



---

This is the **published version** of the bachelor thesis:

Gomez Nicolas, Angel Antonio; Martínez Vidal, Ruben, tut. Website para centro médico. 2025. (Enginyeria Informàtica)

---

This version is available at <https://ddd.uab.cat/record/308780>

under the terms of the  license

# Website para centro médico

Ángel A. Gómez Nicolás

**Resumen**—Aplicativo basado en un website dedicado a ser la herramienta de gestión para las consultas realizadas en un centro médico.

El núcleo del proyecto se motivó por cubrir la necesidad de un profesional de la nutrición en cuanto a la gestión informatizada de su actividad profesional (gestión de pacientes, gestión de consultas y programación de éstas) y la de ofrecer al paciente un aplicativo propio donde gestionar sus visitas, programaciones e históricos.

Esta motivación de fondo se abstrae a un nivel superior, bajo la figura de un administrador, para extender la solución no sólo a un nutricionista sino a profesionales sanitarios de otras especialidades que den servicio en el mismo centro médico.

Aunque este proyecto sólo desarrollará la funcionalidad para un nutricionista, la incorporación de otros sanitarios al sistema es independiente y modular. Esto nos permite implementaciones específicas para cada uno de ellos además de agilidad y sencillez para su incorporación al sistema, siempre bajo la gestión del mismo administrador.

El desarrollo se basa en hitos de entrega que permite obtener un aplicativo temprano y funcionalidades terminadas tras completar cada uno de estos a modo de actualizaciones. Se lleva a cabo en lenguaje XML y Javascript utilizando REACT como aplicativo vehicular, BBDD no relacional y hosting mediante FIREBASE. La documentación y seguimiento bajo metodología SCRUMBAN utilizando Jira y Confluence.

**Palabras clave**— React, Firebase, consultas médicas, centro médico, nutrición, paciente, website, Visual Studio, XML, Javascript, Git, Scrumban, Jira, Confluence, Gantt, Whimsical, RGPD, LOPD.

**Summary**- Application based on a website dedicated to being the management tool for consultations carried out in a medical centre.

The core of the project was motivated by covering the need of a nutrition professional in terms of the computerised management of his professional activity (patient management, consultation management and scheduling of these) and that of offering the patient his own application where he can manage his visits, schedules and history.

This underlying motivation is abstracted to a higher level, under the figure of an administrator, to extend the solution not only to a nutritionist but also to health professionals from other specialities who provide services in the same medical centre.

Although this project will only develop the functionality for a nutritionist, the incorporation of other healthcare professionals into the system is independent and modular. This allows us specific implementations for each one of them as well as agility and simplicity for their incorporation into the system, always under the management of the same administrator.

The development is based on delivery milestones that allow us to obtain an early application and finished functionalities after the completion of each of these as updates. It is carried out in XML and Javascript language using REACT as a vehicle application, non-relational database and FIREBASE hosting. Documentation and monitoring under SCRUMBAN methodology using Jira and Confluence.

**Keywords**- React, Firebase, medical consultations, medical centre, nutrition, patient, website, Visual Studio, XML, Javascript, Git, Scrumban, Jira, Confluence, Gantt, Whimsical, RGPD, LOPD.



## 1 INTRODUCCIÓN

La necesidad que se pretende cubrir es ofrecer un aplicativo basado en un Website dedicado a ser la herramienta de gestión entre profesionales sanitarios y sus pacientes. En concreto a la realización de las consultas que ofrece una profesional de la nutrición (en adelante

nutricionista) a sus pacientes y el aplicativo que se les proporciona a ambos.

Actualmente la nutricionista dispone de un aplicativo que le cubre la gestión y confección de las dietas que pauta a sus pacientes (NUTRIUM[1], DIETOPRO[2], etc.) pero carece de Website propio donde desarrollar tanto la parte de gestión como la parte comercial y divulgativa propia del negocio. No dispone de gestión automatizada de agenda/calendario para reserva de citas ni aplicativo que ofrecerles a sus pacientes como parte del servicio.

- E-mail de contacto: 1360293@uab.cat
- Mención realizada: Tecnologías de la Información.
- Trabajo tutorizado por: Rubén Martínez Vidal (Departamento de Ingeniería de la Información y las Comunicaciones).

Actualmente encontramos varios software online disponibles que cubren parte de las necesidades que este desarrollo implementa como son, por ejemplo, DietoPro y Nutrium en cuanto a la gestión de las visitas nutricionales y control de pacientes.

Por otro lado, existen en el mercado soluciones ERP como SAP[3] u Odoo[4] para pymes que integran funciones de website, aplicaciones de ventas, comunicación y marketing.

Estas soluciones son genéricas, independientes y con un mayor coste tanto de integración como económico. Nuestro aplicativo le ofrece una solución que integra toda la funcionalidad que el cliente necesita con bajo coste, una personalización total del producto y sin dependencias de software externos.

## 2 OBJETIVOS

El fondo de nuestro aplicativo viene motivado por ofrecer una herramienta para la gestión informatizada de la actividad profesional de la nutricionista y a la vez una herramienta de consulta y comunicación para los pacientes.

Por parte de la nutricionista trata de darle un aplicativo donde gestione de una forma informatizada su trabajo: gestión de pacientes y visitas, gestión de agenda/calendario, importación de dietas confeccionadas, histórico de visitas, etc.

En detalle, la herramienta le permite dar de alta a pacientes de nueva consulta, editar y gestionar los existentes en cuanto a datos personales, toma de valores y hábitos actuales y registrar los objetivos inicialmente planteados. De esta misma forma puede informatizar todo lo relevante a realizar las consultas, ya sean primeras o de seguimiento, ya que se le proporciona un formulario donde ir registrando todos los valores nutricionales, mediciones y datos recogidos, así como otras cuestiones planteadas en consulta y que necesite recoger como estado de ánimo, sensaciones, motivación, dificultad, etc. Además se le provee de una agenda/horario donde gestionar las visitas a realizar, un planning diario y gestión de las consultas y días disponibles aún por agendar, es decir, una agenda informatizada y sincronizada con sus visitas/pacientes.

Por el lado de los pacientes se implementa el aplicativo para que puedan gestionar y consultar de todo lo referente a sus visitas, consultar las dietas/pautas que deben seguir, sus datos y valores nutricionales, histórico de consultas, etc. Además de poder mantener comunicación con la nutricionista, puede solicitar nuevas visitas y/o gestionar las visitas ya agendadas.

Con esta base funcional de fondo y abstrayendo el aplicativo un grado más para su uso general en centros médicos, se crea la figura y funcionalidad de Administrador que gestiona todo lo referente a los profesionales que dan servicio en el centro y la gestión administrativa, haciendo extensiva la aplicación a otros servicios sanitarios.

## 3 METODOLOGÍA

La metodología que se utiliza para el desarrollo del proyecto es SCRUMBAN[5][6], que se inspira en los pro-

cesos Scrum[7] y Kanban[8], proporcionando una combinación de lo mejor de ambos métodos. Esto nos brinda el control sobre las distintas etapas de desarrollo y nos permite tener una primera versión funcional del proyecto, añadiendo más funcionalidad en versiones posteriores y permitiendo modificaciones ante nuevas necesidades o variación de las prioridades.

Scrum basa el método en el desarrollo evolutivo de los requisitos en bloques temporales, cortos y fijos. Se priorizan las tareas con más valor para el cliente, y se define a un equipo de trabajo con el que sincronizar y adaptar de forma periódica los desarrollos y las adaptaciones.

Kanban centra su método basado en una filosofía de planificación adaptativa, entrega temprana y mejora continua. Las tareas individuales, dentro de las etapas del desarrollo, se representan en columnas ofreciendo el estado en que se encuentra (pendientes, en curso y finalizadas).

## 4 DESARROLLO

### 4.1 Retrospectiva

Se planificó una implementación incremental donde se han realizado entregas basadas en Sprints y donde cada una de ellas ha permitido tener una funcionalidad completada.

Partiamos de un proyecto que ya tenía implementado el hosting y la base de la autenticación en gran parte. Se actualizó la implementación adaptándola a las nuevas versiones del lenguaje para su optimización y utilizándola como punto de partida conceptual.

Tras el análisis y adaptación a nuestras necesidades en cuanto a la implementación de nuestros roles de usuario planteados, se definieron las siguientes etapas que han permitido obtener hitos funcionales asociados:

- **Entrada al aplicativo:** Login y registro de usuarios para los 3 roles definidos.
- **Gestión de administrador:** Implementación del rol de administrador para la gestión de permisos y usuarios del mismo tipo y profesionales.
- **Gestión de profesional:** Implementación del rol profesional para la gestión de pacientes, visitas y dietas.
- **Gestión de pacientes:** Implementación del rol usuario/paciente para la gestión de sus consultas.
- **Calendario asíncrono:** Implementación de la gestión de un calendario profesional/paciente.

El detalle de las tareas individuales de cada sprint se encuentran disponible en el proyecto Jira Software creado a tal fin en el siguiente enlace (<https://1360293.atlassian.net/jira/software/projects/SCRUM/bboards/1/timeline>) tras solicitar acceso al e-mail de contacto (1360293@uab.cat).

### 4.2 Roles y funcionalidades

Para la implementación de toda la funcionalidad necesaria se definen los 3 roles según el tipo de usuario que ingresa en el sistema y las distintas funcionalidades que se implementan y que cubre cada uno de ellos:

- **Rol Administrador:** Gestiona la creación de los distintos profesionales que pasan consulta y los permisos que se les otorga. También es el encargado de la gestión de pacientes cuando existan cambios de los profesionales que los visitan.  
Este rol nos permite desarrollar el aplicativo con la intención de proporcionar una herramienta "multi-profesional" donde se pueda implantar en un centro de salud y varios profesionales le den uso de forma simultánea si se desea. Además nos permite introducir nuevos roles de forma modular como personal administrativo u otros en función de las necesidades que surjan a futuro.
- **Rol Profesional:** Gestiona todo lo referente a las consultas que realiza, la creación y gestión de pacientes, los valores y registros tomados en cada consulta, las pautas a seguir, recomendaciones, etc.  
Por otro lado gestiona la importación al sistema de dietas previamente confeccionadas, el planning recomendado para cada paciente y el envío y comunicación directa con estos.  
También gestiona su horario y la agenda/calendario que pone a disposición de los pacientes para programar las visitas.
- **Rol Paciente:** Gestiona todo lo referente a las visitas que ha realizado, el acceso a las pautas/dietas que el sanitario le proporciona en cada visita, su histórico, sus datos personales, etc.  
También puede solicitar/gestionar sus visitas de forma asíncrona con la agenda/calendario que el profesional pone a su disposición.

## 4.3 Tecnologías

La implementación se lleva a cabo bajo el lenguaje JavaScript y HTML, concretamente utilizando la extensión JSX, librería integrada y utilizada por la biblioteca de React JS[9].

Esta librería nos permite crear tanto el front-end como el back-end de forma conjunta y basada en componentes simples como clases y funciones igual que en otros lenguajes tipo C++ o python pero con toda la potencia y funcionalidad de JavaScript. Además de tratarse de una librería basada en la modularidad en cuanto a la programación integra Virtual DOM que nos renderiza el aplicativo de forma dinámica y óptima en cuanto al uso de memoria.

Las herramientas utilizadas son Visual Studio Code[10] para la implementación, GitHub[11] para el control de versiones y sincronización con el hosting, Jira[12] para la planificación y gestión del proyecto y Confluence[13] para la documentación del desarrollo.

## 4.3.1 FrontEnd

### 4.3.1.1 React, ¿por qué?

React JS se concibe con el sentido de aunar la estructura de HTML, la apariencia de CSS y la funcionalidad de JavaScript en un sólo elemento, llamado "components", siendo cada uno de estos las piezas que forman el conjunto de todo el website.

Las principales ventajas que nos ofrece React en cuanto a la implementación son las siguientes:

- **Aprendizaje fácil:** Nos brinda una curva de aprendizaje corta si nuestro conocimiento en JavaScript es sólido.
- **Flexible:** Creación de funciones y módulos personalizados en distintos SO y dispositivos.
- **Rendimiento:** VDOM nos brinda la optimización del website y del uso de memoria.
- **Enlace de datos:** Flujo de datos unidireccional, de componentes principales a secundarios, ofreciendo estabilidad y robustez funcional.
- **Reutilizable:** Basado en componentes nos otorga la modularidad para reutilizar código.
- **Renderizado:** La compilación es inmediata por la interpretación de cualquier navegador, permitiendo visualizar cambios simplemente con guardar los cambios en el código.

### 4.3.1.2 Patrón redux

El patrón Redux nos permite desarrollar nuestro aplicativo desde donde obtener en todo momento y desde cualquier punto de nuestra app la información sobre el estado de esta. Es decir, desde cualquier ventana de navegación en la que nos encontremos podremos recuperar, mostrar y/o gestionar aquellos datos y valores que necesitamos mostrar/guardar según la acción que el usuario realice.

Redux es independiente de React y no es un patrón de desarrollo de software como tal, sino una biblioteca para la gestión del estado global de aplicaciones. Sin embargo, se comporta como un contenedor predecible, otorgándonos esos datos "globales" y accesibles desde cualquier punto de la aplicación. Además, nos permite resolver tanto las funciones síncronas como las asíncronas con la ayuda de herramientas adicionales, como los middlewares, y su implementación puede parecer transparente si nuestro conocimiento en esta tecnología no es demasiado profundo.

Esta funcionalidad es posible gracias al componente STORE, que sí es propio de React y que, a nivel conceptual, actúa como "un almacén global" donde podemos guardar cualquier valor, estado y/o contenido de un componente de forma dinámica. Junto a los llamados "thunks", que son funciones específicas y desarrolladas para satisfacer necesidades concretas, nos permite detonar los eventos que queramos dependiendo de las acciones que haya llevado a cabo el usuario o de las funciones que se estén solicitando.

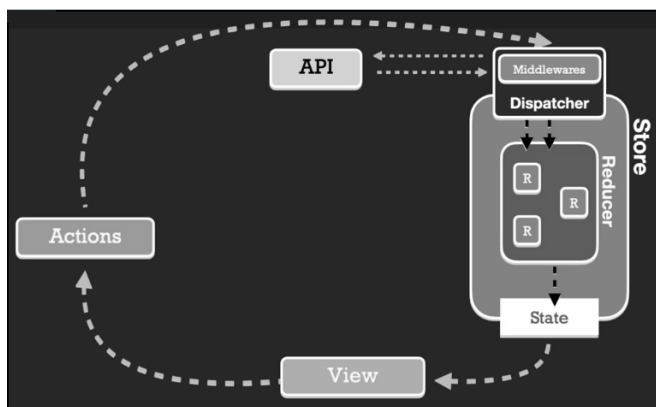


Fig. 1. Representación del flujo de datos entre el Store, los componentes (View) y las "Actions" del usuario sobre estos (Thunks)

#### 4.3.1.3 Desventajas

Las principales desventajas encontradas incluyen que, si bien React tiene una documentación oficial extensa, esta puede resultar insuficiente para casos avanzados, lo que obliga a buscar soluciones en fuentes externas.

Además, no establece patrones claros de desarrollo, dejando la organización del proyecto a criterio del desarrollador, lo que puede ser confuso para principiantes.

Por último, React requiere un conocimiento sólido de JavaScript, ya que muchos conceptos clave de este lenguaje son esenciales para su correcto uso y para evitar que el desarrollo sea lento y frustrante.

#### 4.3.2 BackEnd

Para cubrir los requerimientos de la base de datos en cuanto a la seguridad y la protección de datos personales en cumplimiento del RGPD y la LOPD se implementa de tipo no-relacional. Esto nos ofrece una flexibilidad, escalabilidad y un rendimiento alto a la vez de la atomización de los datos.

En cuanto al hosting del aplicativo nos apoyamos en un hosting cloud, mucho más sencillo, liviano y rápido de implementar con las soluciones existentes actuales.

Estas dos grandes necesidades, además de otras, lo implementamos bajo el aplicativo Firebase[14]. Actualmente nuestro website, y desde el sprint\_2, se encuentra alojado y disponible en el cloud desde cualquier dispositivo en la siguiente url: <https://naturalisanna.web.app>.

##### 4.3.2.1 Firebase, ¿qué es?

Es una plataforma en la nube para el desarrollo de software multiplataforma, creado en 2011 y adquirido en 2014 por Google. La función principal función principal es facilitar la creación de apps tanto webs como móviles en cualquier sistema (IOs, Android y web) utilizando todos los SDK de productos Google, incluido el hosting. Estas funcionalidades que nos ofrece se pueden desglosar en 4 bloques principales:

**compilación, supervisión, análisis y participación.**

##### 4.3.2.2 Firebase, pros y contras

Las ventajas principales cubiertas son la gestión, seguridad y compartición de datos en tiempo real bajo una BBDD de SQL No relacional sin necesidad de la creación de una BBDD de gestión propia e independiente.

Hosting del aplicativo y soporte con CVS, ofreciéndonos de una forma automatizada la conexión directa entre nuestro aplicativo Git con el propio Hosting.

Documentación, tutoriales y soporte gratuito de fácil acceso, ampliamente desarrollados y con gran profundidad de detalle. Gestión del desarrollo bajo un único panel/aplicativo facilitando y optimizando tanto el desarrollo de todas las etapas como la optimización del tiempo empleado para ello.

Funciones de testing, análisis de rendimiento y control de notificaciones proporcionando el típico entorno de pruebas pero incluyendo analíticas de rendimiento, latencias, datos sobre el uso, detección de fallos recurrentes y control de las notificaciones generadas.

##### 4.3.2.3 Funcionalidades de compilación.

El aplicativo nos ofrece soluciones específicas en cada parte de la implementación y se detallan a continuación las utilizadas para el desarrollo de nuestro proyecto:

- **Hosting:** Alojamiento en la nube de aplicativos.
- **Cloud Firestore:** Actualizaciones en tiempo real, consultas ágiles y escalado.
- **Realtime Database:** Almacenaje y sincronización de datos en tiempo real (*no utilizada*)
- **Autenticación:** Implementación automatizada del sistema de login y autenticación desde diferentes aplicativos
- **Storage:** Almacenaje y recuperación de contenido de los usuarios.

##### 4.3.2.4 Funcionalidades de lanzamiento y supervisión

Además, para cubrir la parte de pruebas y distribución de nuestro aplicativo, Firebase nos brinda funcionalidades que implementan los siguientes bloques:

- **Test Lab:** Pruebas de testing.
- **App Distribution:** Distribución de versiones previas a verificadores de confianza.
- **Remote Config:** Personalización y gestión de versiones.
- **Crashlytics:** Seguimiento en tiempo real de estabilidad y calidad del aplicativo.
- **Performance:** Estadísticas de rendimiento y latencias.

#### 4.4 Decisiones y cambios

Inicialmente, tras la extensa tarea de análisis de la base inicial del proyecto, adaptaciones a la nueva versión de React 6, configuración de aplicativos y estructuración de componentes y carpetas de navegación sentamos una base sólida de desarrollo. Pero esto, junto a la creación del prototipo de la aplicación, incorporado el uso de la he-

rramienta Whimsical[15] para el prototipado de los dashboard de navegación, y sentar las bases en cuanto al conocimiento de las distintas funcionalidades nos requirieron más horas de las previstas en los sprints iniciales para no ver afectada la planificación.

Tal como se ha gestionado durante el transcurso de toda la implementación del proyecto, no ha sido necesario realizar modificaciones en cuanto a la fecha de entrega ya que se ha implementado la funcionalidad dentro del tiempo previsto y siguiendo la planificación. Tal como se reflejó en informes anteriores, se han realizado modificaciones en la planificación pero por necesitar ordenar la prioridad de las tareas ya que de forma inicial algunas no estaban previstas en el mismo orden y otras, no contempladas de inicio, se han incorporado al hilo de la implementación de alguna funcionalidad planifica y definida da de inicio.

En resumen, la planificación en cuanto a la implementación del aplicativo no ha sufrido variaciones respecto al tiempo de entrega, sino al orden de implementación de las funcionalidades. La versión definitiva del planificador de tareas ejecutadas, así como el orden real de las mismas, queda reflejada en el documento adjunto en el anexo de este documento.

Las decisiones en cuanto al diseño del aplicativo se ha mantenido en constante colaboración con el cliente tras recoger sus necesidades y valorar el prototipo propuesto. La parte más de diseño visual se ha podido implementar desde inicio según su gusto y necesidad hasta el detalle. También se ha ido definiendo con cliente la política de permisos para cada rol o cuestiones más técnicas como la gestión de las Copias de Seguridad.

Estas decisiones eran coincidentes con la finalización de cada sprint, donde se le entregaba el producto y se planteaban las siguientes cuestiones a decidir para la pr entrega.

Tal como se detalla en los siguientes apartados, para cada una de ellas se presentaban las alternativas existentes, sus ventajas y virtudes brindando el asesoramiento técnico necesario para que el cliente además de implicarse en el desarrollo del aplicativo tuviese conocimiento de las características de su aplicación.

#### 4.5 Presentación de resultados

Las decisiones en cuanto a la selección de un aplicativo u otro para la implementación de cierta funcionalidad del BackEnd se han ido exponiendo a cliente y asesorado sobre la idoneidad de las mismas. Se decidió la implementación utilizando el Storage de Firebase [16] para la subida de documentación y almacenamiento de la misma (pautas y dossiers), en último término también se decidió que la gestión en cuanto a las copias de seguridad se implemente bajo este BackEnd.

##### 4.5.1 Alternativas funcionales

Para la toma de decisiones se expuso de forma detallada, pero a alto nivel, las alternativas existentes a Firebase:

**Supabase[17]:** Alternativa de código abierto a Firebase basada en PostgreSQL, con autenticación, almacenamiento y API en tiempo real. Ofrece un plan gratuito competitivo y es fácil de integrar con React. Sin embargo, su comunidad es más pequeña y el rendimiento en tiempo real puede ser menor que Firestore.

**Appwrite[18]:** Plataforma autoalojada de código abierto con bases de datos, autenticación, almacenamiento y funciones backend. Permite personalización total y control de costos al alojarse en servidores propios. Es más complejo de configurar y no cuenta con hosting gestionado.

**AWS Amplify[19]:** Solución de AWS con bases de datos, almacenamiento, API GraphQL y autenticación. Ofrece escalabilidad infinita y planes gratuitos atractivos, pero tiene una curva de aprendizaje alta. Es ideal para proyectos que requieren integración completa con AWS.

**Back4App[20]:** Plataforma basada en Parse que simplifica el desarrollo backend con bases de datos y notificaciones push. Es fácil de usar, ideal para proyectos pequeños, con un plan gratuito incluido. Sin embargo, tiene menos funcionalidades avanzadas y una comunidad más pequeña.

Servicio	Firebase	Supabase	Appwrite	AWS Amplify	Back4App
Base de datos	Firestore (NoSQL)	PostgreSQL (relacional)	MongoDB/SQLite	DynamoDB/Aurora	MongoDB
Autenticación	Sí	Sí	Sí	Cognito	Sí
Almacenamiento	Firebase Storage	Sí	Sí	S3	Sí
Tiempo real	Firestore/RTDB	Sí	Sí	Opcional con GraphQL	No
Costo inicial	Gratis hasta 50,000 lecturas	Gratis hasta 500 MB	Gratis si autoalojas	Gratis (limitado)	Gratis hasta 1 GB
Escalabilidad	Alta	Alta	Media	Alta	Media
Curva de aprendizaje	Baja	Media	Alta	Alta	Baja

Fig. 2. Comparativa de servicios y funcionalidad de los distintos servicios planteados para la implementación del BackEnd.

#### 4.5.2 Previsión de coste y uso

Otro aspecto que se ha tenido en cuenta para la selección de un BackEnd ha sido el uso y coste que podemos hacer de él mientras se sigue mantenido por debajo de la funcionalidad que ofrece.

Para este cálculo se ha tenido en cuenta toda la funcionalidad que utilizamos de Firebase (Hosting, Autenticación, Firestore, Storage y Copias de Seguridad) y el uso al alza de nuestro aplicativo. Se realizó una simulación de coste definiendo el uso del aplicativo por 10 profesionales de forma paralela, un total de 1000 pacientes registrados y 8 consultas diarias por profesional. Además se ha tuvo en cuenta un crecimiento exponencial en cuanto a las operaciones y uso del aplicativo junto a que la Ley General de Sanidad[21] establece que los documentos médicos deben conservarse durante al menos 5 años después de la última consulta o intervención. La siguiente figura muestra el resultado de esta simulación:



Año	Storage (GB)	Firestore (Operaciones)	Autenticación (Usuarios/semana)	Copias de seguridad	Descargas	Coste total anual (\$)
1	2 GB	40,000	350	104	52	13.584
2	4 GB	80,000	700	208	104	27.168
3	8 GB	160,000	1,400	416	208	54.336
4	16 GB	320,000	2,800	832	416	108.672
5	32 GB	640,000	5,600	1,664	832	217.344

Fig. 3. Comparativa de servicios y funcionalidad de los distintos servicios planteados para la implementación del BackEnd.

La conclusión final tras simular el uso/coste del aplicativo arroja que con el volumen actual de datos y uso de la aplicación, el costo de almacenamiento y operaciones en Firebase no debería ser un problema durante al menos 5-7 años, lo cual es más que suficiente para cumplir con la legislación de retención de datos. Sí existe la dependencia ineludible de que Firebase mantenga los costes en estos precios y no implemente una subida que modifique esta previsión.

#### 4.5.3 Discusión de resultados.

Los resultados obtenidos son muy satisfactorios ya que son los previstos inicialmente a pesar de haber invertido más horas de las valoradas de inicio.

El aplicativo se encuentra disponible en [www.naturalisanna.web.app](http://www.naturalisanna.web.app) y ya es completamente funcional para el cliente a nivel de los tres roles, pudiendo crear pacientes de nueva visita así como incluir en el sistema los ya existentes. Además registrar toda la actividad tanto en el anamnesis como en las posteriores visitas de seguimiento.

También se permite la creación de varios profesionales, en este caso nutricionistas, que pasan consulta de forma simultánea e independiente cada uno de sus pacientes con el acceso del Administrador.

El sistema gestiona todo lo relativo a los pacientes y sus visitas, manteniendo la privacidad y el acceso de cada uno de ellos sólo a los profesionales que les visitan, manteniendo la privacidad de los datos.

Por otro lado ya pueden ofrecerle a los pacientes un aplicativo propio donde consultar tanto la última visita y pauta/dossier disponible como todo el histórico de visitas, además de poder realizar su descarga. También se garantiza en el registro tanto la validez de la cuenta de correo del paciente como la integridad y seguridad del password.

En cuanto a la aplicación de los conocimientos adquiridos en las asignaturas de la mención esta aplicación nos ha permitido aplicar de forma directa múltiples conocimientos adquiridos durante todo el grado, destacando principalmente la de gestión de proyectos y otras áreas como el diseño e implementación de sistemas distribuidos, la gestión de bases de datos, la programación orientada a objetos, el diseño de interfaces de usuario, comunicaciones síncronas y asíncronas, integración entre distintos aplicativos y/o los conocimientos en el ámbito de la ética y la moral.

La elección de herramientas como React para el frontend y Firebase para el backend refleja la capacidad de evaluación, selección y gestión de tecnologías que son competencias esenciales en el perfil profesional del Ingeniero Informático. Otros aspectos como la implementación de funcionalidades clave, la autenticación de usuarios, la gestión de datos sensibles en tiempo real, la planificación de copias de seguridad o el cumplimiento de las normativas legales ha puesto en práctica capacidades adquiridas como el análisis, la planificación, la gestión, la documentación y la comunicación entre otras muchas.

Este proyecto ha sido, en esencia, un ejercicio práctico que bien refleja la capacidad de transformar los conocimientos teóricos en soluciones tecnológicas utilizando aplicativo disponible y actual, cumpliendo con estándares de calidad, seguridad y eficiencia esperados en la profesión. Además ha proporcionado un indicador personal de fortalezas y debilidades actuales en cuanto a nuestra madurez como Ingeniero Informático ayudándonos a enfocarnos en el crecimiento dentro de nuestra profesión.

Se muestra a continuación una captura del dashboard general, así como los iniciales tras la correcta autenticación del usuario correspondiente a cada uno de los roles implementados:



Fig. 4. Dashboard general del site para usuarios sin autenticar.



Fig. 5. Dashboard inicial al autenticar como Administrador.

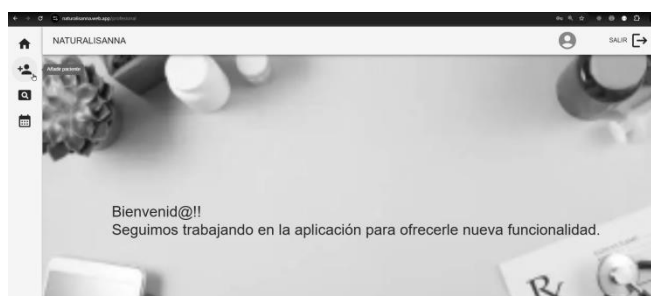


Fig. 6. Dashboard inicial al autenticar como Profesional.



Fig. 7. Dashboard inicial al autenticar como Paciente.

## 5 CONCLUSIONES

En cuanto a la gestión del proyecto hemos podido comprobar que la elección de la metodología ScrumBan es todo un acierto ya que nos ha permitido tener una base funcional al final de cada uno de los sprints. Esta base característica, además de validar los hitos, nos ha permitido mostrar el resultado a cliente y evaluar tanto la funcionalidad como los elementos visuales, permitiendo introducir modificaciones en el siguiente sprint con la seguridad que la implementación ya contaba con su visto bueno.

También, por supuesto, en cada sprint hemos podido introducir cambios o modificaciones no previstas inicialmente o reconfigurar la planificación para agrupar tareas menores, manteniendo la misma base, en otros momentos del desarrollo haciéndolo más eficiente.

Por otro lado, poder entregar de forma periódica la evolución del aplicativo a cliente nos ha permitido mantener el interés y la implicación del mismo además de, como se comenta anteriormente, disponer del criterio de validación de forma temprana y una toma de decisiones en conjunto y asesorada. Esto ha sido muy importante ya que este aplicativo, aunque no se reflejará en este TFG, seguirá su evolución ampliando la funcionalidad y sienta las bases de funcionamiento, siendo fundamental de cara al desarrollo futuro.

Además, utilizando como ejemplo la decisión del uso del storage para la subida de ficheros, nos ha permitido tomar estas decisiones con cliente en el momento justo de la implementación, pudiendo plantear otras opciones independientes si hubiese sido necesario.

En cuanto a la implementación, el uso de Firebase, desde luego es todo un acierto ya que nos permite tener el aplicativo disponible desde cualquier navegador y que cliente ya pueda estar haciendo uso de él. Además de proporcionar una autenticación, una BBDD fácil de manejar y robusta en cuanto a la seguridad y un storage donde gestionar las imagenes que se cargan y descargan entre pacientes y profesionales, su uso es muy intuitivo y modular. Nos permite realizar la implementación sobre la marcha e introducir las modificaciones según necesidad sin tener que realizar grandes modificaciones en la estructura.

Por último, sí se ha de destacar que las horas previstas no son ajustadas a la realidad ya que los plazos de entrega se han completado pero porque se han invertido en cada uno de los sprints más horas de las previstas inicialmente. En este aspecto quizá se debería haber previsto o bien la implementación con dos programadores o bien

recortar la implementación porque las horas de desarrollo invertidas superan con creces las iniciales para completar este TFG. (¡Aunque bien valieron la pena!)

## 6 IMPLEMENTACIÓN FUTURA. VERSIÓN\_2.0

El desarrollo de esta aplicación no es más que la versión 1.0 del aplicativo, sentando la base funcional para las posteriores versiones que seguirán siendo incrementales. No existe funcionalidad básica que haya quedado pendiente en cuanto a los objetivos iniciales, todos quedan cubiertos. Sólo podríamos destacar ciertas aspectos de robustez, como por ejemplo, mantener activa la sesión de usuario al forzar el refresh el navegador u otras más visuales como mostrar/permitir imagen de perfil que dejaremos fuera del desarrollo por no ser prioritarias.

Todos esos aspectos, por supuesto igual de importantes, pero más secundarios para esta primera versión, quedarán implementados en la versión 2.0 juntamente con otra funcionalidad que el cliente ya tiene prevista. Se detalla a continuación algunas de ellas:

1. **Funcionalidad móvil para pacientes:** Se migrará el desarrollo una aplicación móvil que permita a los pacientes acceder de forma más ágil a sus datos médicos, pautas y documentos, además de recibir notificaciones y recordatorios directamente en sus dispositivos.
2. **Gestión avanzada para administradores:** Se incorporará una funcionalidad que permita al administrador reasignar pacientes de un profesional a otro. Selección de proveedor de SMTP. Gestión automatizada de nuevas solicitudes.
3. **Optimización de la vista general comercial:** Se rediseñará la interfaz web para que sea más atractiva y comercial, mejorando la experiencia del usuario y fortaleciendo la presencia de la aplicación en el mercado.
4. **Alternativas al backend actual:** Se proveerá alternativa al backend de Firebase, buscando reducir la dependencia tecnológica y asegurar una mayor flexibilidad en la gestión de la infraestructura.
5. **Comunicación directa profesional-paciente:** Se añadirá una funcionalidad que permita a los profesionales redactar y enviar correos electrónicos directamente desde la aplicación, facilitando la comunicación personalizada con los pacientes.
6. **Nuevas configuraciones y ajustes:** Gestión/limitación de horarios de los profesionales adaptando el horario de servicio de consultas. Protección de la edición de datos sensibles, asegurando que un administrador no pueda modificar su propio perfil. En caso de que un administrador cambie su rol a profesional, el sistema aplicará automáticamente un logout() para reforzar la seguridad.
7. **Incorporación de nuevas funcionalidades sugeridas por el cliente:** Se implementará cualquier funcionalidad que el cliente valore y sugiera tras



el uso diario del aplicativo, adaptándose de forma proactiva a las necesidades detectadas en el día a día y fortaleciendo la experiencia del usuario final.

De forma análoga a la implementación de la Versión\_1, se ha creado un nuevo cronograma donde se reflejan estas implementaciones y servirá de herramienta para el desarrollo del mismo.

Este nuevo cronograma se encuentra disponible en <https://1360293.atlassian.net/jira/software/projects/NISV2/boards/3/backlog?selectedIssue=NISV2-8>

## 7 AGRADECIMIENTOS

Siempre y para siempre a mi mujer, por hacerme grande y orgulloso. A mis peques por ser mis tesoros, mis motores y mis ejemplos.

A mi familia por quererme por y para siempre, con todo. A mi hermano "from another mother" que siempre me confirma que es un tesoro.

A mis profesores por la paciencia y la generosidad de darme de conocimientos y capacidad.

## Bibliografía

- [1] Nutrium, *Elaboración de Dietas. El más innovador del mercado*. [En línea]. Disponible en: <https://www.nutrium.io>. [Accedido: 15-oct-2024].
- [2] DietoPro, Software nutricional para dietistas-nutricionistas. [En línea]. Disponible en: <https://www.dietopro.com>. [Accedido: 15-oct-2024].
- [3] SAP (2021, Oct. 28) SAP S/4HANA, el software de gestión para pymes en... [Online]. Available: <https://news.sap.com/spain/2021/10/sap-s-4hana-el-software-de-gestion-para-pymes-en-crecimiento/>
- [4] OpenWebinars(2020, Sep. 29) Qué es Odoo, características y versiones. [Online]. Available: <https://openwebinars.net/blog/que-es-odoo/>
- [5] P. Narang y P. Mittal, "Software Development Methodologies: Trending from Traditional to DOSE - An Empirical Study," 11-13 de febrero de 2022. [En línea]. Disponible en: <https://ieeexplore.ieee.org/xpl/conhome/9752769/proceeding>
- [6] Asana, Project management methodologies. [En línea]. Disponible en: <https://www.asana.com>. [Accedido: 15-oct-2024].
- [7] Metodología Scrum, Desarrollo del software. [En línea]. Disponible en: <https://www.scrum.org>. [Accedido: 15-oct-2024].
- [8] Metodología Kanban, Desarrollo del software. [En línea]. Disponible en: <https://www.kanban.org>. [Accedido: 15-oct-2024].
- [9] Hostinger (2023, Jun.29) Qué es React: Definición, características y funcionamiento [Online]. Available: <https://www.hostinger.es/tutoriales/que-es-react>
- [10] OpenWebinars (2022, Jul. 22) Qué es Visual Studio Code y qué ventajas ofrece. [Online]. Available: <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>
- [11] Xataka (2019, Oct. 30) Qué es Github y qué es lo que ofrece a los desarrolladores[Online]. Available: <https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores>
- [12] AppMaster (2022, Oct. 11) ¿Qué es JIRA? Descripción general y guía completa. [Online]. Available: <https://appmaster.io/es/blog/que-es-jira>
- [13] Indeed (2022, Sep. 30) Qué es Confluence. [Online]. Available: <https://mx.indeed.com/orientacion-profesional/desarrollo-profesional/que-es-confluence>
- [14] Seidor (2023, Feb. 20) ¿Qué es Firebase? ¿Qué ventajas ofrece en 2023 a ... [Online]. Available: <https://www.seidor.com/blog/firebase-que-es>
- [15] Whimsical, "The iterative workspace for product teams". [En línea]. Disponible en: <https://whimsical.com/>.
- [16] Storage, "The Cloud Storage for Firebase - Google". [En línea]. Disponible en <https://firebase.google.com/docs/storage>
- [17] Supabase, "The open-source Firebase alternative". [En línea]. Disponible en: <https://supabase.com/>.
- [18] Appwrite, "End-to-end backend server for web and mobile apps". [En línea]. Disponible en: <https://appwrite.io/>.
- [19] AWS Amplify, "Build scalable full-stack applications powered by AWS". [En línea]. Disponible en: <https://aws.amazon.com/amplify/>.
- [20] Back4App, "Parse hosting platform". [En línea]. Disponible en: <https://www.back4app.com/>.
- [21] Ley General de Sanidad, "Ley 14/1986, de 25 de abril, de derechos y garantías de la dignidad de la persona en la atención sanitaria". [En línea]. Disponible en: <https://www.boe.es/buscar/doc.php?id=BOE-A-1986-10465>.

ANEXO

		OCT '24	NOV '24	DEC '24	JAN
Sprints		Sprint 1 - Reinicio Proyecto	Sprint 2 - Func. Rol Profesio	Sprint 3 - Func. Rol Profesio	Sprint 4 - Func. Rol Profesio
Sprint 1: REINICIO DEL PROYECTO					
<input type="checkbox"/>	✓  SCRUM-24 SPRINT_1: REINICIO DEL PROYECTO	FINALIZADA			
<input type="checkbox"/>	SCRUM-25 Recuperación de la BBDD anterior para el backend en Firebase	FINALIZADA			
<input type="checkbox"/>	SCRUM-27 Creación de nueva cuenta en Confluence y Jira para documentación y planificación del proyecto.	FINALIZADA			
<input type="checkbox"/>	SCRUM-26 Recuperación del repositorio/código existente del Login y entrada principal al aplicativo.	FINALIZADA			
<input type="checkbox"/>	SCRUM-28 Sincronización VS+Git+Firebase. Verificación de updates al hosting.	FINALIZADA			
<input type="checkbox"/>	SCRUM-29 Modificación/preparación de la BBDD para el Login según el tipo de usuario	FINALIZADA			
<input type="checkbox"/>	SCRUM-30 Implementación FrontEnd de entrada y muestra según usuario logueado.	FINALIZADA			
<input type="checkbox"/>	SCRUM-68 Migración a nuevo repositorio. Migración a nuevo hosting.	FINALIZADA			
<input type="checkbox"/>	SCRUM-70 Reinicio del proyecto para adaptarlo a React 6.	FINALIZADA			
<input type="checkbox"/>	SCRUM-71 Creación de prototipo de layouts iniciales con Whimsical	FINALIZADA			
<input type="checkbox"/>	SCRUM-69 Creación de nuevo espacio para documentar el desarrollo.	FINALIZADA			
Sprint 2: FUNCIONALIDAD ROL ADMINISTRADOR					
<input type="checkbox"/>	✓  SCRUM-31 SPRINT_2: FUNCIONALIDAD ROL ADMINISTRADOR	FINALIZADA			
<input type="checkbox"/>	SCRUM-32 Dashboard gestión de profesionales. Creación de profesionales.	FINALIZADA			
<input type="checkbox"/>	SCRUM-33 Creación de tablas y campos en BBDD	FINALIZADA			
<input type="checkbox"/>	SCRUM-34 Testing de funcionalidad, guardado y recuperación de datos.	FINALIZADA			
<input type="checkbox"/>	SCRUM-62 Añadir botón y funcionalidad de búsqueda de profesionales.	FINALIZADA			
Sprint 3: FUNCIONALIDAD ROL PROFESIONAL					
<input type="checkbox"/>	✓  SCRUM-35 SPRINT_3: FUNCIONALIDAD ROL PROFESIONAL	FINALIZADA			
<input type="checkbox"/>	SCRUM-61 Completar funcionalidad y dashboard para datos de profesionales. Permitir modificaciones en los datos.	FINALIZADA			
<input type="checkbox"/>	SCRUM-65 Testing de funcionalidad de guardado y recuperación de datos de profesional.	FINALIZADA			
<input type="checkbox"/>	SCRUM-63 Implementación de visualización correcta del usuario autenticado y función de logout desde cada Dashboard.	FINALIZADA			
<input type="checkbox"/>	SCRUM-37 Alta y verificación de pacientes.	FINALIZADA			
<input type="checkbox"/>	SCRUM-76 Forzado de cambio de password para el primer ingreso del paciente	FINALIZADA			
<input type="checkbox"/>	SCRUM-75 Modificación y gestión de datos personales del paciente	FINALIZADA			

		OCT '24	NOV '24	DEC '24	JAN
Sprints		inicio Proyecto	Sprint 2 - Func...	Sprint 3 - Func. Rol Profesio	Sprint 4 - Fun... Sprint 5 - Fu... Sprint 6 - Tareas ...
<input type="checkbox"/>	▼ <b>SERUM-36 SPRINT_4: COMPLETAR FUNCIONALIDAD PROFESIONAL</b>	FINALIZADA			
<input type="checkbox"/>	✓ <b>SERUM-66</b> Modificar botón Home en los dashboard y mostrar texto que corresponda.	FINALIZADA			
<input type="checkbox"/>	✗ <b>SERUM-65</b> Añadir check box al cambiar contraseña de paciente para aceptar la LOPD.	FINALIZADA			
<input type="checkbox"/>	✓ <b>SERUM-38</b> Anamnesis, recogida de datos de primera visita.	FINALIZADA			
<input type="checkbox"/>	✓ <b>SERUM-64</b> Modificación de anamnesis	FINALIZADA			
<input type="checkbox"/>	✓ <b>SERUM-44</b> Testing de funcionalidad anamnesis	FINALIZADA			
<input type="checkbox"/>	✓ <b>SERUM-72</b> Testing de coherencia y control al insertar valores. Verificar la solución al error "full call stack"	FINALIZADA			
<input type="checkbox"/>	✗ <b>SERUM-67</b> Gestión del control de pacientes sólo por el profesional asignado	FINALIZADA			
<input type="checkbox"/>	✓ <b>SERUM-68</b> Formulario visita de seguimiento. Registro y guardado en BBDD	FINALIZADA			
<input type="checkbox"/>	✓ <b>SERUM-46</b> Visualización histórico. Listar visitas realizadas. Editar la visita seleccionada.	FINALIZADA			
<input type="checkbox"/>	✓ <b>SERUM-62</b> Testing funcionalidad visita seguimiento	FINALIZADA			
<input type="checkbox"/>	▼ <b>SERUM-44 SPRINT_5: FUNCIONALIDAD ROL PACIENTE. TAREAS PENDIENTES</b>	FINALIZADA			
<input type="checkbox"/>	✓ <b>SERUM-39</b> Subida de ficheros (planificación) y volcado en BBDD.	FINALIZADA			
<input type="checkbox"/>	✗ <b>SERUM-92</b> Corregir errores de funcionalidad en Editar Visita no detectados anteriormente	FINALIZADA			
<input type="checkbox"/>	✓ <b>SERUM-42</b> Visualización de última dieta pautada. Permitir la descarga	FINALIZADA			
<input type="checkbox"/>	✓ <b>SERUM-43</b> Visualización y consulta del histórico de dietas.	FINALIZADA			
<input type="checkbox"/>	✗ <b>SERUM-64</b> Solucionar navegación cuando un usuario de desloguea. Igual cuando creamos un nuevo usuario.	FINALIZADA			
<input type="checkbox"/>	✗ <b>SERUM-76</b> Añadir mensajes de advertencia y control en autenticación.	FINALIZADA			
<input type="checkbox"/>	✗ <b>SERUM-66</b> Control de valores en formularios de registro. Admin, Prof y Paciente	FINALIZADA			
<input type="checkbox"/>	✗ <b>SERUM-64</b> Modificación de contraseña temporal al crear paciente para que sea aleatoria	FINALIZADA			
<input type="checkbox"/>	✓ <b>SERUM-77</b> Recuperación / restauración del password	FINALIZADA			

<input type="checkbox"/>	✓	SPRINT_45	SPRINT_6: FUNCIONALIDAD CALENDARIO SÍNCRONO. NUEVA FUNCIONALIDAD.	FINALIZADA			
<input type="checkbox"/>		SCRUM-46	Gestión de calendario síncrono rol Profesional. Visualización de citas programadas y confirmación de reservas	FINALIZADA	2024/10/26	2024/11/08 14 d	
<input type="checkbox"/>		SCRUM-47	Gestión de calendario síncrono rol Paciente. Reserva de nuevas visitas. Recepción email de confirmación.	FINALIZADA			
<input type="checkbox"/>		SCRUM-99	Calendario síncrono Profesional. Reserva de visitas para un paciente concreto.	FINALIZADA			
<input type="checkbox"/>		SCRUM-48	Testing de funcionalidad.	FINALIZADA			
<input type="checkbox"/>		SCRUM-189	Implementar Home principal y dar estilos CSS a todo el aplicativo.	FINALIZADA			
<input type="checkbox"/>		SCRUM-191	Adaptar creación de usuarios al formato standard (igual que la creación de pacientes)	FINALIZADA			
<input type="checkbox"/>		SCRUM-98	Mantener la sesión activa al refrescar el navegador (F5)	FINALIZADA			
<input type="checkbox"/>		SCRUM-96	Indicar si existe anamnesis previa. Proteger la edición...	FINALIZADA			
<input type="checkbox"/>		SCRUM-192	Pérdida de nuevos datos en anamnesis si escribimos, no guardamos y clickamos en Info.	FINALIZADA			
<input type="checkbox"/>		SCRUM-193	Unificar formato de ventanas emergentes. Añadir spinner de carga en todos los casos.	FINALIZADA			
<input type="checkbox"/>		SCRUM-51	Añadir formulario para solicitar visita/info de usuarios no registrados	FINALIZADA			
<input type="checkbox"/>		SCRUM-54	Testing de funcionalidad general. Verificación de casos de uso para los 3 roles.	FINALIZADA			
<input type="checkbox"/>		SCRUM-97	Mostrar lista de usuario/pacientes por defecto y luego ir filtrando según la selección.	FINALIZADA			
<input type="checkbox"/>		SCRUM-66	Hacer el site responsive, de forma que se adapte a cualquier dispositivo.	FINALIZADA			