DOCUMENTO DE DISEÑO DE SOFTWARE

Aplicativo Web NCHms

Versión 1.0

Preparado por: Josue Peña Atencio
Mauricio Cortes Diaz
Jeffrey Steven García

Entregado a: Gustavo Salazar G.
Profesor

27 de Abril 2020

Introducción

Propósito

El propósito de este documento es presentar una descripción detallada de los diseños del Sistema Web para la Gestión de Historias Clınicas de Nutrición, nombrado ”NCHms” por el equipo de trabajo. Fue realizado para el curso “Procesos de ingeniería de Software” dictado en la Pontificia Universidad Javeriana Cali.

Por otra parte, este documento cumple con el papel de documentar (principalmente mediante los diagramas y diseños desarrollados) lo realizado por los estudiantes que conforman el grupo de trabajo para el proyecto de la materia. Este documento hace parte de los requisitos de entrega del proyecto. También, este documento es para el profesor, Gustavo Salazar, que asume el papel de cliente y profesor evaluador del proyecto.

En este documento se encuentra la información necesaria para proporcionar una descripción de los detalles, toma de decisiones y caminos a seguir a la hora de diseñar el sistema de software a ser implementado.

Por último, este documento podría usarse para diseñadores que intentan actualizar o modificar el diseño actual del sistema web de historias clínicas.

Alcance

Lo que se busca en las secciones de este documento es ilustrar a profundidad el diseño y arquitectura de software del Sistema Web para la Gestión de Historias Clınicas de Nutrición. Especifica la función, interfaces, estructura y el diseño de algunos de los módulos presentes en el diagrama de paquetes preliminar en el documento SRS redactado con anterioridad.

Se incorporan también algunos de los casos de uso que mediante diagramas secuenciales, de despliegue y de actividad. En los diagramas de clase se esclarece cómo el equipo de programación implementaría el módulo específico.

Adicionalmente, se anexa el enlace a la segunda versión del prototipo enseñado en el documento SRS, el cual cuenta con todas las vistas del sistema y el mapa de navegabilidad en su totalidad. Los patrones de diseño de interfaces de usuario y demás información pertinente están incluidas en el documento.

Definiciones, acrónimos y abreviaciones

Término	**Definición**
NCHms	Nutritional Clinical History management system
HC	Historia clínica

Referencias

[1] Salazar, G., Conferencia de clase, Tema: "*Diseño de Software I & II*", Pontificia Universidad Javeriana Cali, Abril 2020.

[2] Salazar, G., Conferencia de clase, Tema: "*Patrones de Diseño de Software*", Pontificia Universidad Javeriana Cali, Abril 2020.

[3] Wayner P., ”*The top 5 software architecture patterns: How to make the right choice*" TeachBeacon. [Online]. Disponible en: https://techbeacon.com/app-dev-testing/top-5-software-architecture-patterns-how-make-right-choice [Accedido 20 de Mayo 2020].

[4] Ashanin, N., ”*Quality attributes in Software Architecture*" Medium. [Online]. Disponible en: https://techbeacon.com/app-dev-testing/top-5-software-architecture-patterns-how-make-right-choice [Accedido 20 de Mayo 2020].

Representación Arquitectónica

Atributos de calidad

Como punto de partida en el desarrollo arquitectónico, el equipo de desarrollo decidió escoger los siguientes atributos de calidad según los requisitos no-funcionales establecidos en el documento SRS:

- Disponibilidad
- Rendimiento

Estos atributos satisfacen gran parte de los requisitos no funcionales. Estos son específicamente los requisitos con IDs: RNF-2-1, RNF-2-2, RNF-3-1 y RNF-3-2, los cuales especifican los tiempos de respuesta esperados de carga luego de alguna acción realizada por el usuario. El equipo de trabajo consideró que para este caso concreto, tener un sistema con buena disponibilidad (pocos tiempos inactividad o tiempos de inactividad programados) y tiempos de respuesta razonables (menor a los 3 segundos para la mayoría de eventos).

En los aspectos que la arquitectura final tendría puntos débiles es en el área de mantenibilidad. Si bien en los requisitos se tiene la necesidad de poder re-establecer el sistema mediante copias de seguridad automáticas (y esto siendo complemento a las decisiones de arquitectura en cuanto al atributo de disponibilidad), no fue el centro de atención a la hora de crear las arquitecturas.

Arquitecturas candidatas

A continuación se adjuntan los diagramas para las 3 arquitecturas candidatas exploradas

Diagrama de arquitectura Cliente-Servidor

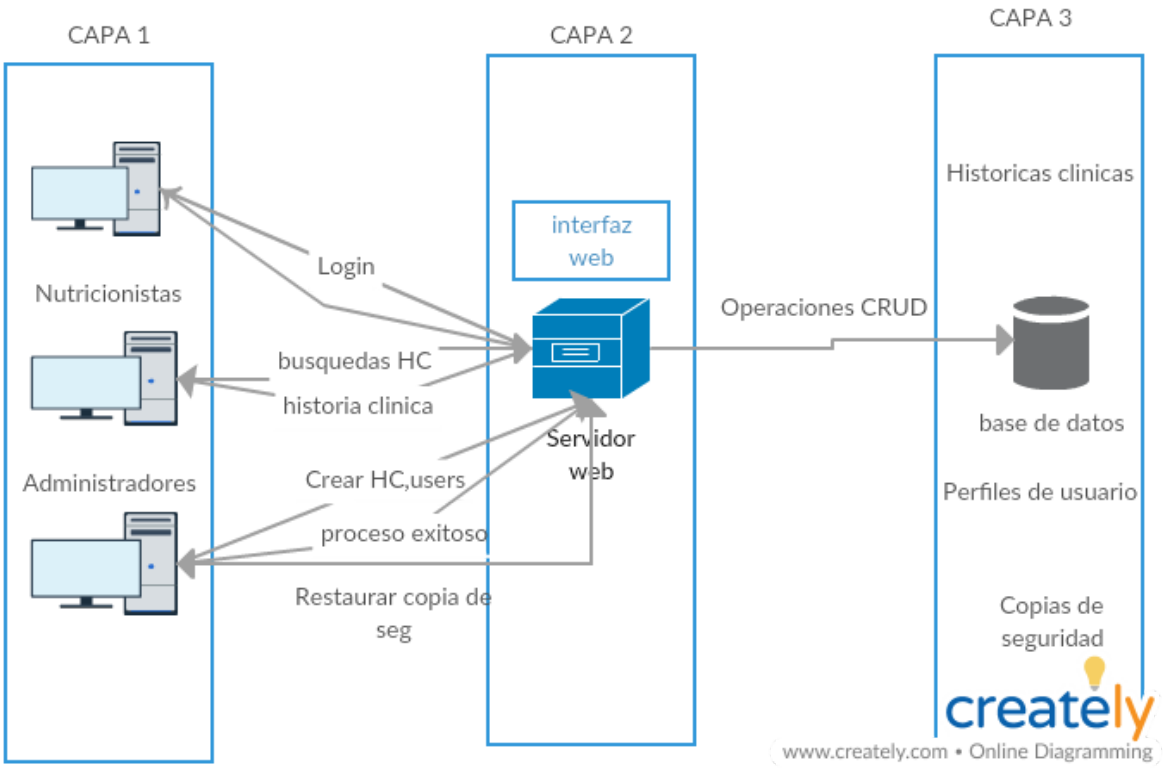


Diagrama patrón de diseño Modelo-Vista-Controlador

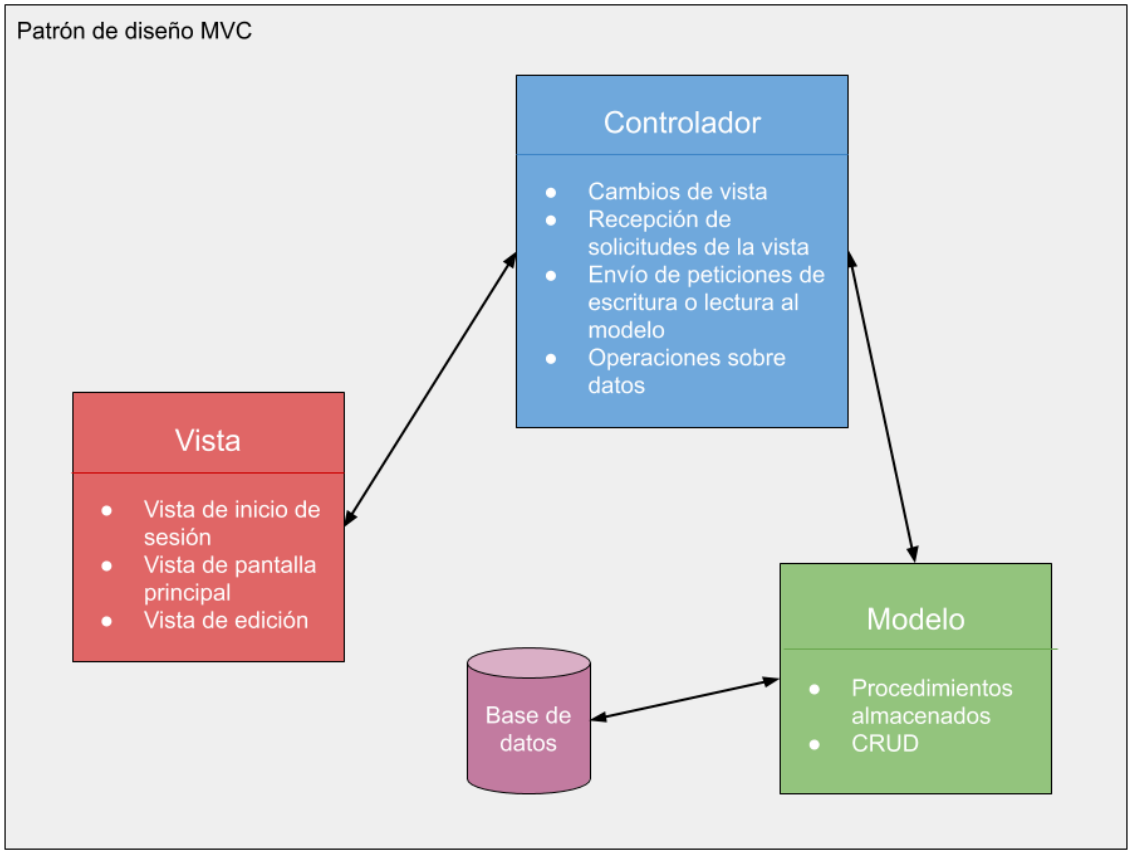
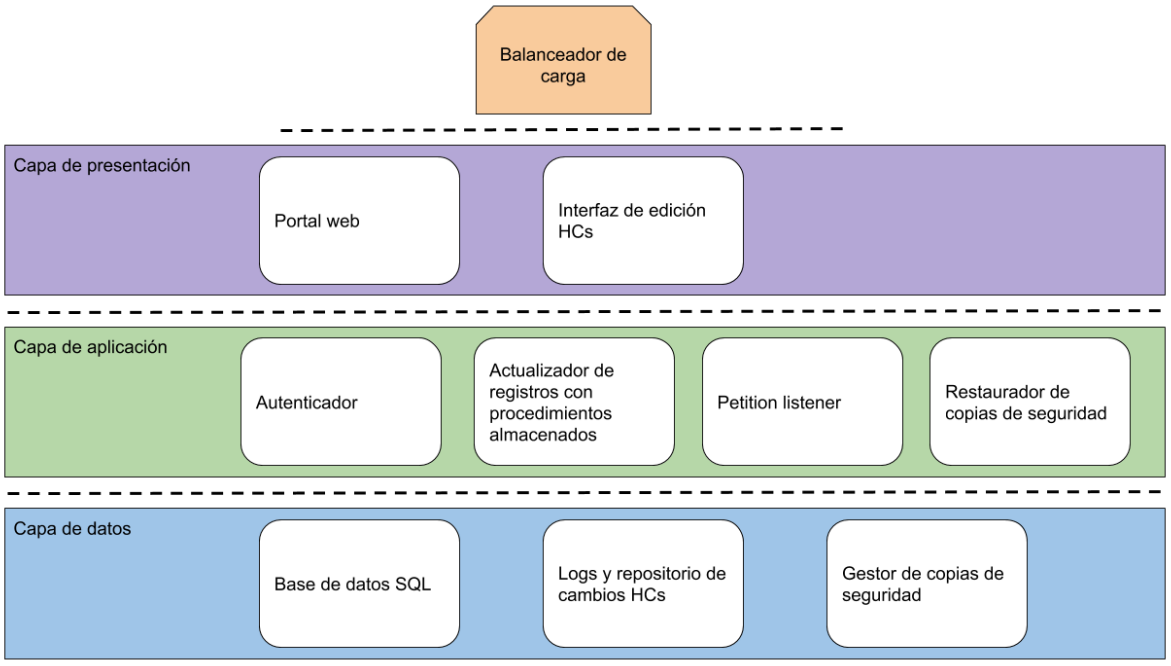


Diagrama de arquitectura por capas (3 capas)



En el siguiente cuadro se realiza la comparación entre las tres arquitecturas en cuanto a sus mayores fortalezas y debilidades.

Atributo de calidad	Capas	Modelo-Vista-Controlador	Cliente-Servidor
Disponibilidad	Al tener que trabajar con la comunicación entre los 3 componentes o capas, es difícil garantizar la disponibilidad del servicio.	Si se tienen componentes independientes de los demás se puede dar una solución rápida en el caso que ocurra una falla en alguno de los elementos de esta arquitectura	Si un servidor, que es el transmite los mensajes, falla, el cliente puede buscar reconexión con otro servidor.
Rendimiento	Al tener que interactuar con tantas capas para conseguir un resultado, el rendimiento se puede ver afectada en el tiempo.	Con el menor overhead de comunicación que tiene esta arquitectura y por la independencia de sus elementos se puede afirmar que su rendimiento es alto	El rendimiento puede verse afectado por factores propios de la red y fallas o retardos en las solicitudes y peticiones.

Según las deliberaciones realizadas con el grupo de diseño de software se llegó a la conclusión de que la arquitectura por capas sería la elegida para realizar el proyecto de Gestión de Historias Clínicas de Nutrición, aunque con ciertas modificaciones.

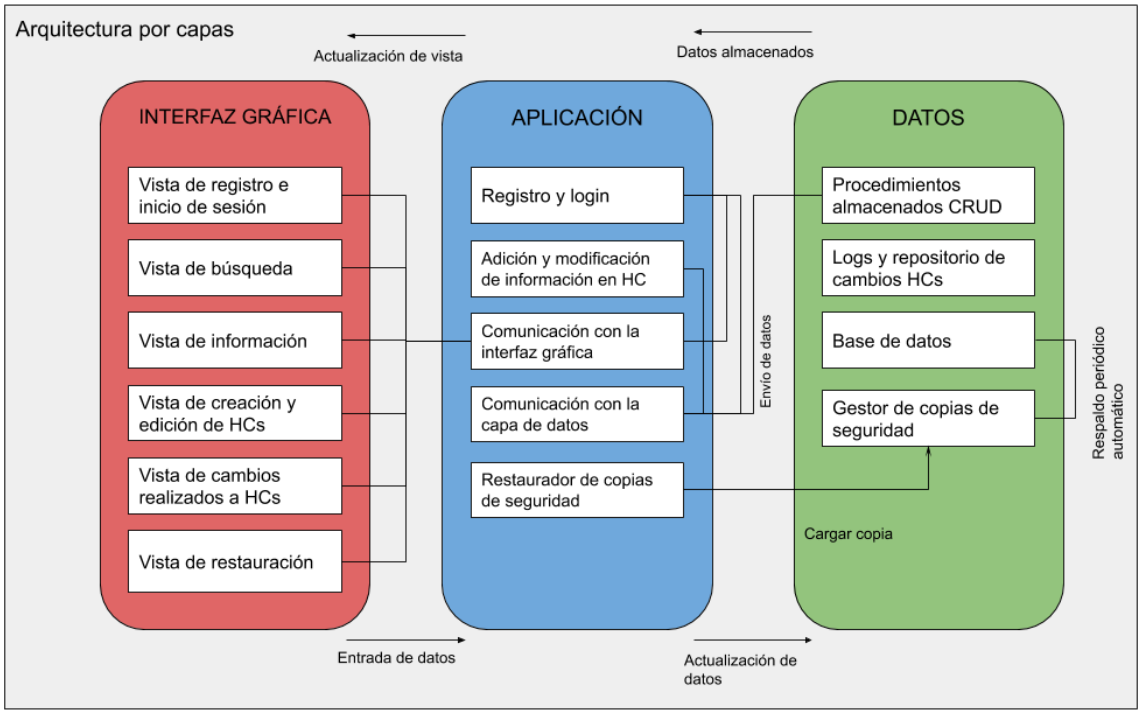
Las razones por las que se escogió esta arquitectura recaen en la comparación con las otras dos arquitecturas propuestas, donde ésta nos proporciona la mayor flexibilidad en el diseño que las otras propuestas que siguen patrones más marcados. Además de eso, ésta consideramos que presenta el mejor equilibrio entre disponibilidad y rendimiento.

Brinda mejor disponibilidad por el margen de operatividad que brinda a los equipos de mantenimiento, permitiendo que el sistema pueda recuperarse rápidamente mediante el sistema de restauración mediante copias de seguridad. De esta forma los problemas con la base de datos podrían revertirse en lugar de ser resueltos de forma manual. Por otro lado, el rendimiento no está sujeto totalmente a la red sino a la comunicación global de la infraestructura y arquitectura física que se posea para el sistema.

Un aspecto que se decidió cambiar de la arquitectura por capas candidata fue la existencia de la capa de balanceo de carga. Si bien esta trae beneficios con respecto a ambos atributos escogidos, su complejidad en términos de implementación es muy elevada para nuestro nivel de familiaridad con el conocimiento necesario para llevar tal tarea a cabo en lo que resulta del proyecto.

A continuación se adjunta el diagrama de la arquitectura por capas con mayor nivel de detalle, al igual que las correcciones y modificaciones realizadas por el equipo.

Diagrama detallado



Objetivos y Restricciones de la Arquitectura

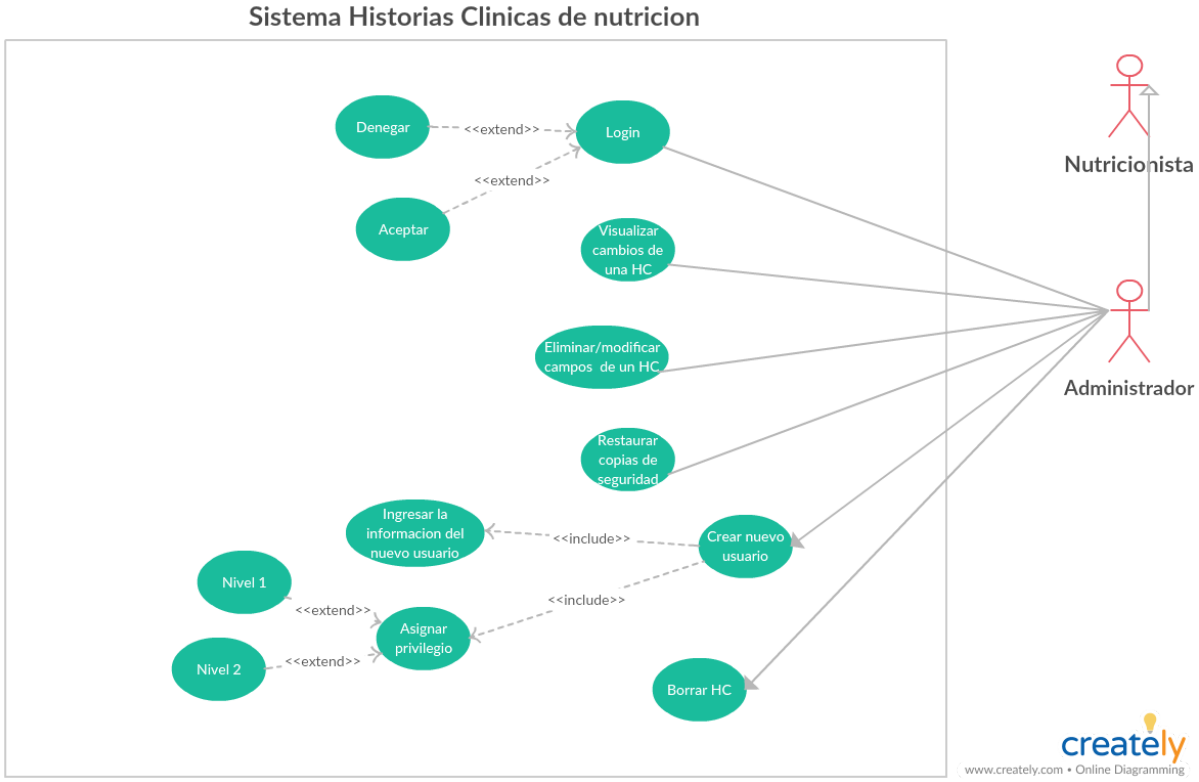
Con la arquitectura escogida se tiene como objetivo que cumpla con los atributos de calidad elegidos anteriormente, principalmente el de disponibilidad (gracias a los componentes de

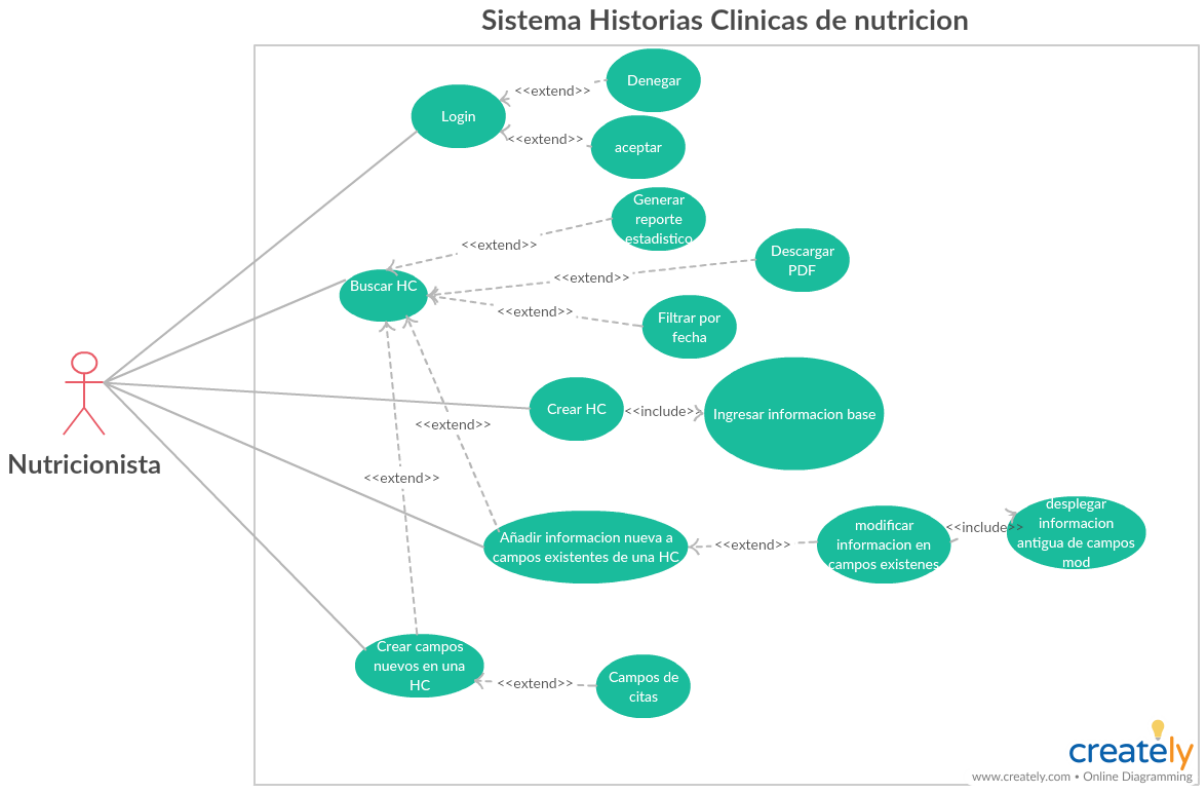
copias de seguridad y logs, los cuales permiten que el sistema pueda ser restaurado de una forma más rápida). La arquitectura tiene un nivel de complejidad reducido tratando de minimizar el overhead de comunicación entre las capas, de tal forma que el atributo de rendimiento puede ser satisfecho en la mejor medida.

En cuanto a limitaciones, esta arquitectura no está pensada para ser escalable. La adición de equipos de cómputo y demás infraestructura no garantiza una escalabilidad en cuanto a los atributos de calidad escogidos. El sistema podría desempeñarse peor en cuanto a rendimiento debido a overheads de comunicación, e incluso tener menos disponibilidad ya que la existencia de más dispositivos añade mayor propensión a los fallos y necesidad de mantenimiento. Esto sin mencionar que las funcionalidades de copias de seguridad tendrían que ser rediseñadas para poder distribuir un volumen de datos abrumante entre diferentes servidores y equipos.

Vista de Casos de Uso

La documentación pertinente para cada caso de uso se encuentra indicada en la sección de anexos.





Vista Lógica

En esta sección se incluyen los diagramas de paquetes correspondientes a cada capa de la arquitectura detallada.

Diagrama de Paquetes para Capa de Aplicación

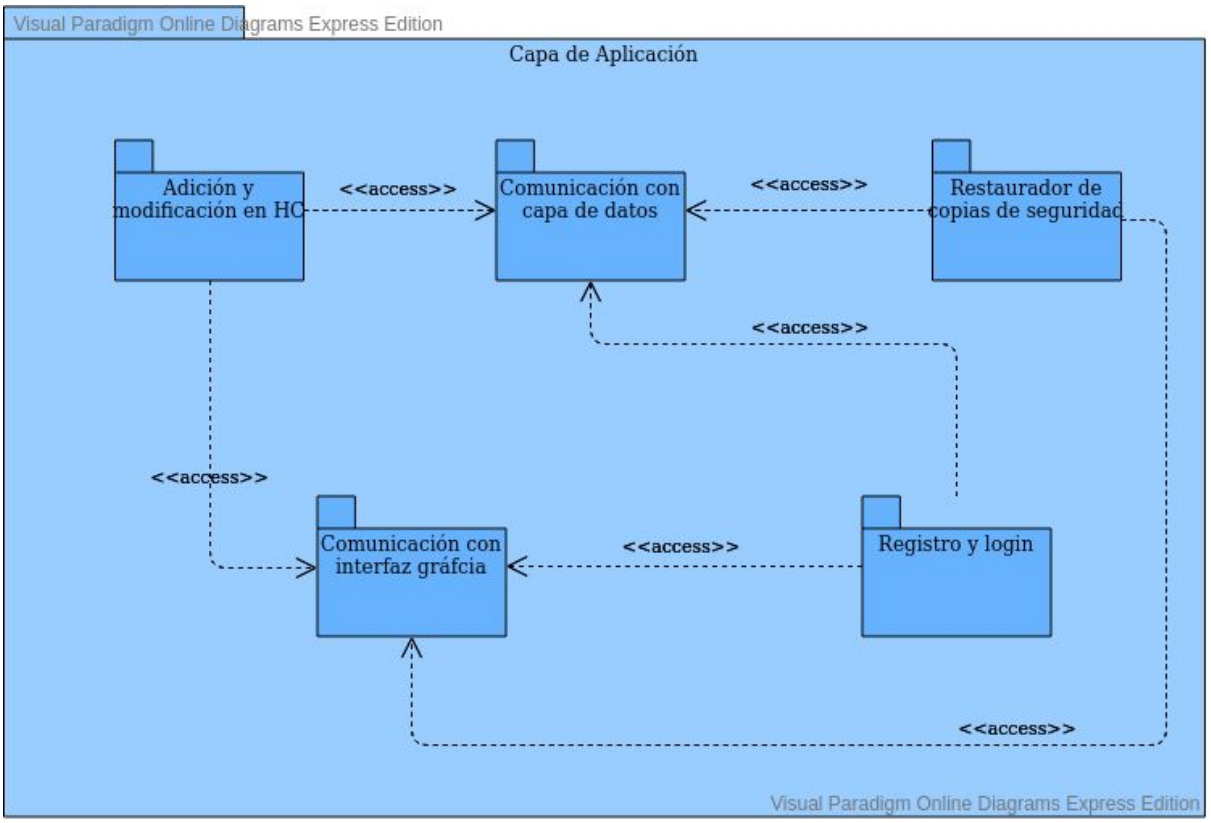


Diagrama de Paquetes para Capa de Interfaz Gráfica o de Presentación

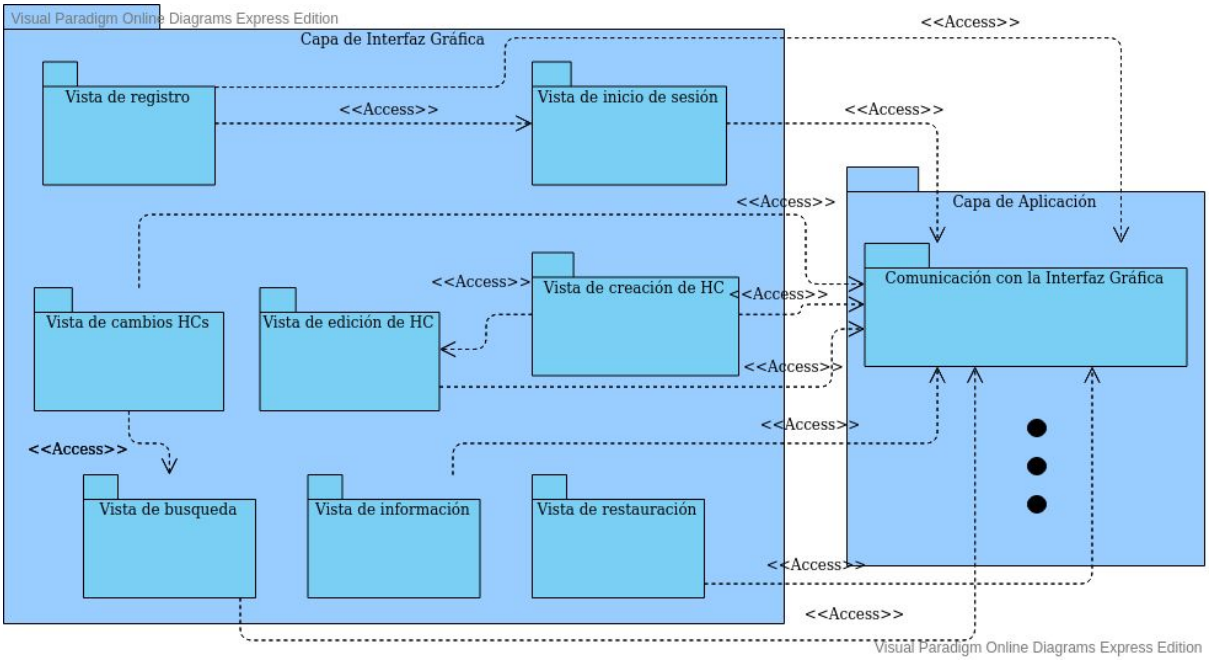
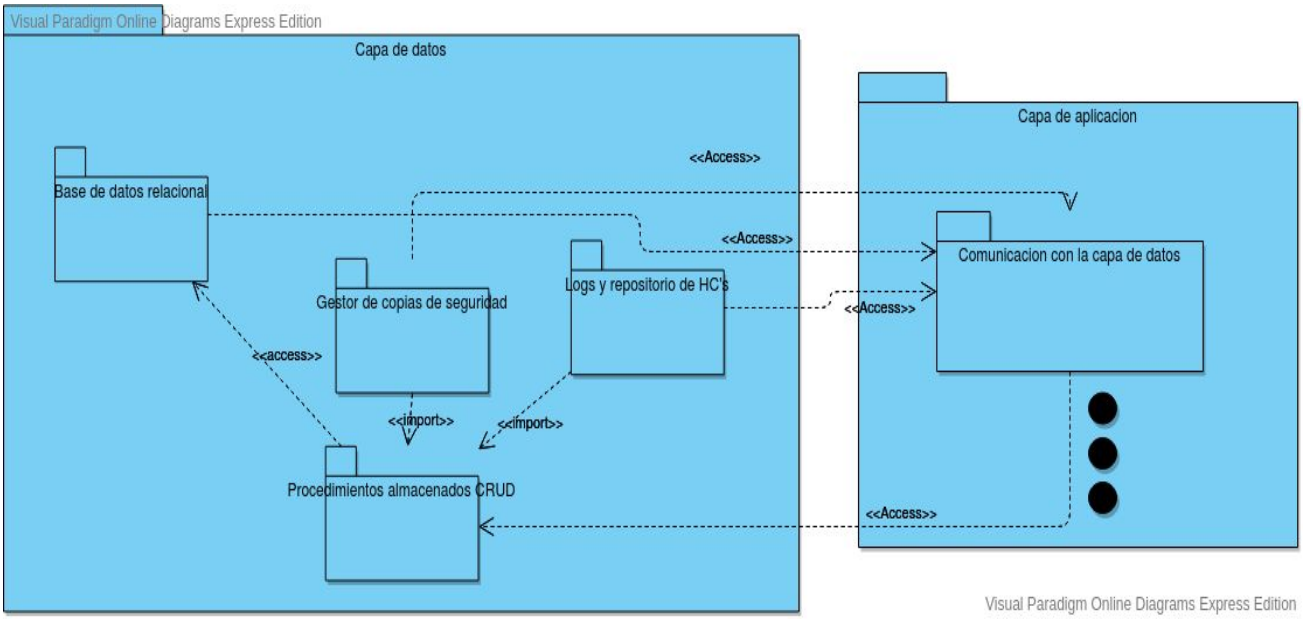
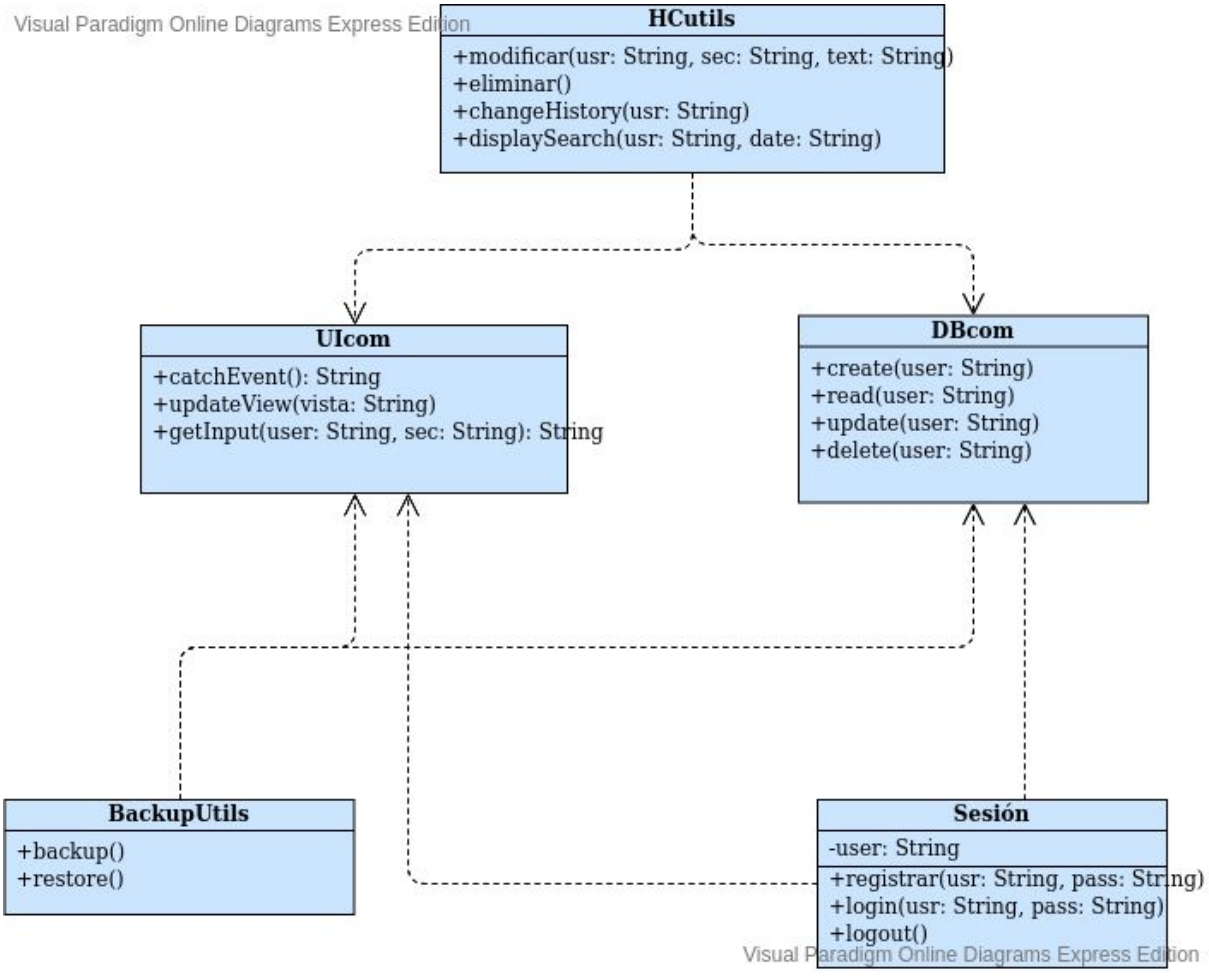


Diagrama de Paquetes para Capa de Datos



A partir de los diagrama de clase anteriores y en las formas que acceden o incorporan las funcionalidades de otros paquetes, se diseñó el siguiente diagrama de clases:



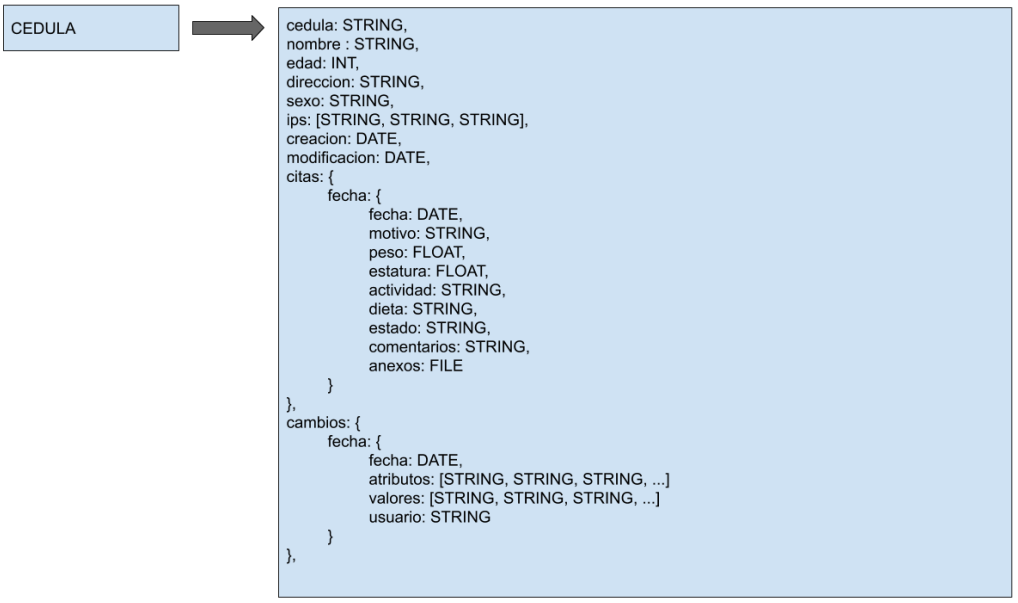
Diseño de base de datos

Se decidió optar por una única base de datos No-SQL basada en documentos. Esta es Cloud Firestore de la plataforma Firebase. Se escogió esta base de datos principalmente porque ayuda enormemente en cuanto al rendimiento de los diferentes eventos del sistema.

Que sea basada en documentos permite que las operaciones de consulta y escritura de registros sean muy rápidas, y gracias a la naturaleza de la aplicación que se está desarrollando (las actividades principales son la consulta y modificación de registros de pacientes), este hecho maximiza el rendimiento de la mayoría de peticiones en el sistema.

De forma secundaria, se escogió esta base de datos debido a su simpleza en implementación y facilidad para diseñar una arquitectura personalizada mediante colecciones de documentos que atendiera sólo las necesidades puntuales de la arquitectura de la aplicación web.

Diagrama de modelo de datos (documentos)

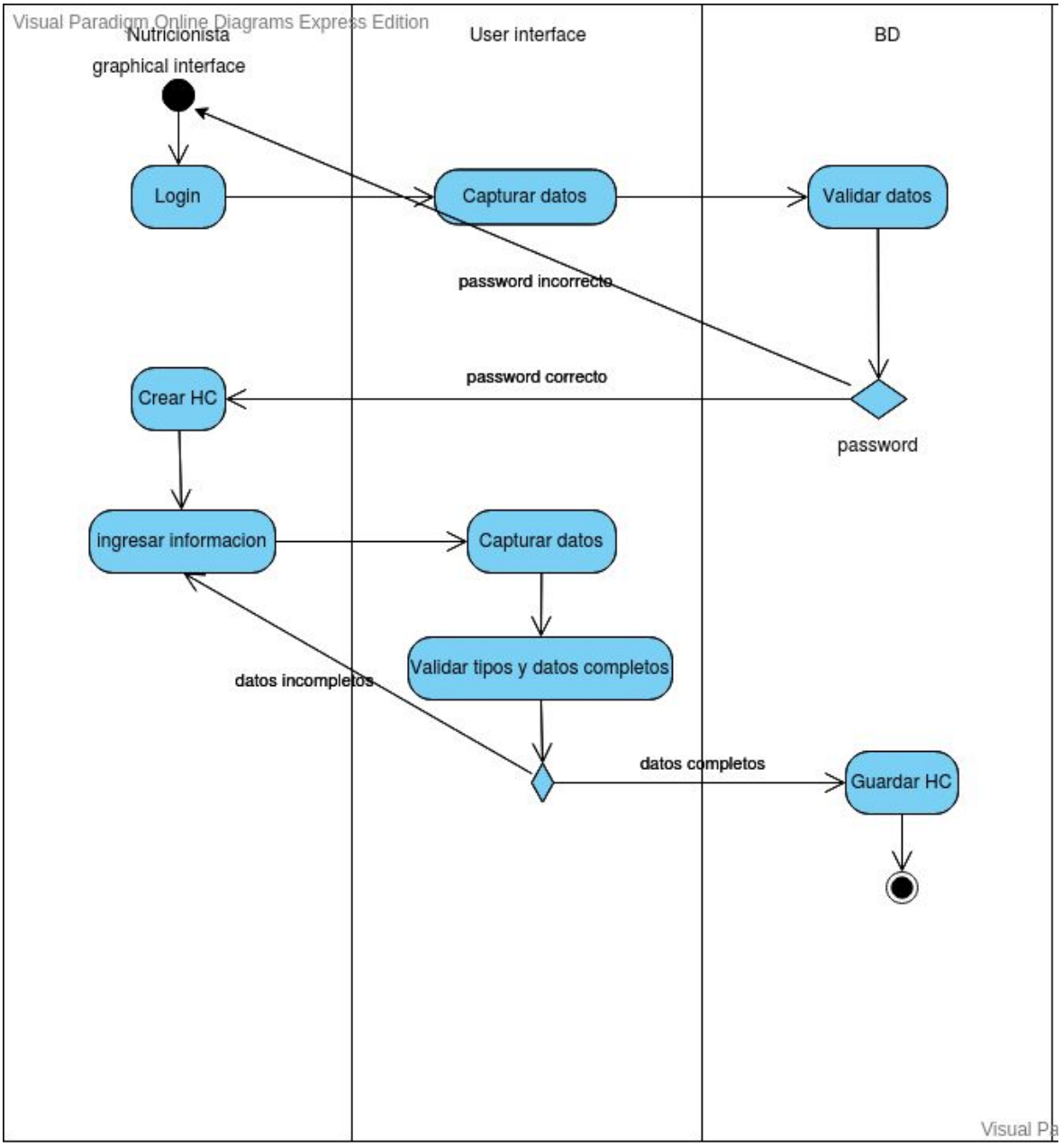


Vista de Procesos

Diagrama de actividades

Los procesos que se eligieron para realizar los dos diagramas de actividades son la funcionalidad del sistema de crear una nueva historia clínica por parte de un nutricionista y la funcionalidad de crear un nuevo usuario por parte de un administrador.

Crear una nueva historia clínica



Crear un nuevo usuario

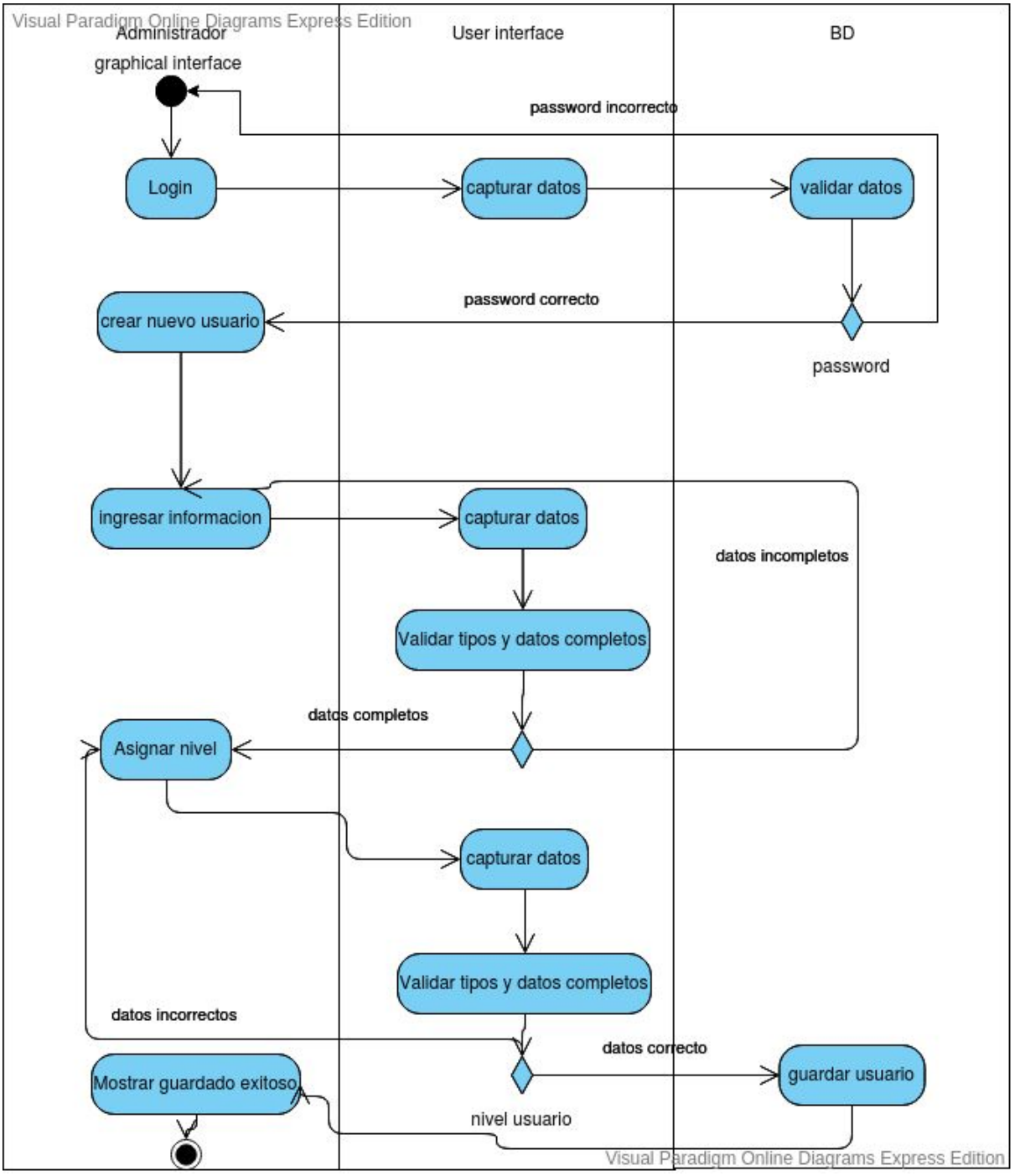
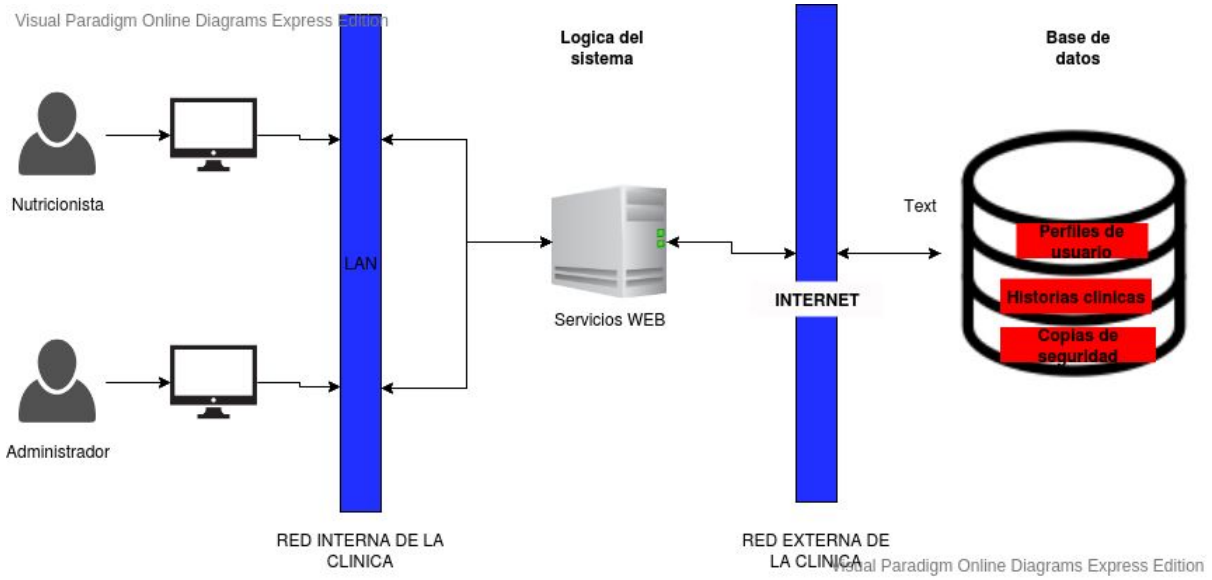


Diagrama de secuencia

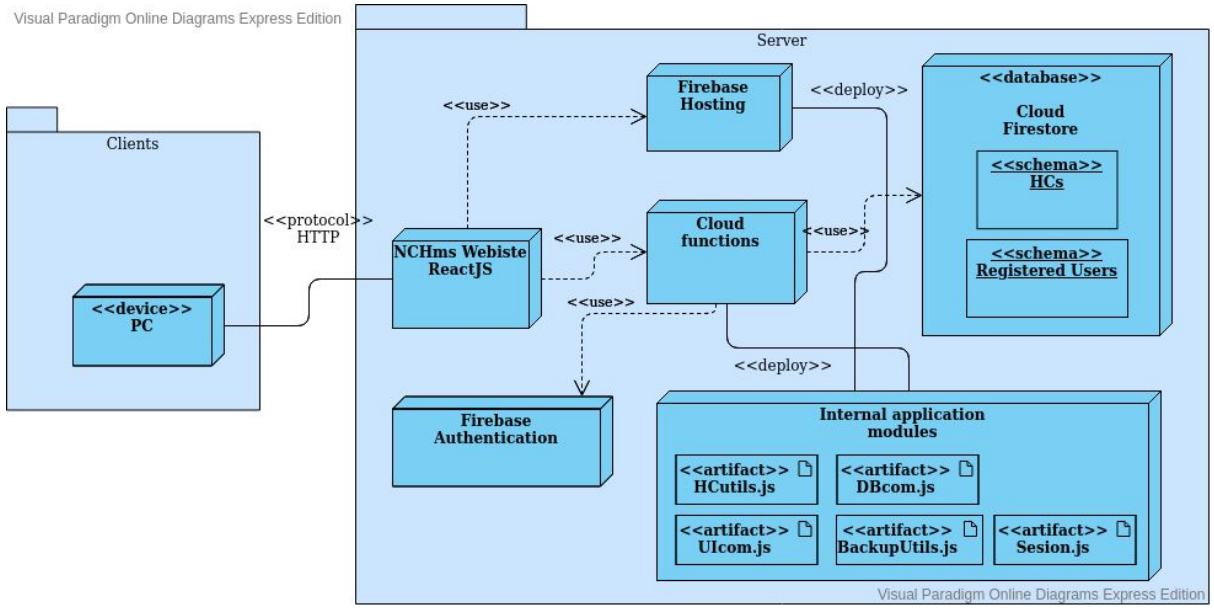
Debido a su complejidad y gran tamaño, los diagramas de secuencia se encuentran anexados a la carpeta del proyecto.

Vista Física

Diagrama de arquitectura física



Vista de Despliegue



Vista de Interfaz de Usuario

Para el desarrollo del front-end del aplicativo web se optó por utilizar la librería de ReactJS para interfaces gráficas.

Anexos

Documento	**Filename**
Diagrama de secuencia nutricionista	diagrama_secuencia_nutricionista.png
Diagrama de secuencia administrador	diagrama_secuencia_administrador.png
Documentación de casos de uso	Documentacion_casos_de_uso.pdf
Mapa de navegabilidad + todas las vistas del sistema.	https://marvelapp.com/89eb83j