



Desarrollo de una aplicación Web e híbrida para la planificación de dietas.

Memoria de Proyecto Final de Máster
Máster en Aplicaciones Multimedia.
Itinerario profesional.

Autor: José Agustín González Pertusa

Consultor: Sergio Schvarstein Liuboschetz
Profesor: Laura Porta Simó.

12/06/2018

Créditos/Copyright



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

C) Copyright

© (el autor/a)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Desarrollo de una aplicación Web, con tecnología headless, y app híbrida, en una versión de prueba, para la planificación de dietas.</i>
Nombre del autor:	<i>José Agustín González Pertusa</i>
Nombre del consultor/la:	<i>Sergio Schvarstein Liuboschetz</i>
Nombre del PRA:	<i>Laura Porta Simó.</i>
Fecha de entrega (mm/aaaa):	<i>Junio/2018</i>
Titulación:	<i>Máster en Aplicaciones Multimedia</i>
Área del Trabajo Final:	<i>Trabajo de Fin de Máster</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Aplicación híbrida, API REST, dieta.</i>
Resumen del Trabajo:	
<p>El presente proyecto describe el desarrollo de una aplicación Web y aplicación híbrida en su versión Alfa, para la creación de planes de dieta a partir de los datos de alimentos servidos desde una Rest API. La aplicación debe servir como base en la realización de aplicaciones con mayor número de funcionalidades, adaptadas a diferentes perfiles de usuario. En el desarrollo del Back-end se utilizó Drupal 8 con su módulo JSONAPI. En el Front-end se eligieron los frameworks Angular4 e Ionic3. Como resultado, la aplicación presenta; Buscador de alimentos obtenidos desde un servicio Web, creador del plan de dieta a partir de estos alimentos, función de guardado y cargado de planes de dieta. Conclusión: La flexibilidad que ofrece las estructuras desacopladas y el ahorro en tiempo y esfuerzo que supone el desarrollo multiplataforma hacen de este planteamiento de trabajo sea una apuesta segura de futuro.</p>	
Abstract:	
<p>In this project, it has been developed an alpha versión of a Web and hybrid app to make diet plans from a Rest API. The application should serve as base for farther development in orther</p>	

to adapt it to different user profiles. Back-end was developed with Drupal 8 and Front-end with Angular4 and Ionic3 frameworks. As a result, the application presents; Search engine, diet plan maker, diet plans store and load. Conclusion: The flexibility of decoupled structures and saving time multiplatform development make this work approach a safe bet for the future.

Agradecimientos

A mi mujer Ángeles, por su paciencia y apoyo.

Abstract

In this project, it has been developed an alpha versión of a Web and hybrid app to make diet plans from a Rest API. The application should serve as base for farther development in orther to adapt it to different user profiles. Back-end was developed with Drupal 8 and Front-end with Angular4 and Ionic3 frameworks. As a result, the application presents; Search engine, diet plan maker, diet plans store and load functionalities. Conclusion: The flexibility of decoupled structures and saving time multiplatform development make this work approach a safe bet for the future.

Resumen

El presente proyecto describe el desarrollo de una aplicación Web y aplicación híbrida en su versión Alfa, para la creación de planes de dieta a partir de los datos de alimentos servidos desde una Rest API. La aplicación debe servir como base en la realización de aplicaciones con mayor número de funcionalidades, adaptadas a diferentes perfiles de usuario. En el desarrollo del Back-end se utilizó Drupal 8 con su módulo JSONAPI. En el Front-end se eligieron los frameworks Angular4 e Ionic3. Como resultado, la aplicación presenta; Buscador de alimentos obtenidos desde un servicio Web, creador del plan de dieta a partir de estos alimentos, función de guardado y cargado de planes de dieta. Conclusión: La flexibilidad que ofrece las estructuras desacopladas y el ahorro en tiempo y esfuerzo que supone el desarrollo multiplataforma hacen de este planteamiento de trabajo sea una apuesta segura de futuro.

Palabras clave

Aplicación híbrida, API REST, dieta.

Notaciones y Convenciones

Código y Urls se presentan en itálica.

Índice

Capítulo 1: Introducción.....	10
1. Prefacio.....	10
1.1 El nicho.....	10
1.2 Expectativas personales.....	11
2. Definición.....	11
1.1 Resumen de la propuesta.....	11
1.2 Justificación.....	13
3. Objetivos generales.....	14
3.1 Objetivos principales.....	14
4. Metodología y proceso de trabajo.....	16
5. Planificación.....	17
5.1 Descripción detallada de cada hito.....	17
5.2 Temporalización.....	27
5.3 Diagramas de Grantt:.....	30
6. Presupuesto.....	32
7. Información en los siguientes capítulos.....	33
Capítulo 2: Análisis.....	33
1. Estado del arte.....	33
2. Análisis del mercado.....	37
3. Análisis de un caso de éxito: myFitnessPal.....	37
Capítulo 3: Diseño.....	39
1. Arquitectura general de la aplicación.....	39
2. Arquitectura de la información y diagramas de navegación.....	41
3. Diseño gráfico e interfaces.....	43
4. Lenguajes de programación y APIs utilizadas.....	46
Capítulo 4: Demostración.....	46
1. Url de demostración.....	46
2. Instrucciones de instalación.....	47
3. Prototipos.....	47
Capítulo 5: Conclusiones y líneas de futuro.....	49

1. Conclusiones.....	49
2. Líneas de futuro.....	50
Bibliografía.....	51
Anexos.....	55
Anexo A: Entregables del proyecto.....	55
Anexo A: Perfil del autor.....	55

Figuras y tablas

Lista de imágenes, tablas, gráficos, diagramas, etc., numeradas, con títulos y las páginas en las que aparecen.

Índice de figuras

Figura 1: Productos procedentes de la agricultura ecológicos.....	10
Figura 2: Tendencia de consumo de productos ecológicos.....	10
Figura 3: Función getInfo.....	19
Figura 4: Diagrama de Gantt para la primera etapa.....	30
Figura 5: Diagrama de Gantt para la segunda etapa.....	31
Figura 6: Diagrama de Gantt para la tercera etapa.....	32
Figura 7: Pantallas de ¿Qué puedo comer?, mySugr LogBook y About.....	35
Figura 8: Pantallas de la app Fooducate con calificación del producto.....	36
Figura 9: Pantallas de las apps Diana Come Sano, Eat Smart y Awesome ats.....	36
Figura 10: Pantallas de la aplicación Fatsecret.....	37
Figura 11: Pantallas de la aplicación myFitnessPal.....	38
Figura 12: Arquitectura general de Equiplan.....	40
Figura 13: Relación de los componentes del Back_end y el Front_end.....	41
Figura 14: Diagramas de navegación. En rosa se encuentran las páginas que todavía no han sido implementadas.....	42
Figura 15: Página planadd.html mostrando la paleta de colores.....	43
Figura 16: Ejemplo de elementos gráficos.....	44
Figura 17: Diseño adaptable.....	44
Figura 18: Elemento emergente.....	45
Figura 19: Ejemplo de elemento deslizante ocultando botones.....	45
Figura 20: Mock-ups de plan.html, planadd.html y nutrientes.html.....	48
Figura 21: Mock-ups de plan.html, guardar.html y cargar.html. La página guardar.html fue desestimada, incluyendo toda la funcionalidad en cargar.html.....	48
Figura 22: Mock-up de la página analisis.html.....	49

Índice de tablas

Tabla 1: Tabla de hitos de la primera etapa.....	27
Tabla 2: Tabla de hitos de la segunda etapa.....	28
Tabla 3: Tabla de hitos de la tercera etapa.....	29

Capítulo 1: Introducción

1. Prefacio

1.1 El nicho



Figura 1: Productos procedentes de la agricultura ecológicos.

Dado el gran interés que suscitan los temas relacionados con la salud y la nutrición, y el uso, cada vez más frecuente por parte de los usuarios de aplicaciones que facilitan el control del estado de salud, de la cantidad de ejercicio, de la dieta ... una aplicación cuya funcionalidad se enmarque dentro de este ámbito dispondrá de un número potencial de usuarios elevado aunque, al mismo tiempo, de un número no menos significativo de competidores. Este último aspecto hace necesaria cierta especialización, que asegure un nicho en el que la competencia sea menor.

La preocupación por una alimentación sana y el consumo de productos ecológicos es una práctica habitual en países de Europa del norte. Aunque menos generalizada en España, está creciendo sin cesar, incluso en los periodos de crisis. Es evidente la existencia de un grupo creciente de usuarios españoles que demandan salud en su dieta diaria. Por otro lado, es de esperar que España siga la tendencia de otros países europeos más desarrollados.

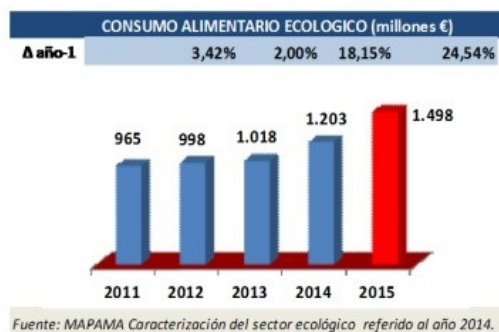


Figura 2: Tendencia de consumo de productos ecológicos.

Esta aplicación puede ser la base para el desarrollo de aplicaciones más complejas; Dirigidas a personas preocupadas por seguir una vida sana, que faciliten el aprendizaje en materia de salud y dieta sana y equilibrada, que permitan el acceso a dietas saludables, que informen con detalle al usuario sobre los alimentos que consume; Dirigidas a un usuario con peculiaridades nutricionales y problemas médicos específicos, que necesite requerimientos nutricionales especiales (cáncer, intolerancias, alergias, diabetes, ulcera, etc.) ; O dirigidas a profesionales de la nutrición.

1.2 Expectativa personal

Por otro lado este proyecto me permite combinar mi experiencia en el ámbito de la nutrición con mis recién adquiridas habilidades en desarrollo multimedia. También me abre la posibilidad de completar mi adiestramiento en materia de aplicaciones multimedia con el desarrollo híbrido, que considero, dado el ahorro de esfuerzo en desarrollo y mantenimiento (y la continua mejora en prestaciones de los dispositivos móviles), una práctica en crecimiento.

Así, mis conocimientos en materia de nutrición y salud facilitan que pueda moverme con cierta agilidad dentro de dicho ámbito, y suponen una ventaja en el desarrollo de este proyecto.

2. Definición

2.1 Resumen de la propuesta:

Se desarrolla una aplicación Web/ app híbrida para realizar un plan de dieta semanal sencillo. Esta aplicación pretende ser una ayuda para personas concienciadas con su salud que deseen planificar su dieta semanal, teniendo en cuenta las necesidades nutricionales (proteínas, carbohidratos, grasas esenciales, vitaminas, sales minerales), y evitar los excesos de sustancias poco recomendables (Grasas saturadas, glúcidos simples). Al mismo tiempo debe dar la oportunidad al usuario de aprender sobre nutrición.

La aplicación presentará las siguientes funcionalidades básicas:

Búsqueda de información nutricional:

- Permite al usuario buscar un grupo de alimentos mediante un cuadro de búsqueda en una base de datos.
- El usuario puede seleccionar uno de los alimentos del listado resultante de la búsqueda para ver sus cualidades nutricionales.

Realización de un plan de dieta semanal:

- El usuario puede elegir día de la semana y categoría temporal (Desayuno, almuerzo, comida, merienda y cena).
- Seleccionado el día y la categoría el usuario puede realizar una búsqueda en la base de datos de alimentos para añadir un alimento a la categoría elegida.
- El usuario puede eliminar un alimento seleccionado en la categoría elegida.
- El usuario puede modificar la cantidad seleccionada de un alimento en concreto.

Guardar y cargar las dietas:

- El usuario puede guardar en formato Json el plan de dieta semanal realizado.
- El usuario puede cargar un plan de dieta ya establecido en formato Json.

Análisis del plan de dieta actual:

- Una vez elegidos los alimentos pertinentes para la semana, la aplicación mostrará información nutricional respecto a la dieta propuesta. Calculando la cantidad total de Kcal, porcentaje de Proteínas, carbohidratos y grasas, proporción de grasas saturadas, proporción de glúcidos simples, cantidad de vitamina C, E, Omega 3 y calcio ...

Otras funcionalidades quedan fuera del alcance de este proyecto:

Búsqueda de información nutricional:

- La búsqueda debe permitir añadir filtros según el tipo de nutriente que proporciona. (ejemplo: Alimentos ricos en calcio, alimentos proteicos, ...)

Exportar información nutricional:

- El usuario puede exportar el plan de dieta de la semana en formato pdf.
- El usuario puede intercambiar planes de dieta con otros usuarios.

(Opcional) Análisis de las necesidades del usuario:

- El usuario puede introducir su peso, altura, edad, régimen de actividad.

- La aplicación puede calcular las necesidades del usuario en base a los datos incluidos y comparar estas con el análisis de la dieta realizada.

(Opcional) Elaboración de la lista de la compra.

(Opcional) Recomendaciones nutricionales que saltan al elegir ciertos alimentos.

(Opcional) Gamificación (puntuar las dietas que mejor se ajusten a un perfil determinado).

(Opcional) Necesidades especiales (avisos para celíacos, intolerantes a la lactosa, diabéticos, ...)

2.2 Justificación:

Situación Actual: En las búsquedas en Internet mediante el buscador de Google con términos tales como “plan de dieta Online, dieta Online, nutrición Online, nutrición y dietética Online, dietética Online, diet Online” es posible encontrar un número muy reducido de aplicaciones Web gratuitas y mínimamente fiables relacionadas con la realización de un plan de dieta. Aún más difícil es localizar alguna en castellano y que permita, sin rodeos, realizar un plan de dieta semanal. En este último grupo abundan las aplicaciones que promueven algún tipo de dieta milagro poco recomendable para la salud.

En cambio, si hablamos de apps móviles, el número de aplicaciones para celulares relacionadas con la dieta y hábitos nutricionales es inmenso. También lo es el interés que suscita este tema en la población de usuarios en países desarrollados.

Es importante, sin embargo, tener en cuenta la poca fiabilidad de estas aplicaciones. Según los estudios realizados en una exhaustiva investigación relacionada con apps de salud “patient apps for improving healthcare” el 45,8% de aplicaciones no estaban realmente relacionadas con la salud (Atkien M.). En otro estudio, al realizar una búsqueda con los términos dietas, menús, recetas, pérdida de peso se encontró que el 40% de aplicaciones ofrecían dietas milagro (Ismael S. M. M. y col.).

En una búsqueda actual en Google Play con el término “app nutrition” encontramos numerosos contadores de calorías y aplicaciones con información en formato curso o base de datos sobre nutrición, así como calculadoras de parámetros relacionados con el metabolismo, peso, etc Aparentemente, se trata de aplicaciones de cierta calidad. Con el término “app diet” aparece un gran número de aplicaciones relacionadas con adelgazamiento y dietas milagro de aparente menor calidad. Sin embargo, “diet apps” presentan mayor porcentaje de compras que “nutrition apps” (Schumer H y col.).

Conclusión: Aunque muchas son las aplicaciones móviles relacionadas con la realización de planes de dieta, el número de estas que ofrecen la seguridad de seguir una dieta equilibrada y saludable se reduce notablemente. Por otro lado, las aplicaciones Web que no requieren la descarga previa son notablemente escasas y sufren del mismo problema de baja fiabilidad. Este mar de aplicaciones relacionadas con el adelgazamiento rápido y las dietas poco saludables no puede cubrir la creciente demanda de un público concienciado con la importancia de la dieta equilibrada.

Por otro lado la sociedad demanda una educación adecuada en relación a la nutrición. Problemas graves como la obesidad, cuya prevalencia crece de forma continua, necesitan que las personas estén concienciadas e informadas para ponerla coto (*Félix Lobo*).

2.3 Propuesta:

La aplicación Web planteada se incluirá como un servicio gratuito proporcionado por un sitio dedicado a la nutrición sana y ecológica. Proporcionara además la posibilidad de descargarla como app para dispositivos móviles. Con un planteamiento de dieta equilibrada no sujeta a ninguna dieta milagro y con el fin de educar en salud al tiempo que proveer de una herramienta fácil y útil para un usuario concienciado.

3. Objetivos generales

Podemos considerar cómo objetivo general el desarrollo de una aplicación Web e híbrida en fase preliminar, pero con las funcionalidades básicas necesarias completamente desarrolladas, que permita la realización de un plan de dieta semanal, con acceso a una base de datos que contenga una pequeña muestra representativa de alimentos. Para alcanzar este objetivo se proponen objetivos más específicos.

3.1 Objetivos principales

Objetivos de la aplicación:

- Servir como punto de partida para el desarrollo de aplicaciones más complejas y con funcionalidades más específicas (Aplicación orientada a productos ecológicos que forme parte de la Web de la asociación Salud Ecológica, aplicación asociada a libro de recetas, aplicación a medida, que facilite la labor de los nutricionistas de la asociación Salud Ecológica, aplicación asociada a los productos de una tienda Online, aplicación específica para intolerancias poco conocidas ...).

- Contar con una base de datos de alimentos ampliable, fácil de manejar desde el Back-end, que permita su actualización y la modificación de sus contenidos de forma ágil y eficiente.
- La aplicación debe permitir la implementación de nuevas páginas que agreguen nuevas funcionalidades.
- Permitir con facilidad la implementación de nuevos campos para las tablas de alimentos en la base de datos, y de este modo, facilitar que la aplicación crezca posteriormente en funcionalidades.
- La base de datos debe incluir un sistema que permita la carga de datos masiva que reduzca el coste temporal del ingreso de estos datos.
- Incluir un número suficiente de datos que permita probar todas las funcionalidades (sin pretender alcanzar la cantidad de alimentos que necesita la aplicación en producción).
- La aplicación debe poder desplegarse como una app Standalone para dispositivos móviles y como una app Web. Las nuevas funcionalidades se implementarán una vez y serán utilizables en las diversas plataformas.
- La aplicación debe mostrar una apariencia similar a la que muestran las aplicaciones nativas de cada dispositivo, permitiendo su futura distribución en las tiendas de aplicaciones Online.

Objetivos para el usuario:

- Ser intuitiva y sencilla de utilizar para aumentar su accesibilidad.
- Mostrar un interfaz de usuario atractivo y amigable, adaptado a las pantallas de diversos dispositivos.

Objetivos personales del autor del TF:

- Integrar conocimientos adquiridos durante el máster relacionados con modelos de arquitectura desacoplada y completarlos con el desarrollo de aplicaciones híbridas.
- Aprender el manejo de frameworks que facilitan el desarrollo de aplicaciones Web (Angular, Ionic).

- Disponer de una experiencia de trabajo cercana al trabajo profesional en desarrollo de aplicaciones.

4. Metodología y proceso de trabajo

Se optó por el desarrollo de un producto nuevo utilizando CMS y frameworks con el fin de agilizar notablemente la implementación.

Con respecto al Back-end se ha elegido utilizar el gestor de contenidos Drupal 8. Este gestor de contenidos ha demostrado su gran potencial en numerosos proyectos realizados por organizaciones de enorme transcendencia como los portales de unicef.es y del Museo Thyssen-Bornemisza .

Entre las ventajas de utilizar Drupal 8 se incluye la gran versatilidad que ofrece para distribuir los contenidos en multitud de formatos, esto es evidente en la creación de apps móviles que se alimenten de los mismos contenidos creados para un canal web. Esta situación es incluso, más evidente si se usa una tecnología headless o desacoplada. Con una arquitectura desacoplada se reducen notablemente los costes de desarrollo y, especialmente, de mantenimiento, ya que los cambios necesarios para que la aplicación mantenga una presencia moderna y adaptada a las necesidades cambiantes del mercado se realizarían solo en el front-end.

En el Fron-end la elección del framework Ionic 3 suponen notables ventajas, por un lado, el uso de Angular JS facilita la inclusión de interactividad en las aplicaciones, presentando, por otro lado, una gran gama de herramientas y servicios que aceleran el desarrollo notablemente.

Sin embargo, la adopción de estas tecnologías ofrecen también una serie de desventajas:

La construcción de un portal con Drupal implica el uso de módulos que no están incorporados en el core del producto. Algunos de estos módulos contribuidos se encuentran en versiones beta y otros no han comenzado a desarrollarse, quedando relegados a Drupal 7. Esto implica que a lo largo del proyecto ha sido necesario buscar alternativas o testar módulos no maduros.

Por otro lado, tanto Ionic 3 como Angular.js son frameworks desconocidos por el equipo de trabajo. Su adopción en este proyecto ha constituido un riesgo considerable y ha dificultado muy notablemente la planificación del esfuerzo requerido y la temporalización.

Todo ello ha marcado notablemente la metodología seguida. Que ha requerido, en todo momento, un periodo previo de investigación, testeo y replanteamiento del problema a resolver.

El desconocimiento de que y que no se podía hacer con Ionic ha dificultado el propio planteamiento de diseño. No es posible diseñar con precisión si se desconocen los elementos disponibles. Esto ha afectado, por ejemplo, a la creación de los esquemas previos (Mock-ups ...) que tuvieron que esperar a que el equipo cogiera experiencia con a primera entrega del producto.

El tipo de modelo seguido en el desarrollo ha sido un modelo simple en cascada, que si bien se trata de un modelo rígido, poco recomendable en proyectos de cierta envergadura, en este caso, con un equipo constituido por una sola persona, se ajusta bien al proyecto.

El tiempo disponible se ha dividido en tres etapas, ajustandose así a las entregas propuestas por el consultor. Al final de cada etapa, el modelo estaba diseñado para mostrar cierta funcionalidad, lo que facilitó el llevar a cabo una revisión final.

En cada hito ha sido necesario el llevar a cabo una investigación previa, de mayor o menor envergadura, antes de acometer la fase de desarrollo. Las fases de cada iteración han seguido la siguiente estructura:

- Análisis de requisitos.
- Investigación y aprendizaje.
- Diseño del item.
- Codificación.
- Pruebas.
- Rediseño si fuera necesario o finalización y paso al siguiente Item.

El ciclo de vida de la aplicación continúa más allá del proyecto que nos ocupa. Finalizando, con este proyecto una primera versión en fase Alfa.

5. Planificación

5.1 Descripción detallada de cada hito:

La planificación se ha realizado en Tres hitos o etapas principales, que coinciden con las entregas realizadas al consultor. Estas tres etapas son:

Etapa1: Buscador de alimentos (Conexión Back-end/Front-end).

Etapa 2: Plan de dieta (Página principal del plan de dieta e información nutricional).

Etapa 3: Guardar y cargar en formato JSON.

Etapa 1: Buscador de alimentos (Conexión Back-end/Front-end):

- **Aprendizaje y pruebas con Ionic/Angular/TypeScript:**
 - Después de un primer análisis sobre el modo de trabajar con Ionic, la propuesta inicial de utilización de Ionic / JQuery / Drupal 8 se modifica, sustituyendo JQuery por AngularJS
 - AngularJS está dirigido a objetos y permite programación Modelo, Vista, Controlador.
 - La mayor parte de los Tutoriales encontrados de Ionic lo combinan con AngularJS.
 - Ionic funciona con TypeScript, Que añade una capa sobre JavaScript.
 - Por todo ello el aprendizaje y pruebas se extiende dos días más de lo previsto.
- **Instalación y configuración de DRUPAL localmente:**
 - Después de las pruebas realizadas con Ionic, se eligió una instalación de Drupal en modo local (Ubuntu y Xamp), en lugar de instalar en un hosting, siendo mucho más rápido implementar los cambios de este modo.
 - Se pospone el contrato de un hosting y la subida de la aplicación a este para la fase final de la entrega 1 (momento en el que la aplicación ya presentará algunas funcionalidades que probar).
 - Se realiza una instalación mínima de Drupal 8.5.0, instalando, posteriormente, El tema Seven como tema del backend, varios plugins del core necesarios para manejar los campos de las entidades, la barra de administración del backend, las taxonomías, todos los pertenecientes al Web Services (*Joe Fender*) y se añaden permisos al role de administrador. Sin incidencias.
- **Creación de entidades (Tipo de Contenido Alimentos):**
 - Se crea el tipo de contenido alimentos con campos de prueba similares a los que presenta la base de datos BEDCA (*Red BEDCA del Ministerio de Ciencia e Innovación*) . Sin incidencias.
- **Instalación del plugin JsonApi para Drupal:**
 - Después de investigar la mejor forma para acceder a los resultados de una búsqueda (con un filtro según un termino de búsqueda) se decide utilizar el plugin JsonApi que implementa la Rest-API (*JsonApi organization*).
 -

- **Aprendizaje y pruebas con el plugin JsonApi:**
 - Dado el desconocimiento del funcionamiento del plugin es necesario realizar unas pruebas con diferentes operadores y filtros (*Matt Glaman*).
 - El aprendizaje y las pruebas conllevan un retraso en el desarrollo de 3 días.

- **Introducción de algunos alimentos utilizando las tablas nutricionales del BEDCA como guía.**
 - Después de sopesar diversas opciones que incluyen tablas descargables en Excel y tablas en PDF fáciles de incluir en la base de datos de Drupal, mediante importación de CSV, se mantiene la decisión inicial de utilizar la base de datos de la BEDCA. Más difícil de pasar a formato CSV pero más completa y fiable.
 - Se instala el plugin Content Import en Drupal para la importación de CSVs
 - Se importan 16 alimentos de prueba.

- **Instalación, configuración y pruebas con el plugin JsonApi Extras en Drupal**
 - Se decide incluir el plugin JsonApi Extras para eliminar campos innecesarios en el Json y reducir la carga de información de la app.

- **Instalación de Ionic y creación/configuración de la app:**
 - Se instaló una aplicación de tipo Tab (*TutorialesEnPDF*)(*Documentación oficial de ionic*) . Sin incidencias.

- **Desarrollo en Ionic: Conectar con la REST API.**
 - Se creó desde el terminal un proveedor Rest y se implementó en Rest.ts la clase RestProvider con el método getInfo que lee los datos del json (*Didin J*). Sin incidentes.

```
18
19     getInfo(finalUrl) {
20         return new Promise(resolve => {
21             this.http.get(this.apiUrl + finalUrl).subscribe(data => {
22                 resolve(data);
23             }, err => {
24                 console.log(err);
25             });
26         });
27     }
```

Figura 3: Función *getInfo*.

- **Desarrollo en Ionic: Implementación básica de la página del buscador de alimentos.**
 - Programación en home.html y home.ts con la implementación de las funciones *getAlimentos* que llama a *restProvider.getInfo* y carga la información del Json en el array de objetos alimentos. Sin incidencias.
 - Programación de la función *filtrarPorTermino*, que llama a la función *getAlimentos* pasándole una url con un filtro con el termino introducido en el control del buscador (*Josh Morony*). Sin incidencias.

- **Desarrollo en Ionic: Mejora de la función del buscador de alimentos.**
 - Se implementa un Observable para introducir una espera que permita que se termine de escribir el término de búsqueda antes de enviar la petición a Drupal (*Josh Morony*). Sin incidentes.

- **Desarrollo en Ionic: Implementación de la página detalle con los nutrientes para cada alimento.**
 - Creación de una nueva página “nutrientes” en el terminal. Implementación de *nutrientes.html* y *nutrientes.ts* con paso de parámetros desde la página home (*Javier García Escobedo*). Sin incidentes.

- **Desarrollo en Ionic: Diseño y mejora de la página de detalle con los nutrientes para cada alimento.**
 - Uso de `<ion-card>` e `<ion-grid>` para mejorar el diseño responsive de la página (*Enrique Oriol*).

- **Diseño de Mockups para las páginas del plan de dieta y de análisis**
 - Páginas con plan de dieta como página raíz
 - Página de Análisis:
- **Elección y contratación de servidor y dominio.**
 - Contratado un hosting en Raiola Networks y un dominio en Namecheap. Se produce un retraso debido a la forma de pago en Raiola Networks.
- **Subida de Drupal desde la instalación local al servidor contratado.**
 - Notable retraso por problemas con el hosting contratado.
- **Creación de una app móvil de prueba.**
 - No se ha podido concluir en esta etapa. Queda pendiente para la próxima. Sin embargo se ha avanzado en el desarrollo de la página de detalle, por lo que no modifica en exceso el plan para la etapa siguiente.

Etapas 2: Plan de dieta (Página principal del plan de dieta e información nutricional):

- **Configuración de hosting: Resolución de problema con DNS.**
 - Con el fin de resolver problemas de conexión con el hosting en el que se aloja el servicio Web implementado en Drupal, se realizan tareas de investigación y configuración de los DNS en el equipo de trabajo local. La resolución de dichos problemas retrasa el desarrollo 4 días.
- **Desarrollo en Ionic: Implementación de la página del buscador de alimentos (paginación).**
 - Implementación del paginador en la página home.html y home.ts (botones anterior y siguiente)(*Documentación oficial de ionic*):
 - Se incluye un offset en el sufijo de la url del proveedor Rest para acceder a los 10 elementos de la página elegida "page[offset]= 10*this.numPag" siendo numPag el número de página. Sin incidencias.

- Implementación de las funciones nuevaPagina() y paginaAnterior() que cambian el valor de la variable numPag. Sin incidencias.
- Se añade a la url del proveedor RestProvider un sufijo para ordenar los alimentos alfabéticamente (*Documentación del módulo jsonapi de Drupal*) al aparecer en el listado de búsqueda.
-
- **Desarrollo en Ionic: Implementación de la página de distribución de alimentos en ingestas y días de la semana (Plan.html).**
- - Creación mediante la línea de comandos de la página “plan” (*Nicolas Molina*). Sin incidencias.
 - Realización inicial del diseño base de la página Plan en html. Sin incidencias.
 - Implementación en paralelo con el proveedor variablesGlobales:
 - Después de crear el proveedor variablesGlobales y una primera aproximación a la variable global planDeDieta se implementa el acceso desde plan.ts a dicha variable. En este punto surgen numerosos problemas que retrasan la implementación relacionados con la creación y manejo de arrays de objetos.
 - Se implementan la función eliminarItem() que ejecuta la función borrarUnAlimento() implementada en variablesGlobales.ts. Sin incidentes.
 - Se implementa la función updateComidaDelDia() que muestra en pantalla los alimentos según el día de la semana elegido. Sin incidencias
 - Implementación en paralelo con planadd.ts:
 - Implementación de la función de navegación abrirPlanaddPage() para acceder a la página planadd.html. Sin incidencias.
 - Implementación de la función eligeCantidad() que dispara un cuadro emergente con un área para introducir una cantidad en gramos (TutorialsPlane.com). Al mismo tiempo fué necesario añadir un campo “field_racion” al tipo de contenido “alimento” en Drupal. Sin incidencias.

- Rediseño de la página para incluir botones ocultos deslizantes. Se prueban varios diseños diferentes optando por los item-sliding de ionic . Sin incidencias.
- Implementación en paralelo con info.html:
 - Implementación de la función de navegación abrirInfoPage() que accede a info.html. Sin incidencias.
- **Desarrollo en Ionic: Nuevo proveedor “variablesGlobales” para la variable global planDeDieta.**
 - Implementación de una variable planDeDieta.
 - Se implementa la variable planDeDieta de una clase creada previamente planSemanal, la cual se construye a partir de las clases comidaDiaria y planDiario (*Alligator.io*) (*w3schools.com*) (*Choly*). Su creación da numerosos problemas al producir errores de variable no definida que retrasan el desarrollo.
 - Uso de esta variable en la implementación de la página plan.html.
 - Sustitución de la variable planDeDieta de la clase planSemanal por planDeDieta de tipo any, la cuál se inicializa con un objeto con la estructura apropiada. Sin excesivas incidencias.
- **Desarrollo en Ionic: Implementación de la página de búsqueda de alimentos para añadir al día (Planadd.html).**
 - Creación mediante la línea de comandos de planadd y copia (con ligeras modificaciones) de la página home.html y home.ts en planadd.html y planadd.ts. Sin incidencias.
 - Diseño diferencial con respecto a home.html. Sin incidencias.
 - Implementación de la función updatePlanDeDietas() que añade un alimento seleccionado al plan de dieta. Sin incidentes.
 - Se crea el array icono que permite cambiar el icono del alimento seleccionado. Sin incidentes.

- **Desarrollo en Ionic: Implementación de la página de información nutricional según cantidad (Info.html).**
 - Creación mediante la línea de comandos de la página “info” y copia (con ligeras modificaciones) de la página nutrientes.html y nutrientes.ts en info.html e info.ts (*Mark Pilgrim*). Sin incidencias.
 - Implementación de info.html para incluir el cálculo de nutrientes según la cantidad de alimento seleccionada (campo ración).
- **Desarrollo en Ionic: Implementación de la página que muestra el cálculo de calorías y nutrientes de la semana (analisis.html).**
 - Creación mediante la línea de comandos de la página “ianalisis” y copia (con ligeras modificaciones) de la página nutrientes.html y nutrientes.ts en analisis.html y analisis.ts (*Mark Pilgrim*). Sin incidencias.
 - Implementación de la función sumaPlanDeDieta() que suma los valores de nutrientes de los alimentos añadidos a la variable global planDeDieta. Sin incidencias.
 - Modificación pertinente de analisis.html para mostrar estos datos. Sin incidencias.

Etapas 3: Guardar y cargar en formato JSON:

- **Desarrollo en Ionic: Implementación de la funcionalidad para guardar en formato JSON en sqlite o local store el plan de la semana.**
 - Investigación previa (*Ionic Academia*) (*Mark Pilgrim*):
 - Por un lado, La necesidad de investigar las opciones posibles de almacenamiento de datos en diferentes dispositivos y de tomar decisiones con respecto a la forma de implementar esta funcionalidad produjo un retraso de 3 días en el desarrollo.
 - Implementación utilizando “ ionic storage “ (*Pontdeveloper*):
 - Se crea un nuevo proveedor: datosdieta.ts, con los métodos setPlan() y removePlan(). Al tiempo se implementa la función guardarPlan() y eliminarPlan() en la página cargar.html.

- Las dificultades para combinar en la implementación dos proveedores; variablesglobales y datosdieta dificultan la implementación y retrasan el desarrollo.
- Surgen numerosos problemas en el desarrollo debidos al manejo de llamadas asincrónicas (promise) que retrasan el proceso de desarrollo (*Asim*)(*Forum*)(*StackOverflow*).
- **Desarrollo en Ionic: Implementación de la funcionalidad para cargar en formato JSON el plan de la semana.**
 - Implementación de la función cargarPlanes() y getTodosLosPlanes():
 - Desarrollo de la función cargarPlanes() en cargar.html que llama a la función getTodosLosPlanes() para mostrar un listado de todos los planes almacenados en el json. Surgen numerosas complicaciones relacionadas con la carga asincrónica de datos (*Asim*)(*Forum*)(*StackOverflow*).
 - Implementación de las funciones removePlanes(), removePlan(), borrarPlanes() y borrarPlan().
 - Implementación de borrarPlanes() en cargar.ts que llama a la función removePlanes() de datosdieta.ts. Sin incidencias.
 - Implementación de eliminaPlan() en cargar.ts que llama a la función removePlan() de datosdieta.ts. Problemas para mostrar los cambios después de realizar el borrado.
- **Desarrollo en Ionic: Gráficos de donut y barras para la página de análisis.**
 - Implementación de dos gráficos en la página análisis.html. Fue necesario instalar chart de Angular 2 y chart.js. Sin incidentes.
- **Diseño en Gimp: Creación de imágenes en Gimp adaptadas a la aplicación y Diseño responsive para la aplicación Web.**
 - Diseño de gráficos en Gimp para los avatares de los listados de alimentos y nutrientes.
 - Diseño responsive (); Según el dispositivo, la barra con las pestañas se muestra en diferente posición. Se crean las variables tabsPlacement y tabsLayout en

tabs.html y tabs.ts. Modificación del constructor de tabs.ts para que detecte el tipo de plataforma. Se incluye en src/theme/variables.scss los puntos de ruptura para grid-fixed.

- **Desarrollo en Ionic y Drupal: Listados de alimento con imágenes cargadas desde la Rest Api de drupal.**
 - Creación de un nuevo campo img de tipo imagen en el tipo de contenido alimento en Drupal. Sin incidencias.
 - En el proveedor rest.ts , en la llamada realizada a jsonapi de Drupal, se incluye la relación con las imagenes y sus Uris. Es necesario hacer muchas pruebas e investigar cómo y dónde encontrar las urls. Se producen retrasos en la planificación.
 - Implementación de la función imagenUrl() para colocar cada imagen en su correspondiente elemento. Sin incidencias.

- **Desarrollo en Ionic: Nueva página home.html.**
 - Impementación de home.html y home.ts tomando como plantilla la que se presenta como ejemplo en ionicframework.org. Sin incidentes.

- **Desarrollo en Ionic: Modificaciones en plan.**
 - Se modifica plan.html para que muestre la información nutricional correspondiente al día elegido. Se implementa sumaPlanDeDietaHoy(). Sin incidencias.
 - Sustitución del icono que indica la posibilidad de descubrir los botones ocultos. Sin incidencias. Para ello se implementa el css en app.scss creando un icono personalizado [12]. Sin incidencias.

- **Desarrollo en Ionic: Modificaciones en planapp.**
 - Se modifica updatePlanDeDieta() en planapp.ts para que al pulsar repetidas veces se añada más cantidad. Para ajustarse al plan de trabajo el esfuerzo en su programación se intensifica.
 - Se implementa marcarSiExisten(); esta función permite que se marquen los iconos de los Items que se encuentran ya seleccionados en el plan de dieta. Su implementación retrasa el despliegue de la aplicación.

- **Despliegue de la primera versión funcional de la aplicación Equiplan.**
 - Se ejecuta ionic cordova android build --prod --release y ionic cordova browser build --prod --release en el terminal. Aunque se utiliza en este ultimo caso --uglify--jsminify --jsoptimizer y --cssminify no se logran dichos efectos (*Damir's Corner*). No queda tiempo para investigar en mayor profundidad.
 - Modificación de index.html en la plataforma browser para activar los workers y desactivar cordova.js (*Josh Morony*). Queda pendiente investigar como eliminar el modo desarrollo para pasar a modo producción.
 - Subida al servidor de la aplicación Web.

5.2 Temporalización:

Temporalización de las tres etapas principales:

		Temporalización		
		Inicio	Final	Días
HITOS	Buscador de Alimentos (Conexión Back-end y Front-end)	27/03/18	23/04/18	28d
	Aprendizaje y pruebas con Ionic/angular.	27/03/18	02/04/18	6d
	Instalación y configuración de DRUPAL localmente	03/04/18	03/04/18	1d
	Creación de entidades (Tipo de Contenido Alimentos, Taxonomías).	04/04/18	04/04/18	1d
	Instalación del plugin JsonApi para drupal	04/04/18	04/04/18	1d
	Aprendizaje y pruebas con el plugin JsonApi	05/04/18	07/04/18	3d
	Introducción de algunos alimentos utilizando las tablas nutricionales del BEDCA como guía.	08/04/18	09/04/18	2d
	Instalación, configuración y pruebas con el plugin JsonApi Extras en Drupal	10/04/18	10/04/18	1d
	Instalación de Ionic y creación/configuración de la app	11/04/18	11/04/18	1d
	Desarrollo en Ionic: Conectar con la REST API.	12/04/18	15/04/18	3d
	Desarrollo en Ionic: Implementación básica de la página del buscador de alimentos.	16/04/18	17/04/18	2d

	Desarrollo en Ionic: Mejora de la función del buscador de alimentos.	17/04/18	18/04/18	2d
	Desarrollo en Ionic: Implementación de la página detalle con los nutrientes para cada alimento.	19/04/18	20/04/18	2d
	Desarrollo en Ionic: Diseño y mejora de la página de detalle con los nutrientes para cada alimento.	21/04/18	21/04/18	1d
	Diseño de Mockups para las páginas del plan de dieta y de análisis	22/04/18	22/04/18	1d
	Elección y contratación de servidor y dominio.	22/04/18	22/04/18	1d
	Subida de Drupal desde la instalación local al servidor contratado.	23/04/18	23/04/18	1d
	Redacción del Informe.	03/04/18	23/04/18	21d

Tabla 1: Tabla de hitos de la primera etapa.

		Temporalización		
		Inicio	Final	Días
HITOS	Plan de Dieta (Página principal y páginas con información)	24/04/18	21/05/18	28d
	Configuración de hosting: Resolución de problema con DNS	24/04/18	27/04/18	4d
	Desarrollo en Ionic: Implementación de la página del buscador de alimentos (paginación).	28/04/18	29/04/18	2d
	Desarrollo en Ionic: Implementación de la página de distribución de alimentos en ingestas y días de la semana (Plan.html).	30/04/18	16/05/18	17d
	Desarrollo en Ionic: Nuevo proveedor "variablesGlobales" para la variable global planDeDieta.	05/04/18	05/05/18	2d
	Desarrollo en Ionic: Implementación de la página de búsqueda de alimentos para añadir al día (Planadd.html).	05/06/18	15/05/18	10d
	Desarrollo en Ionic: Implementación de la página de información nutricional según cantidad (Info.html).	16/05/18	16/05/18	1d

	Desarrollo en Ionic: Implementación de la página que muestra el cálculo de calorías y nutrientes de la semana (analisis.html).	17/05/18	19/05/18	3d
	Redacción del informe	24/04/18	21/05/18	28d

Tabla 2: Tabla de hitos de la segunda etapa.

		Temporalización		
		Inicio	Final	Días
HITOS	Cierre: Guardar y cargar en formato JSON	22/05/18	06/11/18	21d
	Desarrollo en Ionic: Implementación de la funcionalidad para guardar en formato JSON en sqLite o local store el plan de la semana.	22/05/18	29/05/18	8d
	Desarrollo en Ionic: Implementación de la funcionalidad para cargar en formato JSON el plan de la semana.	30/05/18	31/05/18	2d
	Desarrollo en Ionic: Gráficos de donut y barras para la página de análisis.	06/01/18	06/01/18	1d
	Diseño en Gimp: Creación de imágenes en Gimp adaptadas a la aplicación y diseño responsive.	06/02/18	06/03/18	2d
	Desarrollo en Ionic y Drupal: Listados de alimento con imágenes cargadas desde la Rest Api de drupa.	06/04/18	06/05/18	2d
	Desarrollo en Ionic: nueva página home.html	06/06/18	06/06/18	1d
	Desarrollo en Ionic: Modificaciones en plan.	06/07/18	06/07/18	1d
	Desarrollo en Ionic: Modificaciones en planapp.	06/08/18	06/10/18	3d
	Despliegue	06/11/18	06/11/18	1d
	Redacción del informe final.	22/05/18	06/11/18	21d

Tabla 3: Tabla de hitos de la tercera etapa.

5.3 Diagramas de Grantt:

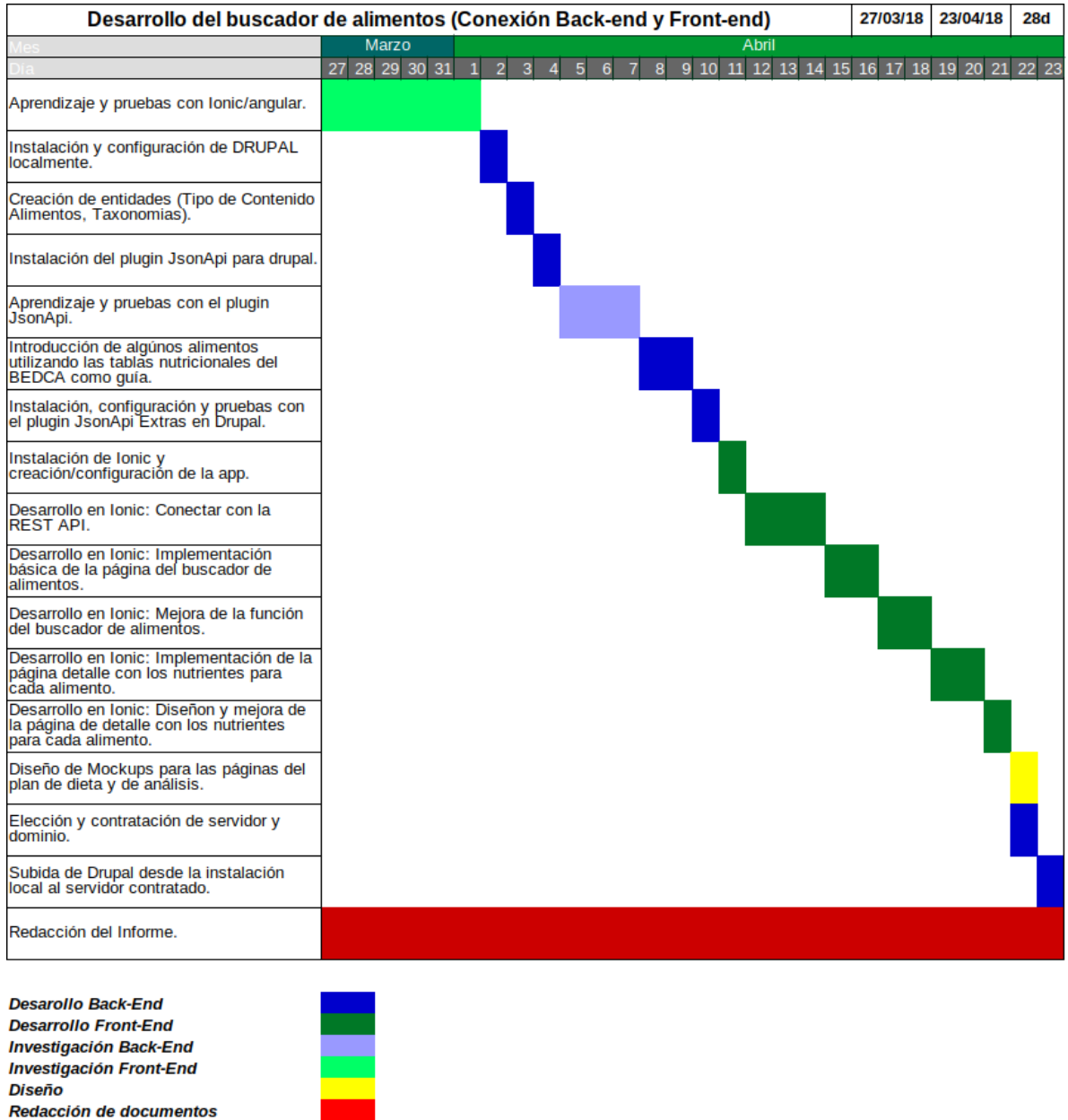


Figura 4: Diagrama de Gantt para la primera etapa.

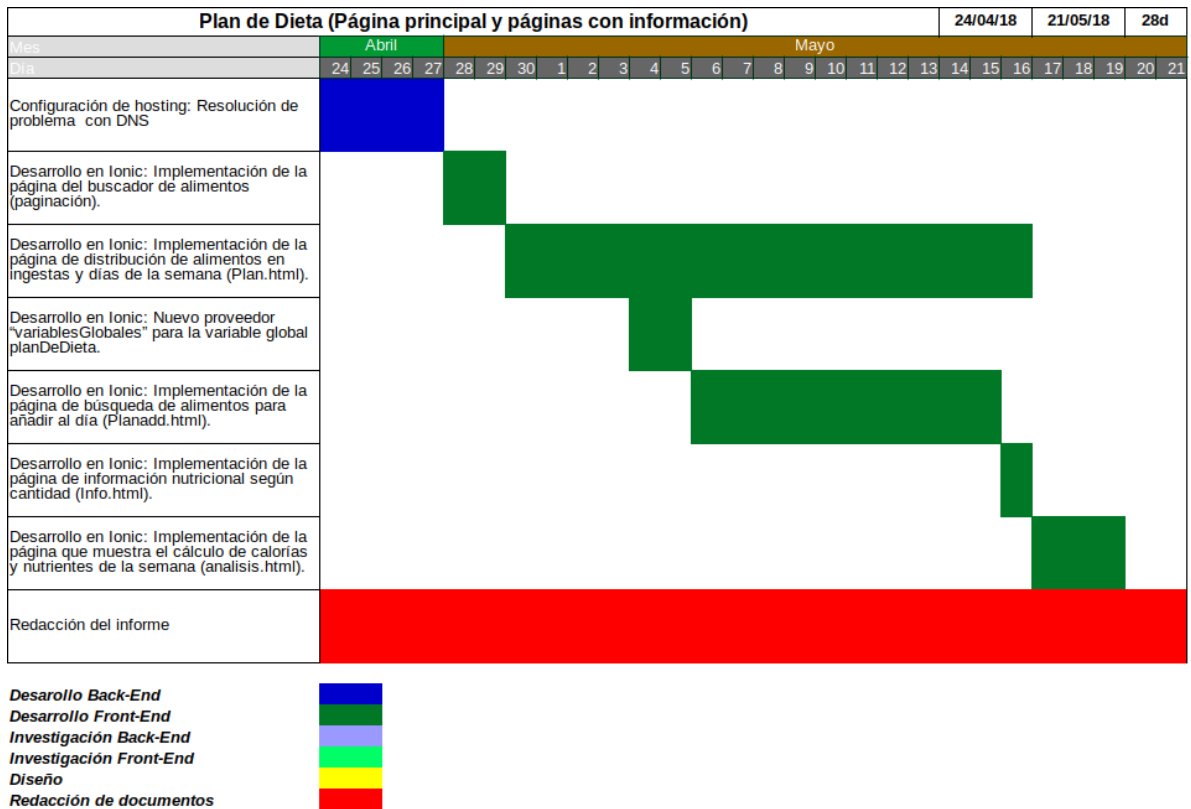


Figura 5: Diagrama de Gantt para la segunda etapa.

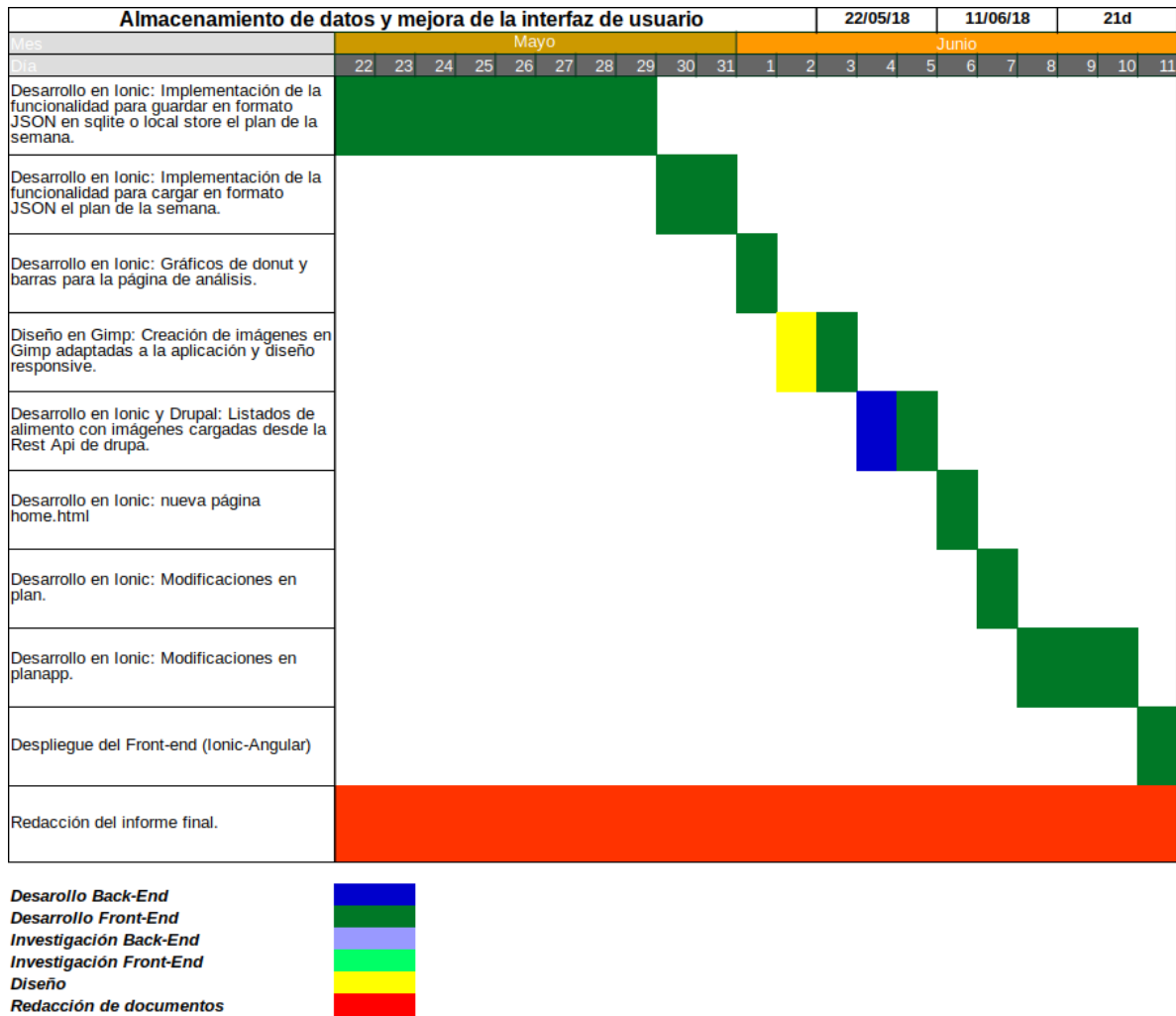


Figura 6: Diagrama de Gantt para la tercera etapa.

6. Presupuesto

Dividido según la etapa o hitos principales:

- Primer hito: Conexión Back-end Front-end.
 - Total de horas: 40.
 - Coste: 800 euros.

- Segundo hito: Página principal.
 - Total de horas: 40.
 - Coste: 800 euros.

- Tercer hito: Almacenamiento de datos y mejora de la interfaz de usuario.
 - Total de horas: 30.

- Coste: 600 euros.

Coste total: 2200 euros.

7. Información en los siguientes capítulos

En los siguientes capítulos se analiza el presente de las aplicaciones en el ámbito de la nutrición y la salud (análisis), La estructura, árbol de navegación y las elecciones en el diseño gráfico de la aplicación (Diseño), los requisitos necesarios para poner en marcha el proyecto y la url de la página en funcionamiento (demostraciones).

Capítulo 2: Análisis

1. Estado del arte

Varios estudios (*Müller AM y col.*, *Semper HM y col.*, *Fabrizio Bert.*, *Steven S. Coughlin y cols.*, *Laura Dennison, y cols.*, *Julia Kolodziejczyk y cols.*) evidencian la gran utilidad que las aplicaciones para dispositivos móviles pueden presentar como herramientas de ayuda en el mantenimiento de una dieta equilibrada, en el mercado podemos encontrar diversos tipos de aplicaciones con esta función. Destacan, por su amplia difusión, las aplicaciones que facilitan el conteo de las calorías y macronutrientes de los alimentos o productos consumidos. Otras, muy abundantes en las tiendas virtuales de aplicaciones, presentan una dieta preestablecida y ayudan a seguirla, algunas aplicaciones son más ambiciosas y ofertan dietas personalizadas con la posibilidad de consultar a un experto, existen también aplicaciones tipo herramienta para personas con intolerancias o enfermedades específicas, por último, se pueden encontrar una gran diversidad de aplicaciones dedicadas a informar y al aprendizaje sobre dieta y nutrición (*Pablo Espeso*).

2. Análisis del mercado

A continuación se analizan los diferentes tipos de herramientas que encontramos en el mercado.

Contar calorías:

Son utilizadas con frecuencia a la hora de seguir una dieta dado que ayudan a hacer un conteo de calorías diarias. Constan de un buscador de alimentos y la posibilidad de incluir la cantidad ingerida del mismo, después, añade las calorías, macronutrientes y micronutrientes al total diario. Algunas pueden leer los códigos de barras de los productos con la cámara y buscarlos directamente en su base de datos (*Julie Weed y col.*).

Un buen ejemplo de este tipo de apps es **MyFitnessPal**

Dieta preestablecida para seguirla:

Dan a elegir entre diferentes dietas para seguir con varios menús y recetas. En esta categoría entran numerosas aplicaciones que se basan en las llamadas dietas milagro.

Dos ejemplos son **Dietas Para Adelgazar, Mi Dieta**.

1. Dietas personalizadas:

Algunas aplicaciones tratan de crear una dieta personalizada y adaptada a las necesidades del usuario. Estas aplicaciones están respaldadas por un equipo de nutricionistas, que elabora una dieta personalizada dando la posibilidad de consultar con ellos vía chat.

Una de estas aplicaciones respaldadas por nutricionistas detrás de la pantalla es **Nootric**.

Herramientas para personas con intolerancias o problemas de salud:

Ejemplos de este tipo de apps son (*Adda Bjarnadottir, MS, José Ángel*):

- **MySugr Logbook:** una app que permite llevar un registro del azúcar que consume el usuario.
- **CeliCity:** Relacionada con una gran comunidad de celíacos, permite a los usuarios encontrar restaurantes, cafeterías, pequeños comercios o supermercados que ofrecen alguna opción libre de gluten.

- **GlutenFreeList:** Ofrece información sobre productos y fabricantes de alimentos sin gluten y, por último.
- **¿Qué puedo comer?:** Contiene información alérgica de más de 20.000 productos entre los que se incluyen ingredientes tales como la leche, los frutos secos, la soja, el pescado, los crustáceos, los huevo, el gluten, los sulfitos, los moluscos, la lactosa y la fructosa.
- **FoodAllergyTranslate:** Permite traducir alérgenos entre el inglés, francés, alemán y español.

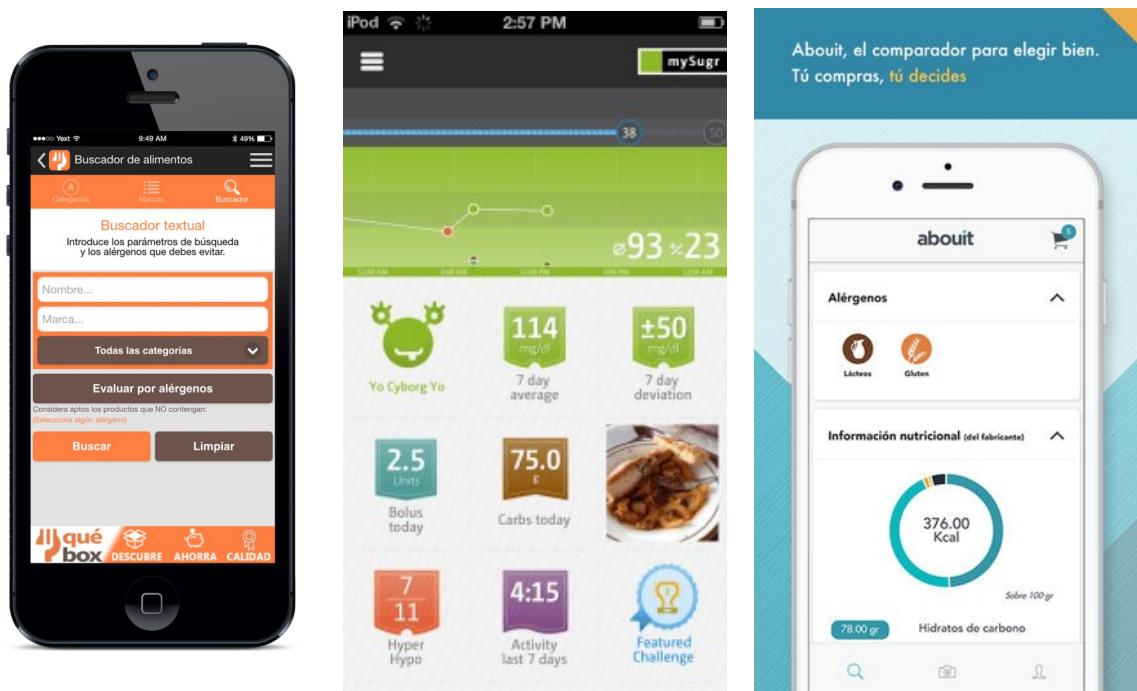


Figura 7: Pantallas de ¿Qué puedo comer?, mySugr LogBook y About.

About (Laura Martínez): Evalúa marcas y productos según el impacto que provocan en la salud, la sociedad y el medio ambiente analizando bienes de consumo de sectores como alimentación, cosmética y cuidado personal, limpieza del hogar, tecnología y electrodomésticos y calificándolos según sean más menos saludables, seguros y sostenibles

Información y aprendizaje en materia de nutrición:

Aplicaciones que facilitan el aprendizaje de qué alimentos son más saludables, cuál es su origen, cómo se cocinan, cómo se almacenan ...

Destacan algunas de ellas (Pablo Espeso).

- **True Food Shoppers Guide:** permite conocer los ingredientes con los que se realizan ciertos alimentos, y si éstos han sido modificados genéricamente.
- **Harvest:** Ofrece trucos y respuestas relacionadas con 120 frutas y verduras.
- **ShopWell:** Da información relativa a todo tipo de productos al escanear su código de barras.
- **Food Truth:** Con información sobre alimentos y con recetas saludables.
- **Fooducate** Para llevar a cabo una dieta sana y equilibrada. Incluye un *registro de salud* y permite escanear los códigos de barras de los alimentos.



Figura 8: Pantallas de la app Fooducate con calificación del producto.

Juegos educativos:

Juegos que facilitan el aprendizaje de como realizar una dieta sana y equilibrada. Algunos ejemplos de este tipo son (*Pablo Espeso*);

- **Diana come Sano:** Con juegos y puzzles.
- **Eat Smart:** Juego de plataformas.

- **Awesome Eats:** Juegos sencillos para niños de 6 o más años.



Figura 9: Pantallas de las apps Diana Come Sano, Eat Smart y Awesome ats.

Versión Web app:

Algunas de estas aplicaciones para móviles presentan una versión Web, ejemplos de estos casos son [9];

- **Fatsecret:** Popular contador de calorías con una enorme base de datos de alimentos.
- **lenva:** Calculadora de dietas de El Centro de Investigación de Endocrinología y Nutrición clínica (CIENC).
- **Sparkpeople:** Controla la nutrición, la actividad y el progreso en la consecución de objetivos nutricionales.

Otros similares son *Loose It*, *Nutritionix Track*, *Cron-o-meter* y *myFitnessPal*.

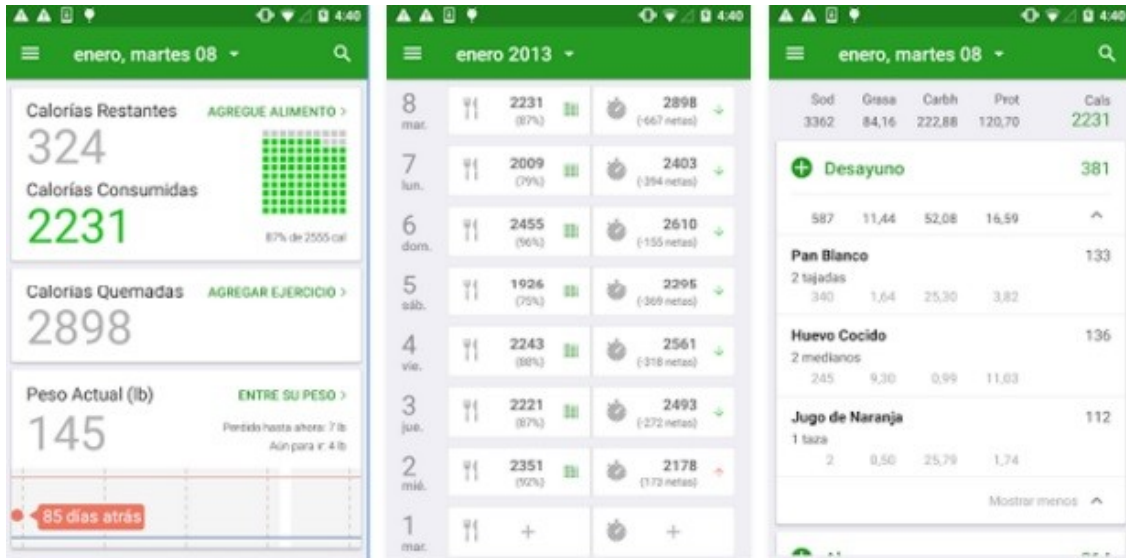


Figura 10: Pantallas de la aplicación Fatsecret.

3. Análisis de un caso de éxito: myFitnessPal

General: Se lanzó para plataforma web en 2005, para iPhone en 2009, para Android en 2010 y para iPad en diciembre de 2011. Se trata de uno de los contadores de calorías más populares. Realiza el seguimiento del peso y calcula la ingesta calórica diaria. Contiene un diario de alimentos y un registro de ejercicios. La página de inicio proporciona una imagen clara de la cantidad de calorías que ha consumido durante el día. También muestra cuánto queda para alcanzar la ingesta recomendada así como las calorías quemadas al hacer ejercicio. Cada día se presenta como un gráfico circular, que muestra la descomposición de carbohidratos, proteínas y grasas. El usuario puede escribir una nota para el día, cómo transcurrió el día o cómo se sintió.

Hardware utilizado: MyFitnessPal puede sincronizarse con dispositivo de seguimiento de actividad física e incluir los datos en el registro de ejercicios. Además, puede ingresar la información nutricional de algunos alimentos envasados mediante escaneo del código de barras.

Dimensión social: Ofrece foros de chat que incluyen conversaciones, recetas, consejos e historias de éxito personal.

Base de datos: Contiene más de cinco millones de alimentos. El usuario puede crear comidas y platos personalizados o descargar recetas de Internet. La mayoría de los alimentos son cargados por otros usuarios, por lo que pueden no ser del todo precisos. También puede haber entradas múltiples para el mismo producto.

Precio: Ofrece una versión gratuita reducida y una versión premium con acceso a todas las funcionalidades por 49.99 \$ al año.

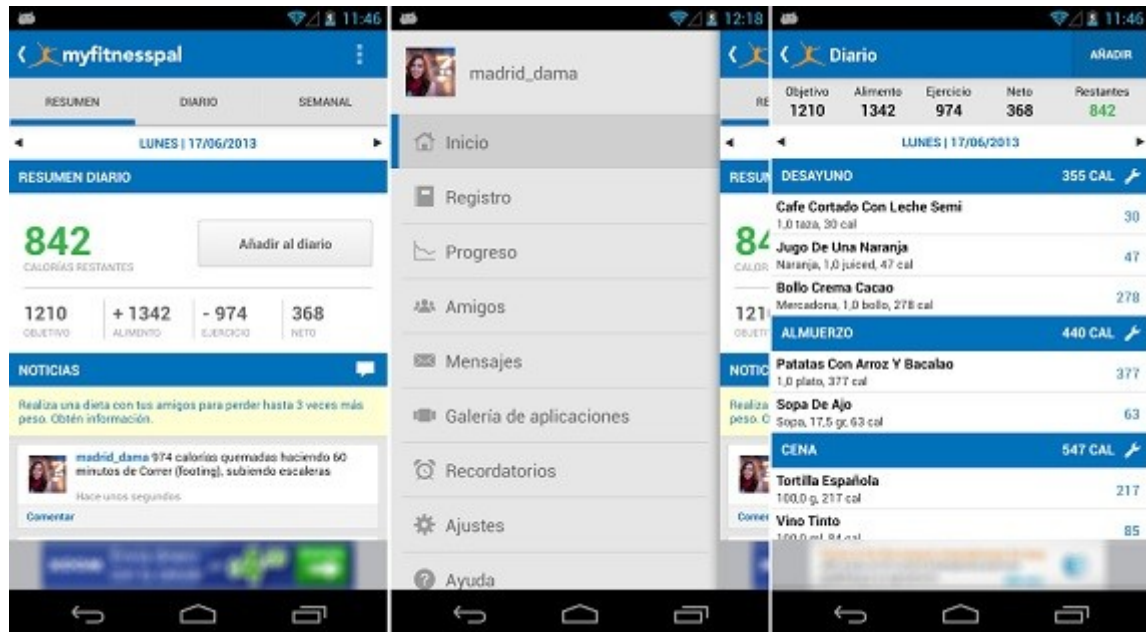


Figura 11: Pantallas de la aplicación myFitnessPal.

El éxito: Avalado por 80 millones de usuarios repartidos por todo el mundo. Alguna de las posibles razones para ello pueden estar relacionadas con la enorme base de datos de alimentos de la que se abastece, su dimensión social con la posibilidad de agregar amigos para fortalecer el refuerzo positivo y su capacidad para crear un ecosistema integral con otras aplicaciones o dispositivos que se conecten por wifi. Todo ello sin olvidar que la aplicación ha sido traducida a 15 idiomas.

El peligro de contar calorías: Muchos son los factores que dificultan el conteo de calorías (Cynthia Graber y cols, Beatriz Portinari):

- Las simples variaciones en las cantidades y porciones utilizadas en la confección de un plato.
- El propio proceso de cocinado, que desata estructuras microscópicas que aumentan la energía de los alimentos, reduciendo el trabajo que hace el tubo digestivo para extraerlos.
- El procesado industrial, que somete los alimentos a muy altas temperaturas y presiones, libera aún más calorías.
- El problema de la inexactitud de las porciones.
- La temperatura de la región (el frío quema calorías per se).
- Incluso algo tan insignificante en apariencia como el momento en que comemos puede afectar la manera en que procesamos la energía.

- Los pesticidas, antibióticos, hormonas de crecimiento en los animales, provocan daños en el intestino e inducen a que las calorías de los alimentos afecten en mayor medida.
- Alimentos muy palatables y adictivos y no saciantes, por lo que en pocas horas volveremos a sentirnos hambrientos.

La mejor opción es enfocarse en otra cosa que no sea la ingesta de energía en sí. Como la saciedad, y tener en cuenta la salud de nuestras vísceras.

Capítulo 3: Diseño

1. Arquitectura general de la aplicación

Para el desarrollo de la aplicación se ha planteado establecer una arquitectura cliente / servidor desacoplada. Es decir, una parte front-end (Vista) donde utilizar la tecnología cliente Angular4 y Ionic3, y otra Back-end con el CMS Drupal 8 y el módulo JSONAPI, de tal forma que el back-end proporciona un conjunto de servicios REST que se comunican mediante el uso de JSON.

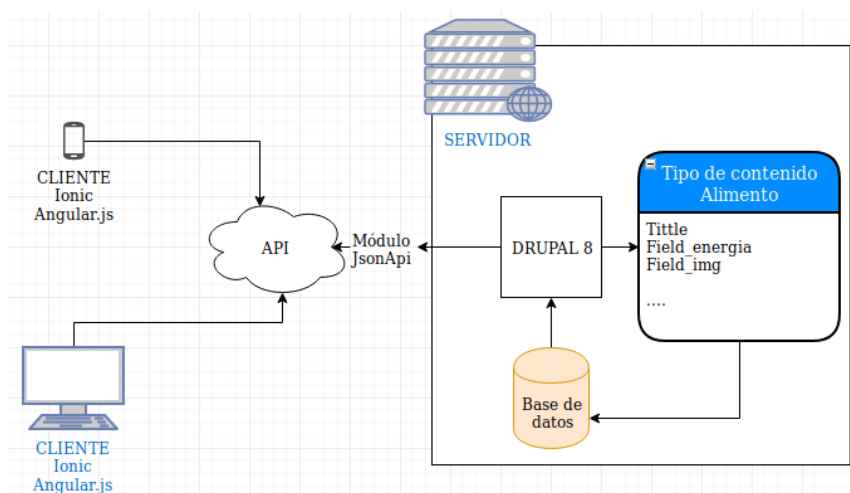


Figura 12: Arquitectura general de Equiplan.

2. Arquitectura de la información y diagramas de navegación

2.1 Arquitectura de la información:

La información relacionada con cada alimento se almacena en la base de datos conectada a la instancia de Drupal 8. Cada contenido del tipo Alimento incluye los campos *title* (contiene el

nombre del alimento), *field_img* (contiene un id que se relaciona con la url de una imagen) y campos numéricos con la información relativa a diferentes nutrientes.

La instalación de Drupal 8 incluye el módulo Jsonapi, cuya función es la de organizar la salida de la información que genera la Rest Api de Drupal (cuyos módulos, que se encuentran en el core de Drupal 8, deben estar activados).

El resultado es un listado de los contenidos con sus campos correspondientes y los datos que estos almacenan en formato Json.

En nuestro caso, el resultado se sirve en esta Url :
“www.nutricioneco/app/drupal/jsonapi/node/alimento”.

Esta salida puede perfilarse mediante el uso de diversos filtros que se añaden a la Url. Así, esta Url mostrará solo 10 alimentos ordenados alfabéticamente por el campo title:
“[www.nutricioneco/app/drupal/jsonapi/node/alimento?page\[limite\]=10&sort=title](http://www.nutricioneco/app/drupal/jsonapi/node/alimento?page[limite]=10&sort=title)”.

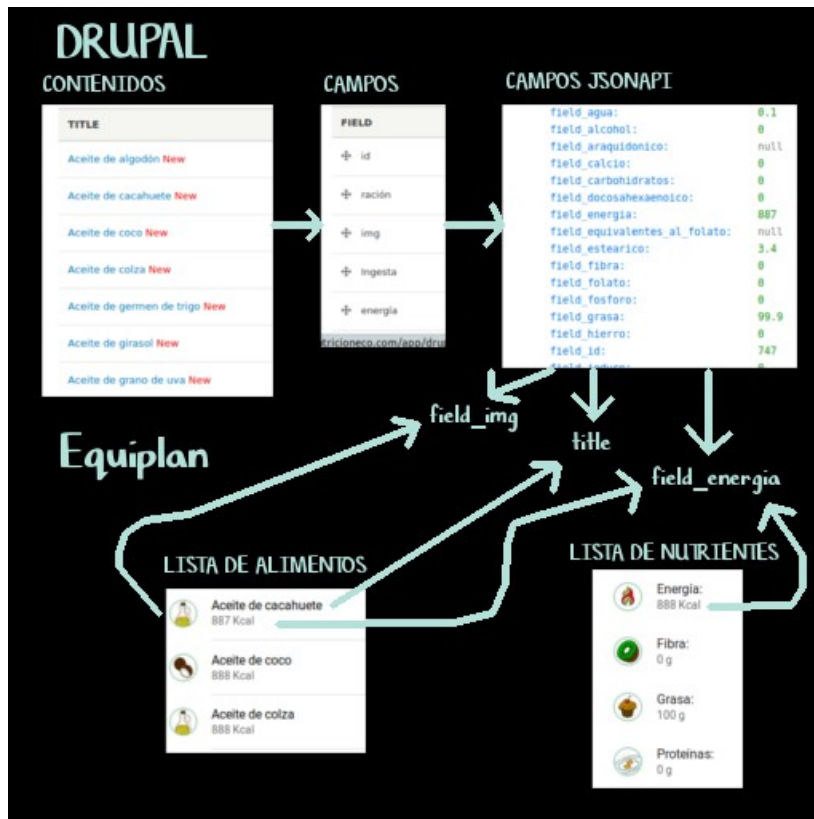


Figura 13: Relación de los componentes del Back_end y el Front_end.

Desde esa Url, la información es consumida de forma asíncrona desde el lado del cliente, almacenándose la parte del Json generado por la API de Drupal que interesa en la variable “*alimentos*”. Angular4 participa en mostrar el contenido de esta variable en pantalla. Otras funciones incluidas en la aplicación utilizan la información consumida de este modo para realizar cálculos, guardar dicha información en una variable global para su posterior manipulación o en SQLite/local storage según lo que esté disponible.

2.2 Diagrama de navegación:

El elemento principal de navegación lo conforma el panel que se presenta en todas las páginas de la aplicación. Este panel incluye cuatro pestañas:

- **Inicio** dirige al usuario hasta `home.html`: Página de presentación con información sobre la aplicación. Esta página contiene un botón que lleva a los planes de dieta guardados por el usuario (`cargar.html`).
- **Buscador** dirige al usuario hasta `buscar.html`: Listado de alimentos por orden alfabético. Cada elemento de la lista es un botón desde el que se accede a información nutricional (`nutrientes.html`).
- **Plan de dieta** dirige al usuario hasta `plan.html`: Página principal de la aplicación en la que se construye el plan de dieta. Desde `plan.html` se accede a `planadd.html`, a `info.html` y a `cargar.html`.
- **Análisis** dirige al usuario hasta `analisis.html`: Muestra los valores nutricionales totales.

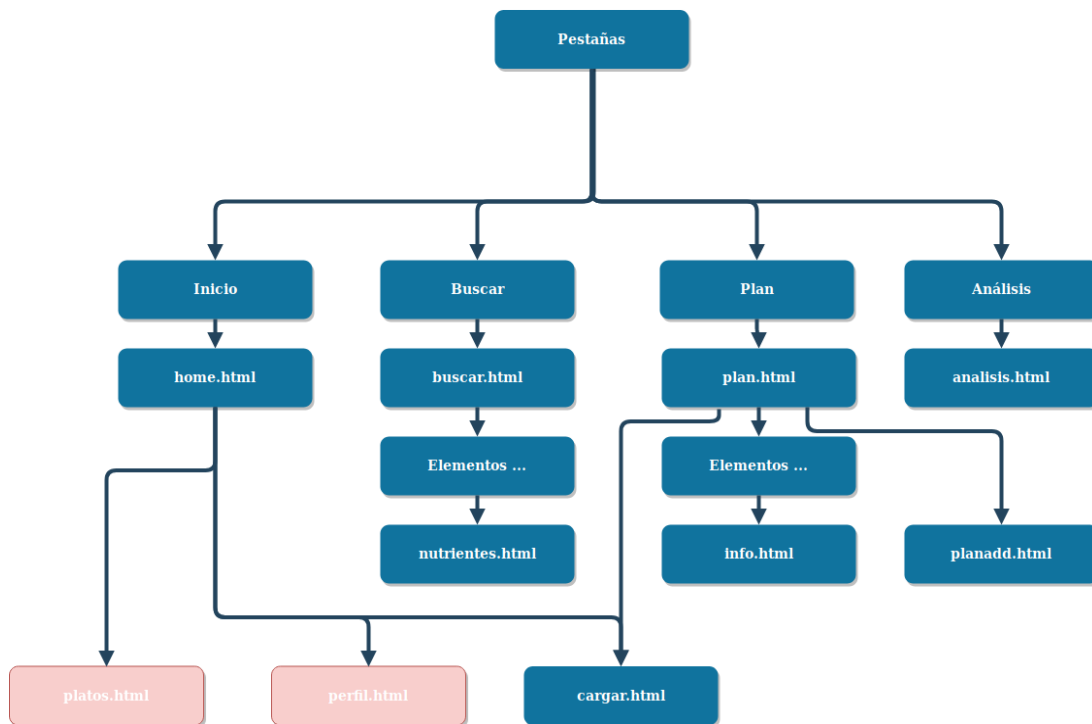


Figura 14: Diagramas de navegación. En rosa se encuentran las páginas que todavía no han sido implementadas.

Desde home.html tres botones conectan con las tres páginas donde el usuario puede cargar o guardar sus creaciones y perfil (Tus planes de dieta (cargar.html), Tus platos combinados y Tu perfil). Las dos últimas páginas no han sido desarrolladas todavía.

Desde plan.html puede navegarse hasta planadd.html, desde dónde es posible añadir alimentos al plan de dieta. Es, además, posible acceder a cargar.html mediante un botón colocado en el panel superior para guardar el plan actual, sin necesidad de viajar a la página home.html. También conecta con info.html, donde se puede examinar los valores nutricionales del alimento elegido, referidos a la cantidad añadida de dicho alimento, y no por 100g, tal como se presentan en nutrientes.html.

3. Diseño gráfico e interfaces

Paleta de colores: La interfaz de usuario combina el color blanco de fondo con el color esmeralda como color primario. Esta elección es apropiada para el tema o ámbito en el que se enmarca la aplicación; Mientras que el blanco simboliza pureza, el verde esmeralda es Bienestar, Vida, Relax, Exuberancia, Luminosidad, Descanso y Salud. Siendo una buena combinación para toda aplicación dedicada a la salud. Como color secundario se utiliza el verde oliva, relacionado con lo tradicional y lo seguro, muy en consonancia con los productos saludables de toda la vida.

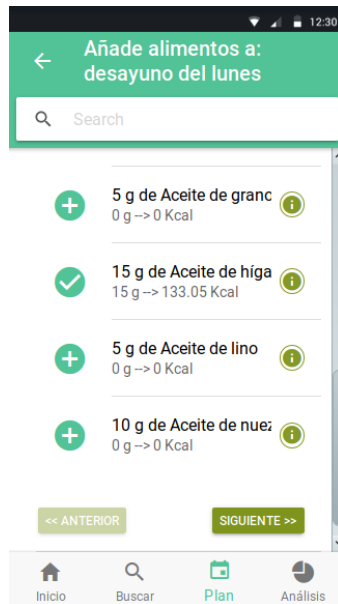


Figura 15: Página planadd.html mostrando la paleta de colores.

Elementos gráficos: Destaca la sencillez y claridad de los gráficos utilizados como avatares de alimentos y nutrientes. Esta sencillez transmite confianza. Se utilizan iconos relacionados con la función del componente en el que se encuentran, para guiar al usuario en la navegación y en el uso de elementos deslizantes.

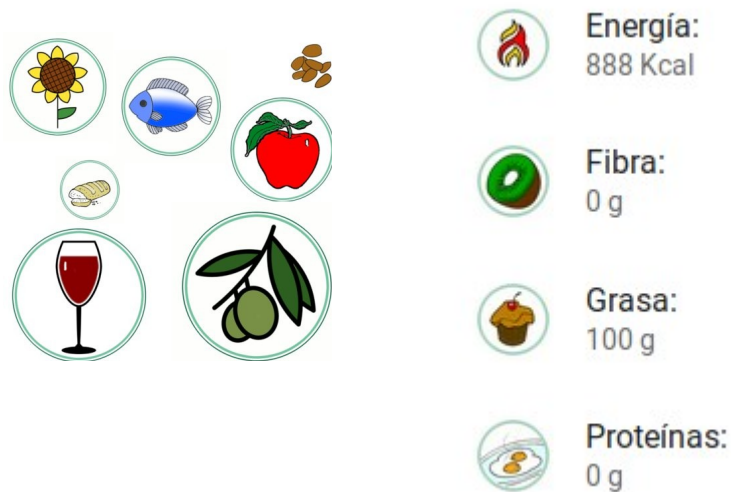


Figura 16: Ejemplo de elementos gráficos.

Diseño adaptable: En los navegadores la barra con las pestañas se sitúa sobre el listado de elementos. En los dispositivos móviles esta barra queda encajada en la base de la pantalla. Además,

se han establecido puntos de ruptura y anchura máxima que evite el crecimiento incontrolado de textos e imágenes hasta tamaños que los hagan difíciles de manejar.

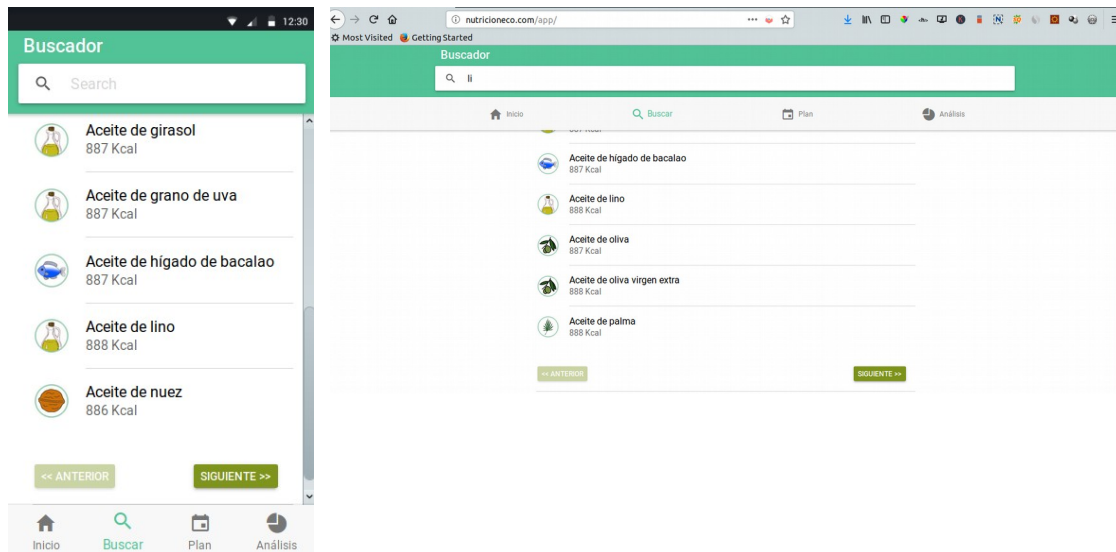


Figura 17: Diseño adaptable.

Paginación: Esta funcionalidad facilita el manejo de los datos por parte de los usuarios de la aplicación Web desde el navegador.

Elementos emergentes: Muestran información y facilitan el manejo de la aplicación en pantallas de pequeño tamaño.

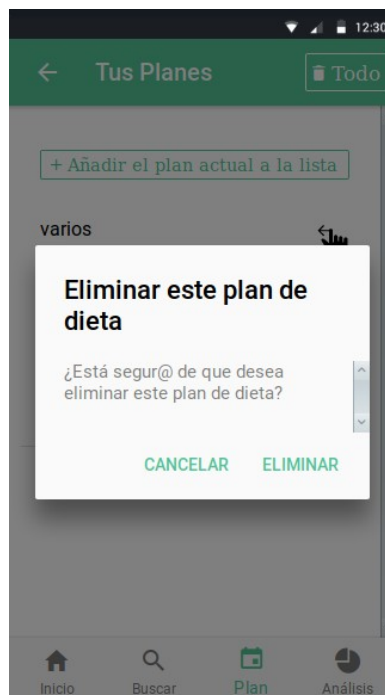


Figura 18: Elemento emergente.

Elementos deslizantes: Para evitar que el usuario pase por alto las funciones que esconden se muestra un icono que explícitamente indica cuál es su función.

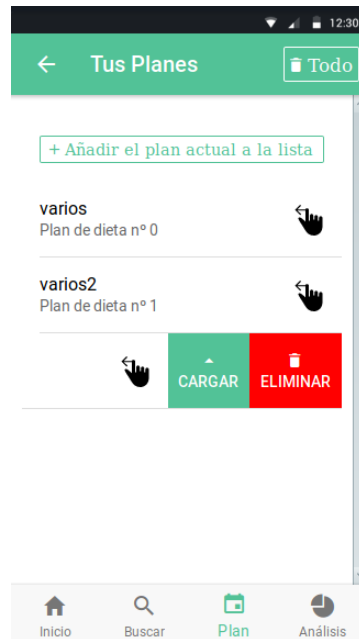


Figura 19: Ejemplo de elemento deslizante ocultando botones.

4. Lenguajes de programación y APIs utilizadas

Descripción del proceso realizado para elegir la plataforma de desarrollo (p.ej. CMS, sistema operativo, lenguaje de programación, herramientas a utilizar, etc.). Enumerar las principales plataformas y herramientas existentes para realizar el producto/servicio y justificar la elección de la solución escogida.

Drupal 8: Se trata de un cms con una comunidad de desarrolladores de gran tamaño, que a madurado desde sus versiones anteriores para constituirse como uno de los CMS más utilizados. Un gran numero de empresas de prestigio utilizan este CMS para desplegar sus contenidos. Por otro lado, es mucho más flexible que WordPress, presentando mucha versatilidad en el manejo de su Rest API. Otras opciones son posibles; CloudCMS, ButterCMS ... orientados por completo a estructuras desacopladas. Estas otras opciones, sin embargo, carecen del extenso número de módulos y ampliación de funcionalidad que presenta Drupal. En una aplicación que puede necesitar nuevas funcionalidades, extensión de su base de datos, carga masiva de datos, un CMS sólido con un ecosistema extenso de módulos que puedan ampliar sus funcionalidades puede ser una buena elección. Es importante considerar también que el equipo tiene experiencia en Drupal 8, mientras que el manejo de los frameworks Angular4 e Ionic3 se escapan de su area de conocimiento. El uso de otro CMS, aumentando la curva de aprendizaje, ya elevada, no se justifica.

Modulo JSONAPI: Aunque el módulo Views permite tener una gran flexibilidad y es plenamente configurable, en lo que respecta al Json de salida que muestra la Rest API de Drupal, no permite exponer los filtros al usuario. Para llevar a cabo una búsqueda mediante un termino introducido por el usuario es conveniente el uso de JSONAPI que si lo permite.

Ionic3: Es uno de los frameworks mas utilizados en el desarrollo de aplicaciones hibridas. React Native es su competencia más reconocida. Ionic, sin embargo, está orientado al desarrollo de aplicaciones Web progresivas, lo que facilita la imlementación de una aplicación para plataformas móviles y, al mismo tiempo, de acceso mediante navegadores.

Angular4: Aunque existen otras tecnologías de peso, como React o Vue, este framework se integra perfectamente con Ionic. Es sencillo encontrar información y tutoriales en los que estas dos tecnologías se unen para alcanzar los objetivos.

Capítulo 4: Demostración

1. Url de demostración

Para una demostración del funcionamiento de la aplicación Web:

- **Web de la aplicación:** www.nutricioneco.com/app

2. Requisitos de instalación en Linux.

Este proyecto se ha realizado desde el principio y en su totalidad sobre Ubuntu, versión 18,04LTS (Bionic Beaver) de 64 bits.

Para poder ejecutar la aplicación desde el proyecto, es necesario tener instalado Node.js y npm.

Desde el terminal (y siempre con conexión a Internet) introducir:

- `sudo npm install -g cordova`
- `sudo npm install -g ionic`
- acceder al directorio del proyecto: `cd '.../equiplan/':`
- `npm install chart.js --save`
- `ionic serve --lab`

3. Prototipos

El Mock-up de la página buscar.html no se realizó debido a la decisión de experimentar primero con el framework Ionic en esta primera página, para hacerse una idea de qué era posible y

cuál era la mejor forma de hacerlo. El Mock-up de la última página implementada home.html tampoco se adjunta. En el plan inicial no se pretendía llegar a realizar esta página dentro de este proyecto.



Figura 20: Mock-ups de plan.html, planadd.html y nutrientes.html



Figura 21: Mock-ups de plan.html, guardar.html y cargar.html. La página guardar.html fue desestimada, incluyendo toda la funcionalidad en cargar.html.



Figura 22: Mock-up de la página analisis.html.

Capítulo 5: Conclusiones y líneas de futuro

1. Conclusiones

La decisión inicial de optar por un proyecto en el que se empleaban tecnologías desconocidas para mí en más de tres cuartos del mismo fue una decisión arriesgada, que en varias ocasiones hizo que me planteara si estaba siendo razonable, si el proyecto no iba a superarme. Sin embargo, me interesaba aprender a desarrollar lo que para mí es presente y futuro del desarrollo Web. Aplicaciones multiplataforma con estructura desacoplada. En este sentido las primeras investigaciones que realicé me indicaron que Ionic podía ser el framework perfecto para un trabajo de desarrollo híbrido, y a la vez, en aplicaciones Web progresivas.

Una vez tomada la decisión, comencé el proyecto perdido en un mar de dudas, sin saber muy bien como abordar los objetivos planteados. Ni siquiera me atreví a diseñar los primeros Mock-ups sin conocer algo más del framework elegido.

La planificación ha sido difícil. ¿Como planificar si no sabes cómo es el camino?. En ocasiones el tiempo para alcanzar uno de los objetivos fijados para la etapa era desproporcionado, otras veces la planificación para otro hito era ridículamente optimista.

En general, los objetivos previstos se han alcanzado en las tres etapas principales en las que se ha dividido el proceso de desarrollo. Algunos objetivos nuevos han surgido según adquiría conocimientos del medio en el que me movía (almacenamiento en sqlite o local store), otros perdían fuerza. Aunque se ha logrado un nivel de desarrollo algo más avanzado de lo planeado en un inicio, algunos puntos fundamentales han quedado en el aire, destacando la posibilidad de exportar los planes de dieta en pdf y Json al disco duro del usuario.

Llegada a esta etapa final del proyecto puedo decir que lo aprendido ha sido mucho, y sin embargo, tengo la sensación de haber penetrado solo la delgada corteza de un sin fin de posibilidades.

Mi opinión inicial se ha reforzado, el presente y el futuro de la Web no puede ignorar las numerosas ventajas que ofrece el desarrollo desacoplado de Back-end y Front-end y, a pesar de sus desventajas en materia de rendimiento frente al desarrollo nativo (desventaja que se reduce conforme la tecnología móvil mejora), poder desarrollar una vez y desplegar el resultado para la Web y para múltiples plataformas móviles no tiene precio.

2. Líneas de futuro

A corto plazo: Durante el desarrollo de la aplicación se han evidenciado varias funcionalidades básicas que deben ser implementadas para aumentar notablemente al utilidad de esta para los usuarios.

- La posibilidad de exportar/importar el plan de dieta en formato Json para guardarlo en la memoria del dispositivo (disco duro, usb, ...) y poder compartirlo con otros usuarios.
- La creación de una página platosCombinados.html que permita guardar combinaciones de alimentos personalizadas para agilizar la creación del plan de dieta. Con acceso desde home.html (para su creación o borrado) y desde plan.html (para añadir el plato al plan de dieta).
- La implementación de una página de perfil de usuario, donde introducir datos de peso, altura, constitución y nivel de ejercicio, para dar consejos nutricionales según el perfil. Accesible desde home.html.

Con estas tres mejoras, la aplicación sería una versión alfa mucho más madura, preparada para adquirir todas las funcionalidades específicas según la línea y el público objetivo.

A largo plazo: Se consideran dos líneas complementarias:

Una línea para el público en general, un público concienciado y decidido a aprender a comer mas saludable. En esta línea, la aplicación debe facilitar la lectura de los datos nutricionales agrupándolos para reducir la confusión, y mostrándolos de manera gráfica, acompañados de consejos nutricionales y de valoraciones. Incluso mostrar una calificación del plan de dieta realizado según lo saludable que sea. Es importante que la aplicación se aleje de transmitir obsesión por el control de las calorías, dando mayor peso a la salud y al consumo de alimentos saciantes, con suficiente fibra y con los oligoelementos necesarios.

Otra línea para el profesional de la nutrición, con mayor cantidad de filtros en la búsqueda de alimentos: Con bajo índice glucémico, con alto contenido en calcio, etc ... Ambas aplicaciones deben compartir un mismo formato de plan de dieta, platos o perfiles para poder así complementarse, y facilitar la labor del profesional, que podrá ofrecer una dieta personalizada lista para correr en la aplicación de su cliente.

Bibliografía

Adda Bjarnadottir, MS (May. 2017) "The 5 Best Calorie Counter Websites and Apps", top stories. <https://www.healthline.com/nutrition/5-best-calorie-counters>

Alligator.io. "JSON.parse() and JSON.stringify()". <https://alligator.io/js/json-parse-stringify/> Visitada el 13/05/2018.

Asim. Blog codeCraft. "Promises". <https://codecraft.tv/courses/angular/es6-typescript/promises/> Visitada el 23/05/2018.

Atkien M. (Oct. 2013). "Patient apps for improved healthcare". IMS Institute for healthcare informatics. http://moodle.univ-lille2.fr/pluginfile.php/128248/mod_resource/content/0/patients%20apps%20for%20improved%20healthcare%20IMS.pdf

Beatriz Portinari, (Dic. 2017) "Contar calorías para perder peso no funciona" - 12:19 CET El País. https://elpais.com/elpais/2017/12/07/buenavida/1512661872_035689.html

Choly, (mar. 2016) "TypeScript: Working with JSON" <http://choly.ca/post/typescript-json/>

Cynthia Graber y cols. Blog Gizmodo. (Ene. 2016) "Por qué contar calorías no funciona en realidad y cómo podemos solucionarlo" <https://es.gizmodo.com/por-que-contar-calorias-no-funciona-realmente-1755653716>

Damir's Corner. (Marzo 31, 2017) Configuring Ionic 2 Web Applications. <http://www.damirscorner.com/blog/posts/20170331-ConfiguringIonic2WebApplications.html>

Didin J, DJAM. (Agosto 2017) "Ionic 3 Consuming REST API using New Angular 4.3 HttpClient" <https://www.djamware.com/post/59924f9080aca768e4d2b12e/ionic-3-consuming-rest-api-using-new-angular-43-httpclient>

Didin J. (Agosto 08, 2017) Creating Beautiful Charts Easily using Ionic 3 and Angular 4. Blog Djam. <https://www.djamware.com/post/598953f880aca768e4d2b12b/creating-beautiful-charts-easily-using-ionic-3-and-angular-4>

Documentación del módulo jsonapi de Drupal. "json API, colectons and sorting". <https://www.drupal.org/docs/8/modules/json-api/collections-and-sorting> Visitada el 11/05/2018.

Documentación oficial de ionic (Componentes). <https://ionicframework.com/docs/components/> Visitada el 25/04/2018.

Enrique Oriol (Febrero 2017) “Personalizar Ionic 2” Visitada el 07/05/2018. <http://blog.enriqueoriol.com/2017/02/personalizar-ionic-2.html>

Fabrizio Bert. (Ene. 2014) “*Smartphones and Health Promotion: A Review of the Evidence*” Journal of Medical Systems, 38:9995 <https://link.springer.com/article/10.1007/s10916-013-9995-7>

Félix Lobo. (Sep./Oct. 2007) “*Políticas públicas para la promoción de la alimentación saludable y la prevención de la obesidad*”. Rev. Esp. Salud Publica vol.81 no.5 Madrid. http://scielo.isciii.es/scielo.php?pid=S1135-57272007000500001&script=sci_arttext&lng=en

Foro Forum. (Julio 2016) “How to get the LocalStorage value out of promise scope”.
<https://forum.ionicframework.com/t/how-to-get-the-localstorage-value-out-of-promise-scope/57296/2>
<https://pointdeveloper.com/use-ionic-storage-save-data-ionic-app/>

Ionic Academia. “How to Store Data inside Ionic Apps”. <https://ionicacademy.com/store-data-inside-ionic/>
Visitada el 19/05/2018.

IonicFramework.org “Get started with Ionic Framework” <https://ionicframework.com/getting-started/>
Visitada el 08/05/2018.

Ismael S. M. M. y col. (jul. 2014) “*Aplicaciones móviles en nutrición, dietética y hábitos saludables; análisis y consecuencia de una tendencia a la alza*”. Nutr. Hosp. vol.30 no.1
http://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S0212-16112014000800002

Javier García Escobedo “Ionic: Navegación entre páginas” <https://javiergarciaescobedo.es/ionic/479-navegacion-entre-paginas-en-ionic> Visitada el 09/05/2018.

Joe Fender, Blog Drupalize.(Ene. 2014) “*An Introduction to RESTful Web Services in Drupal 8*”.
<https://drupalize.me/blog/201401/introduction-restful-web-services-drupal-8>

José Ángel. (Jul. 2017) “*Cinco ‘apps’ para controlar tus alergias y los alimentos que no toleras*” 20:52 CEST El País. https://elpais.com/tecnologia/2017/07/05/actualidad/1499250588_505661.html

Josh Morony (Enero 2018) “High Performance List Filtering in Ionic 2 & 3” <https://www.joshmorony.com/high-performance-list-filtering-in-ionic-2/>

Josh Morony (Enero 25, 2018) Preparing a Progressive Web Application for Production.
<https://www.joshmorony.com/preparing-a-progressive-web-application-for-production>

Josh Morony, (enero 25, 2018) Adding Responsive Charts & Graphs to Ionic 2 & 3 Applications. Blog personal. <https://www.joshmorony.com/adding-responsive-charts-graphs-to-ionic-2-applications/>

JsonApi organization. “[Json:API] A specification for building APIs in JSON” <http://jsonapi.org/>
Visitada 30/04/2018.

Julia Kolodziejczyk y cols. (Jun 2014) “Effectiveness of a Web- and Mobile Phone-Based Intervention to Promote Physical Activity and Healthy Eating in Middle-Aged Males: Randomized Controlled Trial of the ManUp Study”. J Med Internet Res; 16(6): e136. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4090375/>

Julie Weed y col. (Sep. 2012) “Eating Right Between Meetings”. Page B4 of the New York edition. <http://www.nytimes.com/2012/09/04/business/apps-for-eating-right-on-the-road.html>

Laura Dennison, y cols. (Abr. 2013) “Opportunities and Challenges for Smartphone Applications in Supporting Health Behavior Change: Qualitative Study” J Med Internet Res; 15(4): e86. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3636318/>

Laura Martínez, (Ene. 2018) “Las 10 mejores ‘apps’ para cuidar el medio ambiente” Revista Online Compromiso Empresarial. <https://www.compromisoempresarial.com/rsc/2018/01/las-10-mejores-apps-para-cuidar-el-medio-ambiente/>

Mark Pilgrim. “The Past, Present & Future of Local Storage for Web Applications. Libro/tutorial online”. <http://diveintohtml5.info/storage.html> Visitada el 20/05/2018.

Matt Glaman (octubre 2015) “Using JSON API to query your Search API indexes” <https://glamanate.com/blog/using-json-api-query-your-search-api-indexes>

Müller AM y col. (Oct. 2016) “Mobile Phone and Web 2.0 Technologies for Weight Management: A Systematic Scoping Review” Int J Behav Nutr Phys Act. 10;13(1):109. Review. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4704945/>

Nicolas Molina (may, 2017) “Ionic generator y buenas prácticas” Blog de ng.classroom <https://blog.ng-classroom.com/blog/tips/ionic-generator/>

Pablo Espeso, (Oct 2016) “15 apps para promover una buena alimentación” Revista online: Educación 3.0. <https://www.educaciontrespuntocero.com/recursos/apps-alimentacion-sana/38020.html>

Prantik Vaghela. (enero 7, 2018) Use Ionic Storage To Save Data In Ionic App. Pontdeveloper.
Red BEDCA del Ministerio de Ciencia e Innovación. “Spanish Food Composition Database” <http://www.bedca.net/bdpub/index.php> Visitada el 25/04/2018.

Schumer H y col. (Ene. 2018) “*Evaluating the Dietary and Nutritional Apps in the Google Play Store.*” Healthc Inform Res; 24(1):38-45. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5820085/>

Semper HM y col. (Sep. 2016)“*A systematic review of the effectiveness of smartphone applications that encourage dietary self-regulatory strategies for weight loss in overweight and obese adults.*” Obes Rev;17(9):895-906. <https://www.ncbi.nlm.nih.gov/pubmed/27192162>

StackOverflow. How to store and retrieve data using local storage in ionic 2. Foro StackOverflow. <https://stackoverflow.com/questions/43988503/how-to-store-and-retrieve-data-using-local-storage-in-ionic-2>

Steven S. Coughlin y cols. (Ene. 2015) “*Smartphone Applications for Promoting Healthy Diet and Nutrition: A Literature Review*” Jacobs J Food Nutr; 2(3): 021. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4725321/>

TutorialesEnPDF. “Construye Apps móviles multiplataforma con ionic 2 desde cero” <http://tutorialesenpdf.com/ionic-framework/> Visitada el 02/05/2018.

TutorialsPlane.com (feb. 2016). “Ionic Popup” <https://www.tutorialspine.com/ionic-popup/>

w3schools.com. “JSON Objects”. https://www.w3schools.com/js/js_json_objects.asp Visitada el 18/05/2018.

Anexos

Anexo A: Entregables del proyecto

Carpeta proyecto: Con los ficheros de desarrollo de la app (del proyecto en Ionic)

informe.odt: Informe de la etapa de cierre.

memoria.odt: Este documento.

Carpeta presentacionAcademica: contiene el video Equiplan.mp4.

Carpeta presentacionPublica: contiene la presentación.

Anexo B: Perfil del autor

Doctorado en biología por la universidad Miguel Hernández, trabajé durante un tiempo como investigador científico en laboratorios relacionados con el ámbito de la nutrición. En la actualidad ejerzo mi labor como educador en la E.S.O. y el bachillerato.