <nombreInfante>,
  direccion: <direccionInfante>,
  fechaNacimiento: <fechaNacimientoInfante>
});

nuevoPadre.save();
```

5.4. Consultas para el Caso de Uso CU4

- **Consulta004** usada para enlazar un padre con su hijo en la guarderia

```
Padre.findOne({ id: <idPadre> }).exec((error, padre) => {
  if (error) {
    console.log(error);
  } else {
    padre.infantes.push(<infante>);
    padre.save();
  }
});
```

5.5. Consultas para el Caso de Uso CU5

- **Consulta005** llenado de datos de una sala para ser registrada

```
const nuevaSala = new Sala({
  capacidad: <numeroCapacidad>,
});

nuevaSala.save();
```

5.6. Consultas para el Caso de Uso CU6

- **Consulta006** registro de un nuevo nutriólogo

```
const nuevoNutriologo = new Nutriologo({
  nombre: <nombreNutriologo>,
  direccion: <direccionNutriologo>,
  numeroTelefono: <numeroTelefonoNutriologo>
});

nuevoNutriologo.save();
```

- **Consulta026** Eliminar nutriólogo

```
nutriologo = Nutriologo.findOne({nombre: <nombreNutriologo> });

Nutriologo.deleteOne({id: <nutriolog.id> });
```

5.7. Consultas para el Caso de Uso CU7

- **Consulta007** registro de comidas para el menu

```
const nuevoMenu = new Menu({
  momentoDelDia: <momentoDelDiaMenu>,
});

nuevoMenu.save();
```

```
const nuevaComida = new Comida({
  nombre: <nombreComida>,
  kCal: <numeroKCalorias>,
  ingredientes: <ingredientesComida>
});
```

```
nuevaComida.save();  
nuevoMenu.comidas.push(nuevaComida);  
nuevoMenu.save();
```

5.8. Consultas para el Caso de Uso CU8

- **Consulta008** generar Cita con tutor

```
const nuevaCita = new Cita({  
  nomreInfante: <nombreInfante>,  
  nomrePadre: <nombrePadre>,  
  fecha: <fechaCita>,  
  hora: <horaCita>,  
});  
nuevaCita.save();  
  
padre.citas.push(nuevaCita);  
padre.save();
```

5.9. Consultas para el Caso de Uso CU9

- **Consulta009** registrar ingesta en el reporte

```
reporteInfanteA.ingesta.push(<Ingesta>);  
reporteInfanteA.save();
```

5.10. Consultas para el Caso de Uso CU11

- **Consulta011** registrar actividades realizadas en el reporte

```
reporteInfanteA.actividades.push(<ActividadRealizada>);  
reporteInfanteA.save();
```

5.11. Consultas para el Caso de Uso CU12

- **Consulta012** registrar tareas en el reporte

```
reporteInfanteA.tareas.push(<tarea>);  
reporteInfanteA.save();
```

5.12. Consultas para el Caso de Uso CU14

- **Consulta014** registra medico en el sistema

```
const nuevoMedico = new Medico({
  nombre: <nombreMedico>,
  direccion: <direccionMedico>,
  numeroTelefono: <numeroTelefonoMedico>,
  numeroLicenciaMedica: <numeroLicenciaMedico>,
  curp: <curpMedico>,
  fechaNacimiento: <fechaNacimientoMedico>
});
```

```
nuevoMedico.save();
```

- **Consulta025** eliminar medico en el sistema

```
medico = Medico.findOne({nombre: <nombreMedico> });
```

```
Medico.deleteOne({id: <medico.id> });
```

5.13. Consultas para el Caso de Uso CU15

- **Consulta015** registrar incidencia medica en el reporte

```
incidencia = ["Incidencia medica:", <Detalles>, <Estado infante>]
reporteInfanteA.informacionAdicional.push(<incidencia>);
```

5.14. Consultas para el Caso de Uso CU17

- **Consulta017** registrar evento escolar

```
const nuevoEvento = new Evento({
  nombre: <nombreEvento>,
  fecha: <fechaEvento>,
  horaInicio: <horaInicioEvento>,
  horaTermino: <horaTerminoEvento>
});
nuevoEvento.save();
calendario.eventosEscolares.push(nuevoEvento);
calendario.save();
```

- **Consulta024** Eliminar evento escolar

```
calendario.updateOne(  
  { $pull: { eventosEscolares: { id: <nuevoEvento.id> } } }  
);
```

5.15. Consultas para el Caso de Uso CU19

- **Consulta019** Para que el profesor registre asistencia de los infantes

```
reporteInfanteA.asistencia = True;  
reporteInfanteA.save();
```

5.16. Consultas para el Caso de Uso CU22

- **Consulta022** El profesor registra el reporte del infante

```
const nuevoReporte = new Reporte({  
  asistencia: <asistenciaInfante>,  
  evacuaciones: <evacuacionesInfate>,  
  ingesta: <ingestaInfante>,  
  cambioRopa: <cambioRopaInfante>,  
  actividades: <actividadesInfante>,  
  tareas: <tareasInfante>,  
  fecha: Date.now  
});  
nuevoReporte.save();  
  
padre.reportes.push(nuevoReporte)
```