



**Escuela de Doctorado
y Estudios de Posgrado**
Universidad de La Laguna

Máster Universitario en Ingeniería Informática

Trabajo Fin de Máster

Lenguaje de dominio específico para recomendaciones dietéticas

Domain-specific language for dietary recommendations

Santiago Villar Vázquez

La Laguna, 5 de julio de 2024

D. **Coromoto León Hernández**, profesora Catedrática de Universidad del área de Lenguajes y Sistemas Informáticos, adscrita al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutora

I N F O R M A

Que la presente memoria titulada:

“Lenguajes de dominio específico para el sistema sanitario”

ha sido realizada bajo su dirección por D. **Santiago Villar Vázquez**.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 5 de julio de 2024

Agradecimientos

A mi tutora Coromoto León Hernández por toda la ayuda en este proyecto. A mi familia y a mis amigos por apoyarme durante todo el máster.

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento 4.0 Internacional.

Resumen

En las últimas décadas, la sociedad se ha dado cuenta de que una dieta saludable es fundamental para la salud física y mental de una persona. Una alimentación adecuada no solo puede prevenir enfermedades, sino que también puede mejorar el bienestar general y la calidad de vida. Debido a esto, surge la necesidad de crear herramientas que puedan asistir a los individuos o grupos en la planificación y gestión de dietas saludables. Por ello, el principal objetivo de este trabajo ha sido desarrollar un lenguaje de dominio específico (DSL) centrado en la nutrición. Este DSL está diseñado para proporcionar recomendaciones de menús tanto a personas individuales como a grupos/organizaciones.

El DSL nutricional ha sido implementado en Ruby, donde se han desarrollado varias funciones con la capacidad de crear menús nutricionales basados en las restricciones de salud del usuario y realizar recomendaciones dietéticas. La información nutricional se recopila de distintas bases de datos externas, como Nutritionix y USDA FoodData Central, utilizando APIs para asegurar tener datos precisos y actualizados.

Palabras clave: Dieta saludable, Bienestar general, Calidad de vida, Lenguaje de dominio específico (DSL), Nutrición

Abstract

In recent decades, society has realized that a healthy diet is crucial for both the physical and mental health of an individual. Proper nutrition can not only prevent diseases but also enhance overall well-being and quality of life. Because of this, there is a growing need to create tools that can assist individuals or groups in planning and managing healthy diets. Therefore, the main objective of this work has been to develop a domain-specific language (DSL) focused on nutrition. This DSL is designed to provide menu recommendations for both individuals and groups/organizations.

The nutritional DSL has been implemented in Ruby, where several functions have been developed with the ability to create nutritional menus based on the user's health restrictions and provide dietary recommendations. Nutritional information is collected from various external databases, such as Nutritionix and USDA FoodData Central, using APIs to ensure accurate and up-to-date data.

Keywords: Healthy diet, General well-being, Quality of life, Domain-specific language (DSL), Nutrition

Índice general

1. Introducción	1
1.1. Contexto	1
1.2. Objetivos	3
1.3. Antecedentes y estado actual del arte	3
2. Diseño del sistema	23
2.1. Arquitectura del sistema	23
2.2. Definición del DSL nutricional	24
2.2.1. Estructura del DSL	25
2.2.2. Palabras reservadas/Tokens	25
2.2.3. Gramática del lenguaje	26
2.2.4. Semántica del lenguaje	29
2.3. Diseño del modelo de datos	30
2.4. Metodología	32
2.5. Tecnologías empleadas	32
2.5.1. Lenguaje de programación Ruby	32
2.5.2. Base de datos con SQLite3	33
2.5.3. Bases de datos externas	33
2.6. Validación y pruebas	34
3. Implementación	35
3.1. Estructura del código	35
3.2. Patrón de diseño	36
3.3. Clase DSL	36
3.4. Restricciones	39
3.5. Usuarios	39
3.6. Bases de datos	40
3.7. Pruebas unitarias	42
4. Conclusiones y líneas futuras	45
5. Summary and Conclusions	47
A. Repositorio del proyecto	49

Índice de Figuras

1.1. Tabla formulación problema dieta	2
1.2. Diagrama de flujo PRISMA	5
2.1. Arquitectura del sistema	24
2.2. Diagrama de clases	32
3.1. Clase Menú	37
3.2. Clase Plato y Alimento	37
3.3. Generación de un informe de un plato	38
3.4. Restricciones de un menú	39
3.5. Etiqueta sales	39
3.6. Clase usuario	40
3.7. base de datos SQLite3	41
3.8. Accediendo a base de datos Nutritionix	42
3.9. Pruebas menú personalizado	43
3.10 Prueba generando un menú con restricciones de usuario	43
3.11 Prueba generando la respuesta al usuario	44

Índice de Tablas

1.1. Búsqueda ACM	8
1.2. Búsqueda IEEE Xplore	9
1.3. Búsqueda Scopus	11
1.4. Búsqueda Web Of Science	14
1.5. Búsqueda Google Scholar	20
2.1. Esquema del modelo de datos de los alimentos en la base de datos . .	31
2.2. Modelo de datos de menú	31
2.3. Modelo de datos del plato	31
2.4. Modelo de datos del alimento	32
3.1. Número estimado de calorías necesarias basado en la edad, el sexo y el nivel de actividad	38

Capítulo 1

Introducción

Uno de los mayores problemas que puede afectar a la salud de las personas es seguir una mala alimentación debido al aumento del consumo de productos procesados. Esto también se ve agravado por el aumento del estilo de vida sedentario, provocando un incremento de casos de sobrepeso, obesidad y enfermedades relacionadas con la nutrición.

En base a esto, surge la necesidad de desarrollar una herramienta que sea capaz de ayudar a los individuos o las organizaciones a planificar y gestionar una dieta saludable. Por ello, el principal objetivo de este trabajo ha sido desarrollar un lenguaje de dominio específico (DSL) centrado en la nutrición, capaz de proporcionar recomendaciones dietéticas basadas en las necesidades y restricciones de los usuarios, incluyendo alergias e intolerancias alimentarias.

1.1. Contexto

La historia de la alimentación humana [2] ha evolucionado significativamente a lo largo de los siglos. Desde sus primeros pasos consumiendo solo lo que ofrecía la naturaleza (fruta, carne, pescado), luego el descubrimiento de la agricultura y ganadería permitiendo una dieta más estable y variada (cereales, productos lácteos), también se ha visto influencia por el carácter social donde las clases altas podían acceder a gran variedad de productos en cambio las bajas solo podían acceder a dietas más simples.

Hasta la llegada de la revolución industrial donde el acceso a alimentos se expandió debido a la creación de alimentos procesados, el último gran cambio llegó con la globalización haciendo llegar la gran abundancia de opciones alimentarias que tenemos actualmente pero esta gran variedad de opciones a llevado a que se creen nuevos problemas como la obesidad o enfermedades crónicas debido a dietas desequilibradas o consumo excesivo de alimentos procesados.

Estos problemas actuales para tener una dieta equilibrada se puede representar de forma matemática ya que podemos buscar minimizar o maximizar el consumo de nutrientes aplicando restricciones, también se podría tener en cuenta el coste del alimento aunque en este proyecto no se tiene en cuenta. Para formular este problema se deben de tener en cuenta las siguientes variables:

Variables

- m : número de alimentos
- n : número de nutrientes
- a_{ij} : cantidad de nutrientes 'j' que tiene el alimento 'i'

—	Nutriente 1	Nutriente 2	...	Nutriente $n - 1$	Nutriente n	Costo del alimento
Alimento 1	$a_{1,1}$	$a_{1,2}$...	$a_{1,n-1}$	$a_{1,n}$	c_1
Alimento 2	$a_{2,1}$	$a_{2,2}$...	$a_{2,n-1}$	$a_{2,n}$	c_2
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Alimento $m-1$	$a_{m-1,1}$	$a_{m-1,2}$...	$a_{m-1,n-1}$	$a_{m-1,n}$	c_{m-1}
Alimento m	$a_{m,1}$	$a_{m,2}$...	$a_{m,n-1}$	$a_{m,n}$	c_m
Nutrientes requeridos	b_1	b_2	...	b_{n-1}	b_n	—

Figura 1.1: Tabla formulación problema dieta

Cada persona tiene el mismo requerimiento mínimo para cada nutriente b_j

Sea x_i cantidad del alimento 'i' que se le asigna a cada individuo, la forma de representar que cada individuo tiene que recibir los nutrientes requeridos por los alimentos, se representa de la siguiente forma:

$$a_{1,j}x_1 + a_{2,j}x_2 + \dots + a_{m,j}x_m \geq b_j \text{ (Calorías)}$$

$$a_{1,j}x_1 + a_{2,j}x_2 + \dots + a_{m,j}x_m \geq b_2 \text{ (Proteínas)}$$

$$a_{1,j}x_1 + a_{2,j}x_2 + \dots + a_{m,j}x_m \geq b_j \text{ (...)}$$

Esto asegura que cada individuo reciba al menos los requerimientos mínimos de cada nutriente.

Como se puede comprobar planificar una dieta equilibrada y saludable puede ser un desafío, debido a la gran variedad de opciones disponibles, poca información al respecto, falta de tiempo de planificación o por restricciones de salud que puede tener una persona debido a alguna enfermedad.

Para poder dar soluciones a este problema se busca crear un DSL nutricional, que sea capaz de ofrecer a una persona o grupo un menú saludable adaptado a sus necesidades, los objetivos de este proyecto son detallados en el siguiente apartado.

1.2. Objetivos

El objetivo de este trabajo es diseñar y desarrollar un lenguaje de dominio específico (DSL), este es un tipo de lenguaje de programación diseñado para una tarea específica o un conjunto de tareas dentro de un dominio en particular, optimizado para ofrecer soluciones eficientes. En el caso de este proyecto el DSL se encarga de elaborar recomendaciones dietéticas y también permitir seleccionar una recomendación en base a una serie de restricciones.

A nivel funcional, se apunta a conseguir los siguientes objetivos concretos:

- La aplicación debe de ser capaz de contar con los el concepto de plato, menú, día para poder tratar los alimentos como un conjunto.
- Se debe acceder a una base de datos externa para recabar información nutricional de una gran variedad de alimentos
- La aplicación debería contar con una gramática que permita especificar restricciones y necesidades dietéticas.
- La aplicación debería ser fácilmente mantenible y escalable, permitiendo un desarrollo ágil y la adición de nuevas características o actualizaciones de forma sencilla.

Al contrario que una aplicación empresarial, se trata de un proyecto con un grado de incertidumbre alto. El problema a tratar, las fuentes de datos y la lógica de negocio son puntos que tienen el mismo denominador común: un estado de inmadurez considerable.

Por ello, se trata de un proyecto muy experimental y resulta fundamental tener en cuenta la escalabilidad, la limpieza y una buena arquitectura, puntos que se tratarán más adelante.

En el siguiente apartado se va a realizar una revisión bibliográfica que proporcionará los antecedentes necesarios para entender el estado actual de los DSL en el campo de la nutrición.

1.3. Antecedentes y estado actual del arte

Para obtener información sobre el estado del arte, se consultaron los principales buscadores de recursos académicos utilizando el método PRISMA [3] (Elementos de informes preferidos para revisiones sistemáticas y meta-análisis). Este método es una guía utilizada para identificar, evaluar e interpretar el trabajo de académicos y profesionales en un campo específico, buscando identificar alguna falta de conocimiento o una necesidad de investigar en un área determinada.

El diagrama de flujo del método prisma contiene 4 fases:

- Identificación
 - Número de registros identificaos a través de BBDD y otras fuentes.

- Número de registros duplicados eliminados.
- Filtrado
 - Número de registros examinados.
 - Número de registros excluidos y su razón.
- Elegibilidad
 - Número de textos evaluados para su selección.
 - Número de textos completos excluido y su razón.
- Inclusión
 - Número de estudios incluidos en la revisión cualitativa.
 - Número de estudios incluidos en la revisión cuantitativa (meta análisis).

El diagrama de flujo es el siguiente:

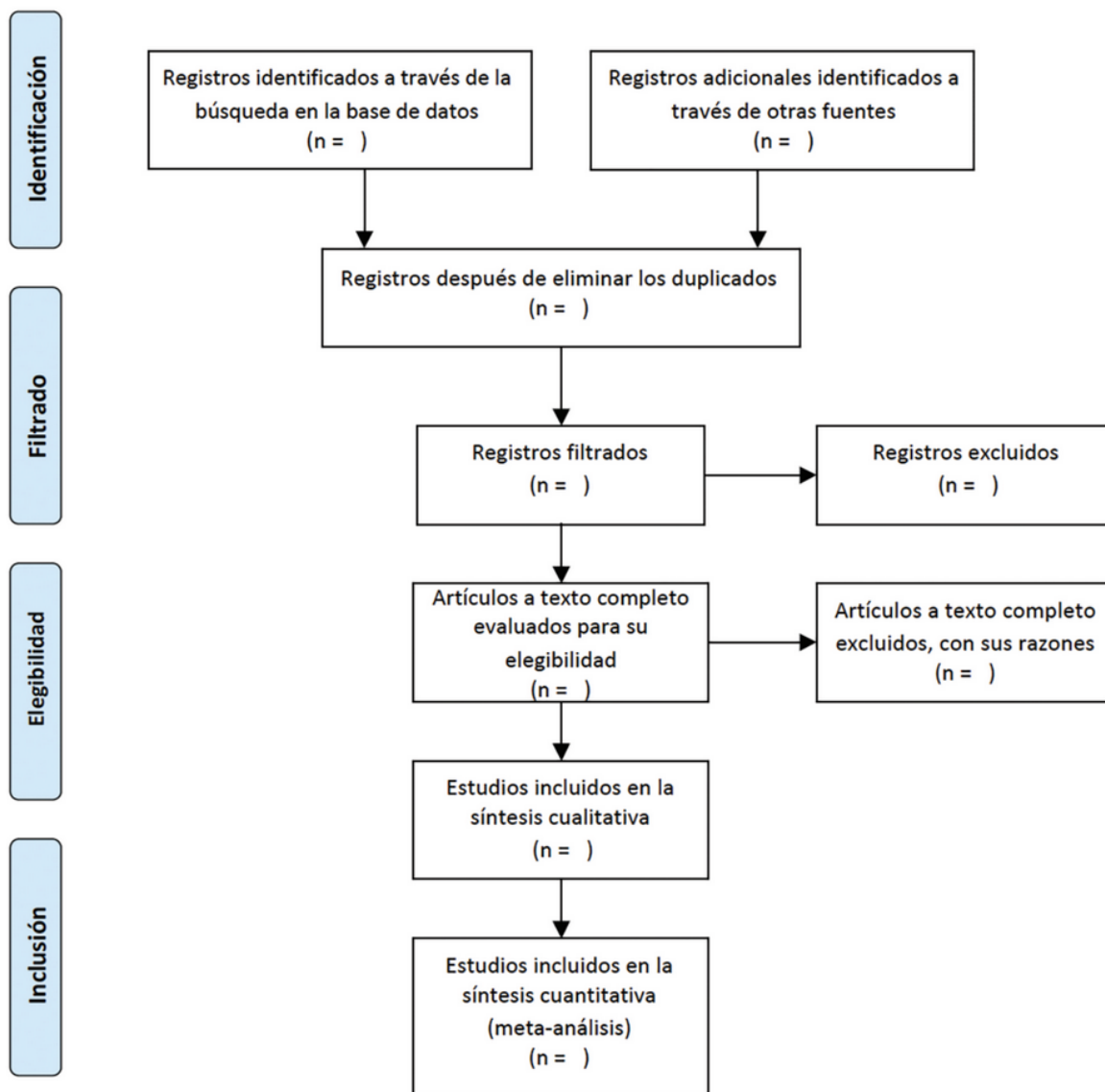


Figura 1.2: Diagrama de flujo PRISMA

Poniendo en práctica la metodología PRISMA[3] se realizó la búsqueda de estudios relacionados con el proyecto del DSL nutricional en diferentes BBDD estas son ACM, IEEE Xplore, Scopus, Web Of Science y Google Scholar, se apuntó el número de registros examinados y el número incluido, así como la razón para su inclusión.

Estas fueron las consultas realizadas y sus resultados:

Patrón: "Domain Specific Languages" AND "Nutritional"

Base de datos: ACM

Número de registros: 12

Se incluyen: 2

Referencia	Abstract	Se incluye
<p>Daniel W. Barowy, Emery D. Berger, Daniel G. Goldstein, and Siddharth Suri. 2017. VoxPL: Programming with the Wisdom of the Crowd. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17). Association for Computing Machinery, New York, NY, USA, 2347-2358.</p>	<p>Se usa VoxPL para crear un DSL capaz de medir las calorías de un plato en base a una foto y luego también se busca detectar caras en una foto.</p>	<p>No</p>
<p>Luca Mainetti, Luigi Patrono, and Roberto Vergallo. 2011. Combining EPCglobal and HL7 to deploy innovative e-health services for patients affected by multiple intolerances. In Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL '11). Association for Computing Machinery, New York, NY, USA, Article 86, 1-6.</p>	<p>Se crea una especie de DSL para ayudar a los pacientes de hospitales a elegir su dieta o medicamentos, ya que algunos medicamentos y comidas pueden reaccionar de forma negativa si se consumen de forma conjunta.</p>	<p>Sí</p>
<p>Mira Vrbaski, Gunter Mussbacher, Dorina Petriu, and Daniel Amyot. 2012. Goal models as run-time entities in context-aware systems. In Proceedings of the 7th Workshop on Models@run.time (MRT '12). Association for Computing Machinery, New York, NY, USA, 3-8.</p>	<p>Se analizan diferentes modelos para crear un DSL, donde también realiza un ejemplo de crear un DSL nutricional para un hospital.</p>	<p>Sí</p>

Referencia	Abstract	Se incluye
2023. Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law. Association for Computing Machinery, New York, NY, USA.	Trata sobre una charla de diferentes temas donde uno de ellos es sobre un DSL sobre hospitales polacos.	No
2022. Proceedings of the 12th International Conference on the Internet of Things. Association for Computing Machinery, New York, NY, USA.	Trata sobre vehículos y una granja.	No
2023. Proceedings of the 22nd Annual ACM Interaction Design and Children Conference. Association for Computing Machinery, New York, NY, USA.	Habla sobre el uso de la VR para que los padres ayuden a sus hijos que sufren maltrato en el colegio por parte de sus compañeros.	No
2022. Companion Proceedings of the Web Conference 2022. Association for Computing Machinery, New York, NY, USA.	Tiene diferentes ejemplos uno de ellos relacionados con la nutrición de una tienda de comida.	No
2023. Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing. Association for Computing Machinery, New York, NY, USA.	Se encuentra un trabajo de machine learning sobre readmisión de niños en los hospitales donde entre todo lo que se controla está la nutrición.	No
2021. Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA.	Muchos trabajos diferentes.	No
2022. Proceedings of the 17th International Conference on Availability, Reliability and Security. Association for Computing Machinery, New York, NY, USA.	Muchos trabajos pero ninguno nutricional lo que más se acerca es una web de venta entre ello comida.	No

Referencia	Abstract	Se incluye
2015. Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. Association for Computing Machinery, New York, NY, USA.	Dispone de numerosos trabajos pero ninguno nutricional.	No
Cláudio Martins, João Rosa, Laerte Franco, Jorge Barbosa, and Eduardo Bezeira. 2012. Towards a model to explore business opportunities in trail-aware environments. In Proceedings of the 18th Brazilian symposium on Multimedia and the web (WebMedia '12). Association for Computing Machinery, New York, NY, USA, 143–150.	Se hace un DSL sobre senderos.	No

Tabla 1.1: Búsqueda ACM

Patrón: "Domain Specific Languages" AND "Nutritional"

Base de datos: IEEE Xplore

Número de registros: 1

Se incluyen: 0

Referencia	Abstract	Incluye
<p>Gowda, S., Hu, Y., & Korpusik, M. (2023, June). Multi-Modal Food Classification in a Diet Tracking System with Spoken and Visual Inputs. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1-5). IEEE.</p>	<p>In this paper, we present multi-modal approaches to diet tracking. As health and well-being become increasingly important, mobile applications for diet tracking attract much interest. However, these applications often require users to log their meals based on relatively unreliable memory recall, thereby underestimating nutritional intake and, thus, undermining the efforts of nutrition tracking. To accurately record dietary intake, there is an increasing need for image computational methods. We investigated multi-modal transfer learning approaches on a novel, food-specific image-text dataset, specifically a Vision-and-Language Transformer that achieves a held-out test set Micro-F1 score of 77.70% and Macro-F1 score of 51.43% for 696 food categories. We aim to give other researchers new insight into the process of developing domain-specific, multi-modal deep learning models with small datasets.</p>	<p>No, porque trata de análisis de señales</p>

Tabla 1.2: Búsqueda IEEE Xplore

Patrón: domain-specific AND languages AND nutritional

Base de datos: Scopus

Número de registros: 7

Se incluyen: 0

Referencia	Abstract	Incluye
Gowda, S., Hu, Y., Korpusik, M. (2023, June). Multi-Modal Food Classification in a Diet Tracking System with Spoken and Visual Inputs. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) - Proceedings (pp. 1-5). IEEE.	In this paper, we present multi-modal approaches to diet tracking. As health and well-being become increasingly important, mobile applications for diet tracking attract much interest. However, these applications often require users to log their meals based on relatively unreliable memory recall, thereby underestimating nutritional intake and, thus, undermining the efforts of nutrition tracking. To accurately record dietary intake, there is an increasing need for image computational methods. We investigated multi-modal transfer learning approaches on a novel, food-specific image-text dataset, specifically a Vision-and-Language Transformer that achieves a held-out test set Micro-F1 score of 77.70% and Macro-F1 score of 51.43% for 696 food categories. We aim to give other researchers new insight into the process of developing domain-specific, multi-modal deep learning models with small datasets.	No, porque trata de análisis de señales
McGrattan, A., van Aller, C., Narytnyk, A., ..., Robinson, L., Siervo, M. (2022). Nutritional interventions for the prevention of cognitive impairment and dementia in developing economies in East-Asia: a systematic review and meta-analysis. <i>Critical Reviews in Food Science and Nutrition</i> , 62(7), 1838–1855.	Trata sobre como afecta la nutrición a las personas con demencia en economías en desarrollo.	No realiza un dsl
Tou, N.X., Wee, S.-L., Pang, B.W.J., ..., Chen, K.K., Ng, T.P. (2021, August 8). Associations of fat mass and muscle function but not lean mass with cognitive impairment: The Yishun Study. <i>PLoS ONE</i> , 16(8 August 2021), e0256702	Trata sobre la relación entre la obesidad sarcopénica y el deterioro cognitivo.	No tiene un dsl

Referencia	Abstract	Incluye
Lee, M., Lim, J.-S., Kim, Y., ..., Lee, B.-C., Yu, K.-H. (2021). Association between geriatric nutritional risk index and post-stroke cognitive outcomes. <i>Nutrients</i> , 13(6), 1776.	Evalúa la relación entre un riesgo nutricional geriátrico y el deterioro cognitivo tras un accidente.	No tiene un dsl
Clairret, N., Lafourcade, M. (2017). Towards the automatic detection of nutritional incompatibilities based on recipe titles. <i>Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)</i> , 10410 LNCS, 346–366.	Se detectan incompatibilidades nutricionales en recetas de cocina por sus títulos que puedan acarrear problemas médicos donde se construye una red de conocimiento para ayudar a los pacientes.	No tiene un dsl
Sojic, A., Terkaj, W., Contini, G., Sacco, M. (2016). Modularising ontology and designing inference patterns to personalise health condition assessment: The case of obesity. <i>Journal of Biomedical Semantics</i> , 7(1), 12.	Destaca el uso de tecnologías inteligentes para prevenir la obesidad, usando diferentes datos.	No tiene un dsl
Laws, K.R., Leeson, V.C., Gale, T.M. (2002). A domain-specific deficit for foodstuffs in patients with Alzheimer's disease. <i>Journal of the International Neuropsychological Society</i> , 8(7), 956–957.	Trata sobre el alzheimer.	No tiene un dsl

Tabla 1.3: Búsqueda Scopus

Patrón: domain-specific AND languages AND nutritional
Base de datos: Web Of Science
Número de registros: 9
Se incluyen: 0

Referencia	Abstract	Incluye
Sojic, A., Terkaj, W., Contini, G., Sacco, M. (2016). Modularising ontology and designing inference patterns to personalise health condition assessment: The case of obesity. <i>Journal of Biomedical Semantics</i> , 7(1), 12.	Destaca el uso de tecnologías inteligentes para prevenir la obesidad, usando diferentes datos	No, realiza un DSL
Clairat, N.,	Lafourcade, M. (2017). Towards the Automatic Detection of Nutritional Incompatibilities Based on Recipe Titles. En 1st IFIP TC 5, WG 8.4, 8.9, 12.9 International Cross-Domain Conference on Machine Learning and Knowledge Extraction (CD-MAKE).	Se detectan incompatibilidades nutricionales en recetas de cocina por sus títulos que puedan acarrear problemas médicos donde se construye una red de conocimiento para ayudar a los pacientes
No, realiza un DSL		

Referencia	Abstract	Incluye
M. A. Beydoun, S. Hossain, ..., A. B. Zonderman, "Vitamin D Status and Intakes and Their Association With Cognitive Trajectory in a Longitudinal Study of Urban Adults, Apr. 2018.	Se investiga la influencia de un suero y de la vitamina D en los resultados cognitivos	No, realiza un DSL
C. Balion, L. E. Griffith, ..., P. Raina, "Vitamin D, cognition, and dementia: A systematic review and meta-analysis," Sep. 2012.	Se examina la asociación entre la función cognitiva y la demencia con la concentración de vitamina D en adultos	No, realiza un DSL
A. McGrattan, C. van Aller, ..., M. Siervo, "Nutritional interventions for the prevention of cognitive impairment and dementia in developing economies in East-Asia: A systematic review and meta-analysis, Apr. 12, 2022.	Trata sobre cómo afecta la nutrición a las personas con demencia en economías en desarrollo	No, realiza un DSL
N. Petruski-Ivleva, A. Kucharska-Newton, ..., G. Heiss, "Association of habitual milk intake with cognitive decline from mid-life to late-life: The Atherosclerosis Risk in Communities (ARIC) Neurocognitive Study (NCS)," presented at the Scientific Sessions of the American Heart Association on Epidemiology and Prevention/Lifestyle and Cardiometabolic Health, Mar. 1, 2016.	https://www-webofscience-com.accedys2.bbtk.ull.es/wos/alldb/full-record/WOS:000381415500295	Vacío

Tabla 1.4: Búsqueda Web Of Science

Patrón: "Domain Specific Languages" AND "Nutritional"

Base de datos: Google Scholar

Número de registros: 71

Se incluyen: 3

Referencia	Abstract	Incluye
Weller, B. (2010). Domain-Specific Languages 101: Lola's Lunch Counter. In Beginning SQL Server Modeling: Model-Driven Application Development in SQL Server 2008 (pp. 33-79). Berkeley, CA: Apress.	Un libro que trata sobre crear dsl usando intellipad	No
Barski, C. (2010). Land of Lisp: learn to program in Lisp, one game at a time!. No starch press.	Libro de programación en Lisp	No
Lizaso Eguileta, A. (2021). Solution for the management of several clinical practice guidelines in a domain-independent decision support system.	Busca crear un programa capaz de ayudar a las personas mayores con su nutrición utilizando sus factores sociales	Si
Krämer, B. 3.2 The Scientific Academy for Service Technology (ServTech).	Busca avanzar en el uso de aplicaciones orientadas a tareas científicas, desarrollo e innovación	No
Liu, S., Chen, Q., Guan, S., Wang, X., & Shi, H. (2013, August). Mining Recipes in microblog. In 2013 International Conference on Asian Language Processing (pp. 29-32). IEEE.	Busca crear un programa capaz de juntar el conocimiento sobre varias recetas	No
Schinle, M., Dietrich, M., Stock, S., Gerdes, M., & Stork, W. (2023). Model-Driven Dementia Prevention and Intervention Platform. CARING IS SHARING-EXPLOITING THE VALUE IN DATA FOR HEALTH AND INNOVATION, 937.	Se crea un dsl para ayudar a las personas con demencia con su nutrición (Pág 937)	Si
Roorda, A. (2021). Corel: A DSL for Cooking Recipes (Doctoral dissertation).	Se crea un dsl capaz de analizar la estructura nutricional de una receta y permite crear una nueva receta	Si

<p>Pérez-Soler, S., Guerra, E., & De Lara, J. (2020, October). Model-driven chatbot development. In International Conference on Conceptual Modeling (pp. 207-222). Cham: Springer International Publishing.</p>	<p>Se crea un DSL para la creación de chatbots personalizados</p>	<p>No</p>
<p>Barowy, D. W., Berger, E. D., Goldstein, D. G., & Suri, S. (2017, May). Voxpl: Programming with the wisdom of the crowd. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (pp. 2347-2358).</p>	<p>Se usa VoxPL para crear un DSL capaz de medir las calorías de un plato en base a una foto y luego también se busca detectar caras en una foto.</p>	<p>No</p>
<p>Rodríguez, J. L. S., Leal, J. P., & Simões, A. (2015). Languages, Applications and Technologies-4th International Symposium, SLATE 2015, Madrid, Spain, June 18-19, 2015, Revised Selected Papers.</p>	<p>Libro sobre lenguajes y tecnologías</p>	<p>No</p>
<p>De Backere, F., Moens, H., Steurbaut, K., Colpaert, K., Decruyenaere, J., & De Turck, F. (2012). Towards automated generation and execution of clinical guidelines: Engine design and implementation through the ICU Modified Schofield use case. Computers in biology and medicine, 42(8), 793-805.</p>	<p>Automatización de servicios clínicos</p>	<p>No</p>

Premarathna, D. (2021, September). Reduce food crop wastage with hyperledger fabric-based food supply chain. In 2021 International Research Conference on Smart Computing and Systems Engineering (SCSE) (Vol. 4, pp. 168-176). IEEE.	Es un estudio que trata de reducir la comida que se tira de sobra en las granjas	No
de Lara, J. Model-driven chatbot development.	Se desarrolla un chatbot utilizando DSL	No
Smyth, S. (2021). Interactive Model-Based Compilation: A Modeller-Driven Development Approach (Doctoral dissertation).	Habla sobre la creciente tendencia de realizar DSL, y recomienda una metodología interactiva	No
Weller, B. (2010). Beginning SQL Server Modeling: Model-Driven Application Development in SQL Server 2008. Apress.	Libro sobre Modelado de SQL Server	No
David, B., & Chalon, R. (2015). Mobile Interaction in Real Augmented Environments: Principles, Platforms, Models, Formalisms, Development Processes, and Applications. In Handbook of Research on Innovations in Systems and Software Engineering (pp. 511-561). IGI Global.	Realidad aumentada para móviles	No
Lee, A. J. (2022). The Smart Ways of Adoption IoT and AI for Sustainable Animal Agriculture.	Explora la adopción de diferentes tecnologías Iot, IIoT, IA	No
Germeier, C. U., & Unger, S. (2019). Modeling crop genetic resources phenotyping information systems. Frontiers in Plant Science, 10, 728.	Modelado de cultivos, cría ecología y evolución	No

Ennas, C. (2023). Effects of anthropogenic and natural disturbance on sedimentary organic matter quantity, nutritional quality and degradation in river, lagoon, and coastal marine ecosystems.	Efectos que molestan en la ecología	No
Miah, S. J., Gammack, J. G., & McKay, J. (2019). A metadesign theory for tailorable decision support. <i>Journal of the Association for Information Systems</i> , 20(5), 4.	Teoría de cómo diseñar un programa a medida para la agricultura	No
Hans-Georg, F. (2009). <i>Visualisation for semantic information systems</i> . Springer Science No	Business Media.	Libro sobre visualización de la semántica en sistemas de información
Konur, S., Gheorghe, M., & Krasnogor, N. (2023). Verifiable biology. <i>Journal of the Royal Society Interface</i> , 20(202), 20230019.	Habla sobre la formación de sistemas biológicos mediante el uso de modelos computacionales	No
Weber, D. (2017). <i>A Constraint-Based Approach to Data Quality in Information Systems</i> (Doctoral dissertation, ETH Zurich).	Aborda el cambio que existe en el volumen, diversidad y acceso a datos durante la última época	No
Oloo, G. J. TRENDS AND INSIGHTS ON TOOLS USED FOR THE DEVELOPMENT OF EXPERT SYSTEMS: A SYSTEMATIC REVIEW OF RESEARCH.	Artículo sobre el desarrollo de sistemas, su implementación e integración con diferentes entornos	No

Rogers, W., Blocker, K., & Dupuy, L. (2020). Current and emerging technologies for supporting successful aging. The Cambridge Handbook of Cognitive Aging: A Life Course Perspective, 717-36.		No
Mining, L. S. C. D. Team LACODAM.	Libro sobre data mining.	No
Fleck, P., Calepso, A. S., Hubenschmid, S., Sedlmair, M., & Schmalstieg, D. (2022). Ragrug: A toolkit for situated analytics. IEEE Transactions on Visualization and Computer Graphics.	Se centra en la realidad aumentada	No
Moad, H. A. N. I. Modeling and Simulating the Resilience of Ecosystems.	Se desarrolla un dsl para ver el efecto de las interacciones humanas en el ecosistema principalmente en los incendios	No
Boers, N. M., Chodos, D., Gburzynski, P., Guirguis, L., Huang, J., Lederer, R., ... & Stroulia, E. (2011). The smart condo project: services for independent living. In E-Health, assistive technologies and applications for assisted living: challenges and solutions (pp. 289-314). IGI Global.	Formas de mejorar la calidad de vida de las personas mayores	No
Liu, R., Gao, M., Wang, L., Wang, X., Xiang, Y., Zhang, A., ... & Chen, S. (2022). Interactive Extended Reality Techniques in Information Visualization. IEEE Transactions on Human-Machine Systems, 52(6), 1338-1351.	Es una encuesta que valora diferentes formas interactivas de acceder a datos como puede ser la VR	No

Lario, R. F. (2023). Method for Structuring Complex Clinical Knowledge and Its Representational Formalisms to Support Knowledge Interoperability in Healthcare (Doctoral dissertation, The University of Utah).	Habla sobre el proceso CK1 que trata sobre documentar, aplicar y aplicar procesos clínicos	No
Valdez, A. C., Adam, L., Assenmacher, D., Burbach, L., Bonart, M., Frischlich, L., & Schaer, P. (2019, July). Computational Methods in Professional Communication. In 2019 IEEE International Professional Communication Conference (ProComm) (pp. 275-285). IEEE.	Métodos computacionales sobre la investigación de comunicaciones profesionales	No
Pierce, M. E. (2022). Integrating Knowledge about Complex Adaptive Systems: Insights from Modelling the Eastern Baltic Cod (Doctoral dissertation, Universität).	Habla sobre el declive del bacalao en el báltico y la complejidad de investigar en el océano	No

Tabla 1.5: Búsqueda Google Scholar

Siguiendo la metodología PRISMA[3] se han analizado 84 referencias bibliográficas. De estas, 12 pertenecen a la base de datos ACM, 1 de IEEE Xplore, 7 de Scopus y 71 de Google Scholar. Finalmente, sólo 5 están relacionados con el tema de investigación, descartando las de IEEE Xplore, Scopus y Web Of Science.

De los 12 artículos de la base de datos ACM se han seleccionado:

- En el artículo "Combining EPCglobal and HL7 to deploy innovative e-health services for patients affected by multiple intolerances"[5]. Se busca ayudar a los pacientes que tienen alguna alergia a poder tener una dieta saludable y poder tomar una medicación adecuada en base a su salud y alergias, para ello se crea un programa que primero pregunta al usuario sobre sus datos para poder recabar su historial médico, luego se recabarán los datos de los diferentes productos clínicos o alimenticios para poder ofrecer una respuesta, todo esto seguirán los estándares EPCglobal y HL7 y está automatizado por si ocurre algún cambio de salud en el paciente o en la fabricación de algún

producto que pueda llegar a afectar de forma negativa ya sea por el uso de un nuevo químico en una granja o que se encuentre una nueva alergia en el paciente.

Este texto es útil ya que en base a unos requisitos médicos y la alimentación disponible se decide crear un programa que es capaz de recomendar diferentes productos de medicinas y de alimentación

- El segundo artículo Goal models as run-time entities in context-aware systems [8]. Se Realiza un programa capaz de adaptarse al usuario en este caso se basará en elegir una dieta apropiada a los pacientes de un hospital basado en sus alergias, también se tiene en cuenta el coste de la comida así como el tiempo de preparación pero no se tiene en cuenta el valor nutricional, el modelo diseñado intentara buscar la solución óptima en base a los 3 entidades afectadas, la cocina del hospital que busca el menor tiempo de preparación y tener a todos los empleados trabajando, la administración del hospital que busca el menor coste económico y el paciente que busca el menor coste pero el mayor valor nutritivo

Se elige este texto debido a que se crea un programa que es capaz en base a diferentes reglas como puede ser el presupuesto, tiempo de trabajo y nutrición. Crear una dieta que pueda ser administrada a un paciente de un hospital.

De los 71 artículos de Google Scholar se han seleccionado:

- El primer artículo Solution for the management of several clinical practice guidelines in a domain-independent decision support system [4]. Se crea un programa que utilice la IA para ayudar a elaborar dietas personalizadas a personas mayores en función a diferentes factores (edad, sexo, enfermedad, alergias, adaptación y seguimiento) donde se crean diferentes guías utilizando XML para poder aumentar la esperanza de vida de estas personas y minimizar el riesgo de contraer alguna enfermedad.

Se selecciona este texto ya que en función a diferentes factores físicos de una persona se crea un programa capaz de recomendar diferentes formas ya sea con dietas o ejercicios físicos que sean capaces de mejorar la esperanza de vida del usuario.

- El segundo artículo Model-Driven Dementia Prevention and Intervention Platform. CARING IS SHARING–EXPLOITING THE VALUE IN DATA FOR HEALTH AND INNOVATION [7]. Se diseña un modelo que ayude a prevenir la demencia entre ellas el alzhéimer, para ello se crea un dsl que ayude al médico a recomendar un tratamiento, este modelo necesita que el médico introduzca un primer diagnóstico y datos del paciente ya sea la edad, peso, falta de ejercicio tras ello el programa empezará a recomendar diferentes dietas/tratamientos a seguir con el paciente durante el curso del tiempo, ayudando a reducir el riesgo de contraer una de estas enfermedades que actualmente no tienen cura. Este trabajo es útil porque se crea un programa que tras un primer diagnóstico por parte de un experto es capaz de analizar y crear diferentes recomendaciones personalizadas en dietas y tratamientos.

- El ultimo artículo Corel: A DSL for Cooking Recipes (Doctoral dissertation) [6]. Busca crear un DSL capaz de analizar y crear recetas de comida, para ello se utilizan 20 recetas como ejemplo donde se analiza la nutrición, ingredientes, cantidades, tiempo de cocción, nombres. Se utiliza el lenguaje de programación Rascal y se necesitan traductores para pasar algunas cantidades de ingredientes a medidas exactas es decir pasar de cucharas a gramos o conversión de medidas más naturales como puede ser la libra a medidas estándar.

Se elige este texto debido a que se crea un DSL donde a partir de ciertas recetas convencionales el programa es capaz de generar recetas totalmente nuevas indicando ingredientes, cantidades y tiempos de cocción.

En total, se han encontrado 2 tipos de DSL entre todos los trabajos. El primero de ellos se basa en indicar recomendaciones de dietas específicas en base a los datos del estado del paciente, mientras que el segundo tipo de DSL es capaz de crear recetas totalmente nuevas a partir de algunos ejemplos de recetas. Por ello, desde el punto de vista informático, se trata de un tema aún sin explotar sobre el cual no existe un número excesivo de estudios. Los estudios existentes se encargan de realizar una recomendación dietética o de seleccionar una recomendación dependiendo de un conjunto de restricciones, pero no se encuentra ningún estudio que aúne ambos dominios. El resto de la memoria se encuentra en el capítulo 2, donde se presentan el diseño y la arquitectura del sistema , además de definirse el DSL y explicarse las tecnologías empleadas. En el capítulo 3 se tratará la implementación del DSL con sus diferentes componentes. Para finalizar, en los capítulo 4 y 5 se reflexionará sobre posibles mejoras en el futuro y las conclusiones del proyecto.

Capítulo 2

Diseño del sistema

En este capítulo se describe el diseño del sistema sin entrar aún en su implementación. Se mostrará una arquitectura del sistema así como del modelo de datos necesario. En definitiva, se realiza el papel de analista definiendo el problema y trasladándolo a un sistema informático. Esta fase de análisis es importante porque sentará las bases del proyecto, permitirá escalarlo o definirá la facilidad o el coste con que se enfrentará a los cambios, entre otras características. Aparte se describen las tecnologías empleadas para la realización de este proyecto y la justificación de la elección realizada. Una buena selección es primordial para garantizar el éxito del desarrollo y la implementación de un lenguaje de dominio específico (DSL) centrado en la nutrición.

2.1. Arquitectura del sistema

Para el desarrollo del proyecto se ha usado la arquitectura Multi-tier Architecture, esta se caracteriza en distribuir las diferentes funcionalidades de la aplicación en capas, donde cada capa realiza una tarea en específico y se comunica con las capas adyacentes, ayudando en mejorar la modularidad, escalabilidad y la mantenibilidad del sistema, dividiendo el código en 3 capas principales.

- Capa de presentación: Son los usuarios que interactúan con la aplicación para obtener los menús nutricionales basados en sus necesidades y restricciones dietéticas.
- Capa de lógica de aplicación: Genera los menús personalizados para los usuarios. Para ello, sigue los siguientes pasos:
 - Obtención del rango calórico del usuario
 - Se calcula el rango calórico adecuado para el usuario en base a su edad, peso, sexo y forma física.
 - Generación del menú
 - Se crea el menú primero comprobando las restricciones alérgicas y luego en base a su rango calórico.
 - Verificación y ajuste de calorías

- Se comprueba que el menú generado cumple con el rango calórico establecido y, si no, se ajusta.
- Salida del menú
 - Se imprimen los platos y comidas con sus calorías.
- Capa de datos: Para poder generar los menús personalizados, es necesario obtener y gestionar los datos de los alimentos y sus valores nutricionales.
 - Datos de alimentos
 - Se utiliza la base de datos de Nutritionix y un archivo Json de FoodData Central para introducir los alimentos necesarios y su información nutricional en la base de datos propias (SQLite3), para poder seleccionar los alimentos que cumplen las restricciones.
 - Clases de la aplicación
 - Menú: Representa un menú de un día específico y contiene un conjunto de platos.
 - Plato: Contiene un conjunto de alimentos.
 - Alimento: Contiene la información nutricional del alimento, como pueden ser las calorías y el peso.

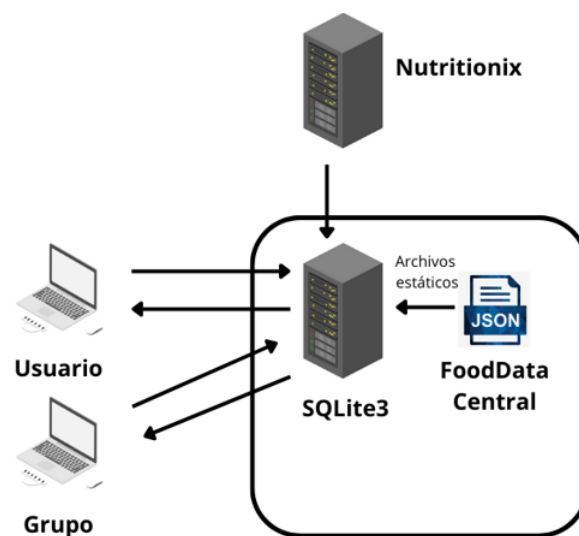


Figura 2.1: Arquitectura del sistema

2.2. Definición del DSL nutricional

A continuación, se presenta la estructura, las palabras reservadas/tokens la gramática y semántica del DSL diseñado para la generación de menús nutricionales basados en las necesidades dietéticas y nutricionales de los usuarios.

2.2.1. Estructura del DSL

La estructura del DSL define los componentes principales que conforman un menú nutricional y sus respectivos detalles.

- Menú: Nombre del menú. Este campo describe de manera breve el menú completo, proporcionando una idea general de su contenido.
- Pautas: Instrucciones a seguir para tomarse los platos del menú (Opcional). Permite incluir recomendaciones de como/cuando tomar los alimentos.
- Datos adicionales (Menú): Datos nutricionales/alérgenos del menú completo. Contiene la información nutricional total del menú y sus posibles alérgenos.
- Plato: Nombre del plato. Descripción de cada plato que se incluye en el menú
- Alimentos: Lista de alimentos. Contiene la lista de todos los alimentos que conforman el menú.
- Datos adicionales (Plato): Datos nutricionales, alérgenos del plato entero. Contiene la información nutricional y alérgenos total del plato.
- Alimentos: Nombre del alimento. Especifica cada ingrediente utilizado.
- Tipo: Tipo de alimento. Clasifica los alimentos según su categoría.
- Datos adicionales (Alimento): Datos nutricionales/alérgenos del alimento. Detalla la información nutricional y los alérgenos del alimento.

2.2.2. Palabras reservadas/Tokens

Las palabras reservadas y tokens del DSL son términos específicos que permiten definir y describir los menús y platos de forma estructurada y consistente. Cada uno de esos tokens tienen su significado y función dentro del DSL.

- Nombre menú: Cómo se llama el menú/recomendación. Define como se llama el menú proporcionando una descripción.
- Pautas: Recomendaciones sobre cómo consumir el menú (Opcional). Incluye instrucciones específicas para la correcta ingesta del menú.
- Nombre plato: Cómo se llama el plato. Especifica el nombre del plato del menú.
- Alimentos: Lista de alimentos utilizados. Indica los ingredientes necesarios para preparar el plato.
- Fructosa: Indica si contiene o no fructosa el alimento
- Lactosa: Indica si contiene o no lactosa el alimento
- Azúcares: Cantidad de azúcares de la receta.

- Sales: Cantidad de sales de la receta.
- Proteínas: Cantidad de proteínas de la receta.
- Carbohidratos: Cantidad de carbohidratos de la receta.
- Calorías: Cantidad de calorías de la receta.
- Grasas: Cantidad de grasas de la receta.
- Grasas saturadas: Cantidad de grasas saturadas de la receta.
- Peso: Cantidad de peso del alimento.

2.2.3. Gramática del lenguaje

En esta sección se define la gramática del lenguaje utilizado para describir menús y platos en el DSL nutricional. La gramática se presenta utilizando la notación Backus-Naur Form (BNF).

Esta notación se lee de la siguiente forma:

- "::=" significa "se define como", la parte derecha define la parte izquierda.
- Los elementos terminales se escriben entre comillas.
- Los elementos no terminales se escriben con Letras mayúsculas.
- Los [] indican una lista de elementos.
- El símbolo | indican que solo se puede elegir uno de los elementos que se separa.

A continuación, se presenta la gramática del lenguaje:

```
Menu ::= Nombre_Menu Pautas Datos_Adicionales_Menu Platos
```

```
Nombre_Menu ::= Nombre_Menu String
```

```
Pautas ::= Pautas String
```

```
Datos_Adicionales_Menu ::= Azucares_Menu Proteinas_Menu Carbohidratos_Menu
    Calorias_Menu Grasas_Menu Sales_Menu Grasas_Saturadas_Menu Lactosa_Menu
    Fructosa_Menu Peso_Menu
```

```
Azucars_Menu ::= sum(Menu[].Azucars)
```

```
Proteinas_Menu ::= sum(Menu[].Proteinas)
```

```
Carbohidratos_Menu ::= sum(Menu[].Carbohidratos)
```

```
Calorias_Menu ::= sum(Menu[].Calorias)
```

```
Grasas_Menu ::= sum(Menu[].Grasas)
```

```
Grasas_Saturadas_Menu ::= sum(Menu[].Grasas_Saturadas)
```

```
Sales_Menu ::= sum(Menu[].Sales)
```

```
Peso_Menu ::= sum(Menu[].Peso)
```

```

Lactosa_Menu ::= Menu[].Lactosa
Fructosa_Menu ::= Menu[].Fructosa

Platos ::= Nombre_plato Alimentos Datos_Adicionales_Platos []

Nombre_plato ::= Nombre_plato String
Alimentos ::= Nombre_alimento Tipo DatosAdicionales []
Tipo ::= Tipo String
Datos adicionales ::= Azucares Proteinas Carbohidratos Calorias Grasas
    Grasas_Saturadas Sales Lactosa Fructosa Peso

Azucares ::= Azucares Integer
Proteinas ::= Proteinas Integer
Carbohidratos ::= Carbohidratos Integer
Calorias ::= Calorias Integer
Grasas ::= Grasas Integer
Grasas_Saturadas ::= Grasas_Saturadas Integer
Sales ::= Sales Integer
Peso ::= Peso Integer
Lactosa ::= Lactosa Boolean
Fructosa ::= Fructosa Boolean

Datos_adicionales_Platos ::= Azucares_Platos Proteinas_Plato Carbohidratos_Plato
    Calorias_Plato Grasas_Plato Grasas_Saturadas_Plato Sales_Platos
    Peso_Plato Lactosa_Plato Fructosa_Plato

Azucares_Platos ::= sum(Alimento[].Azucares)
Proteinas_Platos ::= sum(Alimento[].Proteinas)
Carbohidratos_Platos ::= sum(Alimento[].Carbohidratos)
Calorias_Platos ::= sum(Alimento[].Calorias)
Grasas_Platos ::= sum(Alimento[].Grasas)
Grasas_Saturadas_Platos ::= sum(Alimento[].Grasas_Saturadas)
Sales_Platos ::= sum(Alimento[].Sales)
Peso_Platos ::= sum(Alimento[].Peso)
Lactosa_Platos ::= Alimento[].Lactosa
Fructosa_Platos ::= Alimento[].Fructosa

```

Definiendo un menú de ejemplo

```

Menu ::=
    Nombre_Menu "Menú Saludable"
    Pautas "Consumir a lo hora del almuerzo el plato."
    Datos_Adicionales_Menu
        Azucares_Menu sum(Menu[].Azucares)

```

```

Proteinas_Menu sum(Menu[].Proteinas)
Carbohidratos_Menu sum(Menu[].Carbohidratos)
Calorias_Menu sum(Menu[].Calorias)
Grasas_Menu sum(Menu[].Grasas)
Grasas_Saturadas_Menu sum(Menu[].Grasas_Saturadas)
Sales_Menu sum(Menu[].Sales)
Peso_Menu sum(Menu[].Peso)
Lactosa_Menu Menu[].Lactosa
Fructosa_Menu Menu[].Fructosa
Platos
Nombre_plato "Sopa de Verduras"
Alimentos
Nombre_alimento "Zanahoria"
Tipo "Vegetal"
DatosAdicionales
Azucares 3
Proteinas 1
Carbohidratos 7
Calorias 30
Grasas 0
Grasas_Saturadas 0
Sales 0
Peso 100
Lactosa false
Fructosa false
Nombre_alimento "Apio"
Tipo "Vegetal"
DatosAdicionales
Azucares 1
Proteinas 0
Carbohidratos 1
Calorias 10
Grasas 0
Grasas_Saturadas 0
Sales 0
Peso 50
Lactosa false
Fructosa false
Datos_Adicionales_Platos
Azucares_Platos sum(Alimento[].Azucares)
Proteinas_Platos sum(Alimento[].Proteinas)
Carbohidratos_Platos sum(Alimento[].Carbohidratos)
Calorias_Platos sum(Alimento[].Calorias)
Grasas_Platos sum(Alimento[].Grasas)
Grasas_Saturadas_Platos sum(Alimento[].Grasas_Saturadas)
Sales_Platos sum(Alimento[].Sales)

```

```
Peso_Platos sum(Alimento[].Peso)
Lactosa_Platos Alimento[].Lactosa
Fructosa_Platos Alimento[].Fructosa
```

En este ejemplo se utiliza la gramática para definir un menú completo, añadiendo un plato compuesto de 2 alimentos con sus datos nutricionales y alérgenos.

2.2.4. Semántica del lenguaje

La semántica del lenguaje describe el significado y la interpretación de los diferentes componentes del lenguaje. A continuación se detalla la semántica junto con ejemplos.

- Menú: Un menú es una colección de platos diseñada para cumplir con ciertas restricciones dietéticas. La semántica de un menú incluye su propósito y cómo se agregan los datos nutricionales de los platos individuales.

Menú: Menú Saludable

Pautas: Consumir los platos en orden.

Datos Nutricionales del Menú:

Azúcares: 10

Proteínas: 30

Carbohidratos: 80

Calorías: 500

Grasas: 15

Grasas Saturadas: 3

Sales: 1

Lactosa: false

Fructosa: true

Peso: 600

Platos:

1. Ensalada de Quinoa

2. Sopa de Verduras

En este ejemplo el menú contiene 2 platos y los valores nutricionales son la suma de ambos platos.

- Plato: Un plato es una combinación de alimentos.

Plato: Ensalada de Quinoa

Alimentos:

1. Quinoa

2. Tomate

Instrucciones: Mezclar todos los ingredientes y servir frío.

Datos Nutricionales del Plato:

Azúcares: 4

Proteínas: 9

Carbohidratos: 25
Calorías: 140
Grasas: 2
Grasas Saturadas: 0
Sales: 0
Lactosa: false
Fructosa: true
Peso: 335

En este ejemplo, el plato esta compuesto por 2 alimentos y los datos nutricionales son la acumulación de ambos alimentos.

- **Alimento:** Un alimento es un ingrediente básico con atributos nutricionales específicos. La semántica de un alimento incluye cómo sus atributos contribuyen al valor nutricional total del plato y del menú.

Alimento: Quinoa
Tipo: Grano
Datos Nutricionales:
Azúcares: 1
Proteínas: 8
Carbohidratos: 21
Calorías: 120
Grasas: 2
Grasas Saturadas: 0
Sales: 0
Lactosa: false
Fructosa: false
Peso: 185

En este ultimo ejemplo se define un alimento con sus datos nutricionales. Como se puede observar a través de los ejemplos los datos nutricionales en el menú son mayores que los del plato y los valores de este ultimo son mayores que los del alimento base, permitiendo crear menús de una forma estructurada y ayudando a planificar una dieta equilibrada.

2.3. Diseño del modelo de datos

En esta sección se detalla el diseño del modelo de datos utilizado para almacenar y gestionar la información relacionada con los alimentos, platos y menús en la base de datos del proyecto. Este diseño es importante para asegurar que la información dietética es accesible, precisa y fácil de manejar por el sistema.

Alimento en la base de datos

El modelo de datos para un alimento almacenado en la base de datos del proyecto incluye información sobre su nombre, contenido calórico y otros datos adicionales que pueden ser relevantes para las restricciones alimenticias de los usuarios como se observa en la Tabla 2.1.

Clave	Tipo
id	Integer
nombre	String
calorias	Integer
proteinas	Integer
carbohidratos	Integer
grasas	Integer
grasas_saturadas	Integer
sales	Integer
azucares	Integer
peso	Integer
lactosa	Boolean
fructosa	Boolean

Tabla 2.1: Esquema del modelo de datos de los alimentos en la base de datos

Para poder crear un menú (Tabla 2.2) hace falta colocarle un nombre y asignarle varios platos, las pautas no son obligatorias añadirlas

Clave	Tipo
nombre	String
pautas	String
plato	plato

Tabla 2.2: Modelo de datos de menú

El menú esta compuesto de varios platos y cada plato tiene que tener un nombre y tener asignado varios alimentos (Tabla 2.3).

Clave	Tipo
nombre	String
alimentos	Alimento

Tabla 2.3: Modelo de datos del plato

Cada alimento tiene un nombre, un tipo y una lista con los datos adicionales, estos datos adicionales son la información que se almacena en la base de datos es

decir (calorías, proteínas, carbohidratos, grasas, grasas saturadas, sales, azúcares, peso) (Tabla 2.4).

Clave	Tipo
nombre	String
tipo	String
datos adicionales	Array

Tabla 2.4: Modelo de datos del alimento

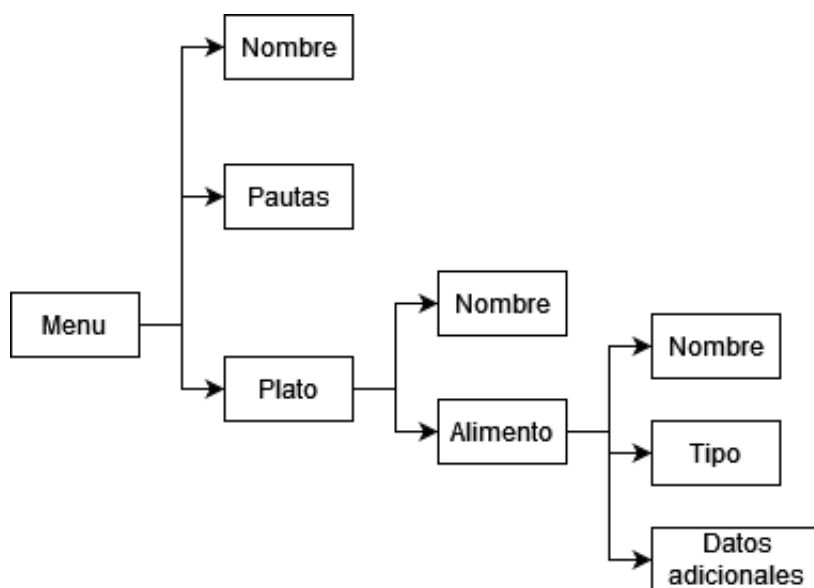


Figura 2.2: Diagrama de clases

2.4. Metodología

La metodología ágil seleccionada para el proyecto se llama SCRUM. Ésta se caracteriza por un desarrollo incremental, dividiendo en pequeñas partes el proyecto. Estas partes se van asignando en diferentes reuniones establecidas a lo largo del desarrollo del proyecto, siendo revisadas y modificadas cada parte a lo largo del proyecto.

2.5. Tecnologías empleadas

2.5.1. Lenguaje de programación Ruby

Ruby ha sido el lenguaje seleccionado para el desarrollo del DSL, debido a que tiene una sintaxis simple y legible que ayuda en el desarrollo posterior del código y su mantenimiento. Algunas de las características principales de Ruby son:

- Simplicidad y legibilidad: Ruby permite escribir código de forma intuitiva y clara, reduciendo la curva de aprendizaje y facilitando la cooperación entre desarrolladores.
- Orientado a objetos.
- Meta-programación: Permite escribir código que se escribe por sí mismo de forma dinámica en tiempo de ejecución.
- Ecosistema y comunidad: Ruby cuenta con una comunidad activa con gran cantidad de gemas (bibliotecas), proporcionando herramientas y extensiones para el desarrollo.
- Open source.

Para este proyecto se ha utilizado Ruby para definir la sintaxis y semántica del lenguaje de dominio específico, permitiendo a los usuarios especificar sus necesidades nutricionales y restricciones.

2.5.2. Base de datos con SQLite3

Para la creación de la base de datos propia del proyecto, se ha utilizado SQLite3. SQLite3 es una biblioteca que implementa una base de datos SQL embebida, auto contenida y de alto rendimiento. Algunas de sus características son:

- Open source.
- Ligereza: No requiere configurar un servidor.
- Portabilidad: Es auto-contenida (sin dependencias externas), permitiendo así distribuir fácilmente el software.
- Fiabilidad: Alta fiabilidad y robustez.
- Rapidez: Puede funcionar enteramente en memoria.

Para este proyecto, SQLite3 se utiliza para almacenar la información de los alimentos más requeridos por los usuarios, almacenando toda la información nutricional necesaria para poder aplicar las restricciones de los usuarios.

2.5.3. Bases de datos externas

Para recopilar información nutricional detallada y actualizada de un alimento que no se encuentra en la base de datos propia, se utilizan dos fuentes externas: Nutritionix y USDA Food Data Center.

- Nutritionix: Esta base de datos proporciona información nutricional sobre una amplia variedad de alimentos, desde alimentos genéricos a alimentos de marcas comerciales.

- API: Nutritionix ofrece una API bien documentada que facilita su integración y acceso a la información nutricional.
 - Información nutricional: Información nutricional sobre macro-nutrientes y micro-nutrientes necesaria para proporcionar recomendaciones dietéticas.
- USDA Food Data Center: La base de datos del USDA (Departamento de Agricultura de los Estados Unidos) es una fuente completa de información nutricional.
- Precisión y detalle: Es una base de datos conocida por su precisión y nivel de detalle, cubriendo una amplia gama de nutrientes.
 - Datos actualizados: Esta base de datos se actualiza de forma regular para reflejar la información más reciente.

Estas bases de datos se usan para recopilar la información de alimentos que no se encuentran en la base de datos propia y luego integrarla en SQLite3 para poder realizar la generación de menús y aplicar restricciones nutricionales.

2.6. Validación y pruebas

Para asegurar el correcto funcionamiento del proyecto se optó por elegir un sistema de comprobación de pruebas unitarias. Esta trata en ir verificando el correcto funcionamiento de cada parte desarrollada de forma independiente, complementándose con la metodología SCRUM utilizada.

En el siguiente capítulo (Capítulo 3 Implementación), se presentarán ejemplos de las pruebas unitarias realizadas.

Capítulo 3

Implementación

En este capítulo se describirá el sistema desde un menor nivel de abstracción y se mostrarán algunos puntos de interés.

Se mostrará y explicará el código del proyecto en su totalidad, ofreciendo una fotografía general del sistema.

En el anexo de esta memoria se adjunta el repositorio que contiene todo el código fuente del proyecto, junto con el tag del último commit realizado en fecha válida para la entrega del presente trabajo.

3.1. Estructura del código

El proyecto se ha estructurado de manera modular para facilitar la extensión y mantenimiento del código. A continuación, se describe la organización de los diferentes componentes y se proporcionan fragmentos del código.

Archivos

- `NutricionalDsl.rb`: Define la estructura principal del DSL y sus métodos.
- `NutricionalDslScript.rb`: Script que ejecuta el DSL para generar menús nutricionales con restricciones de usuarios.
- `Restricciones.rb`: Define las restricciones y etiquetas que se les asignan a los alimentos.
- `Usuarios.rb`: Define la clase `Usuario` con sus atributos (Nombre, Sexo, Edad, Nivel de actividad) y sus restricciones nutricionales.
- `ApiNutritionix.rb`: Permite acceder a la API de Nutritionix para retornar los datos nutricionales de un alimento.
- `DataFoodJson.rb`: Maneja la información del archivo JSON de la base de datos de USDA FoodData Centre.
- `Database.rb`: Este archivo se encarga de interactuar con la base de datos de SQLite3.

3.2. Patrón de diseño

El patrón de diseño utilizado es Embedded Interpretation. Este patrón permite definir un lenguaje específico dentro de un lenguaje de programación más general, Sus características son:

- DSL interpretado: Al ser interpretado las definiciones del DSL son ejecutadas directamente en el intérprete del lenguaje anfitrión.
- DSL interna: Se construye utilizando las estructuras y sintaxis del lenguaje anfitrión.

Las ventajas de usar este patrón son:

- Simplicidad: No necesitas un compilador o traductor adicional.
- Integración: El DSL puede aprovechar las características del lenguaje anfitrión(gemas)
- Flexibilidad: Se puede mezclar el código DSL con el lenguaje anfitrión.

3.3. Clase DSL

Esta es la clase principal del proyecto, permite definir menús y generar menús con restricciones. Un menú está compuesto por tres componentes: menú, plato y alimento. Un menú contiene el nombre de este y una lista de platos; el componente plato contiene igualmente un nombre y una lista de alimentos; y el último componente está compuesto por un nombre, tipo y la información nutricional del alimento, en este caso (calorías, proteínas, carbohidratos, grasas, peso, grasas saturadas, sales, azúcares).

Las partes más importantes de esta clase son:

- Clase Menú, Plato y Alimento: Contiene el modelo de datos del proyecto que ya se explico en el capítulo 2 (Figura 3.1 y 3.2).

```

class Menu
  attr_accessor :name, :platos, :pautas

  def initialize(name, pautas)
    @name = name
    @platos = []
    @pautas = pautas
  end

  def plato(name, &block)
    plato = Plato.new(name)
    plato.instance_eval(&block)
    @platos << plato
  end
end

```

Figura 3.1: Clase Menú

```

class Plato
  attr_accessor :name, :comidas

  def initialize(name)
    @name = name
    @comidas = []
  end

  def comida(name, tipo, &block)
    comida = Alimento.new(name, tipo)
    comida.instance_eval(&block)
    @comidas << comida
  end
end

class Alimento
  include Restricciones
  attr_accessor :name, :tipo, :datos_adicionales

  def initialize(name, tipo)
    @name = name
    @tipo = tipo
    @datos_adicionales = {}
  end
end

```

Figura 3.2: Clase Plato y Alimento

- Generar menú: Este método se encarga de crear un menú para un número específico de días y de platos, este número lo debe de introducir el usuario, tras ello la función genera el menú dietético del usuario pero teniendo en cuenta las restricciones que tiene su usuario y su rango calórico.
- Generar informe: Se encarga de mostrar al usuario o varios usuarios su menú de forma detallada enseñando los platos que debe de tomar cada día y la información nutricional de cada plato.

```
Restricciones más comunes y filtradas: ["Moderado en carbohidratos", "Alto en sales"]
Rango calórico: 2200..2600
Datos adicionales menú: ["Moderado en grasas", "Bajo en grasas saturadas", "Alto en sales", "Bajo en azúcar", "Moderado en calorías", "Moderado en carbohidratos", "Bajo en proteínas", "Contiene lactosa", "Contiene fructosa"]
Usuario: Grupo
Sexo: H
Edad: Adulto
Sedentario: No
Menú: Día 1
Pautas: Menú generado
Plato: Plato 1
Alimento: Oatmeal
Tipo: General
Datos adicionales: {calorías=>166.14, :proteínas=>5.94, :carbohidratos=>28.08, :grasas=>3.56, :peso=>234.0, :grasas_saturadas=>0.73, :sales=>9.36, :azucares=>0.63}
Alimento: Avocado
Tipo: General
Datos adicionales: {calorías=>321.6, :proteínas=>4.02, :carbohidratos=>17.15, :grasas=>29.47, :peso=>201.0, :grasas_saturadas=>4.27, :sales=>14.07, :azucares=>1.33}
Total del plato - Calorías: 487.74, Proteínas: 9.96, Carbohidratos: 45.23, Grasas: 33.03, Peso: 435.0
```

Figura 3.3: Generación de un informe de un plato

- Verificar restricciones: Como su nombre indica se encarga de comprobar si el menú dietético que se esta generando cumple con las restricciones del usuario, para ello verifica las etiquetas de restricciones que se generan del menú, con las etiquetas de restricciones que tiene el usuario.
- Obtener rango calórico: Esta función devuelve el rango calórico del usuario para poder ajustar el menú ya sea añadiendo más alimentos o mayor cantidad de algún alimento del menú para llegar a las calorías necesarias en base a su constitución física, en la Tabla 3.1 se puede observar como se elige el rango calórico.

Sexo	Sedentario	Activo
Niño	1000-1400	1000-1800
Niña	1000-1200	1000-1600
Adolescente (H)	1400-2400	1600-3200
Adolescente (M)	1200-1800	1600-2400
Adulto (H)	2200-2600	2400-3000
Adulto (M)	1600-2000	1800-2400
Anciano	2000-2200	2200-2600
Anciana	1600-1800	1800-2000

Tabla 3.1: Número estimado de calorías necesarias basado en la edad, el sexo y el nivel de actividad

Es importante tener en cuenta que el funcionamiento de generar menú cambia con respecto al número de usuarios, si es solo un usuario pues se genera el menú en base a sus restricciones de forma normal, en cambio si se introducen varios se realiza una media de las etiquetas más comunes y del rango calórico que más se repite, pero en el caso de intolerancias desde que uno de los usuarios no pueda

tomar lactosa o fructosa, entonces esta restricción se aplicara tanto a las persona con el problema de salud como al resto, ya que se entiende que si se introducen varios usuarios los alimentos se prepararan en común.

3.4. Restricciones

Las restricciones en este proyecto se dividen en dos tipos. El primero de ellos consiste en indicar si un alimento contiene o no ciertos componentes, como es el caso de la lactosa y la fructosa. El otro tipo es etiquetar la cantidad del componente nutricional, es decir, si para estos componentes (calorías, proteínas, carbohidratos, grasas, peso, grasas saturadas, sales, azúcares) se encuentran en niveles bajos, moderados o altos (Figura 3.4).

```
Datos adicionales menu: ["Moderado en grasas", "Bajo en grasas saturadas", "Alto en sales", "Bajo en azucar", "Moderado en calorías", "Moderado en carbohidratos", "Bajo en proteínas", "Contiene lactosa", "Contiene fructosa"]
```

Figura 3.4: Restricciones de un menú

El funcionamiento de esta clase se base en recoger la información nutricional del menú, realizar unas formulas matemáticas para cada nutrientes en base a la cantidad que se encuentra de ese nutriente en relación a su peso, para poder clasificarlo en niveles siguiendo las instrucciones del gobierno de EEUU sobre regulación de alimentos [1], de esta forma luego con las restricciones que tiene el usuario se podrá identificar de forma exacta si el menú es apto para su consumo y permite mayor flexibilidad en la generación de menús, debido a que no se identifica si un alimento tiene altos o bajos niveles de un nutriente, sino se identifica en base a su cantidad de consumo.

```
def etiqueta_sales(sales, peso)
  if sales == 0
    return "Sin sales"
  end
  relacion = (sales / peso) * 100
  if relacion > 1.25
    return "Alto en sales"
  elsif relacion < 0.25
    return "Bajo en sales"
  else
    return "Moderado en sales"
  end
end
```

Figura 3.5: Etiqueta sales

3.5. Usuarios

La clase Usuarios, además del nombre, contiene etiquetas de sus restricciones nutricionales, su sexo, edad (Niño, Adolescente, Adulto, Anciano) y si lleva un estilo

de vida sedentario o no. Esta información se pide para poder realizar una estimación del rango calórico (Tabla 3.1) que debe consumir la persona en su menú diario propuesto, esta clase se puede observar en la Figura 3.6.

```
class Usuario
  attr_accessor :nombre, :etiquetas_restriccion, :sexo, :edad, :sedentario

  def initialize(nombre, sexo, edad, sedentario)
    @nombre = nombre
    @sexo = sexo
    @edad = edad
    @sedentario = sedentario
    @etiquetas_restriccion = []
  end

  def agregar_restriccion(etiqueta)
    @etiquetas_restriccion << etiqueta
  end
end
```

Figura 3.6: Clase usuario

3.6. Bases de datos

Para la base de datos propia se utilizó SQLite3 para almacenar toda la información de los alimentos, primero se deben introducir una lista de nombre de alimentos para poder tener varios alimentos en el sistema que generan menú, tras introducir estos nombres, de forma automática se buscará toda su información nutricional en bases de datos externas en este caso será en Nutritionix y USDA FoodData Central, se recopilará toda la información necesaria de estas BBDD y se guardará en un archivo para poder realizar las búsquedas de forma mucho más rápida.

```
$db = SQLite3::Database.new "nutricion.db"

$db.execute <<-SQL
CREATE TABLE IF NOT EXISTS alimentos (
  id INTEGER PRIMARY KEY,
  nombre VARCHAR(255),
  calorias INTEGER DEFAULT 0,
  proteinas INTEGER DEFAULT 0,
  carbohidratos INTEGER DEFAULT 0,
  grasas INTEGER DEFAULT 0,
  grasas_saturadas INTEGER DEFAULT 0,
  sales INTEGER DEFAULT 0,
  azucares INTEGER DEFAULT 0,
  peso INTEGER DEFAULT 0,
  lactosa BOOLEAN DEFAULT false,
  fructosa BOOLEAN DEFAULT false
);
SQL
```

Figura 3.7: base de datos SQLite3

Se accede a la API de nutritionix (Figura 3.8) para recopilar la información de todos los nutrientes de los alimentos introducidos, así como su peso para poder realizar el etiquetado de las restricciones, también se accede a un archivo Json descargado de USDA FoodData Central para recopilar la información adicional.

```

response = HTTParty.post('https://trackapi.nutritionix.com/v2/natural/nutrients', options)

if response.code == 200
  parsed_response = JSON.parse(response.body)

  if parsed_response && parsed_response["foods"] && parsed_response["foods"][0]
    calorías = parsed_response["foods"][0]["nf_calories"].to_f
    proteínas = parsed_response["foods"][0]["nf_protein"].to_f
    carbohidratos = parsed_response["foods"][0]["nf_total_carbohydrate"].to_f
    grasas = parsed_response["foods"][0]["nf_total_fat"].to_f
    grasas_saturadas = parsed_response["foods"][0]["nf_saturated_fat"].to_f
    sales = parsed_response["foods"][0]["nf_sodium"].to_f
    azúcares = parsed_response["foods"][0]["nf_sugars"].to_f
    peso = parsed_response["foods"][0]["serving_weight_grams"].to_f
    atributos = parsed_response['full_nutrients']

    return {
      calorías: calorías,
      proteínas: proteínas,
      carbohidratos: carbohidratos,
      grasas: grasas,
      grasas_saturadas: grasas_saturadas,
      sales: sales,
      azúcares: azúcares,
      peso: peso
    }
  else
    puts "Error: No se encontraron datos nutricionales para #{food_name}"

```

Figura 3.8: Accediendo a base de datos Nutritionix

3.7. Pruebas unitarias

Para la realización de las pruebas unitarias se ha usado el framework RSpec, a continuación se muestran capturas de algunas pruebas unitarias realizadas para generar menús dentro del sistema y una prueba de la respuesta de menú que recibiría un usuario.

```

describe '#menu' do
  it 'crea un menú personalizado' do
    dsl.menu("Lunes", "Pautas para el menú del lunes: ") do
      plato("Desayuno") do
        comida("Apple", "Fruta") do
          calorias 52
          proteinas 0.3
          carbohidratos 14
          grasas 0.2
          peso 182
          grasas_saturadas 0.03
          sales 1
          azucares 10
        end
      end
    end
  end

  menu = dsl.instance_variable_get(:@menus).first
  expect(dsl.instance_variable_get(:@menus).size).to eq(1)
  expect(menu.name).to eq("Lunes")
  expect(menu.platos.size).to eq(1)
  expect(menu.platos.first.name).to eq("Desayuno")
  expect(menu.platos.first.comidas.size).to eq(1)
  expect(menu.platos.first.comidas.first.name).to eq("Apple")
  expect(menu.platos.first.comidas.first.datos_adicionales[:calorias]).to eq(52)
end
end

```

Figura 3.9: Pruebas menú personalizado

```

describe '#generar_menus' do
  it 'genera un menú en base a restricciones del usuario' do
    usuario.etiquetas_restriccion = ["Moderado en carbohidratos"]
    dsl.generar_menus(platos_por_dia: 3, dias: 1, usuario: usuario)

    expect(dsl.instance_variable_get(:@menus).size).to eq(1)
    menu = dsl.instance_variable_get(:@menus).first
    expect(menu.platos.size).to eq(3)
    expect(menu.platos.first.comidas.size).to be > 0
  end
end

```

Figura 3.10: Prueba generando un menú con restricciones de usuario

```

describe '#generate_report' do
  before do
    dsl.menu("Lunes", "Pautas para el menú del lunes: ") do
      plato("Desayuno") do
        comida("Apple", "Fruta") do
          calorias 52
          proteinas 0.3
          carbohidratos 14
          grasas 0.2
          peso 182
          grasas_saturadas 0.03
          sales 1
          azucares 10
        end
      end
    end
  end

  it 'genera la salida del menú' do
    expect { dsl.generate_report([usuario]) }.to output(/Usuario: Juan/).to_stdout
  end
end

```

Figura 3.11: Prueba generando la respuesta al usuario

Capítulo 4

Conclusiones y líneas futuras

A pesar del enorme recorrido que supone desarrollar un software hasta el punto de ser apto para su explotación y que este proyecto aún necesitaría en dicho caso, no es aventurado decir que hasta este punto, el proyecto ha conseguido sus objetivos, los cuales fueron descritos en el Capítulo 1:

- Se pretendía lograr una aplicación que al contrario de muchas existentes, no fuese una mera tabla de valores, ajustándose dinámicamente a las tolerancias del usuario y a lo que ha comido a lo largo del día, objetivo que ha sido cumplido satisfactoriamente.
- Se ha introducido el concepto de plato con el que ninguna aplicación cuenta a fecha de realización de este trabajo, permitiendo agrupar y consumir múltiples alimentos frecuentes de una sola vez, mejorando la comodidad del usuario y dejando la puerta abierta a potenciales mejoras como podría ser un manual de recetas.
- La aplicación se ha desarrollado siguiendo una arquitectura muy modularizada que le permite ser modificada o escalada con seguridad y simplicidad, pudiendo adaptarse rápidamente a cambios o novedades.

Como posibles líneas futuras del proyecto si se dispusiera de más tiempo, las divido en dos categorías: las de carácter técnico y las de carácter funcional.

A nivel técnico:

- Versionar el código adecuadamente utilizando versionado semántico. Separar entornos de desarrollo y producción, comenzar a diseñar y utilizar una estrategia de branching, así como en herramientas de testing, integración y despliegue continuo antes de que el proyecto escale demasiado.
- Dotar a la aplicación de un escáner de código de barras.

A nivel funcional:

- Implementar una interfaz de usuario amigable.
- Ampliar la base de datos.

En conclusión, ha sido un proyecto muy interesante y ameno de realizar en el que además han intervenido todos o muchos niveles de la pila de desarrollo necesaria para crear un producto, temática precisamente propia de la titulación a la que se opta con la presentación de esta memoria. Como primer prototipo, podría decirse que el proyecto ha sido un éxito, ya que se han cumplido todos los objetivos propuestos en el planteamiento y además de la forma en que se pretendían cumplir.

Capítulo 5

Summary and Conclusions

Despite the enormous journey involved in developing software to the point of being suitable for exploitation, and that this project would still need in such a case, it is not bold to say that up to this point, the project has achieved its objectives, which were described in Chapter 1:

- The aim was to achieve an application that, unlike many existing ones, was not a mere table of values, dynamically adjusting to the user's tolerances and what they have eaten throughout the day, a goal that has been satisfactorily achieved.
- The concept of a dish has been introduced, which no application has to date, allowing the grouping and consumption of multiple frequent foods at once, improving user convenience and leaving the door open to potential improvements such as a recipe manual.
- The application has been developed following a very modular architecture that allows it to be modified or scaled with security and simplicity, being able to quickly adapt to changes or new features.

As potential future lines of the project if more time were available, I divide them into two categories: technical and functional.

At a technical level:

- Properly versioning the code using semantic versioning. Separating development and production environments, starting to design and use a branching strategy, as well as tools for testing, continuous integration, and deployment before the project scales too much.
- Equipping the application with a barcode scanner.

At a functional level:

- Implementing a user-friendly interface.
- Expanding the database.

In conclusion, it has been a very interesting and enjoyable project to carry out, in which all or many levels of the development stack necessary to create a product have been involved, a theme precisely characteristic of the degree aimed for with the presentation of this report. As a first prototype, it could be said that the project has been a success, as all the proposed objectives have been met and in the way they were intended to be fulfilled.

Apéndice A

Repositorio del proyecto

Se adjunta la dirección del repositorio Github que contiene el proyecto junto con el tag del último commit, que garantiza el estado del mismo en fecha previa al límite de entrega del presente trabajo.

https://github.com/SantiagoVV/DSL_Master Tag: v1.0.0

Bibliografía

- [1] Administración de Alimentos y Medicamentos. Administración de alimentos y medicamentos. <https://www.fda.gov/media/80651/download>, 2023. Agencia del Gobierno de los Estados Unidos responsable de la regulación de alimentos, medicamentos, cosméticos, aparatos médicos, productos biológicos y derivados sanguíneos.
- [2] Jean-Louis Flandrin and Massimo Montanari. Historia de la alimentación. *Gijón: Trea*, pages 1960–1990, 2004.
- [3] Francisco José García Peñalvo et al. Desarrollo de estados de la cuestión robustos: Revisiones sistemáticas de literatura. *Education in the knowledge society: EKS*, 2022.
- [4] Ainhoa Lizaso Eguileta. Solution for the management of several clinical practice guidelines in a domain-independent decision support system. 2021.
- [5] Luca Mainetti, Luigi Patrono, and Roberto Vergallo. Combining epcglobal and hl7 to deploy innovative e-health services for patients affected by multiple intolerances. In *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, pages 1–6, 2011.
- [6] Auke Roorda. *Corel: A DSL for Cooking Recipes*. PhD thesis, 2021.
- [7] Markus Schinle, Michael Dietrich, Simon Stock, Marius Gerdes, and Wilhelm Stork. Model-driven dementia prevention and intervention platform. In *Caring is Sharing—Exploiting the Value in Data for Health and Innovation*, pages 937–941. IOS Press, 2023.
- [8] Mira Vrbaski, Gunter Mussbacher, Dorina Petriu, and Daniel Amyot. Goal models as run-time entities in context-aware systems. In *Proceedings of the 7th Workshop on Models@ run. time*, pages 3–8, 2012.