



UNIVERSIDAD DE MÁLAGA



E.T.S. INGENIERÍA  
**INFORMÁTICA**  
UNIVERSIDAD DE MÁLAGA

Graduado en Ingeniería de la Salud

## Desarrollo de una aplicación para la gestión del tratamiento de la enterocolitis necrotizante (ECN)

Development of an application for the management of necrotizing  
enterocolitis (NEC) treatment

Realizado por  
Samuel Pérez Fernández

Tutorizado por  
José María Álvarez Palomo  
Carlos Rossi Jiménez

Departamento  
Lenguajes y Ciencias de la Computación  
UNIVERSIDAD DE MÁLAGA

MÁLAGA, septiembre de 2021



UNIVERSIDAD  
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA  
Graduado en Ingeniería de la Salud

**Desarrollo de una aplicación para la gestión del  
tratamiento de la enterocolitis necrotizante (ECN)**

**Development of an application for the management of  
necrotizing enterocolitis (NEC) treatment**

Realizado por  
**Samuel Pérez Fernández**

Tutorizado por  
**José María Álvarez Palomo**  
**Carlos Rossi Jiménez**

Departamento  
**Lenguajes y Ciencias de la computación**

UNIVERSIDAD DE MÁLAGA  
MÁLAGA, SEPTIEMBRE DE  
2021

Fecha defensa: septiembre de 2021

# Abstract

Necrotizing enterocolitis (NEC) is an intestinal disease that frequently affects newborns and especially premature infants. Planning the treatment aspects and their adaptation to the patients is vital for overcoming the disease. Therefore, this project proposes the development of an application to manage therapy guidelines, to facilitate the organization of treatment and to standardize the different steps of the treatment of NEC: evaluation of symptoms, adjustment of the clinical response (calculation of the milk intake based on the characteristics of the patient, control of the corresponding intakes, choice of a feeding method and the appropriate type of milk, etc.), as well as the recording of the information obtained to promote feedback and its subsequent statistical analysis. An incremental iterative methodology will be followed; it consists of an initial phase and a series of iterations in which the application software will be increasingly developed. To implement this project, different technologies will be used, including Angular, NodeJS and Google Firebase to deploy the web application, add-ons such as FullCalendar and Echarts for specific functionalities within the application, and Visual Studio Code as a development environment and code editor.

**Keywords:** Angular, Necrotizing enterocolitis, NodeJS, treatment, web application

# Resumen

La enterocolitis necrotizante (ECN) es una enfermedad intestinal que afecta de manera frecuente a recién nacidos y sobre todo a prematuros. La planificación de los aspectos del tratamiento y su adecuación a los pacientes es vital para la superación de la enfermedad. Por lo tanto, en este proyecto se propone el desarrollo de una aplicación web para gestionar las pautas de la terapia, de manera que se facilite la organización del tratamiento y se estandarice los diferentes pasos del tratamiento de la ECN: evaluación de los síntomas, ajuste de la respuesta clínica (cálculo de los aportes de leche en función de las características del paciente, control de las ingestas correspondientes y la sintomatología, etc), así como el registro de la información obtenida para favorecer la retroalimentación y su posterior análisis estadístico. Para el desarrollo de dicha aplicación se empleará una metodología iterativa incremental, que consta de una fase inicial y una serie de iteraciones en las que se desarrollará de manera creciente el software de la aplicación. Para implementar este proyecto se hará uso de distintas tecnologías, entre ellas Angular, NodeJS y Google Firebase para desplegar la aplicación web, complementos como FullCalendar y Echarts para funcionalidades específicas dentro de la aplicación y Visual Studio Code como entorno de desarrollo y editor de código.

**Palabras clave:** Angular, aplicación web, Enterocolitis necrotizante, NodeJS, tratamiento.



# Índice

<b>1. Introducción</b>	<b>7</b>
1.1. Motivación y contexto . . . . .	7
1.2. Objetivos . . . . .	8
1.3. Metodología . . . . .	9
1.4. Estructura del documento . . . . .	10
<b>2. Requisitos</b>	<b>11</b>
2.1. Requisitos funcionales . . . . .	11
2.2. Requisitos no funcionales . . . . .	12
2.3. Casos de uso . . . . .	13
2.4. Pruebas y testeo . . . . .	18
<b>3. Tecnologías empleadas</b>	<b>21</b>
<b>4. Desarrollo e implementación</b>	<b>25</b>
4.1. Fase inicial: Investigación previa . . . . .	25
4.2. Fase 1: <i>frontend</i> de la aplicación . . . . .	27
4.3. Fase 2: <i>backend</i> de la aplicación . . . . .	34
4.4. Fase 3: Lógica de la aplicación y algoritmos . . . . .	39
<b>5. Conclusiones y Líneas Futuras</b>	<b>41</b>
5.1. Conclusiones . . . . .	41
5.2. Líneas Futuras . . . . .	42
<b>Apéndice A. Manual de Instalación</b>	<b>47</b>
<b>Apéndice B. Manual de usuario</b>	<b>49</b>
<b>Apéndice C. Diseño de la interfaz en Balsamiq</b>	<b>59</b>



# 1

# Introducción

## 1.1. Motivación y contexto

La enterocolitis necrosante o necrotizante (ECN) es una enfermedad digestiva que afecta a los bebés neonatos, caracterizada por la inflamación del tejido del intestino grueso, la isquemia y el crecimiento bacteriano; estos síntomas desembocan en la necrosis del tejido intestinal[8], la cual puede conllevar un fallo multiorgánico. La mortalidad de esta enfermedad alcanza un 20-30 % en países como Estados Unidos y Canadá[13]; esta alta incidencia ha provocado que sea la tercera causa de muerte en las unidades de cuidados intensivos neonatales estadounidenses[9].

A pesar de que esta enfermedad es conocida desde 1823[14], su origen permanece incierto a día de hoy, lo que hace que su prevención sea complicada de establecer. Sin embargo, a lo largo del tiempo ciertos factores han sido asociados de manera evidente a esta enfermedad, como la prematuridad del infante, el tipo de alimentación recibida o el peso. En efecto, la mayoría de los pacientes afectados por ECN son prematuros (han nacido antes de las 37 semanas de gestación)[5]. El peso al nacer también es un factor de riesgo a tener en cuenta: se calcula que el 5 % de los recién nacidos de muy bajo peso (menos de 1500 g) está afectado por ECN, mientras que entre los recién nacidos de peso extremadamente bajo (menos de 1000 g) la incidencia es del 10 %[16].

Debido a la gravedad de la ECN, un diagnóstico temprano y un tratamiento eficaz son claves para su superación. Respecto al tratamiento, se ha demostrado que la incorporación de protocolos y medidas alimenticias en los recién nacidos de bajo peso disminuye la incidencia de ECN, reduciendo la mortalidad en casi un 10 %[19], así como la necesidad de realizar intervenciones quirúrgicas asociadas. Por lo tanto, resulta imperante la aplicación adecuada de estos protocolos alimenticios y su estandarización a los recién nacidos susceptibles de padecer

ECN; de esta necesidad de gestión surge la idea de desarrollar una software que ayude a los profesionales sanitarios a aplicar dichos protocolos.

Dicho software consistirá en una aplicación que siga los pasos propuestos por la guía de práctica clínica sobre nutrición enteral del recién nacido de muy bajo peso al nacimiento, planteada por el grupo multidisciplinar de investigación pediátrica de la Unidad de Gestión Clínica de Neonatología del Hospital Regional Universitario de Málaga[18]. En esta guía se discuten ciertas decisiones que afectan a la alimentación del paciente, como el tipo de leche, el momento para empezar la nutrición, la velocidad de las ingestas o el volumen de los incrementos de las ingestas; estas decisiones repercuten en mayor o menor medida en el desarrollo de la aplicación y sus características. En esta guía se establece que la alimentación de los pacientes se divide en dos partes diferenciadas, una primera parte en la que se administrará al paciente una nutrición enteral trófica o mínima (NET/NEM) y una segunda parte en la que se le administrará una nutrición enteral completa (NEC). Además será necesario tener otras situaciones específicas en cuenta, como por ejemplo si el paciente presenta un crecimiento intrauterino retardado (CIR). Estas y otras características se explicarán en los siguientes capítulos.

## **1.2. Objetivos**

El objetivo principal de este trabajo es desarrollar una aplicación web capaz de estandarizar los protocolos de un régimen alimenticio en los recién nacidos que necesiten un tratamiento de este tipo. Para alcanzar este objetivo se han planteado los siguientes objetivos específicos:

- Identificar los síntomas de estos pacientes y las características del régimen alimentario, de manera que se puedan definir unos requisitos para cubrir las necesidades del software.
- Desplegar una base de datos en la que poder realizar las operaciones CRUD (Create, Read, Update and Delete, en español Crear, Leer, Actualizar y Borrar) básicas con la información relativa al estado de los pacientes y su evolución.
- Desarrollar una interfaz de usuario fácil e intuitiva.
- Recopilar datos estadísticos sobre la información recogida para su posterior estudio y representación gráfica.

- Diseñar un algoritmo que permita en función de los datos del paciente calcular la respuesta alimentaria óptima para su mejora.

### 1.3. Metodología

Para la realización de este proyecto se sigue un desarrollo iterativo e incremental; el proyecto se divide en diferentes iteraciones, en las cuales el software se va desarrollando progresivamente. En cada iteración se completan una serie de requisitos, de manera que el resultado de nuestro trabajo va mejorando y evolucionando conforme las iteraciones avanzan[21]. Antes del proceso iterativo existe una fase inicial en la cual se hace un estudio de las cuestiones generales del proyecto, como pueden ser el contexto de la aplicación, antecedentes de la guía alimenticia, recopilación de información sobre el tratamiento de la ECN, posibles usuarios de la aplicación, requisitos, restricciones, etc.). Una vez establecidos los principales datos e ideas del proyecto se puede comenzar con la serie de iteraciones; en cada una de ellas se escogen un conjunto de requisitos a desarrollar (detallados en capítulos siguientes). Para esos requisitos se crearán los modelos, diseños, pruebas y elementos necesarios para construir el código que implemente dichos requisitos, integrados sobre el resultado de la iteración previa. Al final de cada iteración, se hará una reunión con el cliente para determinar la satisfacción de este con la implementación resultante de la iteración y para evaluar y planificar el trabajo del resto del proyecto.

Debido a las características de las tecnologías de desarrollo web que vamos a emplear durante este proyecto, que permiten trabajar de manera independiente los componentes de la aplicación, las iteraciones pueden ser clasificadas en función del tipo de requisitos que van a abordar, o la parte del software a la que están enfocadas. De esta manera se clasifican las iteraciones según si están dirigidas a desarrollar el lado del cliente o frontend (interfaz, navegación, formularios, representación estadística, etc.), el lado del servidor o backend (conexión con la base de datos, conversión de los datos, autenticación y protección de la información, diseño de la API, etc.) o la lógica interna de nuestra aplicación (desarrollo de algoritmos para computar la alimentación protocolizada y cálculo de estadísticas).

Las ventajas que ofrece esta metodología son diversas: desde las primeras iteraciones se puede ir utilizando y testeando el software implementado, la carga de trabajo se reparte espaciada en el tiempo, las tareas se pueden separar en función de los requisitos a abarcar, los errores y

fallos se pueden detectar con mayor facilidad debido al testeado y pruebas regulares, el diseño incremental permite añadir nuevas funciones que no estuviesen planeadas con anterioridad, etc[21].

#### **1.4. Estructura del documento**

La memoria de este proyecto se divide en los siguientes capítulos:

Una introducción, en la que se expone la motivación para la realización de este trabajo; se hace una descripción de los objetivos a conseguir; la metodología, que explica la planificación que se ha seguido durante el transcurso de este trabajo, las fases o iteraciones en las que se ha dividido el proyecto y el por qué se ha seleccionado este tipo de metodología a seguir; y la estructura que va a seguir este documento.

Un capítulo donde se explican las tecnologías empleadas junto a sus características principales.

Un capítulo donde se detallan los requisitos de la aplicación presentada en este trabajo, así como los casos de uso donde se describirán las acciones que se pueden llevar a cabo dentro de dicha aplicación.

Una sección sobre el desarrollo y la implementación, donde se detalla el proceso de trabajo seguido durante cada iteración de manera exhaustiva, indicando los requisitos resueltos en estas iteraciones y los problemas que puedan haber surgido en cada una de ellas.

Unas conclusiones y líneas futuras de trabajo, donde se exponen los resultados obtenidos durante este proyecto, se comenta una visión global del trabajo realizado y se proponen futuras líneas en las que este proyecto podría continuar.

# 2

## Requisitos

A continuación se detallan los requisitos encontrados a la hora de desarrollar esta aplicación; dichos requisitos se dividen en dos grupos, los requisitos funcionales (RF) y los requisitos no funcionales (RNF). Los requisitos funcionales especifican las características que debe satisfacer el software que se va a implementar para su correcto funcionamiento, mientras que los requisitos no funcionales abordan aspectos cualitativos como el rendimiento de nuestra aplicación, necesidades externas (requisitos legales, gubernamentales, etc.). Los requisitos funcionales se agrupan según la funcionalidad que vayan a desempeñar, mientras que los requisitos no funcionales se agrupan según la característica pertinente a la que afecte dicho requisito, como puede ser rendimiento, usabilidad, seguridad, etc.

### 2.1. Requisitos funcionales

Gestión del almacenamiento de los pacientes.

- RF-01: Alta del paciente. El usuario puede introducir a un nuevo paciente en el programa de alimentación protocolizada; para ello, el usuario deberá introducir información básica sobre el paciente, como nombre, apellidos, peso al nacer, número de semanas de gestación, etc.
- RF-02: Consulta de datos de un paciente. El usuario puede consultar los datos relativos a un paciente incluido en el programa de alimentación protocolizada.
- RF-03: Actualización de datos personales del paciente. El usuario puede modificar los datos de un paciente incluido en el programa de alimentación protocolizada.

Gestión de los datos relativos a la alimentación protocolizada

- RF-04: Inicio del protocolo alimenticio. El usuario puede activar la alimentación protocolizada en un paciente completando la información necesaria para ello, como la ausencia de asfixia, estabilidad hemodinámica en las últimas 12 horas, el tipo de leche disponible, etc.
- RF-05: Eliminación del paciente del protocolo. El usuario puede sacar un paciente del programa de alimentación protocolizada, indicando el motivo por el que el paciente ha finalizado dicho protocolo.
- RF-06: Informe de evolución diaria. El usuario puede incluir información sobre la evolución del paciente en cada día, como por ejemplo si las deposiciones son normales o si se han producido vómitos.
- RF-07: Proporcionar la cantidad de leche que puede ser administrada cada día. El usuario puede solicitar al sistema que calcule la cantidad que debe proporcionar al paciente en función de su estado clínico y las condiciones tenidas en cuenta en el protocolo de alimentación.

#### Desarrollo de estudios estadísticos

- RF-08: Estadísticas comparativas. El usuario puede obtener un informe con estadísticas de las características de los pacientes (peso, semanas de gestación, presencia de catéter umbilical, etc.).
- RF-09: Exportación de datos. El sistema es capaz de exportar la información de los pacientes para su posterior estudio estadístico con otra herramienta.

## **2.2. Requisitos no funcionales**

#### Usabilidad

- RNF-01: El sistema muestra de forma clara en que apartado se encuentra el usuario mediante botones de distintos colores en una barra de navegación.
- RNF-02: El sistema notifica al usuario cuando un campo de un formulario no se ha completado correctamente.

## Seguridad

- RNF-03: El sistema permite almacenar la información de los usuarios que pueden entrar a la base de datos, siendo dicha información accesible a un administrador de la aplicación.
- RNF-04: El sistema proporciona un forma de iniciar y cerrar sesión, de manera que solo los usuarios registrados por el administrador pueden hacer uso de la aplicación.

## Rendimiento y Fiabilidad

- RNF-05: El sistema es capaz de acceder a la información de la base de datos en tiempo real, de manera rápida y actualizada.

## Interfaz

- RNF-06: La interfaz hace uso de los colores de manera intuitiva para indicar la funcionalidad de los botones.

## 2.3. Casos de uso

A continuación se van a explicar los casos de uso más relevantes dentro de la aplicación; en la figura 1 se puede observar el diagrama de estos casos de uso. En cada caso se detallan las precondiciones del sistema, el resultado o postcondiciones, el escenario principal, el actor principal y los posibles escenarios alternativos.

- Iniciar sesión en la aplicación:
  - Actor principal: personal sanitario/usuario.
  - Precondiciones: el usuario debe estar registrado en la base de datos de usuarios con acceso a la aplicación.
  - Escenario principal:
    1. El sistema carga la página de inicio de sesión.
    2. El usuario completa los campos del formulario necesarios para iniciar sesión (correo electrónico y contraseña).

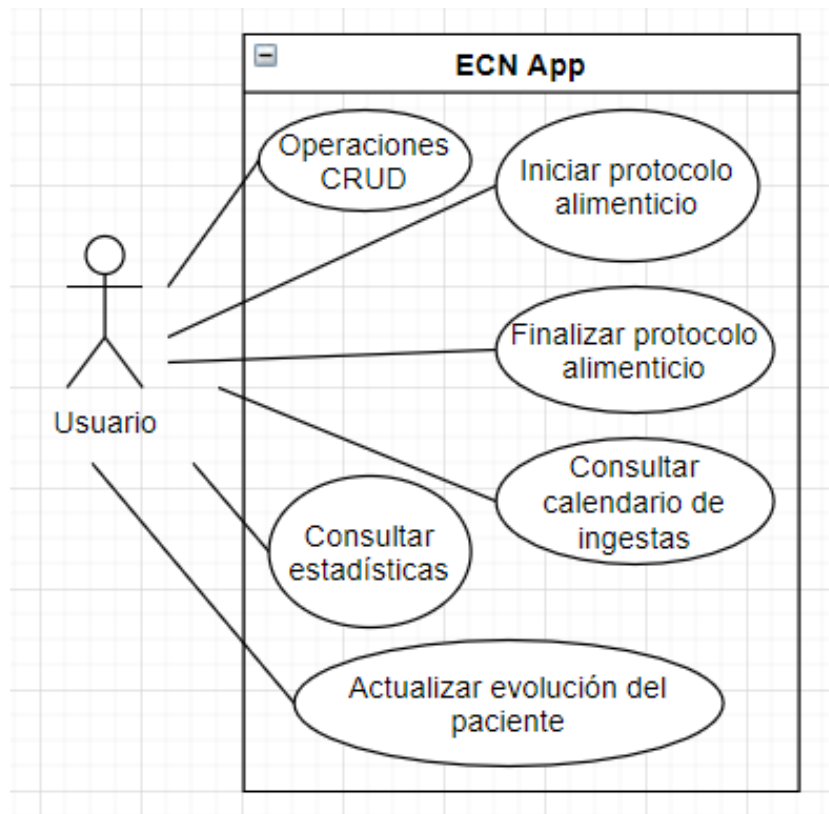


Figura 1: Diagrama de casos de uso disponibles.

3. El sistema comprueba la validez de estos campos.
  4. El sistema carga la página principal de la aplicación.
- Resultado: el usuario ingresa con éxito en la aplicación.
  - Escenarios alternativos:
    - 3.1 El sistema notifica al usuario de que alguno de los campos introducidos no es correcto.
  - Añadir un nuevo paciente a la aplicación:
    - Actor principal: personal sanitario/usuario.
    - Precondiciones: el usuario ha iniciado sesión correctamente.
    - Escenario principal:
      1. El usuario clic en el botón de 'añadir paciente'.
      2. El sistema muestra un formulario con los campos necesarios.
      3. El usuario rellena los campos y pulsa el botón de enviar.

4. El sistema notifica al usuario de que el paciente se ha añadido con éxito y redirige al usuario a la página principal.
- Resultado: el paciente nuevo es ingresado con éxito en la aplicación.
  - Escenarios alternativos:
    - 3.1 El usuario no ha completado los campos necesarios o ha rellenado incorrectamente algún campo.
    - 3.2 El sistema notifica al usuario del campo incorrecto.
- Consultar la información de un paciente:
    - Actor principal: personal sanitario/usuario.
    - Precondiciones: el usuario ha iniciado sesión correctamente, el paciente que se quiere consultar está en la base de datos.
    - Escenario principal:
      1. El usuario clic en el botón ‘más información’ del paciente cuya información quiere consultarse, ya sea un paciente activo o uno inactivo.
      2. El sistema carga la información del paciente seleccionado.
    - Resultado: La información del paciente es mostrada correctamente.
  - Editar la información de un paciente:
    - Actor principal: personal sanitario/usuario.
    - Precondiciones: el usuario ha iniciado sesión correctamente, el paciente que se quiere editar está en la base de datos.
    - Escenario principal:
      1. El usuario clic en el botón ‘editar paciente’ del paciente cuya información quiere editarse.
      2. El sistema carga la información del paciente seleccionado dentro de un formulario.
      3. El usuario edita la información que se desee cambiar y pulsa el botón de ‘guardar cambios’.

4. El sistema graba la nueva información en el paciente correspondiente y redirige al usuario a la página principal.
- Resultado: los cambios introducidos en la información del paciente se guardan correctamente.
  - Escenarios alternativos:
    - 3.1 El usuario ha rellenado incorrectamente algún campo.
    - 3.2 El sistema notifica al usuario del campo incorrecto.
- Iniciar el protocolo alimenticio de un paciente:
    - Actor principal: personal sanitario/usuario.
    - Precondiciones: el usuario ha iniciado sesión correctamente, el paciente deseado está en la base de datos.
    - Escenario principal:
      1. El usuario clic en el botón 'iniciar protocolo' del paciente deseado.
      2. El sistema muestra un formulario con los campos necesarios para iniciar la alimentación protocolizada.
      3. El usuario rellena los campos y pulsa el botón de enviar.
      4. El sistema notifica al usuario de las precauciones específicas del paciente (si las hubiera), la alimentación protocolizada se inicia con éxito y se redirige al usuario a la página principal.
      5. El botón 'iniciar protocolo' deja de estar disponible para este paciente.
    - Resultado: El calendario del protocolo alimenticio está disponible para consultar para este paciente.
    - Escenarios alternativos:
      - 3.1 El usuario no ha completado los campos necesarios o ha rellenado incorrectamente algún campo.
      - 3.2 El sistema notifica al usuario del campo incorrecto.
  - Eliminar a un paciente del protocolo alimenticio:

- Actor principal: personal sanitario/usuario.
  - Precondiciones: el usuario ha iniciado sesión correctamente, el paciente que se quiere eliminar está en la base de datos, el protocolo alimenticio ha sido iniciado con éxito
  - Escenario principal:
    1. El usuario clicca en el botón 'finalizar protocolo' del paciente que desea eliminar.
    2. El sistema muestra un formulario para justificar el motivo de finalización del protocolo.
    3. El usuario rellena el formulario y confirma la eliminación del paciente de la base de datos de pacientes activos.
    4. El sistema añade el paciente eliminado de la lista de pacientes activos a la lista de pacientes inactivos.
  - Resultado: El paciente es eliminado de la lista de pacientes activos/pacientes con el protocolo alimenticio iniciado.
- Consultar el calendario de la alimentación protocolizada:
    - Actor principal: personal sanitario/usuario.
    - Precondiciones: el usuario ha iniciado sesión correctamente, el paciente deseado está en la base de datos, el protocolo alimenticio ha sido iniciado con éxito.
    - Escenario principal:
      1. El usuario clicca en el botón 'calendario' del paciente deseado.
      2. El sistema carga el calendario correspondiente al paciente clicado.
    - Resultado: El calendario del protocolo alimenticio del paciente seleccionado se muestra correctamente.
- Actualizar la evolución de un paciente:
    - Actor principal: personal sanitario/usuario

- Precondiciones: el usuario ha iniciado sesión correctamente, el paciente deseado está en la base de datos, el protocolo alimenticio ha sido iniciado con éxito, el calendario del protocolo alimenticio ha sido abierto por el usuario.
- Escenario principal:
  1. El usuario selecciona la fecha y la ingesta/toma relativa a la evolución que se quiere registrar.
  2. El sistema muestra un formulario con los campos propios de la evolución del paciente.
  3. El usuario rellena los campos y pulsa el botón de enviar.
  4. El sistema actualiza la información del paciente y se redirige al usuario a la página principal.
- Resultado: La evolución del paciente relativa a la fecha e ingesta/toma seleccionada queda registrada en la base de datos.
- Consultar las estadísticas de la aplicación:
  - Actor principal: personal sanitario/usuario
  - Precondiciones: el usuario ha iniciado sesión correctamente.
  - Escenario principal:
    1. El usuario clicla en el botón ‘estadísticas’.
    2. El sistema carga la página correspondiente a la información estadística.
    3. El usuario puede descargar un informe con los datos estadísticos correspondientes a pacientes activos e inactivos hasta ese momento almacenados en la base de datos.
  - Resultado: Las representaciones gráficas de los datos estadísticos se muestran correctamente; se obtiene un informe de los datos estadísticos.

## 2.4. Pruebas y testeo

Durante la realización de este proyecto se ha ido comprobado que las propiedades del software desarrollado cumplían los requisitos necesarios. Estas comprobaciones se han llevado a

cabo mediante pruebas manuales, ya que se ha considerado que debido a la cantidad de software a implementar y el tiempo disponible, esta era una mejor opción que invertir recursos y tiempo en preparar pruebas automáticas. Las pruebas realizadas pueden dividirse en dos tipos: las pruebas funcionales, destinadas a chequear los requisitos funcionales de la aplicación, y las pruebas no funcionales, para revisar los requisitos no funcionales del proyecto.

Las pruebas funcionales que se han realizado son las siguientes:

- Pruebas exploratorias: En las primeras iteraciones de desarrollo se han aplicado estas pruebas, ya que aportan un feedback rápido que ayuda a encontrar fallos en el software y no se necesita una planificación premeditada para llevarlas a cabo. Estas exploraciones se han aplicado a todos los componentes desarrollados en el proyecto (desde RF-01 hasta RF-09).
- Pruebas de regresión: Conforme han avanzado las iteraciones y el software se ha ido volviendo más complejo se han puesto en marcha este tipo de pruebas, que han resultado muy útiles a la hora de evaluar los algoritmos diseñados para el cómputo de ingestas de los pacientes, ya que estas funcionalidades eran las más complicadas de la aplicación y han sufrido muchos cambios (RF-04, RF-05, RF-06 y RF-07).

Las pruebas no funcionales que se han llevado a cabo son las siguientes:

- Pruebas de carga: Mediante estas pruebas se ha evaluado la capacidad de la aplicación para recibir accesos simultáneos de varios usuarios (RNF-05), comprobando la eficiencia del tipo de base de datos empleada (base de datos en tiempo real Firebase de Google).
- Pruebas de seguridad: A través de estas pruebas se ha intentado acceder a la información de la aplicación sin estar registrado, para así poner a prueba el sistema de autenticación de la misma (autenticación de Google Firebase) (RNF-04).
- Pruebas de usabilidad: Usando estas pruebas se ha pretendido mejorar la experiencia de usuario al navegar por la aplicación y la interacción entre usuario y sistema; uso de botones, diseño de la aplicación (colores, posición de elementos, etc.), uso de ventanas emergentes, notificaciones de éxito, etc (RNF-01, RNF-02 y RNF-06).



# 3

## Tecnologías empleadas

A continuación se describen las principales tecnologías empleadas durante el desarrollo de este proyecto. NodeJS es un entorno de ejecución de JavaScript orientado a eventos asíncronos, basado en el motor de JavaScript V8 de Google y diseñado para crear aplicaciones network escalables[4]. NodeJS permite utilizar JavaScript en el lado del servidor, facilitando así el desarrollo de la API necesaria para esta aplicación. Otra de las ventajas de NodeJS es que tiene un sistema de gestión de paquetes (Node Package Manager, npm) muy versátil, el cual permite instalar y manejar dependencias de manera intuitiva y rápida.

Angular es un framework para aplicaciones web, de código abierto, desarrollado y mantenido por Google, que posibilita crear aplicaciones de página única (Single Page Applications, SPAs). Angular sigue el patrón MVC (Modelo Vista Controlador), separando los componentes del proyecto según su función (plantillas HTML, hojas de estilo CSS, servicios, etc.) de manera que le otorga una gran escalabilidad a las SPAs[20]. Angular emplea TypeScript como lenguaje de programación; TypeScript es un superconjunto de JavaScript, es decir, añade una capa funcionalidad sobre Javascript, permitiendo tipar variables, funciones, etc., declarar interfaces, enumeraciones, etc. Typescript es de código abierto y está desarrollado y mantenido por Microsoft. Este lenguaje permite desarrollar el código tanto en el lado del servidor (gracias a NodeJS) como en el lado del cliente, lo cual ayuda en la simplificación de la complejidad este proyecto[2].

Bootstrap es un kit de herramientas (HTML, CSS y JavaScript) de código abierto, creado inicialmente por Twitter y mantenido actualmente por GitHub. Bootstrap ofrece gran variedad de componentes en su librería; ventanas, cuadros, botones, desplegables, etc.[22] Gracias a Bootstrap el diseño de la página web se convierte en una tarea sencilla y apta para no expertos

en el campo del frontend.

Google Firebase es una plataforma de desarrollo de aplicaciones (web, móviles, etc.) que ofrece distintos servicios para los programadores, como bases de datos en la nube, hosting, autenticación de Google, etc. En concreto en este trabajo se han empleado una base de datos en tiempo real de Firebase (Firebase Realtime Database), que almacena los datos de nuestra aplicación en formato JSON y se sincroniza en tiempo real con cada cliente conectado; y la Autenticación de Firebase (Firebase Authentication), que aporta la robustez de Google en cuanto a protección y la facilidad para que los usuarios puedan entrar a través de alguna de sus cuentas ya existentes de Google[3]. Para este proyecto se ha optado por una opción más convencional; el administrador de la aplicación, que tiene acceso a Firebase, es el encargado de registrar los correos electrónicos de los usuarios en la aplicación.

FullCalendar es un plugin jQuery de código abierto y gratuito (aunque tiene una versión premium de pago con funcionalidades extra) que permite incorporar un calendario interactivo a la aplicación, con un aspecto moderno y llamativo, en el cual es posible crear eventos de manera programática [11].

Echarts es una librería de gráficas y estadísticas de código abierto y respaldada por Apache; gracias a su módulo para Angular ngx-charts se pueden implementar gráficas de todo tipo (gráficos circulares, diagramas de barras, árboles, etc). Estos gráficos son interactivos y tienen animaciones, obteniéndose un aspecto fluido en su representación[10].

Latex es un software gratuito para la redacción de documentos con una composición tipográfica de calidad, de manera que el autor solo tenga que preocuparse por los contenidos de su documento. Latex facilita el uso de fórmulas, tablas, bibliografías, referencias, estilos, etc[17]. Para la realización de este documento se ha empleado Overleaf, que es un editor de Latex online, el cual permite manejar los archivos de Latex así como imágenes, portadas, bibliografías, etc. Tanto Overleaf y latex están orientados a la escritura de textos científicos; aún así, su versatilidad hace que sea útil para cualquier tipo de publicación[15]. Este editor es gratuito y cuenta con una gran documentación, lo que proporciona muchas posibilidades a la hora de configurar informes y textos.

Visual Studio Code (VS Code) es un editor de código gratuito desarrollado por Microsoft para Windows, macOS y Linux, que incluye soporte para JavaScript, TypeScript y NodeJS; además VS Code permite instalar una amplia gama de componentes y extensiones, como

pueden ser lenguajes de programación como Python o C, o frameworks como es el caso de Angular[12].

Balsamiq es una herramienta de pago que permite diseñar interfaces de usuario y esquemas de páginas web (UI wireframing), de manera que podemos representar fácilmente los elementos de la página web y manipularlos con soltura[1]. Balsamiq ofrece una versión integrada con Google drive, que es la que se ha empleado para este proyecto; esta versión es la recomendada para usuarios casuales, ya que tiene un periodo de prueba y el precio de la suscripción es razonable.

Git es una herramienta gratuita y de código abierto para la gestión de versiones y el control de los cambios en el código, permitiendo a varios desarrolladores trabajar a la vez en el mismo proyecto[6]. Por otro lado, Github es un repositorio donde se pueden albergar proyectos controlados por Git, además de trabajar conjuntamente en ellos con otras personas y compartirlos con el resto del mundo[7].

Por último, Taiga es una herramienta de código abierto, enfocada a las metodologías ágiles, que permite la gestión y/o administración de proyectos, de manera que se puede realizar un seguimiento de las tareas, objetivos y demás características detalladamente. Taiga permite dividir dichas tareas, añadir plazos y definir intervalos de trabajo, facilitando el trabajo en grupo y la revisión de los progresos.[23]



# 4

## Desarrollo e implementación

A continuación se va a explicar el proceso seguido en cada fase del desarrollo de esta aplicación, así como los contenidos implementados durante dichas fases.

### 4.1. Fase inicial: Investigación previa

En esta fase se ha investigado sobre la ECN y se ha estudiado el protocolo alimenticio presentado en la guía de práctica clínica sobre nutrición enteral del recién nacido de muy bajo peso al nacimiento, propuesta por la Unidad de Gestión Clínica de Neonatología del Hospital Regional Universitario de Málaga[18], con el objetivo de definir su sintomatología y las necesidades de su tratamiento, para satisfacer los requisitos relativos a la gestión de los datos de la alimentación protocolizada (RF-04, RF-05, RF-06 y RF-07). Gracias a esta investigación se extraen los siguientes resultados ajustables a la aplicación:

- Como se comentaba en el apartado de Motivación y contexto, existen dos tipos de nutrición en este protocolo, la NET y la NEC. La diferencia principal entre ambas nutriciones es la cantidad de leche administrada al paciente y la frecuencia de los incrementos de dicha cantidad.
- Hay ciertos síntomas que contraindican la aplicación de este protocolo, como la inestabilidad hemodinámica, una mala perfusión o la presencia de asfisia.
- Ciertas situaciones también se consideran poco favorables para la aplicación de este protocolo, como la ausencia de leche materna o calostro, que el peso del paciente sea mayor de 1500 g o que su periodo de gestación haya sido mayor a las 32 semanas.

- Existen otros factores como la presencia de catéteres umbilicales o que un paciente sea de crecimiento intrauterino restringido (CIR) que en principio no afectan al desarrollo de la alimentación protocolaria, pero deben ser tenidos en cuenta en el caso de que se complique la evolución del paciente.
- Según el peso del paciente y el resto de características anteriormente mencionadas se le aplica un ritmo de NET distinto y una duración distinta de la misma a cada paciente. Por consiguiente, se pueden clasificar a los pacientes en pacientes de alto riesgo, pacientes de riesgo medio y pacientes de riesgo bajo.
  - Los pacientes de alto riesgo reciben una NET de una semana de duración, en la que los 3 primeros días se realizan incrementos de  $0.3 \text{ ml de leche} \times \text{kilogramo} \times \text{hora}$ ; los 3 siguientes días los incrementos son de  $0.5 \text{ ml de leche} \times \text{kilogramo} \times \text{hora}$ , y el último día de esa semana el incremento a seguir es de  $1 \text{ ml de leche} \times \text{kilogramo} \times \text{día}$ .
  - Los pacientes de riesgo medio reciben una NET de 6 días de duración, en la que los 3 primeros días se realizan incrementos de  $0.5 \text{ ml de leche} \times \text{kilogramo} \times \text{hora}$  y los 3 siguientes días los incrementos son de  $1 \text{ ml de leche} \times \text{kilogramo} \times \text{hora}$ .
  - Los pacientes de bajo riesgo reciben una NET de 5 días de duración, en la que el incremento es de  $1 \text{ ml de leche} \times \text{kilogramo} \times \text{hora}$ ; además, si estos pacientes de bajo riesgo presentan una evolución favorable durante la NET (deposiciones normales, sin vómitos, etc.), esta misma podrá acortar su periodo de 5 días a 3 días.
- Al terminar la NET los pacientes prosiguen con la nutrición enteral completa (NEC), en la que reciben  $10 \text{ ml} \times \text{kilogramo}$  cada 12 horas si no superan los 1000 gramos de peso, o  $15 \text{ ml} \times \text{kilogramo}$  cada 12 horas si sí superan los 1000 gramos de peso.
- Los incrementos en el volumen de las ingestas se mantiene hasta que se alcance el volumen máximo recomendado para estos pacientes, que es de  $160 \text{ ml} \times \text{kilogramo} \times \text{día}$ . Por lo tanto, la duración de la NEC dependerá de las características del paciente y de su registro anterior de tomas.

## 4.2. Fase 1: *frontend* de la aplicación

Durante las iteraciones de esta fase se desarrolla el lado del cliente de la aplicación. Antes de empezar a implementar código se utilizó la herramienta Balsamiq para esbozar el aspecto de la aplicación; en el apéndice C se encuentran dichos esquemas. Esta fase aborda los requisitos funcionales desde RF-01 hasta RF-07, ya que todos estos requisitos necesitan de una interacción con el usuario para poder ser completados; también se llevan a cabo los requisitos no funcionales correspondientes a la usabilidad y la interfaz; RNF-01, RNF-02 y RNF-06.

### 4.2.1. Inicio de sesión o login

En la figura 2 se puede ver la página en la que el usuario debe de iniciar sesión para poder ingresar en la aplicación; como se comentaba en el RNF-03, la persona designada como administradora es la encargada de registrar los correos electrónicos de los usuarios que tienen acceso a la aplicación (más detallado en la fase 2 sobre backend). Al iniciar sesión correctamente el usuario entra en la página principal de la aplicación, la cual se corresponde con la lista de pacientes activos de la aplicación (figura 3). Desde cualquier página de la aplicación podemos acceder a las otras páginas disponibles de la misma gracias a la barra de navegación superior. Dichas páginas disponibles son la lista de pacientes activos, la lista de pacientes inactivos, la página de estadísticas, la página para añadir un nuevo paciente y la página para cerrar sesión.

### 4.2.2. Pacientes activos e información del paciente

Cada paciente de la lista de pacientes activos se visualiza como una tarjeta, donde se puede observar el nombre y apellidos del paciente, su fecha de nacimiento y las opciones posibles, como editar la información básica del paciente, consultar la información del paciente o iniciar la alimentación protocolizada (figura 3).

Al consultar la información de un paciente se pueden ver los datos relativos al paciente así como los pertinentes a la alimentación protocolizada que está siguiendo; fecha de inicio del protocolo, evolución del paciente en cada ingesta, motivo de finalización del paciente, etc. (figura 5) Según si el paciente ha iniciado o no el protocolo, o si lo ha finalizado, la información a mostrar será distinta para cada paciente.

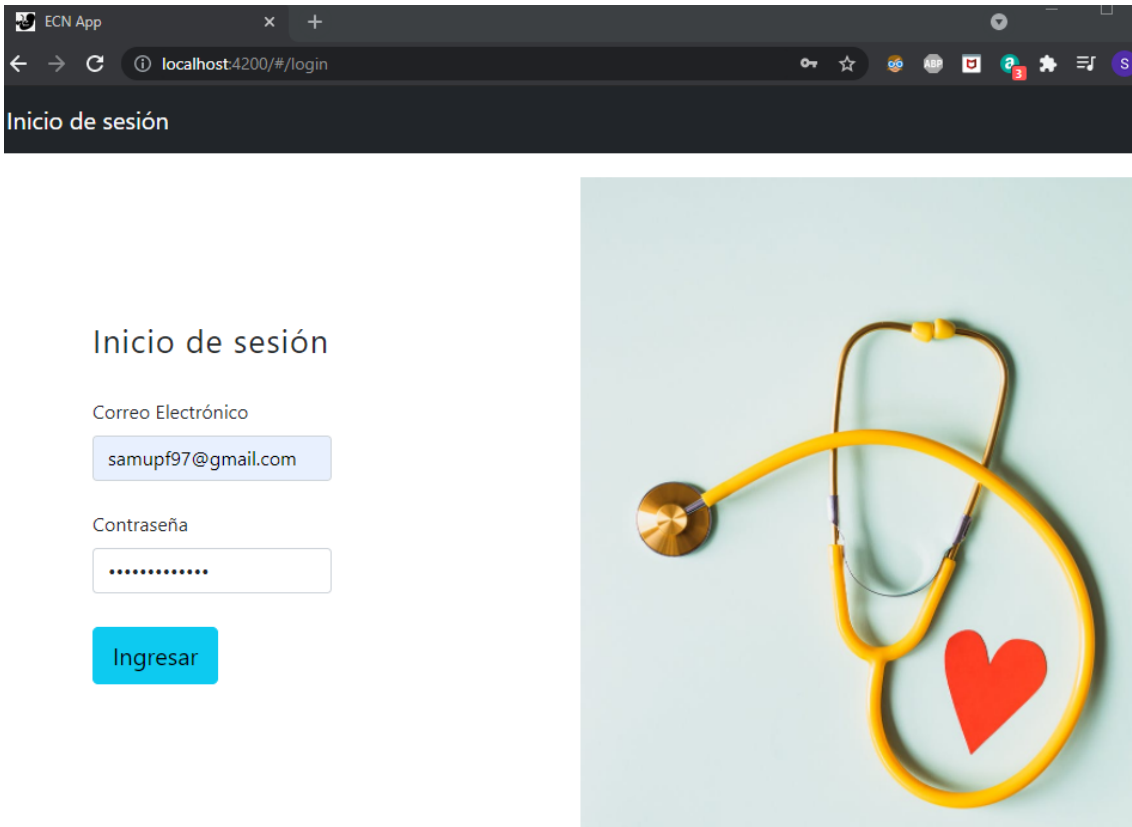


Figura 2: Página para iniciar sesión de la aplicación.

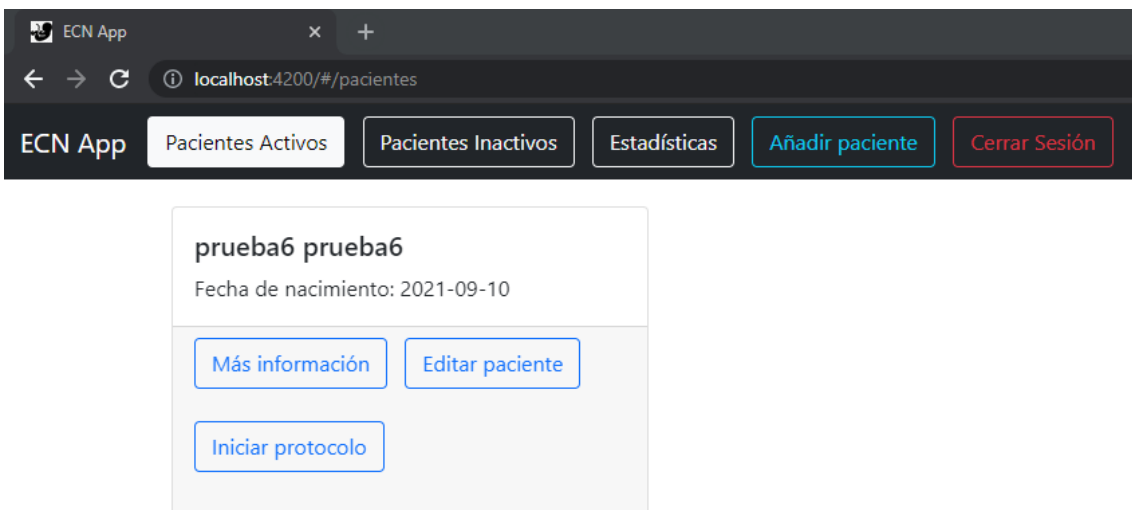


Figura 3: Página principal de la aplicación con la lista de pacientes activos.

### 4.2.3. Pacientes inactivos

En cuanto a los pacientes inactivos, al clicar sobre la página podemos observar la lista de pacientes cuya alimentación protocolizada ha concluido (figura 4); en cada uno de estos pacientes podemos observar sus características (donde se incluye el motivo por el cual han finalizado la alimentación) y el calendario que siguieron durante el protocolo.

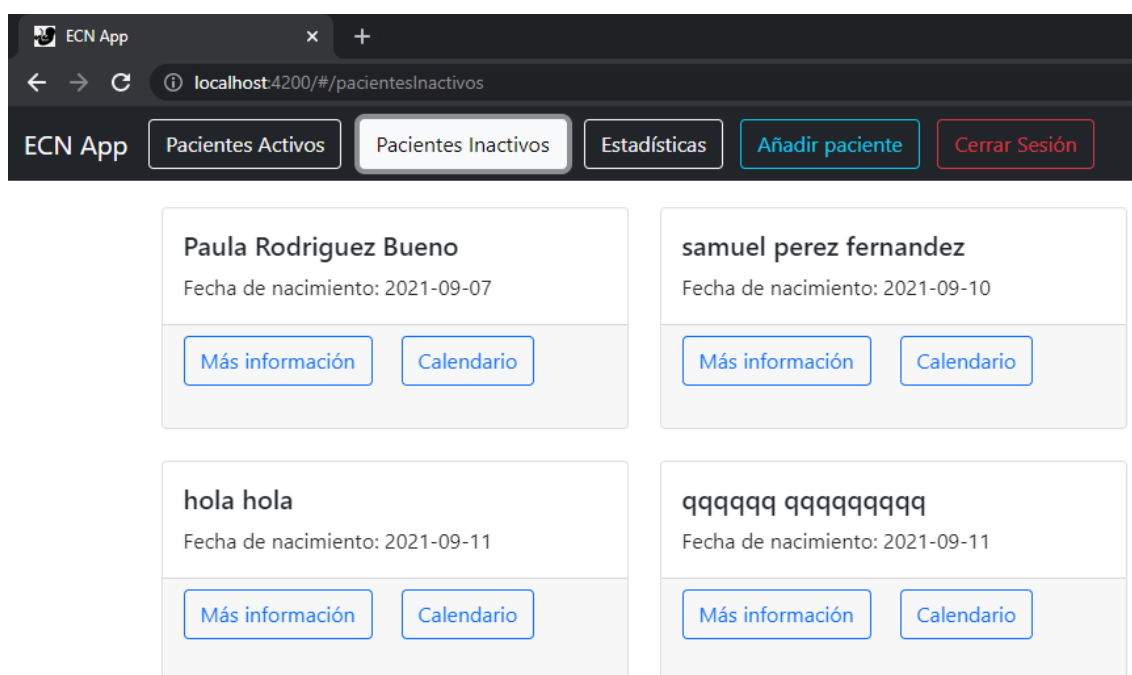


Figura 4: Página con la lista de pacientes inactivos.

### 4.2.4. Nuevo paciente y nuevo protocolo alimenticio

En la figura 6 se distingue el formulario para añadir un nuevo paciente a los pacientes activos; en dicho formulario se debe completar la información básica relativa al paciente como el nombre, apellidos, fecha de nacimiento, sexo, semanas de gestación y peso al nacer. Si se completa algún campo de manera incorrecta el formulario avisa al usuario, indicando el fallo correspondiente (RNF-02).

En la figura 7 se ve el formulario con el que iniciar el protocolo alimentario del paciente en cuestión; en dicho formulario se deben completar unos campos necesarios relativos al estado del paciente: la estabilidad hemodinámica, perfusión estable, ausencia de asfixia, presencia de leche materna o de cateter umbilical, etc. Algunas campos del formulario despliegan avisos

Paula Rodriguez Bueno			
Fecha de nacimiento:	2021-09-07		
Sexo:	F		
Semanas de gestación:	30 semanas		
Peso al nacer:	2000 g		
Protocolo alimenticio:	Iniciado		
Fecha de inicio del protocolo:	7/9/2021		
Calostro disponible al iniciar el protocolo:	Sí		
Paciente CIR:	No		
Presencia de catéter umbilical:	No		
Protocolo finalizado:	Sí		
Motivo de finalización del protocolo:	Paciente recuperado exitosamente		
Evolución del paciente	Deposiciones normales	Vómitos	Abdomen normal
Dosis 1: 2 ml, fecha: 2021-09-08 00:00	Sí	Sí	No

Figura 5: Información de un paciente.

sobre el usuario antes de iniciar la alimentación cuando las condiciones del paciente no son las apropiadas, como por ejemplo la ausencia de leche materna o un peso inadecuado para el tratamiento[18]; sin embargo, la decisión última corresponde al usuario, que puede iniciar el tratamiento de todos modos. Al completar el formulario, el calendario de las ingestas estará disponible para consultar desde la tarjeta de cada paciente.

Cuando ya se ha iniciado la alimentación protocolizada se puede consultar el calendario de ingestas de dicho protocolo (figura 8).

#### 4.2.5. Calendario de ingestas y evolución del paciente

El calendario puede visualizarse de diferentes maneras según el intervalo de tiempo; existe una vista mensual, una semanal, una diaria y una vista en forma de listado. Existen marcas de tiempo llamadas eventos que indican el inicio y fin de la NET, el inicio y final de la NEC y las ingestas correspondientes a ambos periodos. El día actual se indica con un fondo ligeramente amarillo y medida que avanzan las horas los eventos cambian de un color azul a gris, para indicar que ya han terminado (RNF-06). Los eventos correspondientes a las dosis/ingestas de los pacientes están marcados con un círculo (en la vista mensual), la hora a la que debe darse la dosis, el número de la dosis y la cantidad de mililitros (ml) de leche de dicha dosis. Cada

ECN App

localhost:4200/#/agregarPaciente

ECN App Pacientes Activos Pacientes Inactivos Estadísticas Añadir paciente Cerrar Sesión

### Información básica

Nombre:

Apellidos:

Fecha de nacimiento:

Sexo:

Semanas de gestación:

Peso al nacer:

Figura 6: Formulario para agregar un paciente.

ECN App

localhost:4200/#/iniciarProtocolo/-Mj0brXUE02IW14shITf

ECN App Pacientes Activos Pacientes Inactivos Estadísticas Añadir paciente Cerrar Sesión

### Requisitos básicos para iniciar el protocolo alimenticio

¿Estabilidad hemodinámica? (presión sanguínea y frecuencia cardíaca normales)  Sí  No

¿Perfusión estable durante las últimas 12 horas?  Sí  No

¿Ausencia de asfixia?  Sí  No

¿Peso inferior a 1500 g o menos de 32 semanas? (Peso del paciente: 1000 g, Semanas del paciente: 28 semanas)  Sí  No

¿Calostro o leche materna disponible?  Sí  No

¿Crecimiento intrauterino retardado? (CIR)  Sí  No

¿Catéter umbilical?  Sí  No

Figura 7: Formulario para iniciar la alimentación protocolizada.

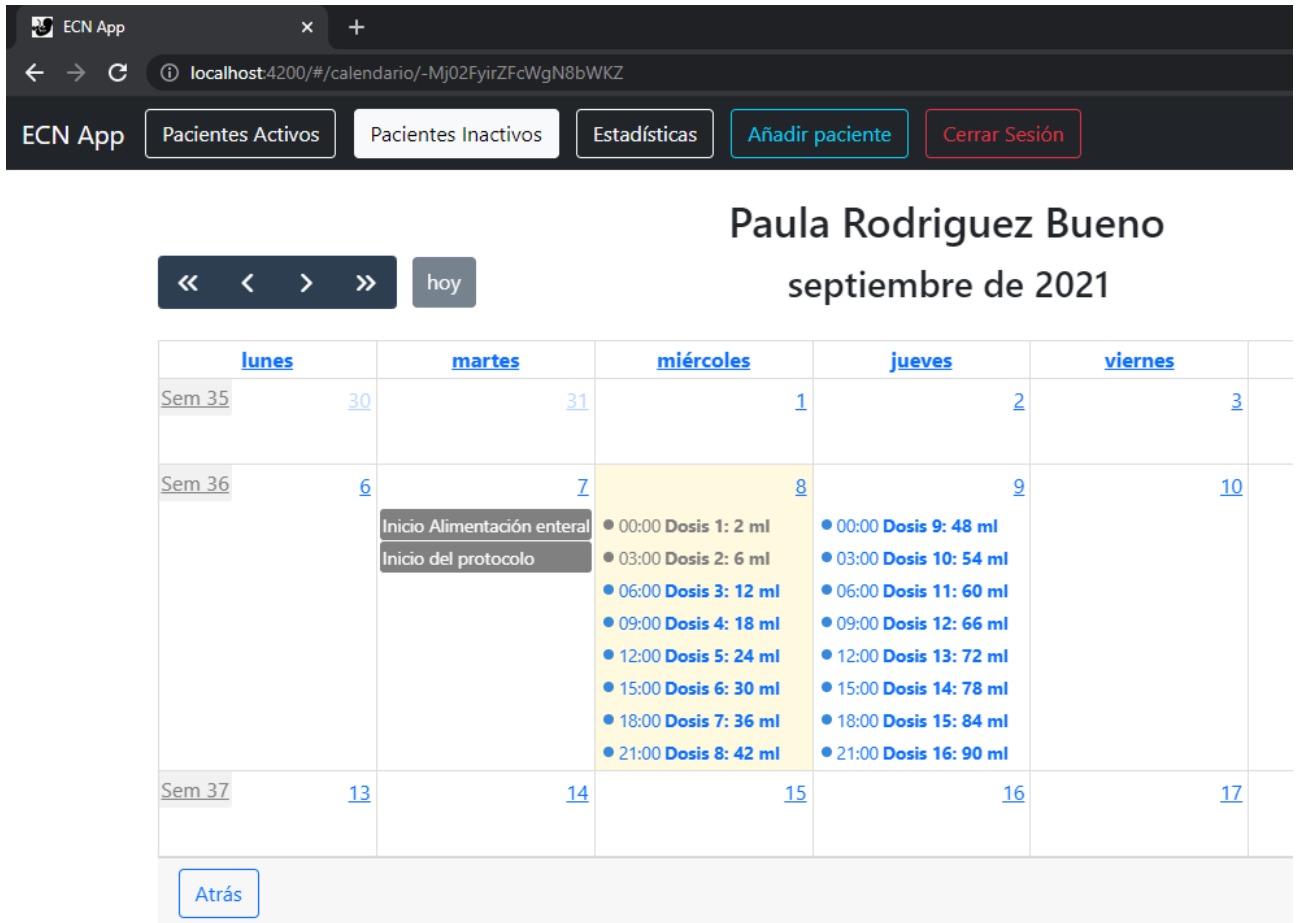


Figura 8: Vista mensual del calendario de la alimentación protocolizada.

evento de dosis es clicable; al hacerlo se despliega una pestaña para registrar la evolución del paciente hasta antes de esa ingesta (figura 9). En esta pestaña se puede indicar si el paciente ha sufrido vómitos, si hace sus deposiciones de manera correcta o si tiene el abdomen de un tamaño normal; los cambios guardados en esta pestaña se pueden consultarlos en la página de información del paciente correspondiente.

#### 4.2.6. Estadísticas

En la figura 10 se observa la pestaña de Estadísticas, donde podemos consultar diferentes gráficos, como un gráfico circular con la clasificación de los pacientes según su progreso en el tratamiento (no finalizado aún, finalizado exitosamente, fallecidos e inacabados), otro gráfico circular con su clasificación según el peso en el que se encuentren (<750g , 750-1000 g, 1000-1500 g y >1500 g), un diagrama de barras que indica el número de semanas de gestación de los pacientes, entre otras gráficas (RF-08). Aparte de mostrar los gráficos, al clicar sobre el botón de Estadísticas el usuario descarga automáticamente un fichero .json donde se encuentra la información de los pacientes de la base de datos hasta ese momento (RF-09). De esta forma, la información puede analizarse fácilmente con software focalizado al estudio estadístico, como podría ser R, por ejemplo.

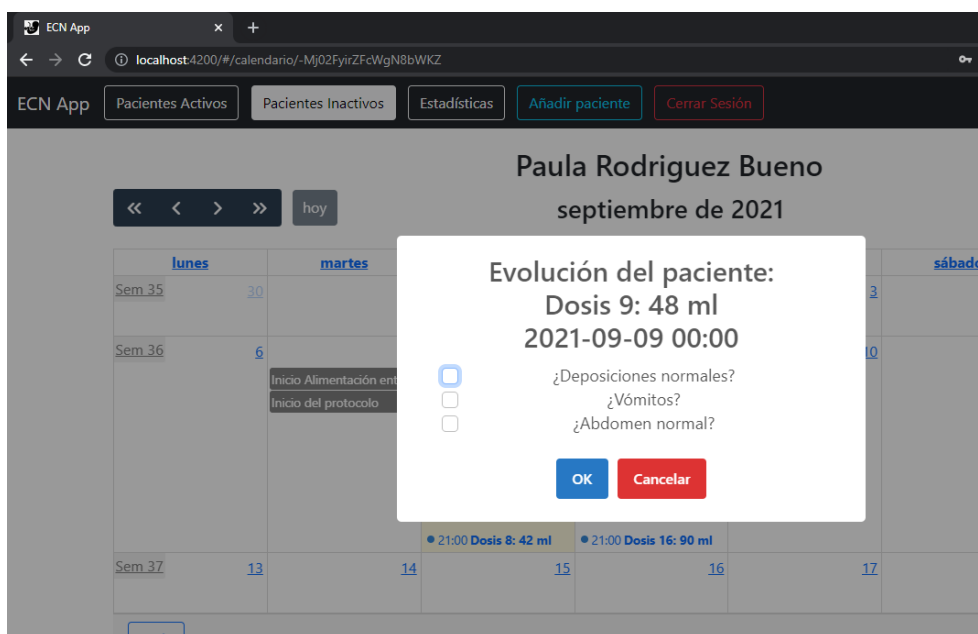


Figura 9: Pestaña para registrar la evolución de un paciente respectiva a una dosis concreta.

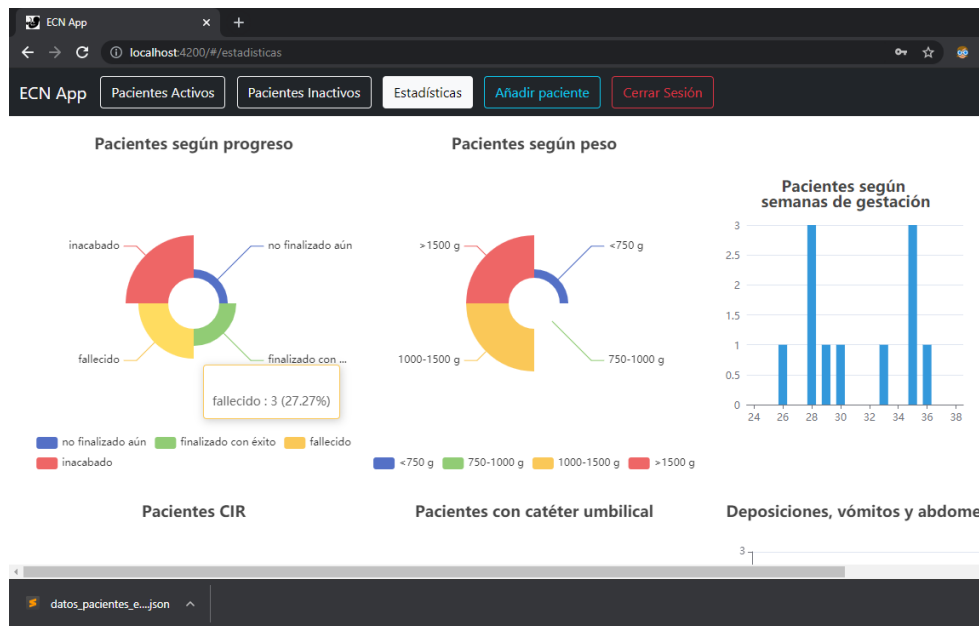


Figura 10: Pestaña para consultar las estadísticas de los pacientes de la aplicación.

## 4.3. Fase 2: *backend* de la aplicación

### 4.3.1. Arquitectura de la aplicación

En la figura 11 se muestran los componentes que conforman la arquitectura de la aplicación; el usuario interactúa con una aplicación web implementada en Angular, que a su vez hace uso de distintos recursos como son Bootstrap[22], con el que se puede diseñar el aspecto de la web con facilidad; ngx-charts[10], que permite incorporar tablas, gráficas, diagramas, etc. y hacerlos interactivos con el usuario; y FullCalendar[11], con el que se consigue incorporar un calendario interactivo en el que añadir el protocolo alimenticio. Además, Angular hace uso de NodeJS[4] para comunicarse con la base de datos no relacional en tiempo real almacenada en Google Firebase[3].

### 4.3.2. Orientación a los servicios

Uno de los primeras partes que se deben implementar en un proyecto de Angular de estas características es un servicio; Angular está orientada a los servicios, lo cual permite separar cómodamente los componentes de la aplicación según si dependen del lado del cliente o del lado del servidor. Los servicios permiten obtener información y transmitirla al resto de componentes de un proyecto; de esta forma se mejora la modularidad y reusabilidad del trabajo.

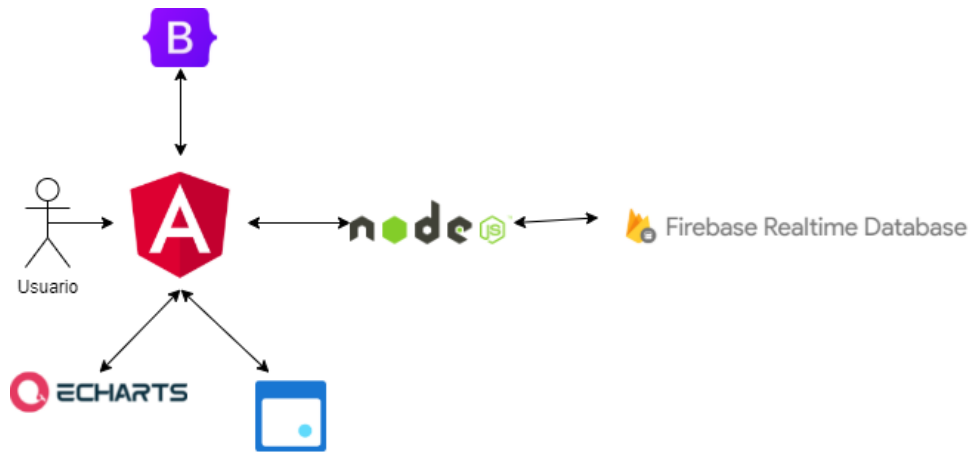


Figura 11: Arquitectura de la aplicación web de este proyecto.

El servicio se encarga de realizar las operaciones CRUD (crear pacientes, leer información de pacientes, actualizar información de pacientes y borrar información de pacientes) sobre la base de datos y de preparar la información de los pacientes para su consulta. Los componentes que necesiten de estos datos hacen peticiones al servicio y este se los proporciona, haciendo cada componente uso de los datos del servicio como se desee. La interacción entre el servicio de la aplicación y la base de datos permite satisfacer los requisitos RF-01, RF-02, RF-03, RF-04 y RF-05.

#### 4.3.3. Base de datos Firebase en tiempo real (Firebase realtime database)

Para almacenar los datos de los pacientes se ha decidido por emplear una base de datos en tiempo real del servicio Firebase de Google. La información de este tipo de base de datos se almacena como un objeto tipo JSON en la nube de Google. A diferencia de una base de datos relacional, en esta base de datos no hay tablas ni relaciones entre ellas, sino que existe un único JSON que actúa como un árbol, en el que cada rama se corresponde con un elemento de la base de datos, en este caso pacientes. Cada elemento/paciente recibe al entrar en la base de datos un identificador único con el que acceder a su información, que se encuentra anidada; Firebase proporciona un máximo de hasta 32 niveles de anidación de datos[3], más que suficiente para los requisitos de esta aplicación. La flexibilidad de un formato como JSON hace que tanto la creación de los pacientes como la actualización de su información sea un proceso fácil de gestionar. Otra de las ventajas de esta base de datos es la instantaneidad que proporciona; como su nombre indica, los cambios que realicemos en ella se sincronizan en tiempo real gracias a

la eficacia de los servidores de Google, obteniéndose así la versión más actualizada de la base de datos cada vez que se realice una acción sobre ella (RNF-05). En la figura 12 se muestra la estructura del JSON que compone la base de datos del proyecto que se ha descrito.



Figura 12: Estructura de la base de datos Firebase en tiempo real.

#### 4.3.4. Autenticación de Firebase (Firebase Authentication)

Además de la base de datos en tiempo real, Google Firebase proporciona un sistema de autenticación para las aplicaciones que utilicen su sistema gestor de base de datos. Este sistema de autenticación permite a los usuarios identificarse de las siguientes maneras: mediante correo electrónico y contraseña, mediante un número de teléfono, mediante accesos anónimos o mediante identidades ya federadas en otro servicio, como pueden ser Google, Facebook, Twitter, Microsoft, etc. Para esta aplicación se ha optado por el método más tradicional, el de un correo electrónico y contraseña. Esta elección se debe al perfil de usuario al que va dirigida esta aplicación, que no es otro que el de personal sanitario de un hospital, encargado de la salud de los recién nacidos en una unidad de neonatología. Por lo tanto, se sabe con anterioridad qué personas que tendrán acceso a la aplicación y qué personas no lo tendrán. Asimismo, se distinguen dos usuarios de la aplicación: el ya citado personal sanitario, que es al que va dirigida la aplicación, y el administrador, que además de poder realizar todas las acciones de un usuario

ordinario puede gestionar el acceso a la aplicación y debe de agregar la información de los usuarios al sistema de Firebase antes de que estos quieran hacer uso de la aplicación (RNF-03). Firebase suministra un sistema de agregación de usuarios con cualquiera de los métodos que se han mencionado anteriormente; también permite controlar restablecer las contraseñas de los usuarios existentes y llevar un registro de los accesos a la aplicación. La forma en la que se facilita acceso a la aplicación es la siguiente: en el momento en el que un usuario entra en la aplicación y se logea, el servicio de Angular manda una petición al endpoint de Firebase encargado de comprobar los accesos de usuarios existentes. Una vez Firebase ha comprobado que la información del usuario es correcta, Firebase crea y asigna un token único al usuario; dicho token permite identificar las acciones del usuario y le proporciona acceso a los recursos de Firebase, entre ellos la firebase realtime database a la que queremos acceder. Una vez decidimos salir de la aplicación, el token que teníamos asignado es destruido y ya no puede volver a ser utilizado para entrar en la aplicación (RNF-04).

#### **4.3.5. Sistema de rutas (Routing) y ventanas emergentes (Pop-ups)**

Como se comentaba en el apartado de tecnologías, Angular permite generar aplicaciones de una sola página o Simple Page Applications (SPAs), que permiten navegar por las distintas pestañas y/o páginas de la aplicación sin tener que refrescar el navegador. Para conseguir esta navegación fluida debemos asignar a cada componente de la aplicación (inicio de sesión, pacientes activos, añadir un paciente, etc.) una ruta relativa dentro de la misma aplicación. De esta forma se consigue rebajar la carga del servidor ya que la navegación se realiza en el lado del cliente: cuando el usuario desea cambiar a otra pestaña, no es necesario ninguna petición al servidor, lo que mejora la velocidad y la experiencia del usuario (RNF-01). Por otro lado, cada una de estas rutas está protegida del acceso mediante la autenticación de Firebase; si no se ha iniciado sesión, ninguna de las rutas será accesible desde el navegador.

En esta aplicación se establece una ruta para cada uno de los siguientes componentes:

- Pacientes.
- Pacientes inactivos.
- Estadísticas.

- Inicio de sesión.
- Agregar paciente.
- Editar paciente.
- Inicio del protocolo.
- Calendario.

Existen acciones en esta aplicación que no necesitan de rutas para su funcionamiento: para cerrar sesión, para registrar la evolución de los pacientes y para finalizar el protocolo alimenticio se emplean ventanas emergentes o pop-ups en lugar de rutas propias. El uso de estas ventanas emergentes permite añadir información a los pacientes o realizar otras acciones aún más rápidamente que con las rutas, además de ser un recurso visual llamativo que se integra con el resto de la pestaña de la aplicación. En la figura 13 se ve un ejemplo de una ventana emergente empleada para cerrar la sesión en la aplicación.

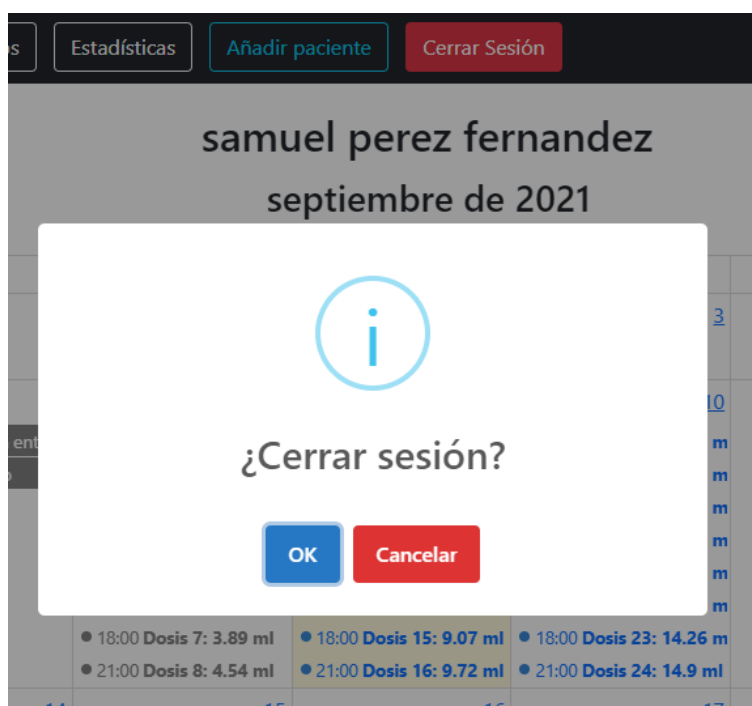


Figura 13: Ejemplo de ventana emergente para cerrar sesión.

## 4.4. Fase 3: Lógica de la aplicación y algoritmos

En este apartado se explica la lógica interna de la aplicación, es decir, las funciones y/o algoritmos que se han desarrollado para poder llevar a cabo el protocolo alimenticio de cada paciente (RF-07). En primer lugar, es necesario completar el formulario de inicio para poder comenzar la alimentación. Una vez que los datos necesarios para el protocolo están definidos en el paciente, es posible generar el protocolo; este se carga en el lado del cliente, en el componente del calendario. Al entrar en la pestaña de calendario, el componente carga la función `calcularProtocolo()`, que a partir de los datos del paciente añade una lista de eventos con las ingestas correspondientes al calendario en el que se ha entrado. La función `calcularProtocolo()` llama a otras dos funciones para calcular estos eventos:

- `generarTrofica()`, que calcula las dosis que corresponden al periodo de NET.
- `generarCompleta()`, que calcula las dosis que equivalen al intervalo de NEC.

### 4.4.1. Cálculo de la Nutrición Enteral Trófica (NET)

La función `generarTrofica()`, mediante los datos del paciente (en concreto, peso del paciente, semanas de gestación y si es CIR o no) selecciona la cantidad de días que va a durar la NET. En pacientes de alto riesgo su duración es de 7 días, en pacientes de riesgo medio es de 6 días y en pacientes de riesgo bajo es de 5 días. A continuación, esta función llama a otras 3 funciones con las que se calcula la cantidad de las dosis; `calcularAltoRiesgo()`, `calcularRiesgoMedio()` y `calcularRiesgoBajo()`. Dichas cantidades se calculan teniendo en cuenta lo planteado en la fase inicial de este capítulo. La frecuencia de las tomas se realiza cada 3 horas, ya que es necesario estimular el sistema digestivo del paciente de manera gradual.

- `calcularRiesgoAlto()` calcula las dosis para los pacientes que se encuentren en alto riesgo.
- `calcularRiesgoMedio()` calcula las dosis para los pacientes que se encuentran en riesgo medio.
- `calcularRiesgoBajo()` calcula las dosis para los pacientes que se encuentran en riesgo bajo. Los pacientes en riesgo bajo que presenten un registro de evolución favorable durante el tratamiento (abdomen normal, deposiciones normales, ausencia de vómitos) disminuirán la duración de su intervalo de NET de 5 días a 3 días.

#### **4.4.2. Cálculo de la Nutrición Enteral Completa (NEC)**

La función generarCompleta(), teniendo en cuenta el peso del paciente y la cantidad de leche que ha ingerido en las tomas anteriores (NET), genera las dosis que le quedan por consumir hasta alcanzar la cantidad máxima recomendada de 160 ml. Este cálculo depende del peso del paciente y de la cantidad de leche consumida anteriormente; por lo tanto, la duración de la NEC está condicionada por las características del paciente y la NET que haya recibido. Como se describía en la fase inicial, los incrementos de la NEC son mayores que los de la NET, por lo que cuanto mejor sean las condiciones del paciente menor será la duración de este periodo.

#### **4.4.3. Registro de la evolución del paciente**

Para registrar los cambios en el pacientes se emplea la función EvolucionDelPaciente(), que despliega una ventana emergente sobre la dosis en la que el usuario haya clicado, mostrando los campos en los que se puede mejorar (abdomen normal, deposiciones normales, ausencia de vómitos, etc.). La evolución del paciente, ya sea favorable o no, queda registrada en la base e datos y puede consultarse en el apartado de información del paciente. Si la evolución es favorable y el paciente pertenece al grupo de riesgo bajo, su periodo de NET se reduce.

#### **4.4.4. Carga del calendario de ingestas**

Una vez generada la lista de eventos para mostrar en el calendario, esta se carga en dicho calendario a través de la función actualizarEventos(). Esta función se encarga de cargar los datos de los eventos, alternar las distintas vistas del calendario, colorear de gris los eventos cuya fecha y/o hora haya pospuesto y de azul los eventos aún por cumplir, así como calcular el intervalo de días de tratamiento que se muestra en el calendario para no sobrecargar de información al usuario.

# 5

## Conclusiones y Líneas Futuras

### 5.1. Conclusiones

Como consecuencia del proyecto realizado y todo el trabajo llevado a cabo, se han extraído una serie de conclusiones que se exponen a continuación. En primer lugar, la redacción de los requisitos de la aplicación ha sido una actividad enriquecedora, ya que al ser un problema de carácter biológico/sanitario que afecta a recién nacidos, ha sido necesario una investigación previa sobre la que cimentar la información necesaria. Dicha investigación ha ayudado a mejorar la comprensión del trastorno que se pretende solventar; en estas situaciones es importante la figura de un bioinformático o ingeniero de la salud, una persona que aún sin ser un experto en ambos campos, pueda entender tanto la parte tecnológica como la parte médica, siendo capaz de trasladar las necesidades biológicas de un ser humano a los requisitos de un software.

Por otra parte, la metodología empleada ha permitido completar los requisitos propuestos, gracias en gran parte a la simbiosis con el tipo de tecnologías empleadas, como Angular. Esta coordinación ha posibilitado la agrupación del desarrollo de las iteraciones en fases distintas y bien definidas, presentadas en el capítulo de Desarrollo e Implementación. Gracias a esta separación de tareas la fluidez y eficiencia del proyecto han sido mayores. Se puede extraer entonces que para el progreso de un proyecto que involucra desarrollo software como es este caso, es tan importante la elección de una metodología adecuada como unas tecnologías apropiadas que acompañen el ritmo impuesto por dichos métodos.

Siguiendo con las tecnologías empleadas, los Trabajos de Fin de Grado son una buena

oportunidad para salir de la zona de confort y aventurarse a usar software desconocido y/o novedoso. En este caso, frameworks de desarrollo web como Angular, el uso de bases de datos no relacionales, extensiones como FullCalendar, etc. eran herramientas ignotas al comienzo de este proyecto, pudiendo resultar en conjunto apabullantes, pero una vez se ha superado la curva de dificultad inicial de cada programa se pueden alcanzar grandes avances en la calidad del desarrollo del software. No obstante, la parte más exigente a desarrollar en esta aplicación no ha sido la parte relativa a las nuevas tecnologías utilizadas, sino sin lugar a duda diseñar los algoritmos con los que adecuar la alimentación a las necesidades de los pacientes y su integración con el resto de la aplicación, ya que es la parte más compleja del proyecto a nivel computacional, además de ser única en sí misma.

## **5.2. Líneas Futuras**

En cuanto a líneas futuras de trabajo, existen diversas características del proyecto las cuales pueden ampliarse y otras que no han sido desarrolladas pero que mejorarían la experiencia de uso de esta aplicación. Una es, por ejemplo, añadir un sistema de alertas, ya sea en el mismo ordenador o mediante una aplicación móvil, que permita avisar al usuario de que la hora de una ingesta está próxima o de que se debería registrar la evolución del paciente en cuestión. Estas alertas podrían añadirse automáticamente al calendario del ordenador o teléfono, o presentarse como notificaciones.

Otra propiedad que podría mejorarse es la presentación del calendario de ingestas de cada paciente; en la aplicación actual cada paciente posee un calendario de ingestas propio, donde se indican sus tomas de manera individual. Sin embargo, podría implementarse un calendario global, donde se incluyeran la información relativa a todos los pacientes activos en ese momento. De esta forma, se podría visualizar de manera rápida y conjunta el estado de cada paciente y el progreso que siguen.

La fortificación de la leche también es otro aspecto a tener en cuenta en futuras revisiones; tanto el uso de fortificante como el de probióticos forman parte del protocolo de alimentación y se mantienen habitualmente una vez terminado los riesgos de ECN. Por lo tanto, podría incluirse al calendario de cada paciente la cantidad de dosis fortificadas y su frecuencia.

La distinción de roles dentro de los usuarios de la aplicación también podría ser un aspecto de desarrollo; en concreto, podría diferenciarse el papel de encargado, que sería responsable de la introducción de los pacientes en la base de datos de la aplicación, así como de iniciar su alimentación protocolizada, y el papel de enfermero, que se ocuparía de administrar las dosis a los pacientes y controlar la evolución de los mismos.



# Referencias

- [1] LLC Balsamiq Studios. Balsamiq wireframes. <https://balsamiq.com/wireframes/>. 2021.
- [2] Jorge Cano. Angular: Mucho más que un framework. <https://sg.com.mx/revista/56/angular>. mayo 2018.
- [3] Google Developers. Firebase. <https://firebase.google.com/?hl=es-419>. 2021-03-25.
- [4] OpenJS Foundation. acerca de nodejs. <https://nodejs.org/es/about/>. Accessed: 2021-08-31.
- [5] Ms Sheila M Gephart, Jacqueline M McGrath, Judith A Effken, and Melissa D Halpern. Necrotizing enterocolitis risk: state of the science. *Advances in Neonatal Care*, 12(2):77, 2012.
- [6] Git. About git. <https://git-scm.com/about>. Accessed: 2021-09-2.
- [7] GitHub. Features. <https://github.com/features>. Accessed: 2021-09-2.
- [8] X Demestre Guasch and F Raspall Torrent. Enterocolitis necrosante. *Servicio de pediatría y neonatología. Protocolos Diagnostico Terapéuticos de la Asociación Española de Pediatría: Neonatología. SCIAS. Hospital de Barcelona. Grup Assistencia. Barcelona. Citado*, pages 26–05, 2008.
- [9] Jack Jacob, Michael Kamitsuka, Reese H Clark, Amy S Kelleher, and Alan R Spitzer. Etiologies of nicu deaths. *Pediatrics*, 135(1):e59–e65, 2015.
- [10] Freaky Jolly. Echarts for angular charts using ngx-echarts tutorial with examples. <https://www.freakyjolly.com/angular-e-charts-using-ngx-echarts-tutorial/>. 2021-07-25.
- [11] FullCalendar LLC. Fullcalendar. <https://fullcalendar.io/>. 2021.
- [12] Microsoft. Getting started. <https://code.visualstudio.com/docs>. 2021-05-8.

- [13] Josef Neu and W Allan Walker. Necrotizing enterocolitis. *New England Journal of Medicine*, 364(3):255–264, 2011.
- [14] Michael Obladen. Necrotizing enterocolitis–150 years of fruitless search for the cause. *Neonatology*, 96(4):203–210, 2009.
- [15] Overleaf. Learn latex in 30 minutes. [https://es.overleaf.com/learn/latex/Learn\\_LaTeX\\_in\\_30\\_minutes](https://es.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes). Accessed: 2021-09-1.
- [16] Bhoomika K Patel and Jigna S Shah. Necrotizing enterocolitis in very low birth weight infants: a systemic review. *International Scholarly Research Notices*, 2012, 2012.
- [17] The Latex Project. An introduction to latex. <https://www.latex-project.org/about/>. Accessed: 2021-09-1.
- [18] Tomás Sánchez-Tamayo, María G Espinosa Fernández, María C Moreno Algarra, Verónica Fernández Romero, José Vallejo Triano, Elías Tapia Moreno, and Enrique Salguero García. Nueva guía de práctica clínica sobre nutrición enteral del recién nacido de muy bajo peso al nacimiento: primera parte. *Nutrición Hospitalaria*, 30(2):321–328, 2014.
- [19] Tomás Sánchez-Tamayo, María Gracia Espinosa Fernández, Laura Affumicato, María González López, Verónica Fernández Romero, María Concepción Moreno Algarra, and Enrique Salguero García. Reducción de la enterocolitis necrosante tras la introducción de un protocolo de alimentación enteral basado en la evidencia en recién nacidos de muy bajo peso. In *Anales de Pediatría*, volume 85, pages 291–299. Elsevier, 2016.
- [20] Holger Schmitz. Build a basic crud app with angular and node. <https://developer.okta.com/blog/2018/10/30/basic-crud-angular-and-node>. 2021-04-12.
- [21] Ian Sommerville. Ingeniería del software séptima edición. *I. Sommerville, Ingeniería del software Séptima edición*, 2005.
- [22] David Suárez. Qué es bootstrap y cómo usarlo. <https://raiolanetworks.es/blog/bootstrap/>. 2021-08-14.
- [23] LLC Taiga Agile. About taiga. <https://www.taiga.io/es>. Accessed: 2021-09-7.

# Apéndice A

## Manual de Instalación

Para poder utilizar la aplicación es necesario instalar primero NodeJS, que incluye el gestor de paquetes npm; ambos se pueden descargar desde la página web de NodeJS. La versión de NodeJS empleada en esta aplicación es la versión 14.16.0:

<https://nodejs.org/en/download/>

A continuación es necesario Angular CLI, una herramienta que permite gestionar aplicaciones de Angular mediante líneas de comando. La versión de Angular CLI empleada para el desarrollo de esta aplicación es la versión 11.2.6 y la versión de Angular es la versión 11.2.7. Se puede instalar siguiendo los pasos de su página web oficial:

<https://angular.io/guide/setup-local>

Además es necesario instalar los paquetes de FullCalendar y Ngx-charts; estos paquetes se pueden instalar fácilmente con los comandos de Angular CLI, descritos respectivamente en los siguientes enlaces:

<https://fullcalendar.io/docs/angular>

<https://xieziyu.github.io/ngx-echarts//welcome>

Para arrancar la aplicación simplemente se abre una línea de comandos en la ubicación de la aplicación de angular y se introduce el comando `ng serve -o`. Este comando abre la aplicación web en el navegador que el usuario tenga predeterminado; Google Chrome fue el navegador empleado para el testeo de esta aplicación, en su versión 93.0.4577.63, por lo tanto es el que se

recomienda para su mismo uso.

Podemos encontrar la última versión del proyecto para descargar, así como las versiones anteriores, en el siguiente repositorio público de Github:

<https://github.com/samuUma/TFGSamuel>

Por otro lado, para poder acceder a la aplicación es necesario ponerse en contacto con el administrador de la aplicación para que este dé acceso al correo del usuario que desee emplear la aplicación.

# Apéndice B

# Manual de usuario

En este apéndice se detalla la forma de utilizar la aplicación y sus diferentes funcionalidades. Las acciones que se describen a continuación siguen el orden en el se deben realizar para el correcto funcionamiento de la aplicación.

## B.1. Inicio de sesión

La primera página de nuestra aplicación es la página para iniciar sesión (figura 2). En esta página se debe completar los campos de correo electrónico y contraseña para entrar en la aplicación (figura 14); antes el usuario debe estar registrado por parte del administrador.

### Inicio de sesión

Correo Electrónico

samupf97@gmail.com

Contraseña

.....

Ingresar

Figura 14: Formulario de inicio de sesión.

## B.2. Pacientes activos

Una vez se ha iniciado sesión correctamente, el usuario accede a la página principal de la aplicación, que es la página de pacientes activos (figura 3). Desde esta página el usuario puede ver una lista de los pacientes activos de la aplicación (pacientes que no han comenzado aún el protocolo alimenticio o pacientes que se encuentran realizando el protocolo alimenticio). Cada paciente presenta las siguientes opciones:

- **Más información:** para consultar los datos del paciente, ya sean relativos al propio paciente, al protocolo alimenticio que está siguiendo o a la evolución de su estado en dicho protocolo.
- **Editar paciente:** para editar la información básica relativa al paciente.
- **Iniciar protocolo:** para iniciar la alimentación protocolizada en el paciente (opción solo disponible si el paciente no ha iniciado aún su alimentación).
- **Calendario:** para consultar las ingestas, los horarios de las mismas y para registrar la evolución de los pacientes (opción solo disponible si el paciente ya ha iniciado su alimentación).
- **Finalizar protocolo:** para indicar que el paciente ya ha terminado la alimentación protocolizada, pasando de activo a inactivo (opción solo disponible si el paciente ya ha iniciado su alimentación).

Además podemos consultar desde la barra superior de navegación las siguientes ventanas:

- **Pacientes inactivos:** para ver los pacientes que han finalizado el protocolo alimenticio.
- **Estadísticas:** para consultar los datos de manera gráfica o obtener un fichero con los mismos para su estudio posterior.
- **Añadir paciente:** para agregar un paciente nuevo a la base de datos de la aplicación.
- **Cerrar sesión:** para terminar la sesión del usuario y cerrar la aplicación.

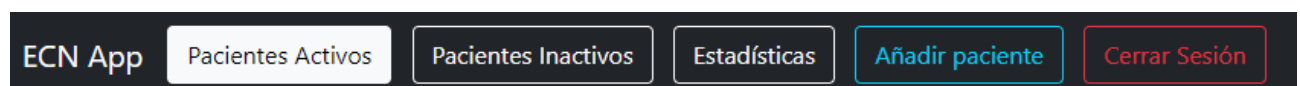


Figura 15: Barra de navegación que da acceso a las distintas páginas de la aplicación.

### B.3. Añadir paciente

Desde esta página se pueden añadir pacientes nuevos a la aplicación (figura 6). Para ello se debe rellenar un formulario con la información básica del paciente:

- Nombre y apellidos.
- Fecha de nacimiento.
- Sexo del paciente.
- Semanas de gestación del embarazo.
- Peso al nacer, en gramos.

**Información básica**

---

Nombre:	<input type="text" value="Nombre"/>
Apellidos:	<input type="text" value="Apellidos"/>
Fecha de nacimiento:	<input type="text" value="dd/09/2021"/>
Sexo:	<input type="text"/>
Semanas de gestación:	<input type="text"/>
Peso al nacer:	<input type="text" value="Peso al nacer"/>

Figura 16: Formulario para añadir un paciente nuevo a la aplicación.

Cuando el formulario se ha rellenado correctamente se habilita la opción de guardar los datos, redirigiendo al usuario a la página de pacientes activos, donde ahora aparecerá el paciente que se acaba de añadir.

## **B.4. Iniciar protocolo alimentario**

Para iniciar el protocolo alimentario de un paciente se debe clicar en el botón de iniciar protocolo, dentro de la tarjeta del paciente correspondiente, en la página de pacientes activos (figura 3). A continuación se debe rellenar el formulario necesario para iniciar la alimentación (figura 17), el cual contiene los siguientes campos:

- Chequeo de la estabilidad hemodinámica del paciente.
- Chequeo de la perfusión estable del paciente durante las últimas 12 horas.
- Chequeo de ausencia de asfixia.
- Chequeo del peso del paciente y de las semanas de gestación que ha tenido.
- Presencia de leche materna o calostro al iniciar la alimentación.
- Presencia de crecimiento intrauterino retardado (CIR).
- Presencia de un catéter umbilical.

Cuando el formulario se ha rellenado correctamente se habilita la opción de guardar los datos, redirigiendo al usuario a la página de pacientes activos, donde ahora aparecerá en el paciente correspondiente la opción de visualizar el calendario de ingestas y la opción de finalizar el protocolo, desapareciendo la opción de iniciar el protocolo.

## **B.5. Consultar calendario de ingestas y evolución del paciente**

Una vez se ha iniciado el protocolo alimenticio del paciente la opción de visualizar su calendario de ingestas pasa a estar disponible (figura 8). El calendario tiene diferentes vistas disponibles; mes, semana, día y lista, siendo la opción mensual la predeterminada. Al entrar al calendario se cargan los eventos en el mismo, siendo cada evento un mensaje indicando el inicio o final de la NET/NEC o indicando la dosis correspondiente, junto a la cantidad y hora. Conforme avanzan los días, los eventos se van añadiendo automáticamente al calendario. Cada evento asociado a una ingesta es clicable, lanzando la ventana desplegable para registrar la evolución del paciente 9. Los registros de la evolución pueden verse en la pestaña de Más

## Requisitos básicos para iniciar el protocolo alimenticio

---

- ¿Estabilidad hemodinámica? (presión sanguínea y frecuencia cardíaca normales)  Sí  No
- ¿Perfusión estable durante las últimas 12 horas?  Sí  No
- ¿Ausencia de asfixia?  Sí  No
- ¿Peso inferior a 1500 g o menos de 32 semanas? (Peso del paciente: 1000 g, Semanas del paciente: 28 semanas)  Sí  No
- ¿Calostro o leche materna disponible?  Sí  No
- ¿Crecimiento intrauterino retardado? (CIR)  Sí  No
- ¿Catéter umbilical?  Sí  No

Guardar

Atrás

Figura 17: Formulario para iniciar el protocolo alimenticio de un paciente.

información de la tarjeta de cada paciente. Para acceder a cualquiera de las otras pestañas puede usarse la barra superior de navegación o el botón de Atrás, que redirige al usuario a la página principal, pacientes activos.

### B.6. Editar un paciente

Un usuario puede editar la información básica de un paciente ya existente en la aplicación; para ello debe clicar en la opción Editar paciente, que se encuentra en la tarjeta de cada paciente en la lista de pacientes activos. Esta opción despliega un formulario similar al de la pestaña de Añadir paciente, pero en este caso el formulario se encuentra completado con los datos previos del paciente (figura 19). Cuando el usuario haya hecho los cambios pertinentes, simplemente pulsa el botón de guardar para salvar los cambios (siempre y cuando estos cambios sean admisibles).

### B.7. Paciente inactivos

Un paciente activo puede terminar su alimentación protocolizada pulsando en el botón Finalizar protocolo, que se encuentra dentro de su tarjeta. Al pulsarlo se abre una ventana

<a href="#">martes</a>	<a href="#">miércoles</a>	<a href="#">jueves</a>	
<a href="#">31</a>	<a href="#">1</a>	<a href="#">2</a>	
<a href="#">7</a>	<a href="#">8</a>	<a href="#">9</a>	
<div style="background-color: #444; color: white; padding: 2px;">Inicio Alimentación enteral</div> <div style="background-color: #444; color: white; padding: 2px;">Inicio del protocolo</div>	<ul style="list-style-type: none"> <li>● 00:00 <b>Dosis 1: 2 ml</b></li> <li>● 03:00 <b>Dosis 2: 6 ml</b></li> <li>● 06:00 <b>Dosis 3: 12 ml</b></li> <li>● 09:00 <b>Dosis 4: 18 ml</b></li> <li>● 12:00 <b>Dosis 5: 24 ml</b></li> <li>● 15:00 <b>Dosis 6: 30 ml</b></li> <li>● 18:00 <b>Dosis 7: 36 ml</b></li> <li>● 21:00 <b>Dosis 8: 42 ml</b></li> </ul>	<ul style="list-style-type: none"> <li>● 00:00 <b>Dosis 9: 48 ml</b></li> <li>● 03:00 <b>Dosis 10: 54 ml</b></li> <li>● 06:00 <b>Dosis 11: 60 ml</b></li> <li>● 09:00 <b>Dosis 12: 66 ml</b></li> <li>● 12:00 <b>Dosis 13: 72 ml</b></li> <li>● 15:00 <b>Dosis 14: 78 ml</b></li> <li>● 18:00 <b>Dosis 15: 84 ml</b></li> <li>● 21:00 <b>Dosis 16: 90 ml</b></li> </ul>	
<a href="#">14</a>	<a href="#">15</a>	<a href="#">16</a>	

Figura 18: Vista mensual de las dosis de un paciente.

### Información básica

Nombre:	<input type="text" value="prueba6"/>
Apellidos:	<input type="text" value="prueba6"/>
Fecha de nacimiento:	<input type="text" value="10/09/2021"/> 
Sexo:	<input type="text" value="Masculino"/> 
Semanas de gestación:	<input type="text" value="28"/> 
Peso al nacer:	<input type="text" value="1000"/>

Figura 19: Formulario para editar los datos de un paciente ya existente.

preguntando el motivo de la finalización: el protocolo puede que haya acabado exitosamente, puede que el paciente haya fallecido o puede que el sanitario haya decidido dejar de aplicar el tratamiento al paciente (figura 20). Al completar esta decisión el paciente desaparece de la lista de pacientes activos para pasar a la de inactivos.

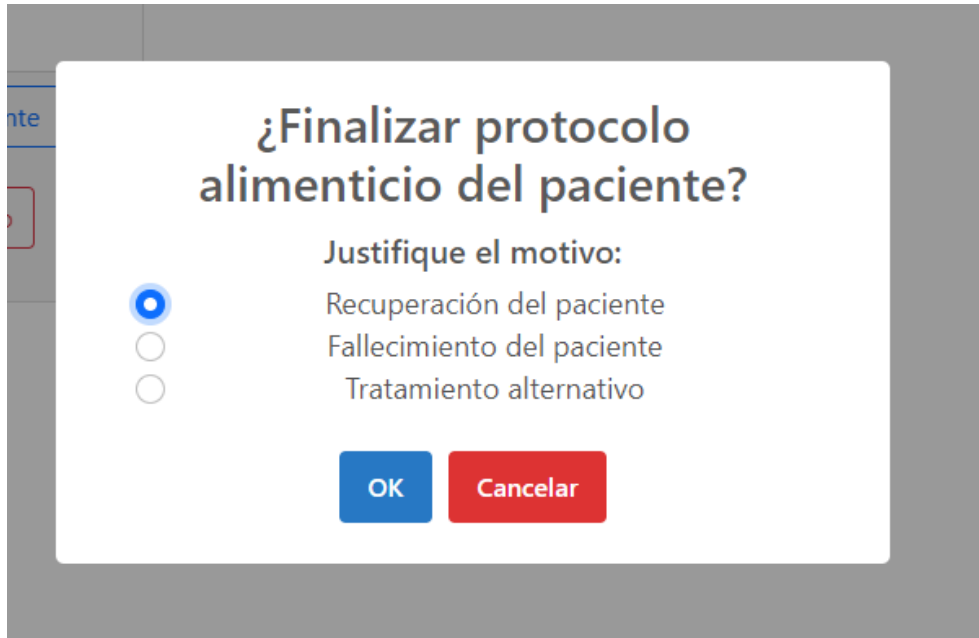


Figura 20: Ventana para confirmar la finalización del protocolo.

## B.8. Más información del paciente

Un usuario puede revisar la información básica y la información de la alimentación de un paciente ya existente en la aplicación; para ello debe clicar en la opción Editar paciente, que se encuentra en la tarjeta de cada paciente en la lista de pacientes activos. Según la situación del paciente respecto a la alimentación, estará disponible una cantidad de información u otra; si no la ha comenzado, solo se podrá ver la información básica. Si la ha empezado, se podrán visualizar los requisitos que se completaron al iniciar la alimentación. Si se han realizado registros de evolución se podrán ver también en esta pestaña (figura 21).

## B.9. Consultar estadísticas de los pacientes

Al pulsar sobre el botón de Estadísticas, el sistema carga 6 gráficas distintas sobre la información de los pacientes tanto activos como inactivos:

Paula Rodriguez Bueno			
Fecha de nacimiento:	2021-09-07		
Sexo:	F		
Semanas de gestación:	30 semanas		
Peso al nacer:	2000 g		
Protocolo alimenticio:	Iniciado		
Fecha de inicio del protocolo:	7/9/2021		
Calostro disponible al iniciar el protocolo:	Sí		
Paciente CIR:	No		
Presencia de catéter umbilical:	No		
Protocolo finalizado:	Sí		
Motivo de finalización del protocolo:	Paciente recuperado exitosamente		
Evolución del paciente	Deposiciones normales	Vómitos	Abdomen normal
Dosis 1: 2 ml, fecha: 2021-09-08 00:00	Sí	Sí	No
Dosis 2: 6 ml, fecha: 2021-09-08 03:00	No	Sí	Sí

Figura 21: Pestaña con la información de un paciente cuya alimentación ha finalizado exitosamente.

- Una gráfica circular donde ver la cantidad de pacientes que han superado la enfermedad exitosamente, cuántos han fallecido, cuántos no finalizaron el tratamiento y cuántos se encuentran aún en el proceso (figura 22).
- Una gráfica circular que clasifica a los pacientes en 4 intervalos de peso: menos de 750 gramos, entre 750 y 1000 gramos, entre 1000 y 1500 gramos y más de 1500 gramos (figura 22).
- Una gráfica circular que divide a los pacientes según si son pacientes CIR o no (figura 23).
- Una gráfica circular que clasifica a los pacientes en si han tenido un catéter umbilical o no (figura 23).
- Una gráfica de barras que permite ver la cantidad de pacientes según las semanas de gestación que han requerido su nacimiento (figura 24).
- Una gráfica de barras donde se puede ver los registros de evolución de los pacientes: la presencia de vómitos, deposiciones normales y la presencia de un abdomen normal

(figura 24).

Además de generarse estas estadísticas, el sistema descarga automáticamente los datos de los pacientes en formato JSON para su posterior estudio estadístico en otro software más especializado en esta cuestión.

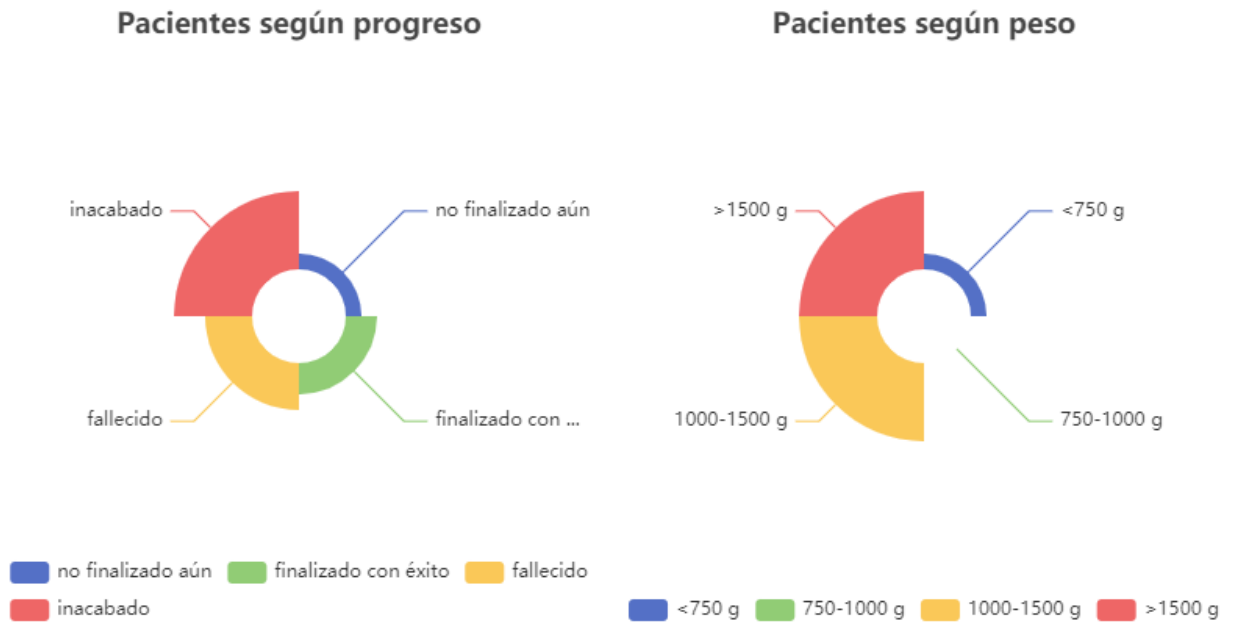


Figura 22: Gráficas que clasifican los pacientes según el progreso y el peso.

## B.10. Cerrar sesión

Para cerrar sesión el usuario simplemente debe pulsar en el botón indicado de la barra de navegación y confirmar su elección (figura 13); una vez cerrada la sesión el sistema se redirige a la página inicial para ingresar en la aplicación (figura 2).

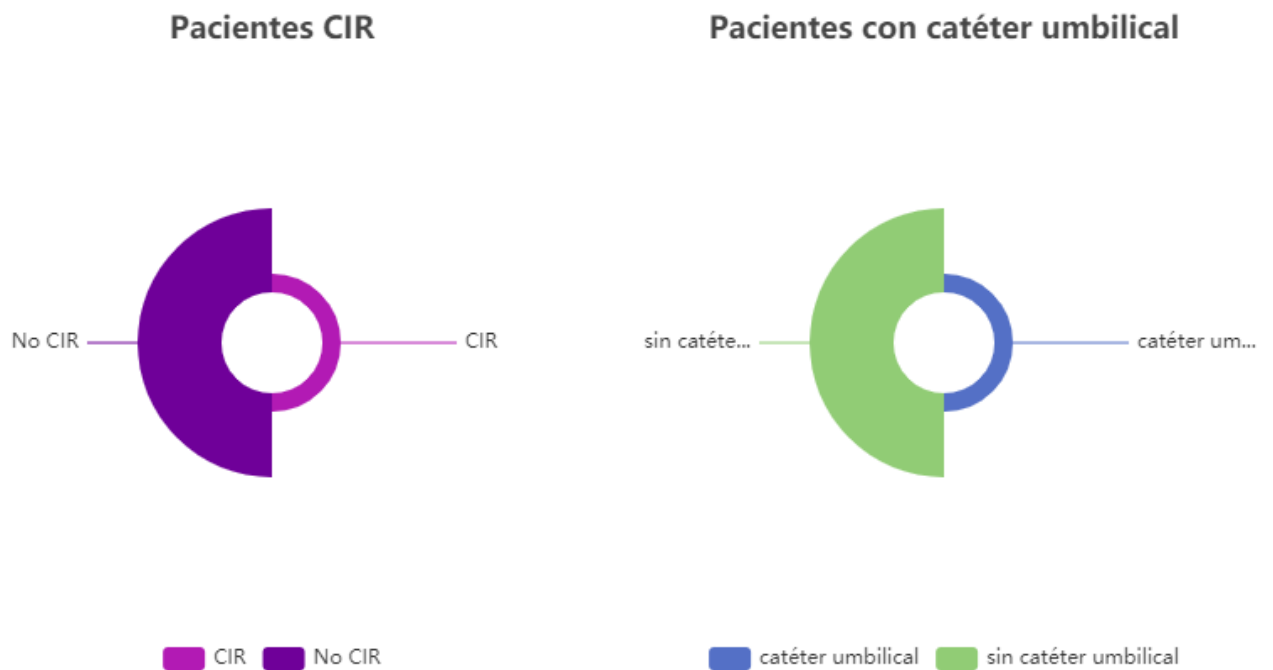


Figura 23: Gráficas que clasifican los pacientes según CIR y catéter umbilical.

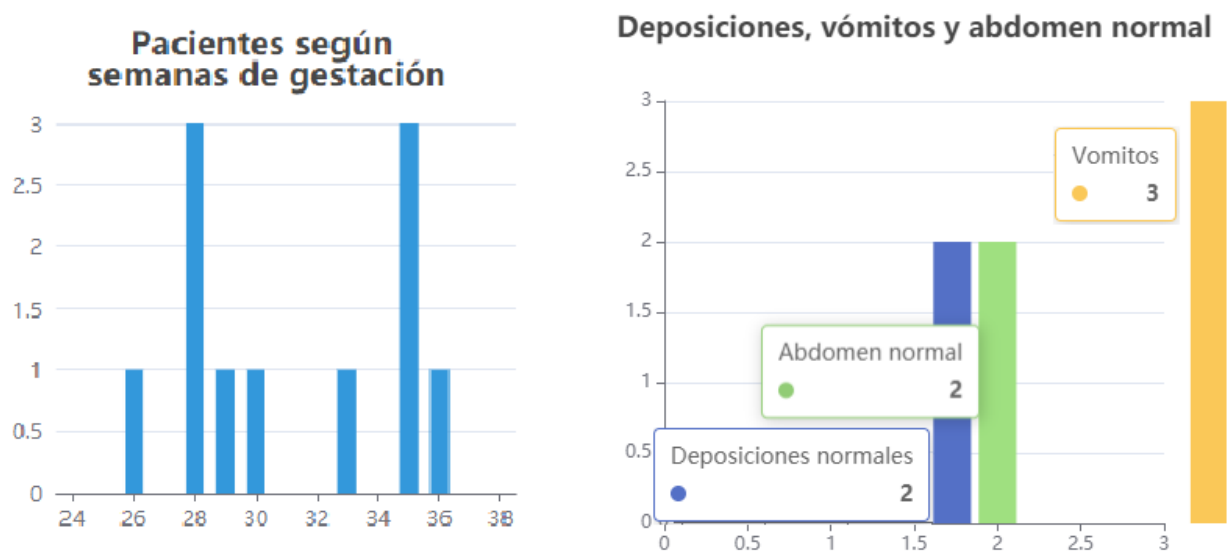


Figura 24: Gráficas que clasifican los pacientes según semanas de gestación y síntomas de evolución.

# Apéndice C

## Diseño de la interfaz en Balsamiq

En este apéndice se incluyen las imágenes correspondientes al diseño de la interfaz de la aplicación con la herramienta Balsamiq; estas imágenes abarcan desde la figura 25 hasta la figura 33.

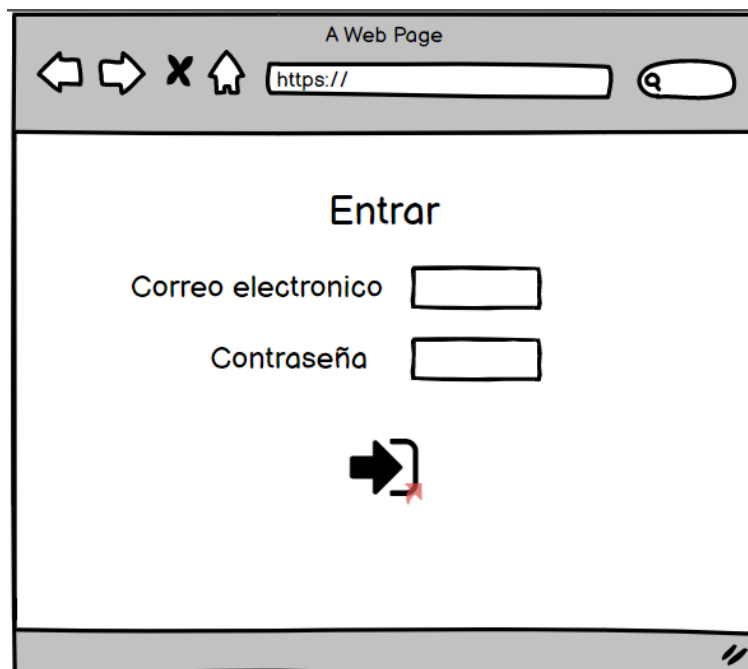


Figura 25: Página para iniciar sesión en la aplicación.

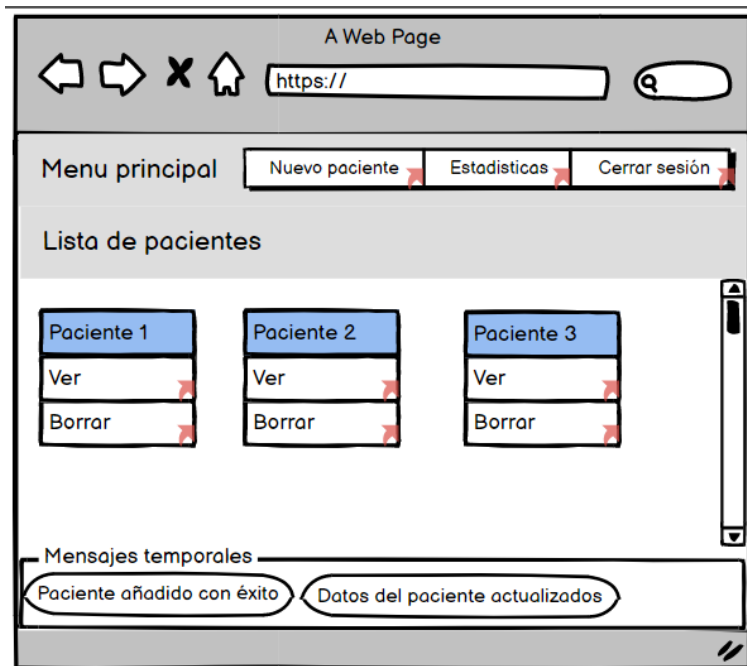


Figura 26: Página principal de la aplicación con la lista de pacientes.

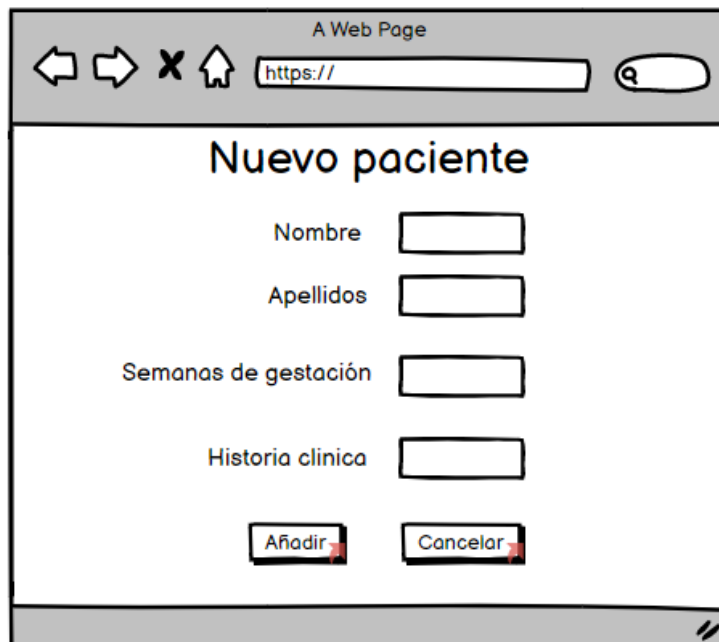


Figura 27: Página para añadir a un nuevo paciente.

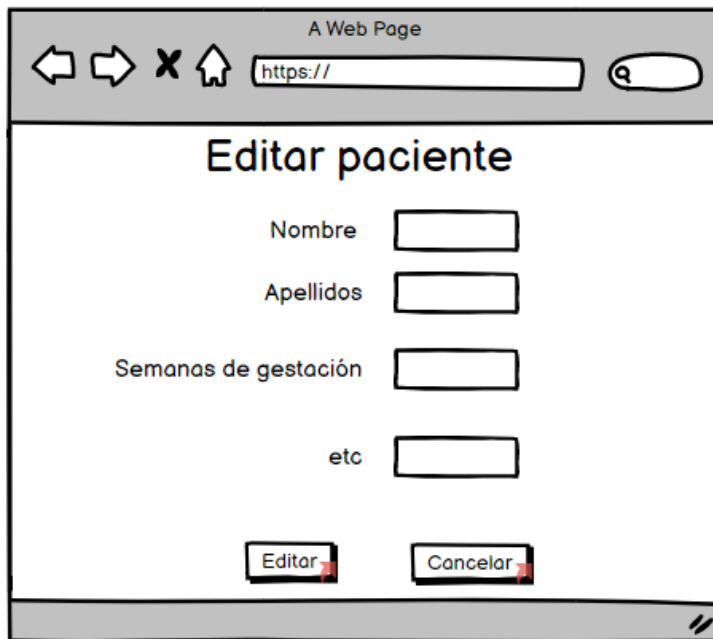


Figura 28: Página para editar los datos de un paciente existente.

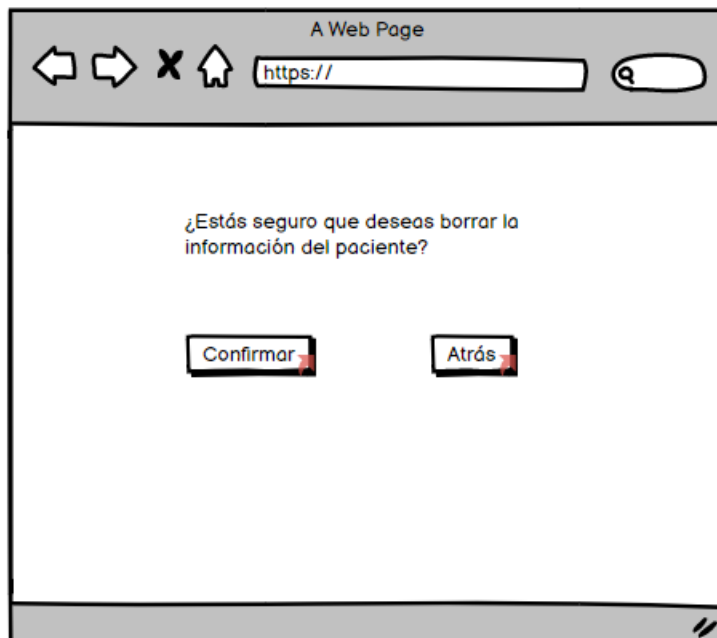


Figura 29: Página para borrar a un paciente de la aplicación.

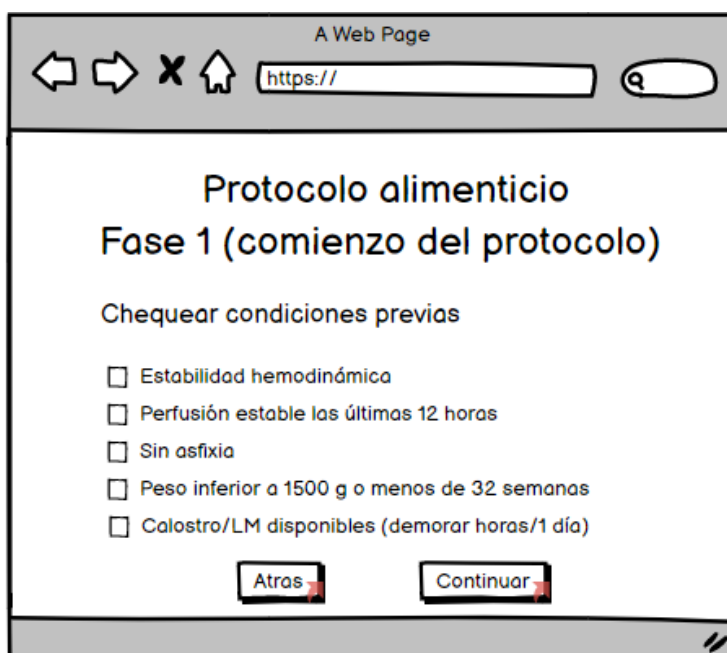


Figura 30: Página para iniciar la alimentación protocolizada en un paciente.



Figura 31: Página para registrar la evolución en la alimentación de un paciente .

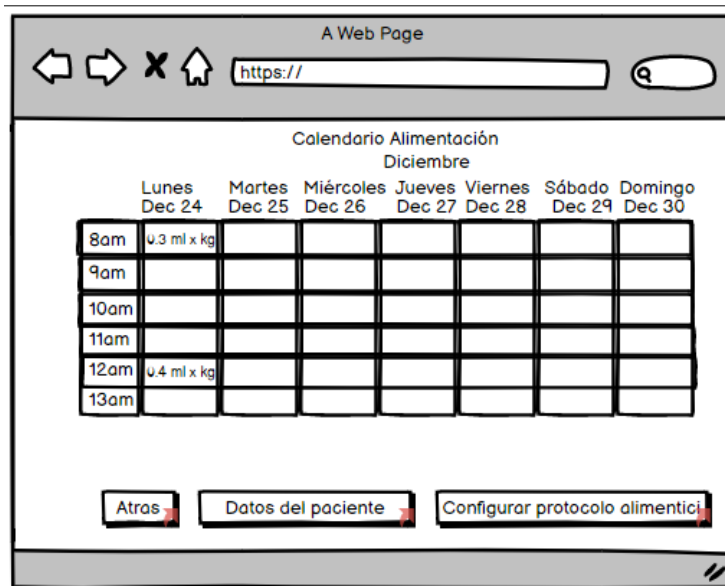


Figura 32: Página para consultar la información de las tomas del paciente dentro del protocolo alimentario.



Figura 33: Página para consultar las estadísticas computadas por la aplicación con la información disponible.



UNIVERSIDAD  
DE MÁLAGA

| [uma.es](http://uma.es)

E.T.S de Ingeniería Informática  
Bulevar Louis Pasteur, 35  
Campus de Teatinos  
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA