



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**Sistema de recomendación  
nutricional  
HEALTHAPP  
Documentación Técnica**



Presentado por Jesús Carro Tomé  
en Universidad de Burgos — 29 de junio  
de 2019

Tutor: Antonio J. Canepa Oneto



---

# Índice general

---

<b>Índice general</b>	<b>I</b>
<b>Índice de figuras</b>	<b>III</b>
<b>Índice de tablas</b>	<b>v</b>
<b>Apéndice A Plan de Proyecto Software</b>	<b>1</b>
A.1. Introducción . . . . .	1
A.2. Planificación temporal . . . . .	1
A.3. Viabilidad . . . . .	3
<b>Apéndice B Especificación de Requisitos</b>	<b>13</b>
B.1. Introducción . . . . .	13
B.2. Objetivos generales . . . . .	13
B.3. Catalogo de requisitos . . . . .	14
B.4. Especificación de requisitos . . . . .	14
<b>Apéndice C Especificación de diseño</b>	<b>17</b>
C.1. Introducción . . . . .	17
C.2. Diseño de datos . . . . .	17
C.3. Diseño procedimental . . . . .	17
C.4. Diseño arquitectónico . . . . .	23
<b>Apéndice D Documentación técnica de programación</b>	<b>29</b>
D.1. Introducción . . . . .	29
D.2. Estructura de directorios . . . . .	29
D.3. Manual del programador . . . . .	30

D.4. Compilación, instalación y ejecución del proyecto . . . . .	36
D.5. Pruebas del sistema . . . . .	38
<b>Apéndice E Documentación de usuario</b>	<b>41</b>
E.1. Introducción . . . . .	41
E.2. Requisitos de usuarios . . . . .	41
E.3. Instalación . . . . .	41
E.4. Manual del usuario . . . . .	43
E.5. Guía de errores, avisos e informes . . . . .	63
<b>Bibliografía</b>	<b>67</b>

---

## Índice de figuras

---

A.1. Cronograma del proyecto completo . . . . .	2
A.2. Matriz Mckensey del plan de negocio . . . . .	5
A.3. Análisis del resultado de la matriz Mckensey . . . . .	5
C.1. Diagrama UML sobre la interacción en el registro o modificación de un usuario . . . . .	19
C.2. Diagrama UML que muestra la interacción entre frames y fun- ciones durante la selección . . . . .	20
C.3. Diagrama de flujo de la acción de refrescar opciones de menú . .	21
C.4. Diagrama UML de la interacción básica del programa con la base de datos. . . . .	22
C.5. Diagrama UML general del programa, una vez se ha iniciado sesión. . . . .	23
C.6. Imagen de la consiguiente arquitectura visual para la correcta navegación . . . . .	24
C.7. Diagrama de las relaciones generales entre los distintos módulos	25
C.8. Diagrama de clases del módulo/clase vista. . . . .	26
C.9. Diagrama de clases del módulo/clase AdminBase. . . . .	27
C.10. Diagrama de clases del módulo/clase CalculosDieta. . . . .	27
C.11. Diagrama interno del Main . . . . .	28
D.1. comando para el uso de auto-py-to-exe . . . . .	32
D.2. Pantalla de la aplicación auto-py-to-exe . . . . .	33
D.3. Interfaz del editor Spyder para Python. . . . .	35
D.4. GitKraken . . . . .	36
D.5. Boton ejecutar de Spyder. . . . .	36
D.6. Pantalla nueva generada por la librería Tkinter (Versión 2.0) . .	37

D.7. Señalización de la terminal interna de Spyder . . . . .	37
E.1. Descomprimir el archivo ".zip" de la aplicación . . . . .	42
E.2. Localización del ejecutable del programa. . . . .	43
E.3. Pantalla de inicio del programa (Versión 2.0) . . . . .	44
E.4. Pantalla de inicio, botón Registrar (Versión 1.0) . . . . .	44
E.5. Ventana de registro del nuevo usuario señalizada para el registro (Version 1.0) . . . . .	45
E.6. Ventana de registro del nuevo usuario señalizada para el cierre (Version 1.0) . . . . .	46
E.7. Inicio de sesión detallado (Versión 2.0) . . . . .	46
E.8. Imagen detallando el cierre del programa (Version 2.0 . . . . .	47
E.9. Página principal de la aplicación (Version 1.0) . . . . .	47
E.10. Pantalla Principal señalizando la opción Archivo (Versión 1.0) . .	48
E.11. Pantalla Principal señalizando la opción estilos (Versión 1.0) . .	49
E.12. Botón para pasar a la ventana de la información del usuario (Versión 1.0) . . . . .	50
E.13. Botón para editar la información (Versión 1.0) . . . . .	51
E.14. Pantalla para editar la información del Usuario (Versión 1.0) . .	51
E.15. Botón para pasar a la dieta diaria (Versión 1.0) . . . . .	52
E.16. Pantalla principal de mostrar dieta (Versión 1.0) . . . . .	53
E.17. Información del plato y selecciones del día (Version 1.0) . . . . .	54
E.18. Selección de comida y guardado de la elección (Version 1.0) . .	55
E.19. Resultados del botón refrescar (Versión 1.0) . . . . .	56
E.20. Calificación Nutriscore . . . . .	56
E.21. Botón añadir alimento (Versión 1.0) . . . . .	57
E.22. Formulario para añadir un nuevo alimento (Versión 2.0) . . . . .	57
E.23. Partes del formulario (Versión 2.0) . . . . .	59
E.24. Opciones del tipo comida (V.1.0) . . . . .	59
E.25. Pantalla de validación de alimento (Versión 1.2) . . . . .	60
E.26. Botón Historial (Versión 1.0) . . . . .	61
E.27. Pantalla del Historial (V.1.0) . . . . .	62
E.28. Pantalla de muestra de la semana ingerida (Versión 1.0) . . . . .	63
E.29. Ventana emergente de información . . . . .	64
E.30. Ventana emergente de avisos . . . . .	64
E.31. Ventana de error . . . . .	65

---

# Índice de tablas

---





## *Apéndice A*

---

# Plan de Proyecto Software

---

### A.1. Introducción

En este Anexo se detallarán las distintas etapas, fases, cambios y procesos que ha pasado el programa hasta finalizar su desarrollo. También dentro de este proceso hay que tener en cuenta la viabilidad de llevar a cabo el proyecto hasta una versión comercial.

Utilizando el marco de trabajo SCRUM [3], para metodología ágil. Se lleva a cabo una evolución estructurada del proyecto. Sabiendo en la medida de lo posible cual iba a ser el avance de este.

### A.2. Planificación temporal

El proyecto fue dividido en diferentes etapas para lograr un desarrollo constante:

- Nacimiento y pulimento de la idea
- Desarrollo:
  - Creación de la estructura de los datos.
  - Carga y manejo de los datos
  - Creación del proyecto por interfaz de comando
  - Traducción del proyecto a interfaz gráfica
  - Pulimento de detalles y nuevas funcionalidades

- Pruebas
- Desarrollo de la documentación pertinente

Planificación principal del proyecto. Sostenida sobre un cronograma para cumplir las metas pautadas.

Cronograma:

Tareas	Responsable	Fecha de inicio	Fecha final	Días	Estado
<b>Pre-requisitos</b>					
Perfeccionamiento de idea	Jesus Carro	11/20	11/30	10	Completado
Definir objetivos	Jesus Carro/Antonio Jesús Canepa	30-nov	03-dic	3	Completado
<b>Inicio</b>					
Determinar los requisitos	Jesus Carro	30-nov	03-dic	3	Completado
Requisitos de Hardware	Jesus Carro	03-dic	05-dic	2	Completado
<b>Desarrollo</b>					
<b>Inicio</b>					
Requisitos técnicos	Jesus Carro	03-dic	03-dic	0	Completado
Desarrollo de base de datos	Jesus Carro	05-dic	10-dic	5	Completado
<b>Desarrollo Shell</b>					
Interfaz de Usuario del Cliente	Jesus Carro	10-dic	08-mar	88	Completado
Prueba	Jesus Carro	08-mar	12-mar	4	Completado
<b>Desarrollo Gráfico</b>					
Completar desarrollo	Jesus Carro	12-mar	20-jun	100	Completado
<b>Operaciones</b>					
Prueba del sistema	Jesus Carro	20-jun	01-jul	11	Completado
<b>Lanzamiento</b>		05-jul	05-jul		

Figura A.1: Cronograma del proyecto completo

## Fases

### Perfeccionamiento de la idea

La primera fase del proyecto fue elegir y perfeccionar una idea propia.

Se llevó a cabo un estudio sobre la demanda actual de los usuarios junto a las ultimas investigaciones realizadas, sumadas al interés del alumno por la ingeniería biomedica. Tras varios días de estudio se llegó a la conclusión de que una aplicación que no impartiese una dieta al usuario sino que ayudase al usuario a llegar por sus propios métodos al fin establecido, parecía un reto en cuanto investigación y avance.

**Definir Objetivos**

Tras la primera reunión, se pautaron los diferentes objetivos. Pautando la meta y requisitos que la aplicación debía cumplir con el paso del tiempo y hasta su etapa final.

**Determinar los requisitos**

En la reunión para hablar de los objetivos, se desarrollaron los requisitos. Ambas partes están relacionadas. Naciendo requisitos del sistema, según el alcance de los objetivos.

**Aplicación en Shell**

Se desarrollaron los algoritmos y las funcionalidades principales, con interfaz de línea de comandos. Permitted un desarrollo completo y centrado en el perfeccionamiento de dichas funciones.

**Pruebas del software**

Antes de avanzar con el desarrollo de la aplicación, se realizaron una serie de pruebas. Se corrigieron los fallos y errores resultantes durante dichas pruebas. Una vez terminados los errores, se pasó a la siguiente fase.

**Completar Desarrollo**

Construcción del programa en el resto de ámbitos, proporcionando al usuario una interfaz visual clara y simple, y dando las funcionalidades posibles de dicha interfaz al usuario como son: los gráficos, la persistencia y la navegabilidad.

**Pruebas del software finales**

Se llevaron a cabo numerosas pruebas del sistema. Comprobando su solidez, y consistencia ante posibles errores. Dando por finalizada la aplicación.

## A.3. Viabilidad

Es necesario que el proyecto sea viable en numerosos ámbitos, tanto en el económico como en el legal, además de una viabilidad consciente a la hora del desarrollo futuro.

## Viabilidad económica

En este apartado se realizará un análisis de la viabilidad económica usando algunas de las técnicas más utilizadas en iniciativas emprendedoras. Este apartado se divide en varias secciones, diferencia dos tipos de análisis:

1. Análisis de la viabilidad empresarial
2. Análisis económico de la viabilidad del proyecto, y margen de beneficios.

### Análisis de la viabilidad empresarial

#### Análisis DAFO

El análisis DAFO [5] es una herramienta de estudio de la situación de un proyecto, en base a sus principales características.

- Oportunidades: Crear algo nuevo y ser el primero en crearlo, siempre opta a ser el cabecilla del mercado
- Fortalezas: Producto innovador que se crea en la cresta de la ola que es la moda por la salud y el fitness
- Amenazas: El mundo de las aplicaciones es un mar lleno de rivales, donde malas aplicaciones triunfan y buenas fallan
- Debilidades: La dificultad de entrada al mercado al ser una aplicación nueva

#### Matriz MCKINSEY

La matriz mckinsey [9] es una herramienta de marketin estratégico que ayuda a la empresa a tomar decisiones sobre la inversión en el proyecto. Ayuda a priorizar la inversión, o llevar una estrategia mas conservadora

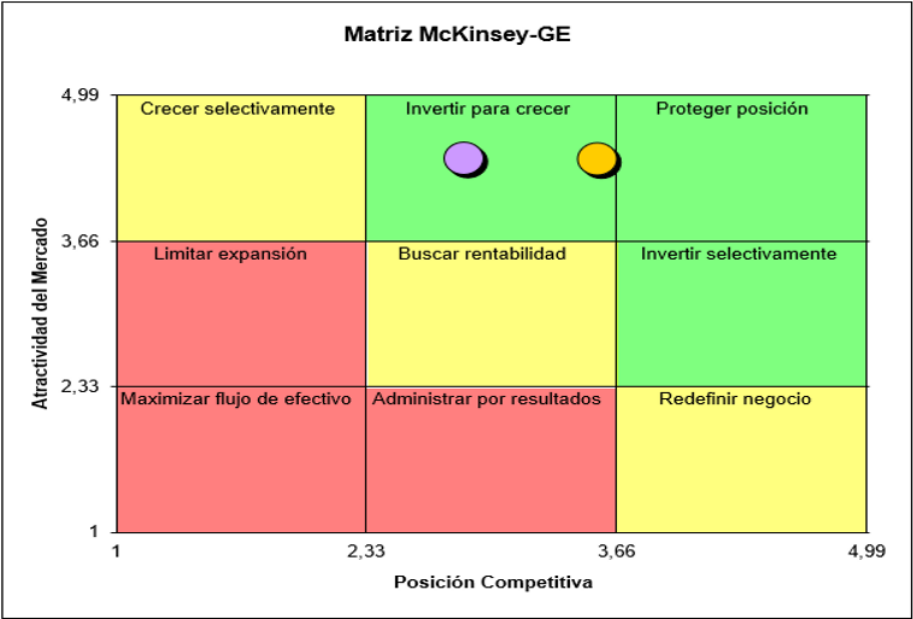


Figura A.2: Matriz Mckensey del plan de negocio

Oportunidad de Negocio						
Posición Competitiva	Factor	Peso	Calificación	Valor	Nueva posición	Nuevo Valor
	1. Posición de mercado	5%	3	0,15	4	0,20
	2. Crecimiento del mercado	10%	5	0,50	5	0,50
	3. Variedad de la oferta	15%	5	0,75	5	0,75
	4. Reputación de marca	20%	2	0,40	2	0,40
	5. Socios de negocios	10%	2	0,20	4	0,40
	6. Conocimiento del mercado	10%	3	0,30	4	0,40
	7. Capacidad de entrega	5%	3	0,15	5	0,25
	8. Imagen del mercado	15%	2	0,30	3	0,45
	9. Estructura organizacional	10%	1	0,10	2	0,20
			-		-	
	100%		2,85		3,55	

Figura A.3: Análisis del resultado de la matriz Mckensey

### Análisis económico

La formula para calcular la Amortización de los diferentes elementos utilizados a lo largo del programa es:

$$Amortización = \frac{(Valor)}{(VidaUtil)} \quad (A.1)$$

#### Costes Hardware:

S Durante el desarrollo del proyecto únicamente ha sido necesario utilizar un ordenador y un router con internet.

- Ordenador portátil: 460euros  
Vida útil de aproximadamente 5 años.  
 $460\text{euros}/(12*5) \text{ meses} = 7,66\text{euros}/\text{mes}.$   
Durante 7 meses de trabajo  $= 7,66*7 = 53.66$  euros brutos.
- Router: 50euros.  
 $50\text{euros}/(12*5)\text{meses}*7 \text{ meses de trabajo} = 5,82$  euros brutos.

#### Costes Software:

La única licencia software de pago usada durante el desarrollo del proyecto ha sido Windows 10 Home, la cual tiene un coste de 120 euros.

Se estimamos su vida útil en 4 años [1]. Por lo tanto:

$120\text{euros}/(12*4)\text{meses} = 1,5\text{euros}/\text{mes}.$

Por 7 meses trabajados son 17,5 euros brutos.

#### Costes Materiales:

Diferentes costes ajenos al software o hardware necesario para la integra realización del proyecto.

- Mobiliario - Oficina CEEI Burgos, con luces e inmuebles: 200euros/mes.  
Durante 7 meses de trabajo  $= 200\text{euros}/\text{mes}*7 = 1.400$  euros brutos
- Cartulinas - 2 euros
- Libretas - 5euros
- Pizarra - 10euros

**Costes de totales:**

Costes del trabajador según el régimen general de la Seguridad Social en 2019. [6]

Porcentajes de cotización:

- Comunes: 23.6 %
- Desempleo: 5.5 %
- Formación profesional: 0.7 %
- Total acumulado: 29,8 %

A continuación se mostrarás las diferentes cotizaciones respecto a la situación del alumno.

Coste real del trabajador:  $4070 \times 1,298 = 5.282$  euros.

Coste total del desarrollo del proyecto:

$5282 + 2 + 5 + 10 + 17.5 + 5.82 + 1400 + 53.66 = 6775.98$  euros brutos

**Beneficios:**

En este apartado se estima los beneficios que se pueden obtener del proyecto, y la amortización en años, en base al número y precio de ventas.

Supongamos que esta herramienta, es usada de manera privada por usuarios, y diferentes clínicas destinadas al reparto entre sus clientes.

Dos posibles paquetes de venta:

- 1 licencia X 130 euros.
- 10 licencias X 1.000 euros.

Suponiendo la venta a 10 personas para uso personal, y 3 paquetes de diez unidades para empresas sanitarias por año:

$10 \text{ unidades} \times 130 \text{ euros} = 1.300 \text{ euros}$

$3 \text{ Packs de } 10 \text{ unidades} \times 1000 \text{ euros} = 3000 \text{ euros}$

Lo que saldría de media a: 4.300 euros/año

Recuperando la inversión en:  $6775.98 \text{ euros} / 4300 \text{ euros} \times \text{año} = 1.53 \text{ años}$ .

A partir de ese periodo, aproximadamente año y medio, empezaríamos a tener beneficios.

## Viabilidad legal

### Licencias

#### Licencias de las librerías

- Matplotlib, licencia BSD.
- Numpy, licencia BSD.
- Pandas, licencia BSD.
- Auto-py-to-exe, licencia MIT.
- webbrowser, open source.
- os-win, licencia Apache software.
- Pillow, licencia *Open source Pil Software license*.
- openpyxl, licencia MIT.

La licencia BSD, es una licencia software libre permisiva, utilizada para los sistemas operativos BSD (Berkeley Software Distribution).

La licencia MIT, es una licencia de software libre permisiva lo que significa que apenas tiene limitaciones en la reutilización de sus recursos.

Según la guía de licencias de Python [7], tanto la licencia BSD como MIT, son adecuadas para este proyecto. No obstante, debido a que la licencia MIT es menos restrictiva, será la usada para el desarrollo. Con la licencia BSD, se ha de citar al creador del software usado y al propio software. La licencia Mit [8]

1. **Condiciones:** La condición es que la nota de copyright y la parte de los derechos se incluya en todas las copias o partes sustanciales del Software. Esta es la condición que invalidaría la licencia en caso de no cumplirse.
2. **Derechos:** sin restricciones; incluyendo usar, copiar, modificar, integrar con otro Software, publicar, sublicenciar o vender copias del Software, y además permitir a las personas a las que se les entregue el Software hacer lo mismo.
3. **Limitación de responsabilidad::** finalmente se tiene un disclaimer o nota de limitación de la responsabilidad habitual en este tipo de licencias.



### **Sociedad Limitada**

Se consideraría una sociedad limitada, al ser el alumno un autónomo presentando el proyecto para lo que sería su propia empresa con la serie de inversiones que esto conlleva, y la viabilidad legal-económica que trae consigo. Siendo al alumno autónomo y considerando este proyecto parte de su empresa. Se considera el proyecto como una sociedad limitada.

#### **Forma jurídica**

Descripción de la forma jurídica

Elección de una sociedad limitada por los motivos que me benefician:

- Capital Social mínimo: 3.000 euros
- N° socios mínimo:1
- 1socio 100 % del Capital Social.
- Limitacion a los bienes de la empresa en caso de deudas o cierre

#### **Trámites para la constitución de la empresa**

Descripción en detalle de los trámites de constitución

Solicitar el Certificado de Denominación Social.

El primer paso hay que dar es obtener el Certificado de Denominación Social. Dicho certificado acredita que no existe ninguna otra sociedad ya constituida que tenga la misma denominación social que la que pretendemos constituir y debe incorporarse a la escritura de constitución. Se obtiene en el Registro Mercantil Central, que tiene su sede en Madrid, y se podrá solicitar el certificado respecto de una o varias denominaciones sociales hasta un máximo de cinco. El certificado debe ir a nombre de cualquiera de las personas físicas o jurídicas que van a constituir la sociedad limitada como socios fundadores de la misma.

El Registro Mercantil Central deberá expedirlo de forma telemática en el plazo máximo de 1 día hábil contado desde la fecha de la solicitud. El Notario está obligado a solicitarlo por vía telemática, a menos que los interesados manifiesten su intención de solicitarlo ellos mismos. En el caso de que sean los propios interesados los que lo soliciten, pueden hacerlo de forma presencial rellenando el impreso oficial que deberán presentar en dicho Registro, o bien solicitarlo “on line” a través de la página web del Registro Una vez

obtenido, el certificado caduca a los dos meses de su fecha. Esto no obstante, se pueden solicitar nuevas certificaciones de la misma denominación social si caducan las anteriores, dado que la denominación social queda reservada a favor del solicitante durante el plazo de quince meses. Firmar la Escritura Pública de Constitución.

La sociedad de responsabilidad limitada se constituye mediante Escritura Pública otorgada ante Notario por la totalidad de los socios fundadores.

**Firmar la Escritura Pública de Constitución.** La sociedad de responsabilidad limitada se constituye mediante Escritura Pública otorgada ante Notario por la totalidad de los socios fundadores.

La escritura de constitución debe contener:

1. La identidad de los socios.
2. La voluntad de constituir una sociedad limitada.
3. La aportación de cada socio y las participaciones asignadas en pago de su aportación.
4. Los estatutos de la sociedad.
5. El sistema de administración que inicialmente se establezca para la sociedad.
6. La identidad de la persona que inicialmente se encargue de la administración y de la representación de la sociedad. Los socios deben elaborar unos Estatutos Sociales que se incorporarán a la Escritura de Constitución y por los que se regirá la sociedad. Dicha tarea

pueden encargarla los socios al Notario que autorizará la escritura de constitución.

Los Estatutos deben contener las siguientes menciones:

1. Denominación de la sociedad, en la que deberá figurar necesariamente la expresión “sociedad de responsabilidad limitada”, “sociedad limitada” o sus abreviaturas “S.R.L.” o “S.L.”.
2. Objeto social, que es la actividad a la que se va a dedicar la sociedad.

3. Fecha de cierre de cada ejercicio social.
4. Domicilio social dentro del territorio español.
5. Capital social, participaciones en que se divida, valor nominal de cada participación y numeración de las mismas.
6. Sistema de administración de la sociedad.

Si los socios realizan aportaciones dinerarias a la sociedad, deberán entregar al Notario un certificado que acredite el depósito en una Entidad de Crédito a nombre de la sociedad de las cantidades aportadas por los socios. La fecha del depósito bancario no podrá ser anterior en más de dos meses a la fecha en la que se firme la escritura de constitución de la sociedad. También cabe la posibilidad de que los socios entreguen directamente al Notario el dinero en que consista su aportación a la sociedad para que sea el propio Notario el que constituya el depósito en el plazo de cinco días hábiles. Si se trata de aportaciones no dinerarias (inmuebles, maquinaria, vehículos, etc), los socios deberá entregar al Notario los títulos de propiedad de tales bienes o la documentación relativa a los mismos.

La sociedad puede dar comienzo a sus operaciones comerciales desde la fecha en que se otorga la Escritura de Constitución, aunque no esté inscrita aún en el Registro Mercantil, salvo que en la propia escritura se haya fijado una fecha posterior para el comienzo de las operaciones de la sociedad.

#### **El Impuesto de Operaciones Societarias.**

Con anterioridad, la constitución de una sociedad limitada generaba para la misma la obligación de pagar el Impuesto de Operaciones Societarias al tipo del uno por ciento (1 %) del capital de la sociedad.

En la actualidad, a partir del Real Decreto-Ley 13/2010, la constitución de una sociedad limitada está exenta del pago del Impuesto de Operaciones Societarias. Además, para inscribirla en el Registro Mercantil, no será necesaria la presentación en dicho Registro del impreso de autoliquidación en el que se alegue la exención.

#### **Solicitud del N.I.F. provisional.**

El mismo día de la firma de la escritura, el Notario solicitará también telemáticamente un N.I.F. provisional para la sociedad, que se convertirá en

definitivo cuando la sociedad se inscriba en el Registro Mercantil correspondiente. Una vez convertido en definitivo el .N.I.F asignado inicialmente, Hacienda deberá comunicarlo por vía telemática al Notario y al propio Registro Mercantil donde está inscrita la sociedad.

**La Inscripción en el Registro Mercantil.** La Escritura de Constitución otorgada ante Notario debe inscribirse obligatoriamente y con carácter constitutivo en el Registro Mercantil de la provincia correspondiente al domicilio de la sociedad. A estos efectos, el Notario debe remitir telemáticamente una copia de la escritura de constitución al Registro correspondiente, a menos que los interesados soliciten lo contrario. El Registrador Mercantil tiene un plazo máximo de 15 días para inscribir la sociedad.

Una vez inscrita, la sociedad adquiere su personalidad jurídica como sociedad de responsabilidad limitada. Es necesario también publicar la inscripción en el B.O.R.M.E., cuyas tasas se pagarán telemáticamente.

Finalmente, para iniciar la actividad del negocio deberán cumplirse otros trámites ante la Hacienda Pública como dar de alta a la sociedad en el Impuesto de Actividades Económicas (I.A.E.), salvo que se trate de sociedades exentas, o realizar la declaración censal, que es el alta de la sociedad a los efectos del I.V.A.

Igualmente será preciso en la mayoría de los casos la obtención de licencia de apertura del establecimiento de la empresa y la licencia de obras, en el caso de que se efectúen obras de reforma en el local en el que se va a realizar la actividad. Ambas licencias deberán tramitarse ante el Ayuntamiento del municipio en el que se encuentra el local de la sociedad. También deberá tramitarse el alta la empresa en la Seguridad Social, que se efectúa en la Tesorería de la misma, y la comunicación de apertura del centro de trabajo, que se realiza en el Ministerio de Trabajo.

## *Apéndice B*

---

# Especificación de Requisitos

---

### B.1. Introducción

En los siguientes apartados de este capítulo se describirá todos los requisitos y objetivos que inicialmente debía cubrir el programa en cuanto a funcionalidad.

### B.2. Objetivos generales

El objetivo principal de este TFG es la realización de un proyecto completo, que ponga a disposición del usuario, un sistema automatizado de planificación alimenticia. Automatizando y facilitando el aprendizaje de la dietoterapia. Desarrollando un sistema de recomendación completo, renovando el calculo TMB, y guiando al usuario a través de múltiples gráficos y medidas. Como proyecto, ha de tener las funcionalidades mínimas para su correcta utilización:

- Alta y registro en la aplicación, además de la personalización de los datos.
- Sistema de recomendación único y preciso, para una buena veracidad de los resultados.
- Sistema de gráficos, y elementos visuales, para la correcta orientación del usuario.
- Cálculos estandarizados y personalizados, para la correcta expansión y funcionamiento.

- Guardado de datos y persistencia.

## B.3. Catalogo de requisitos

### Requisitos funcionales

1. **Requisito 1.** Probar correcto inicio de sesión y registro de nuevos usuarios.
2. **Requisito 2.** Probar que añadir un nuevo alimento funciona correctamente.
3. **Requisito 3.** Probar la correcta distribución alimenticia según la patología del usuario.
4. **Requisito 4.** Probar el correcto funcionamiento del sistema de recomendación
5. **Requisito 5.** Verificar la correcta actualización de los Frames, cada vez que se actualizan los resultados.
6. **Requisito 6.** Calibrar la formula del sistema de recomendación para que sea lo mas precisa posible.
7. **Requisito 7.** Comprobar el correcto almacenamiento del progreso, y la persistencia de los datos.

### Requisitos no funcionales

- **Requisito 8.** Crear diferentes estilos visuales para la ergonomía con el usuario.
- **Requisito 9.** Mejorar la navegabilidad y velocidad del software.
- **Requisito 10.** Para futuras extensiones el software ha de ser modular

## B.4. Especificación de requisitos

### Requisito 1 - Inicio y Registro

Versión: 1.0

Importancia: Media

Descripción:

Para un correcto uso de la aplicación debían poderse añadir diferentes usuarios e iniciar sesión. Se crearon las condiciones necesarias y el formulario, para la correcta inscripción en el programa, haciendo las comprobaciones pertinentes para respetar siempre la consistencia de la base de datos.

## **Requisito 2 - Añadir Alimento**

Versión: 2.0

Importancia: Alta

Descripción:

Debido a la limitación de la base de datos, era necesaria la posibilidad de añadir nuevos alimentos, para ello se llegaron a crear dos versiones:

versión 1: El usuario debía meter la información completa del menú además de añadir él, la calidad que viese pertinente. Resultaba un trabajo arduo para el usuario y una posibilidad de romper la funcionalidad del programa haciendo que cualquier usuario pudiera poner cualquier comida con una buena calidad.

version 2: El usuario inserta el menú por cada alimento y su información nutricional extraída de Bedca, la cual, esta valorada en 100 gramos. Calcula automáticamente el valor del menú y la calidad de dicho menú.

## **Requisito 3 - Correcta distribución**

Versión: 1.0

Importancia: Baja

Descripción:

Cada patología viene inscrita con un tipo de dieta. A la hora de realizar los cálculos pertinentes en cuanto al reparto calórico ,el tipo de dieta es vital. . Esto se resolvió mediante clausulas condicionales que permitía crear esa variedad dietética.

## **Requisito 4 y 6 - Sistema de recomendación: Pruebas y calibre**

Versión: 3.0

Importancia: Alta

Descripción:

El algoritmo de recomendación es el pilar de la aplicación.

Primero para su versión inicial, se ideo un sistema basado en recomendar solo a través del alimento menos ingerido (Para crear variedad) y la calidad,

lo que provoca un desequilibrio nutricional grave, que había que evitar. En la segunda versión, se añadió el atributo "dif" que hacía referencia a la diferencia calórica con la necesidad del usuario. Esto provocaba un menor desajuste, pero no tenía en cuenta los macronutrientes, no teniendo en cuenta el tipo de dieta del usuario y creando un importante desajuste nutricional. Versión 3: Se corresponde con la formula actual, la cual permite que el alimento se ajuste lo máximo posible a las necesidades del usuario basandose en las características que este busca en ese momento y en las características del menú recomendado.

### **Requisito 5 - Actualización de los Frames**

Versión: 1.0

Importancia: Alta

Descripción:

Para que el sistema de recomendación fuera lo más preciso posible, había que asegurarse que con cada elección, la recomendación pudiera ser diferente, ajustándose de esta manera más al usuario y dando una experiencia mas cercana a la personalización. De esta manera cada vez que se selecciona un alimento, el programa cambia todas las recomendaciones posibles.

Debido a las limitaciones puestas por la interfaz gráfica. Para lograr la correcta actualización de la información, cada vez que se hace una elección, se reconstruye desde cero de nuevo el Frame.

### **Requisito 7 - Persistencia**

Versión: 1.0

Importancia: Media

Descripción:

Se ha de guardar el progreso actual cada vez que el usuario cierre la aplicación. Para ello se crea un función encargada de guardar la base de datos y los mismos elementos de esta.

Se realizaron varias pruebas para ver el correcto almacenamiento y carga de los datos. Nacieron problemas de la necesidad de guardar un solo elemento de la base de datos. Para ello se separó la base de datos en distintos archivos ".xlsx", consiguiendo así la persistencia que se buscaba.



## *Apéndice C*

---

# Especificación de diseño

---

### C.1. Introducción

En este capítulo veremos todo lo relacionado con el diseño de la herramienta.

### C.2. Diseño de datos

El diseño tanto visual como de los datos, se decidió antes de empezar a realizar el proyecto. Se basa en:

- Diseño visual organizado en Frames y ordenado por paquetes (Packs de Tkinter [4]). Los packs permite subdividir cada pantalla en pequeñas proporciones y estas en sub-proporciones y así sucesivamente. .
- Alimentos tratados según el estándar de medida nutricional, por cada 100 gramos [2].
- Bases de datos en archivos ".xlsx"
- Manejo interno de los datos a través de DataFrames

### C.3. Diseño procedimental

#### Introducción

Se ha definido la estructura del sistema, y a continuación se definirá la forma de funcionamiento de las grandes acciones realizadas por la aplicación

a pro de la utilidad de dicha. Todos los diseños han sido realizados con la herramienta "Draw.io"[10]

### **Alta o modificación de usuarios**

La alta y modificación de usuarios parten de la misma estructura del funcionamiento variando simplemente en un pequeño paso que será indicado en su momento. Antes de comenzar con el análisis detallado se mostrará a grandes rasgos, la estructura principal externa de este proceso:

1. El usuario rellena sus datos y valida
2. El programa comprueba que todo es correcto. Si es correcto lo añade, sino lanza un error.

Así es como funcionaría el algoritmo paso a paso:

- Inicio
- Solicitar los datos a rellenar
- Enviar dichos campos al modulo de administración (AdminBase) para su comprobación.
- Si algún dato es erróneo se muestra el mensaje de error, sino sigue su ejecución.
- Si es alta de usuario añade información del nuevo usuario a la base de datos.
- Si es Modificación de datos Busca al usuario existente y remplacea.

Se muestra el diagrama de interacción de los diferentes objetos.

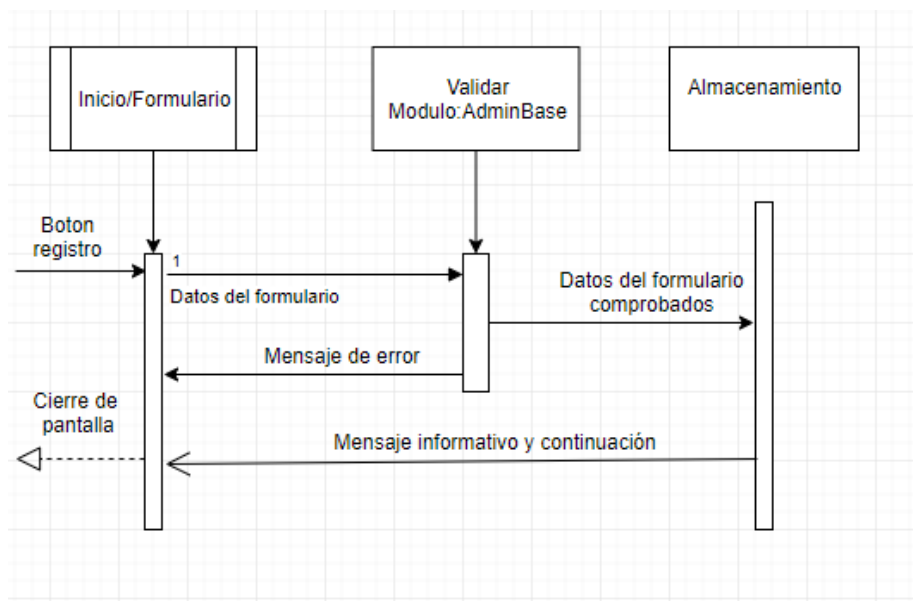


Figura C.1: Diagrama UML sobre la interacción en el registro o modificación de un usuario

## Elegir Alimento

En este caso, es algo más complicado que lo anterior, pues hay varias vertientes. Es posible refrescar hasta encontrar el alimento buscado, y cuando es seleccionado se actualizan el resto de Frames.

- El Usuario refresca la página hasta encontrar su elección.
- El usuario elige y se almacena en su registro diario esa elección
- El programa actualiza automáticamente el resto de Frames.

A continuación se detalla cada paso del procedimiento explicado anteriormente:

1. El Usuario se incorpora al Frame de selección.
2. El Usuario decide si escoger una opción o buscar otra.
  - Si escoge una opción se almacenan los datos, y se actualizan el resto de posibles elecciones.

- Si refresca la página en busca de otros alimentos, se suma los valores de uso a esas comidas y se recomiendan unas nuevas.
3. Sigue el programa, hasta que el usuario salga de la ventana, y se mantienen las elecciones registradas

Diagrama UML de la interacción del programa en la selección:

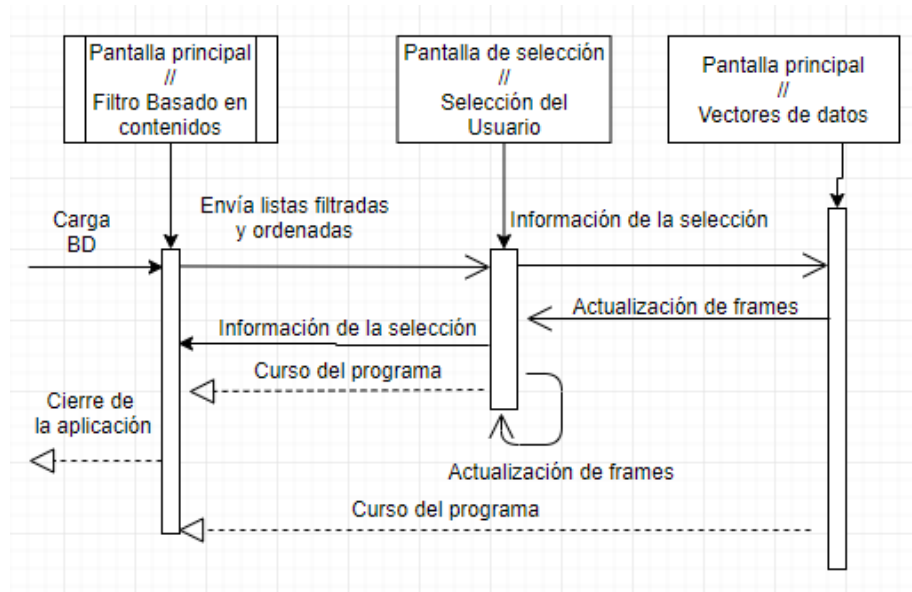


Figura C.2: Diagrama UML que muestra la interacción entre frames y funciones durante la selección

A continuación se visualizará un diagrama de flujo similar al anterior pero sobre el refrescar de las opciones y con otro método diferente a UML:



Figura C.3: Diagrama de flujo de la acción de refrescar opciones de menú

## Carga Y almacenamiento bases de datos

Teniendo en cuenta la similitud entre el funcionamiento de la carga y almacenamiento de datos de las diferentes colecciones, se pasa a describir un modelo general que engloba a todos ellos:

- El usuario inicia la aplicación
- Se carga automáticamente los datos
  - Se abre los diferentes Excel que simulan la base de datos.
  - Se traducen a DataFrame con el nombre de columna como el nombre del objeto de la colección
- Se tratan los arrays por separado evitando romper la consistencia de la base de datos.
- Cuando el usuario guarda los datos, se sustituye la base de datos por el DataFrame actual.

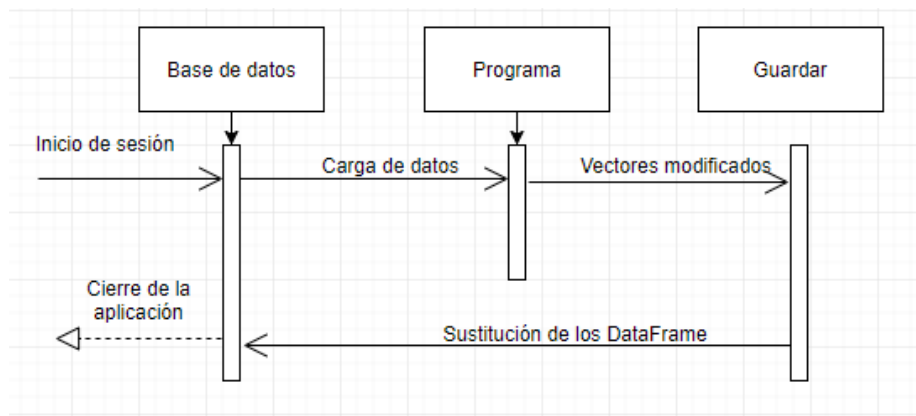


Figura C.4: Diagrama UML de la interacción básica del programa con la base de datos.

## Estructura general del programa

Diagrama del funcionamiento e interacción de la estructura del programa completo.

- Se crea la ventana de Inicio.

- El usuario inicia sesión o se registra
  - Registra: Añade toda la información si es correcta se le añade a la base de datos, sino se le muestra un mensaje de error.
  - Inicia Sesión:
    - Se le muestra la pantalla principal con tres vertientes.
      - ◇ Información de Usuario
      - ◇ Dieta
      - ◇ Historial
    - El usuario hace el correcto uso de la aplicación
    - Se guardan los diferentes cambios a voluntad del usuario
    - Se cierra la aplicación.

A continuación se mostrará un diagrama UML básico de la interacción del usuario con el programa en caso de haber iniciado sesión.

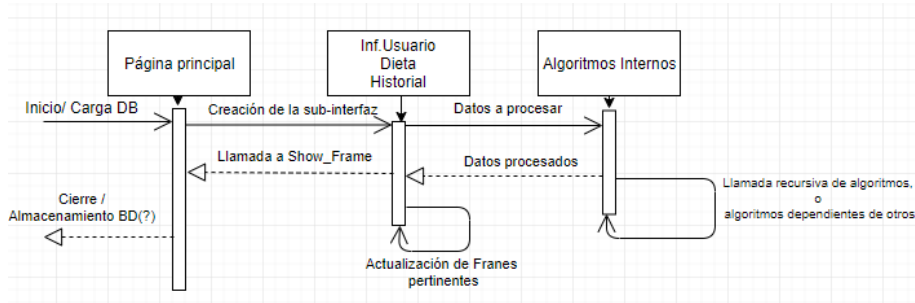


Figura C.5: Diagrama UML general del programa, una vez se ha iniciado sesión.

## C.4. Diseño arquitectónico

El diseño arquitectónico lo separaremos en dos partes principales para mejorar su entendimiento. La idea principal es separar la arquitectura visual entre Frames, el cual contiene un sistema de herencia en la navegabilidad; y la estructura modular, con las diferentes funciones internas.

## Arquitectura visual

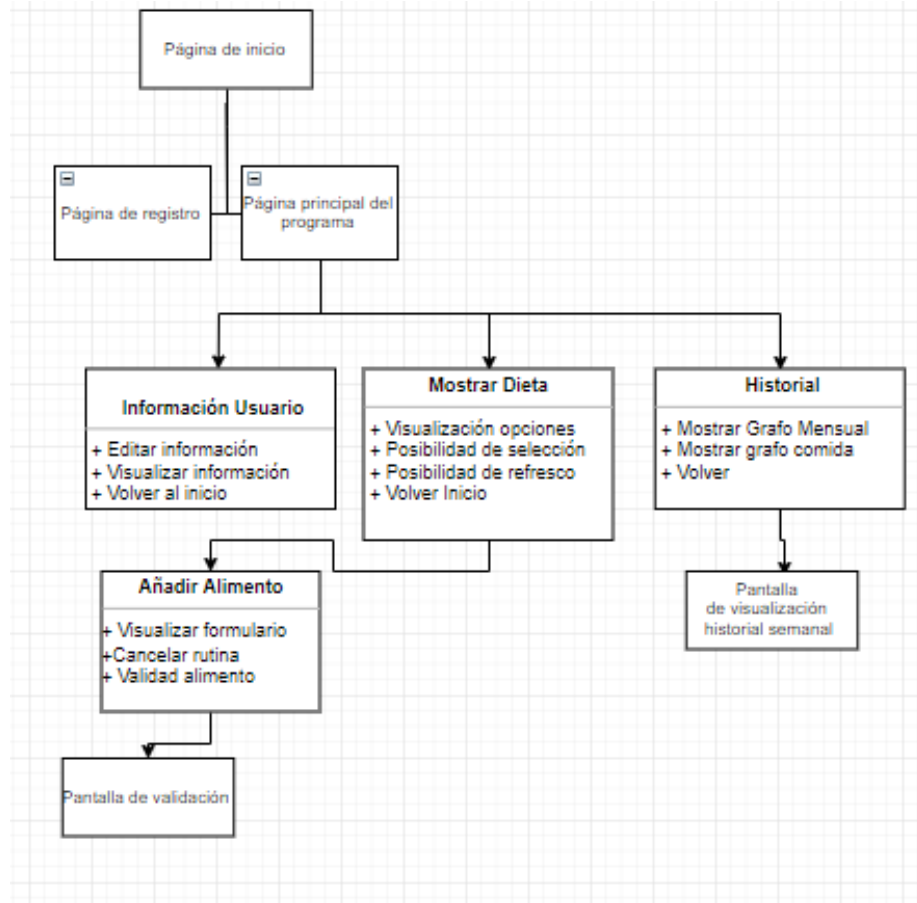


Figura C.6: Imagen de la consiguiente arquitectura visual para la correcta navegación

## Arquitectura Modular

Para esta arquitectura se tendrá en cuenta dos diagramas principales, dado la complejidad del Main, se realizará un diagrama general y otro particular para el interior del Main. Debido al tamaño de cada modulo, y para una visualización más clara primero se visualizará el diagrama de conexiones y luego el diagrama de clases de cada módulo.



**Diagrama de módulos**

Antes de ver la imagen de relaciones entre módulos, aclarar que cuando la flecha sale de un modulo al siguiente es que ese modulo hace uso del otro, pero no viceversa (Excepto en flechas bidireccionales).

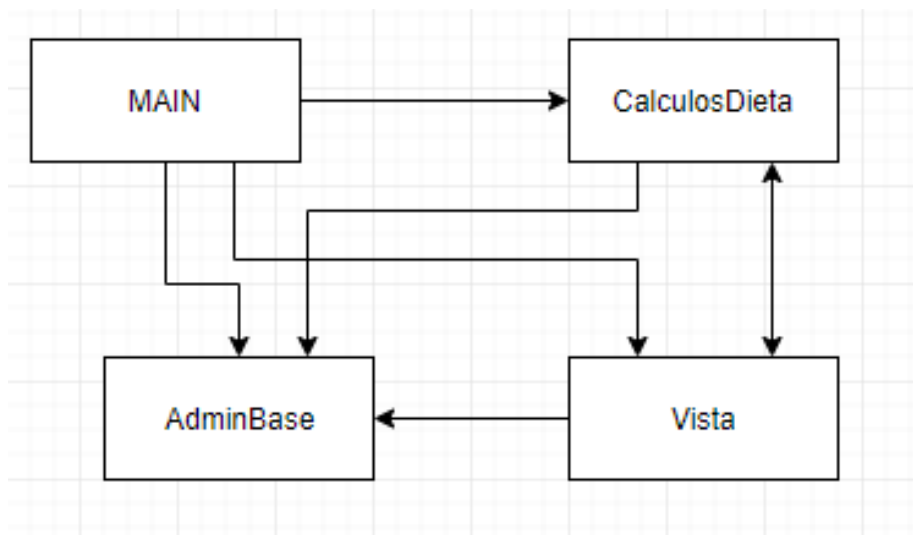


Figura C.7: Diagrama de las relaciones generales entre los distintos módulos

## Vista

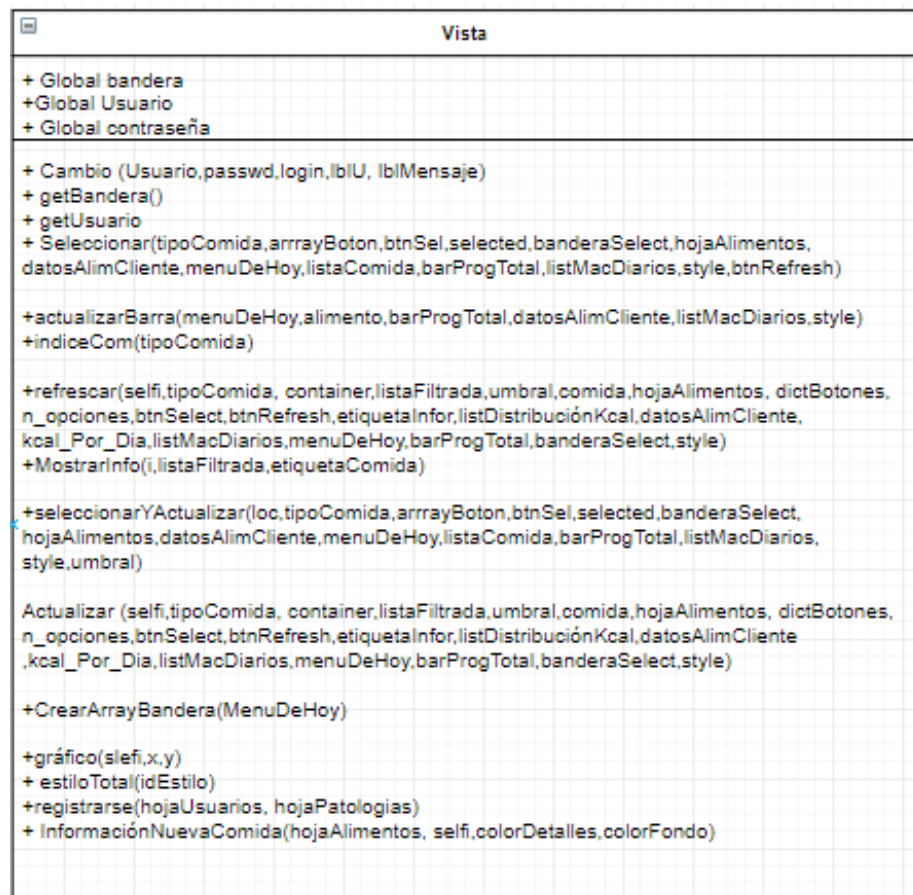


Figura C.8: Diagrama de clases del módulo/clase vista.

## AdminBase

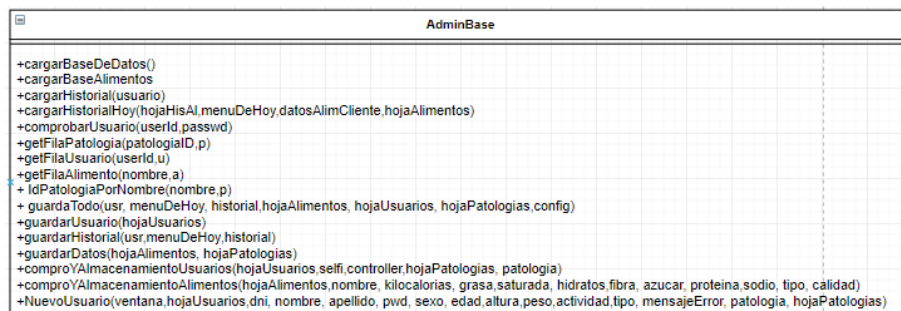


Figura C.9: Diagrama de clases del módulo/clase AdminBase.

## CalculosDieta

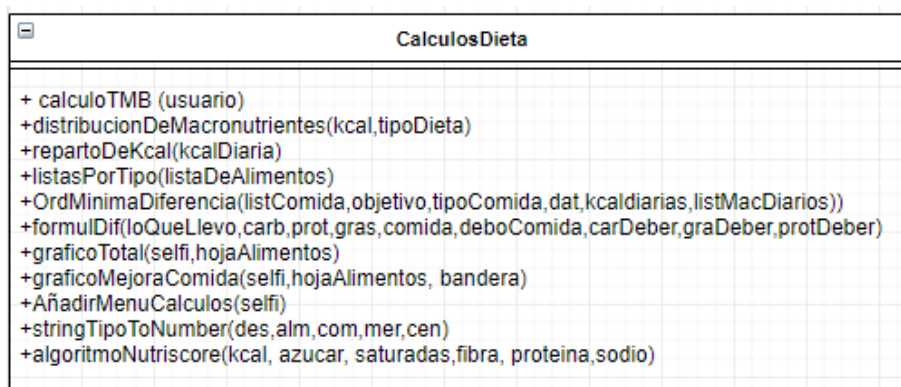


Figura C.10: Diagrama de clases del módulo/clase CalculosDieta.

### Diagrama Interno del Main

Debido a la mecánica de la librería Tkinter, fue necesario crear cada Frame, dentro de la misma clase, para que esto creara el flujo que principalmente se usa durante el programa. Por ello hay una jerarquía propia de clases dentro del propio Main.

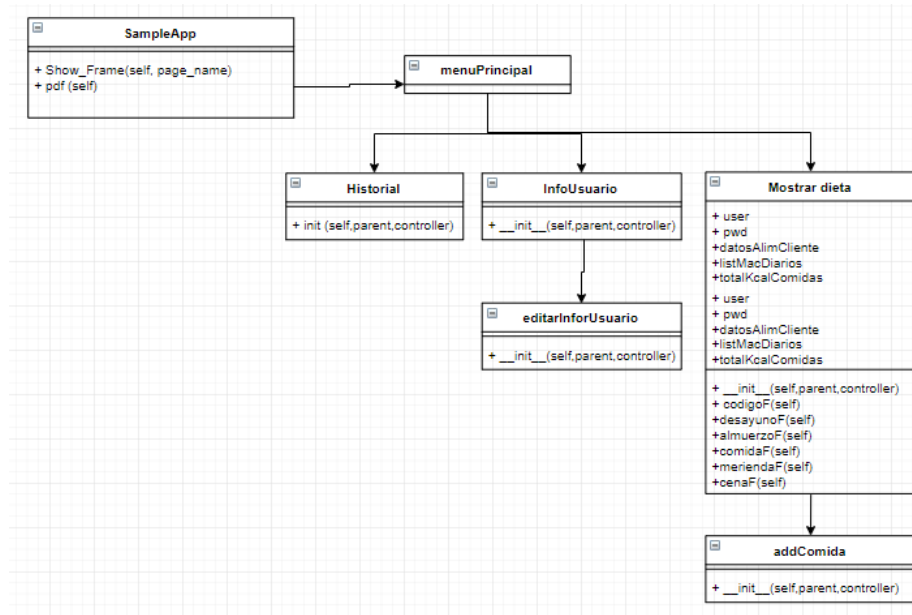


Figura C.11: Diagrama interno del Main

## *Apéndice D*

---

# Documentación técnica de programación

---

### D.1. Introducción

En el siguiente apartado se detallaran, requisitos, herramientas, pautas... Para trabajar con este proyecto.

El proyecto se puede descargar desde el: [Repositorio GitHub](#)

### D.2. Estructura de directorios

Los directorios siguen la siguiente estructura:

- Carpeta principal / Inicial
  - Directorio: assets
    - Manual.pdf
    - Logotipo.PNG
    - logo.ico
    - A.png
    - B.png
    - C.png
    - D.png
    - E.png
    - caraRoja.png

## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

- caraVerde.png
- Dorectprio: Dat
  - BaseDeDatosDeAlimentos.xlsx
  - BaseDeDatosUsuarios.xlsx
  - Historial.xlsx
  - config.txt
  - RegistroHistorial
- Directorio: Memorias
  - Anexos.pdf
  - Memorias.pdf
  - Directorio: img (Almacenamiento de imágenes)
  - Directorio: LatEx (Memorias y anexos en LatEx.
- Directorio: Poster (Poster del programa).
- Directorio: ForDevelopments.
  - requirements.txt
  - Intructions.txt
- AdminBase.py
- Main.py
- CalculosDieta.py
- Vista.py

### D.3. Manual del programador

A continuación veremos una pequeña guía para preparar el entorno de programación.

#### **Python**

El lenguaje usado durante este proyecto es Python, en su versión 3.6 (También funcional para 3.7), para ello tendremos que descargar e instalar el interprete de Python. Lo podremos hacer desde el siguiente enlace [Python 3.6.8](#). Desde la misma página podremos descargar si lo deseamos la versión 3.7.

A continuación instalaremos Anaconda (No es estrictamente necesario, pero es el sistema usado para el desarrollo del proyecto), el cual nos dará una serie de funciones y programas, además de una powershell propia, muy útiles. Link: [Anaconda](#). Con esto, se nos instalará automáticamente tanto Spyder,

como Notebook, y VisualCode. Cualquiera de estos tres editores son muy potentes y funcionan a la perfección para ejecutar el proyecto (Aunque se aconseja que no usar NoteBook). Esta herramienta además viene con una serie de librerías principales ya instaladas y que ahorran trabajo al programador.

En caso de no instalar anaconda, se debería instalar un editor, para su posterior ejecución. Editores recomendados para Python:

- PyCharm
- VisualCode
- Spyder
- Eclipse con API de Python

Recordar que si se escoge un editor el cual no tenga la opción de ejecutar directamente desde el editor, se deberá hacer a través de la consola de comandos, para ello vaya a la carpeta donde tenga descargado el proyecto, y en la parte superior (Donde aparece la ruta del directorio), escriba `cmd` y se cargará la powershell desde la carpeta actual, acto seguido escriba `HealthApp.py` y el programa se ejecutará para su prueba o test.

Las siguientes librerías son las librerías principales usadas durante el proyecto:

- matplotlib
- numpy
- pandas
- auto-py-to-exe
- webbrowser
- os-win
- Pillow
- functools
- xlrd

## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

### ■ openpyxl

Auto-py-to-exe, es una librería que sirve para crear archivos ejecutables desde un archivo con extensión ".py", sencilla de usar, la cual usa pyinstaller de manera interna, y nos da una interfaz gráfica bastante intuitiva para crear el ejecutable. El resto de librerías son las usadas para que el programa corra con normalidad.

### Auto-py-to-exe

Para usar la aplicación auto-py-to-exe, abriremos la consola de comandos con **cmd**, y escribiremos el comando: **auto-py-to-exe**

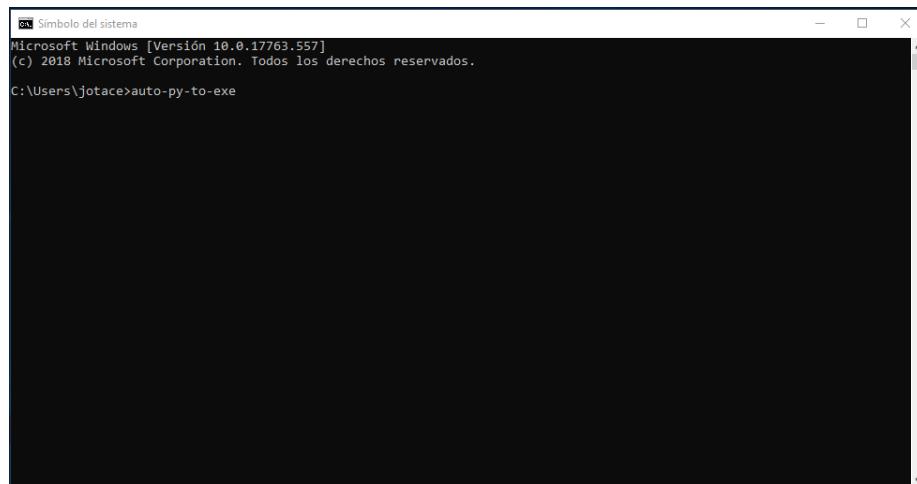


Figura D.1: comando para el uso de auto-py-to-exe

Cuando insertemos ese comando se nos abrirá una ventana como la siguiente:



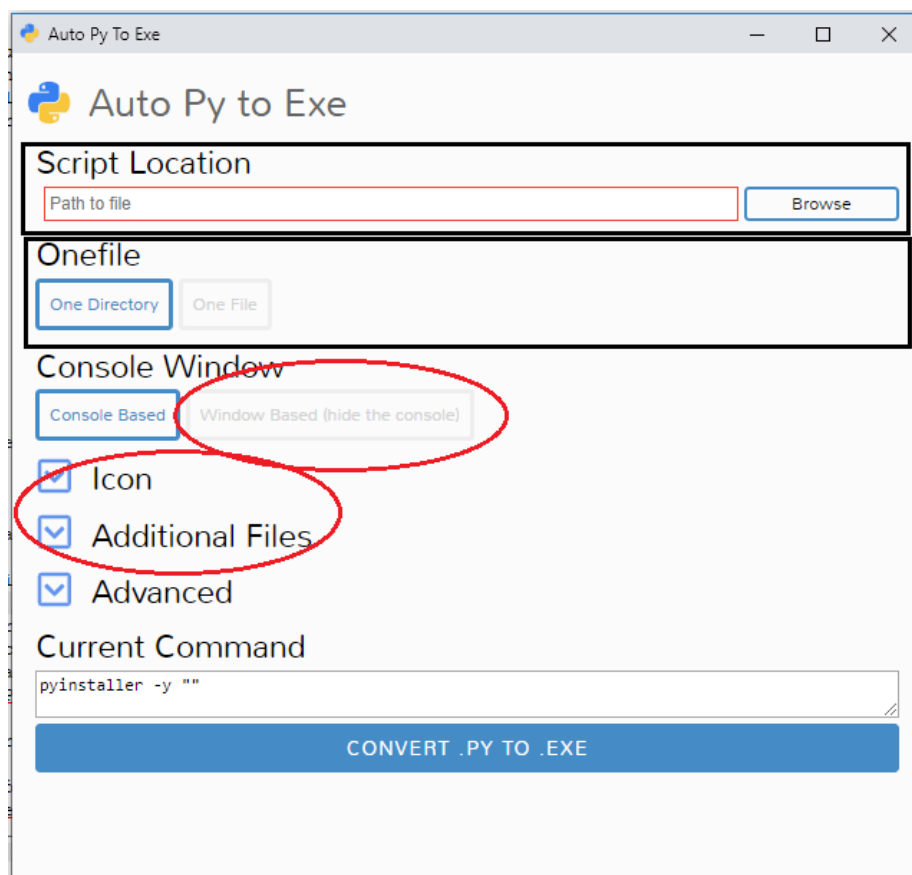


Figura D.2: Pantalla de la aplicación auto-py-to-exe

Para su correcto uso se deberá añadir en **path file** el archivo principal (HealthApp) que ejecuta todo nuestro programa. Para evitar problemas con los archivos extra, se marcará la opción **One Directory**, y más adelante la opción **Windows Based (console hidden)** para evitar que se reproduzca la consola cada vez que lo ejecutemos (Si deseamos depurar el programa se aconseja usar la otra opción).

Para terminar añadimos el icono de la aplicación en **Icon** y en **Additional Files** se elegirá la opción **add folder** y se añadirás las carpetas de `.assets` y `Data`, una vez esto, se pulsará sobre el botón **Convert .py to .exe**. Si no hemos escogido carpeta de salida se hará sobre la carpeta propia del proyecto, si deseamos que se cree en cualquier otra carpeta específica se deberá añadir la ruta dentro de las opciones que se encuentran en **Advanced**

### INSTALACIÓN DE LAS LIBRERÍAS

## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

Para comodidad del desarrollador se dejará preparado el documento: **requirements.txt** donde estarán almacenadas todas las librerías necesarias, si se desea instalar a través de este documento se ha de ejecutar el siguiente comando en la Shell: **pip3 install -r requirements.txt**.

Además existirá la carpeta: **ForDevelopmnet**, donde encontraras dicho archivo (requirements.txt), además de un resumen sobre como trabajar con el proyecto.

En caso de existir algún fallo con alguna librería, instalar manualmente la librería concreta con el comando: **pip3 install NombreLibreria** Una vez tenemos preparado el entorno de Python podemos pasar a la instalación del IDE.

### IDE

En este apartado se hablará de como descargar y preparar el entorno para trabajar, como se realizó durante estos meses.

#### S Spyder

Si se ha realizado la correcta instalación de anaconda, ya tendrá instalado este programa por defecto.

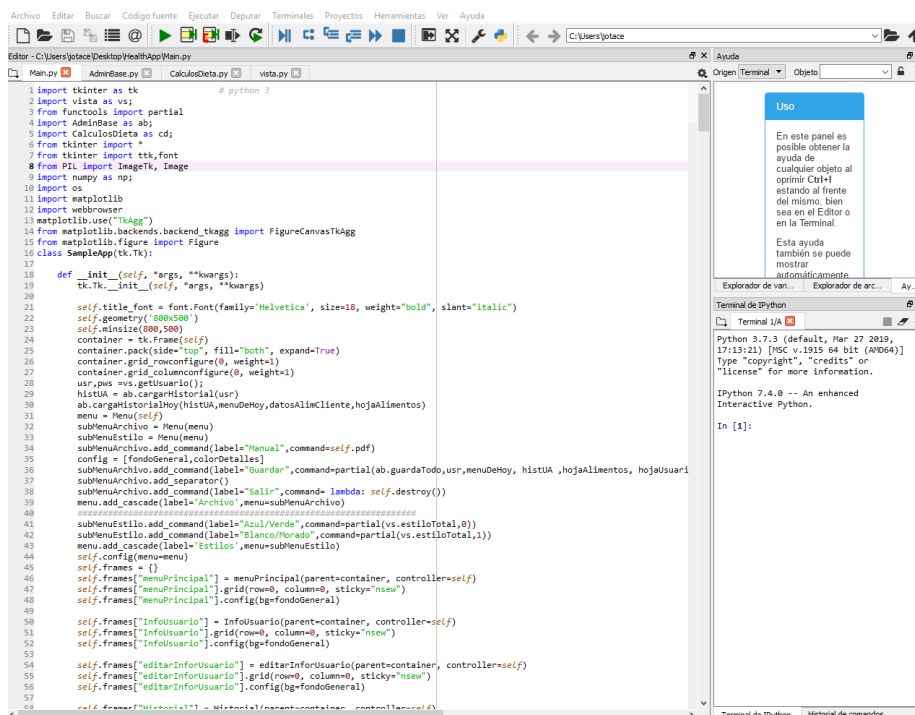


Figura D.3: Interfaz del editor Spyder para Python.

## Git

Sistema de control versiones seleccionado para este proyecto. Windows no lo trae instalado por lo que deberemos descargarlo e instalarlo desde el siguiente enlace: [Git](#)

## GitKraken

Para una mejor gestión, hemos usado la herramienta de escritorio Gitkraken. Si se desea descargar se puede hacer entrando en [GitKraken](#). Descargamos el ejecutable y lo instalamos una vez abierto el programa deberemos ir a: File/Clone Repo. Y añadir la URL del proyecto GitHub. Si ya lo hemos clonado tenemos que dar a la opción: Open Repo y buscar la carpeta donde lo hayamos descargado previamente.

## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

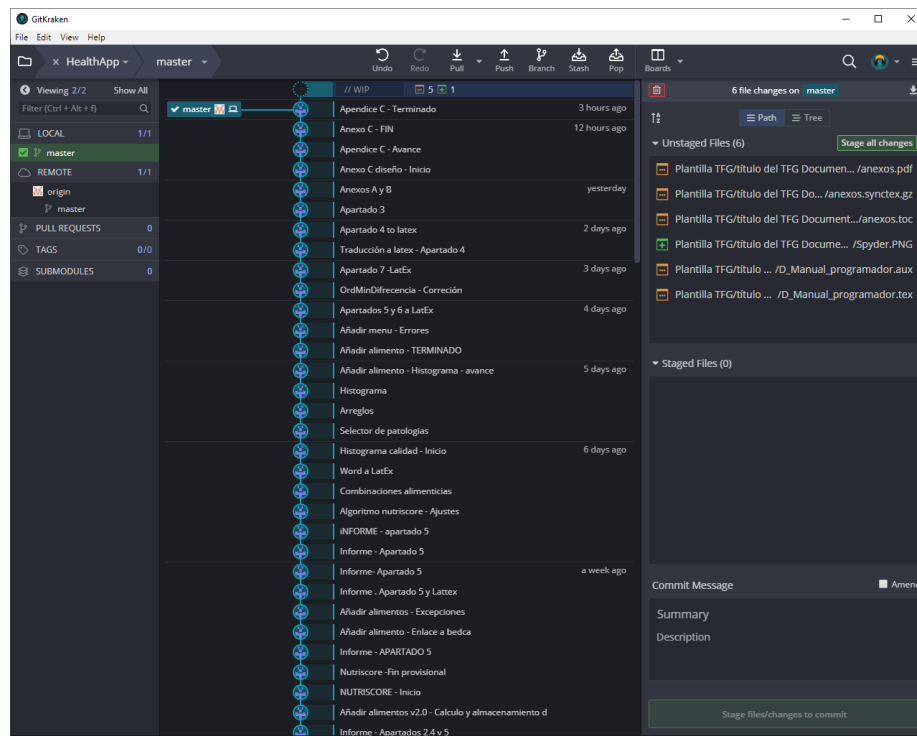


Figura D.4: GitKraken

### D.4. Compilación, instalación y ejecución del proyecto

Lo primero que hay que hacer, es abrir el editor, en el caso de este proyecto Spyder. Una vez abierto el editor debemos abrir los archivos \*.py. Para ello pulsamos en archivo -> Abrir.

No es necesario abrir todos los archivos, basta con abrir el archivo Main (HealthApp) para su ejecución. Pulsamos el botón ejecutar, que se encuentra en la parte superior, como se muestra a continuación:



Figura D.5: Botón ejecutar de Spyder.

Una vez pulsado, se abrirá en otra pantalla generada por Tkinter, el programa principal (Imagen: D.6)

**Terminal**

#### D.4. COMPILACIÓN, INSTALACIÓN Y EJECUCIÓN DEL PROYECTO

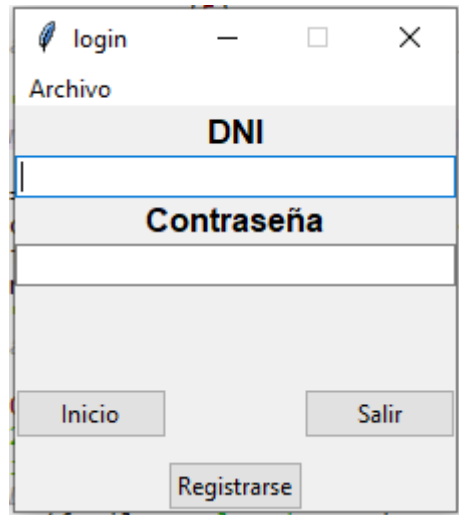


Figura D.6: Pantalla nueva generada por la librería Tkinter (Versión 2.0)

Ante cualquier prueba que se desee realizar. Aparecerá en la interfaz de Spyder, la terminal. Por defecto viene posicionada abajo a la derecha:

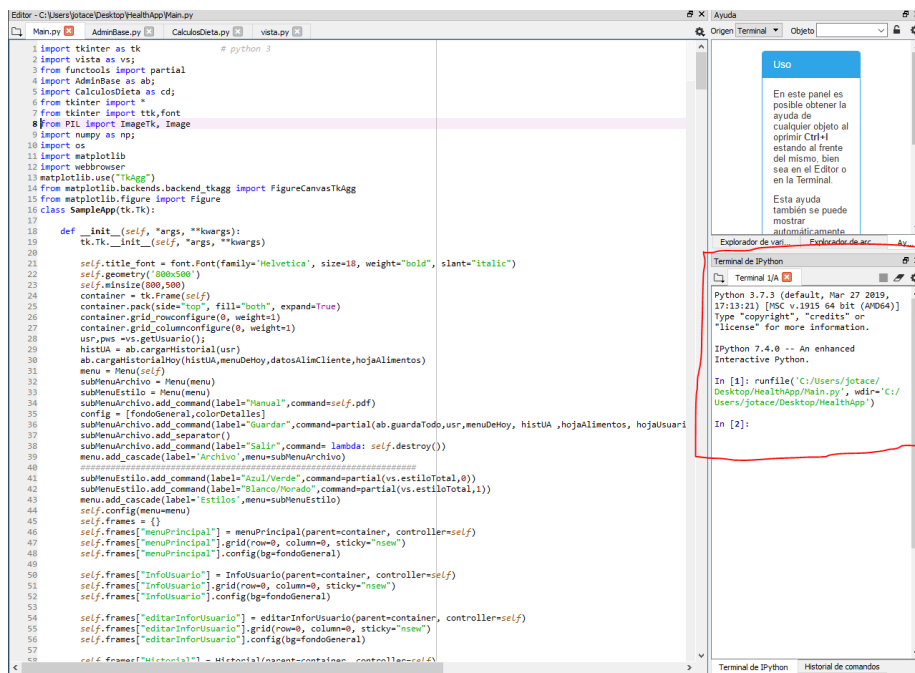


Figura D.7: Señalización de la terminal interna de Spyder

## **D.5. Pruebas del sistema**

Las pruebas del sistema son pruebas de la correcta salida de información, del transcurso entre frames de manera adecuada, de la veracidad de los resultados de los cálculos internos, etcétera.

### **Almacenamiento y carga de los datos**

Consisten en una serie de pruebas donde se han abordado todas las posibles combinaciones de carga y almacenamiento de los datos. Por ejemplo:

- Nuevo Usuario, Nuevo Alimento, Editar Usuario, Hacer selección y refrescar selecciones todo de manera independiente.
- Combinaciones varias entre las opciones anterior, editando un usuario que acabo de crear, añadiendo un alimento y haciendo una selección, editando un usuario y haciendo una seleccion, etcétera. Comprobando acto seguido que se había guardado correctamente en la base de datos.

#### **Problemas encontrados:**

Los DataFrames en ocasiones, se ordenaban alfabéticamente a la hora del almacenamiento provocando una inconsistencia de los datos con el programa. Los datos se guardan como objetos de Python en vez de como valores. Además, se tuvo que eliminar los índices de fila, debido a un problema de compatibilidad en la carga de los datos en otros ordenadores.

### **Navegabilidad**

Se estuvo reiteradamente navegando por la interfaz gráfica haciendo uso de todas las funciones posibles del programa comprobando que este fuera fluido y no diera ningún tipo de problema a la hora de cambiar de Frame o generar nuevas ventanas.

#### **Observaciones:**

Se percibieron pequeños tiempos de espera. Los Frames, son creados al inicio y mantiene su forma durante toda la navegabilidad del programa, haciéndolo una vez creado más rápido, pero dando problemas en cuanto a cambios gráficos se refiriese. Debido a la actualización de Frames por cada selección, se muestra una pequeña ralentización del programa, mientras crea de nuevos los Frames.

## Algoritmos

Se llevaron a cabo las comprobaciones necesarias para ver que el sistema de recomendación y de reparto de datos funcionara correctamente. Para ello se llevó a cabo el siguiente tipo de pruebas:

- Comprobar el Calculo TMB para personas con diferentes capacidades físicas.
- Comprobar las recomendaciones resultantes a una serie de individuos específicos, comprobando todas las opciones recomendadas.
- Comprobar la distribución calórica de todos los tipos de dietas posibles.
- Comprobar la correcta actualización de los datos en cuanto a la selecciones

### Problemas encontrados:

Se encontraron una serie de problemas que fueron corregidos en el apto. El sistema de recomendación fallaba, pues se quedaba con el alimento con menor diferencia, pero, para que se entienda, es tan mala una diferencia de 800, que de -800, para ello se halló el valor absoluto de la formula.

Resulta que se hallaban bien los tipos de dietas pero no eran llamados en ningún momento en el programa, siendo un programa estático. Como solución se añadió este reparto a la formula principal de recomendación.

## Eliminación y apertura de las bases de datos

Se ha llevado a cabo una serie de pruebas respecto a las bases de datos, como la eliminación previa a la ejecución y durante la ejecución de las bases de datos.

- Se prueba a eliminar las bases de datos y el fichero configuración antes de abrir el programa
- Se prueba a cerrar todos los archivos de datos mencionados antes durante la ejecución
- Se prueba a abrir/ocupar las bases previo la ejecución.
- Se prueba a abrir/ocupar los archivos previo a la ejecución.

## ~~4~~APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

### **Problemas encontrados**

Si los archivos necesarios por la aplicación eran eliminados antes de la apertura de la aplicación, la aplicación se bloqueaba impidiendo la continuidad sin avisar al usuario de que podía estar pasando. Si se eliminaba durante la ejecución, mientras no se guardara el progreso, se podía ejecutar sin ningún problema, en caso contrario se bloqueaba sin avisar de que estaba ocurriendo. Si se abren los archivos antes o durante el inicio de la aplicación, es el mismo caso que el anterior, no existe ningún problema hasta que se vaya a guardar el progreso.

Los errores mostrados derivaban de: Permisos denegados, y archivos no encontrados. Para solucionar estos problemas, se crea un tratamiento de excepciones, que recoge la excepción lanzada por el sistema, y la convierte en un cuadro informativo al usuario. De esta forma, se explica al usuario a grandes rasgos, cual es la razón del problema.



## *Apéndice E*

---

# Documentación de usuario

---

### E.1. Introducción

En este apartado se muestra el manual de usuario. En su mayoría las imágenes o capturas, corresponden a la versión 1.0 de la aplicación, puede haber pequeños cambios visuales en la aplicación. En caso de un cambio mayor se habrá actualizado el manual.

### E.2. Requisitos de usuarios

HealthApp no requiere de grandes requisitos. Exclusivamente, tener el ejecutable y un ordenador desde el cual ejecutarlo.

### E.3. Instalación

La aplicación no requiere instalación, es una versión portable. Para su uso, descargue la aplicación del siguiente enlace: ENLACE  
Una vez descargada descomprima el archivo , pulse el botón derecho del ratón y escoja la opción: ".extraer aquí..."

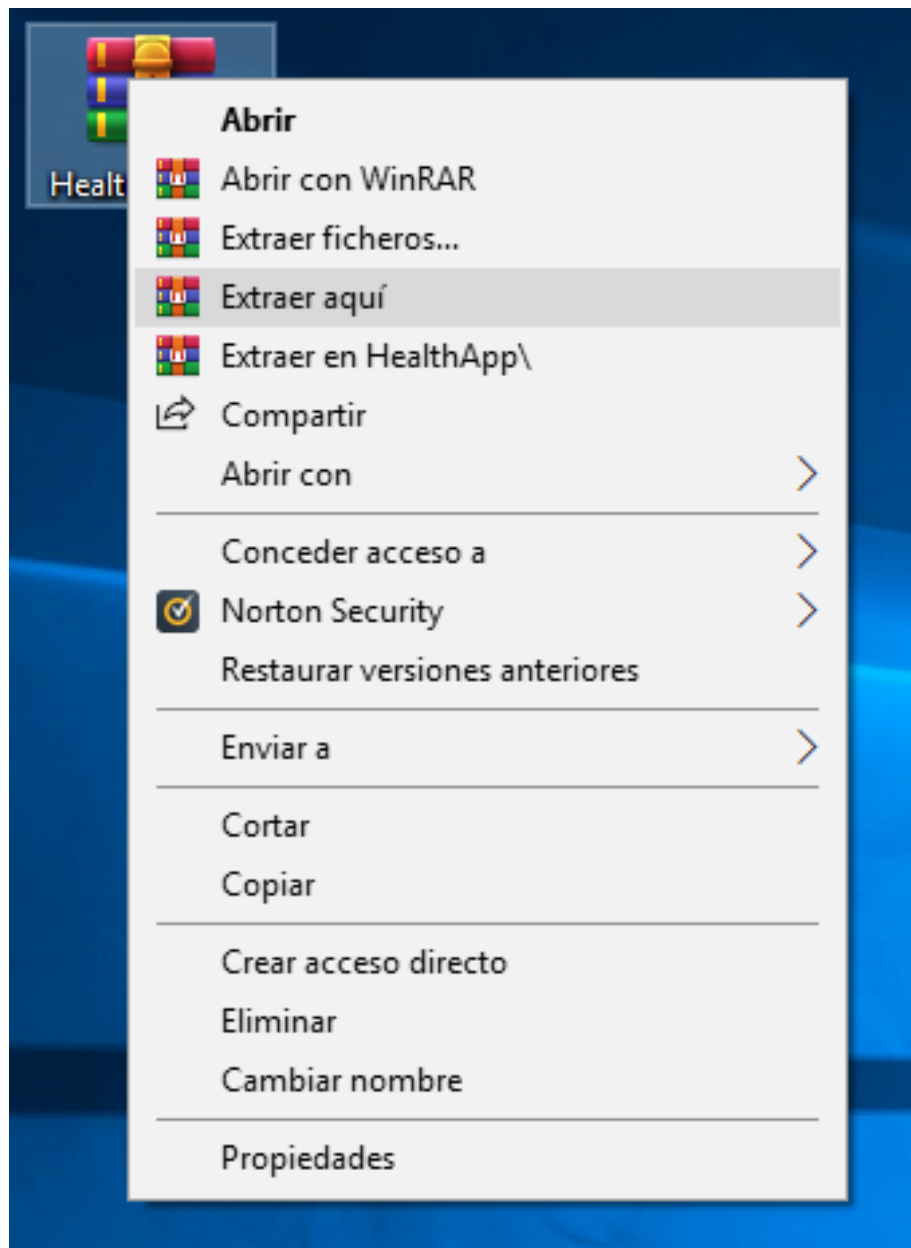


Figura E.1: Descomprimir el archivo ".zip" de la aplicación

Una vez descomprimido iremos a la carpeta del proyecto y buscaremos el archivo **HealthApp.exe**

_types.pyd	19/06/2019 17:42	Python Extension ...	100 KB
_decimal.pyd	19/06/2019 17:42	Python Extension ...	222 KB
_elementtree.pyd	19/06/2019 17:42	Python Extension ...	167 KB
_hashlib.pyd	14/06/2019 23:07	Python Extension ...	32 KB
_lzma.pyd	14/06/2019 23:07	Python Extension ...	181 KB
_multiprocessing.pyd	19/06/2019 17:42	Python Extension ...	25 KB
_queue.pyd	19/06/2019 17:42	Python Extension ...	24 KB
_socket.pyd	14/06/2019 23:07	Python Extension ...	66 KB
_sqlite3.pyd	19/06/2019 17:42	Python Extension ...	66 KB
_ssl.pyd	14/06/2019 23:07	Python Extension ...	102 KB
_testcapi.pyd	19/06/2019 17:42	Python Extension ...	81 KB
_tkinter.pyd	14/06/2019 23:07	Python Extension ...	57 KB
_win32sysloader.pyd	19/06/2019 19:44	Python Extension ...	11 KB
base_library.zip	24/06/2019 17:05	Archivo WinRAR Z...	768 KB
greenlet.cp37-win32.pyd	19/06/2019 17:42	Python Extension ...	22 KB
<b>HealthApp.exe</b>	<b>24/06/2019 17:07</b>	<b>Aplicación</b>	<b>8.668 KB</b>
HealthApp.exe.manifest	24/06/2019 17:07	Archivo MANIFEST	2 KB
kiwisolver.cp37-win32.pyd	19/06/2019 19:37	Python Extension ...	107 KB
libcrypto-1_1.dll	14/06/2019 23:07	Extensión de la apl...	1.866 KB
libopenblas.FN5FF57TWHUYLRG54LA6B3...	19/06/2019 17:42	Extensión de la apl...	27.109 KB
libssl-1_1.dll	14/06/2019 23:07	Extensión de la apl...	397 KB
mfc140u.dll	19/06/2019 19:44	Extensión de la apl...	4.341 KB
MSVCP140.dll	19/06/2019 17:42	Extensión de la apl...	439 KB

Figura E.2: Localización del ejecutable del programa.

Una vez localizado, haremos doble click izquierdo sobre el archivo para iniciar el programa.

## E.4. Manual del usuario

### Inicio

#### Crear un Usuario/Registrarse

Una vez abierto el programa, haga clic en el botón de “**Registrarse**”.

- **DNI:** escriba su número del documento de DNI sin la letra, el cual será su usuario para iniciar sesión posteriormente. Ej: 71257992
- **Nombre:** escriba su nombre solo con letras, sin símbolos o signos.
- **Apellido:** escriba su nombre solo con letras, sin símbolos o signos.
- **Contraseña:** escriba la contraseña que quiere para iniciar sesión en la aplicación, sin límite de caracteres y/o símbolos.

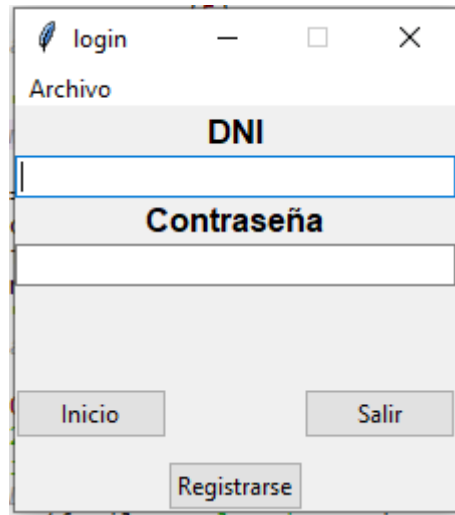


Figura E.3: Pantalla de inicio del programa (Versión 2.0)

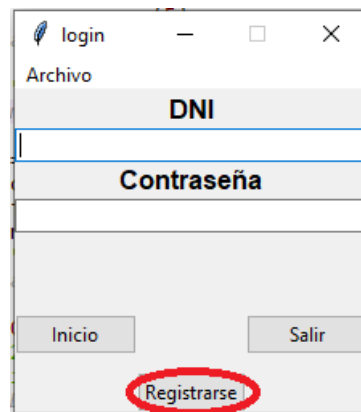


Figura E.4: Pantalla de inicio, botón Registrar (Versión 1.0)

- **Sexo:** marque su sexo haciendo clic encima del correspondiente.
- **Edad:** escriba su edad en números, sin letras o símbolos.
- **Altura:** escriba su altura en centímetros en números, sin letras o símbolos.
- **Peso:** escriba su peso en kilogramos en números, pudiendo incluir comas y decimales, pero no letras. Ej: 65,8
- **Actividad:** marque haciendo clic encima del número correspondiente al nivel de actividad que suele realizar. El significado de cada uno es:

- 1 - poco o ningún ejercicio.
  - 2 - ejercicio ligero (de 1 a 3 días por semana).
  - 3 - ejercicio moderado (de 3 a 5 días por semana).
  - 4 - ejercicio fuerte (6 días por semana).
  - 5 - ejercicio profesional o extremo.
- **Patología:** haga clic en el desplegable y seleccione si sufre alguna de las patologías sugeridas o “sin patología” en el caso de que no sea así.
  - **Tipo:** marque haciendo clic encima del tipo de dieta que quiere, siendo:
    - Bajar - una dieta para bajar de peso de forma saludable.
    - Mantener - una dieta para mantener el peso de forma saludable.
    - Subir - una dieta para subir de peso de forma saludable.

Cuando haya rellenado todos los datos, pulse en el botón de “Aceptar y Guardar” para crear el usuario y guardar todos los datos que ha registrado (si falta algún campo por completar el programa dará error y no le permitirá guardar los cambios). Se abrirá la siguiente ventana con los datos que deberá rellenar:

Registro

Rellene los datos

DNI

Nombre

Apellido

Contraseña

Sexo: ☒ Hombre ☐ Mujer

Edad

Altura

Peso

Actividad: ☐ 1 ☐ 2 ☐ 3 ☐ 4

Patología: [dropdown]

Tipo: ☐ bajar ☐ mantener ☐ subir

Aceptar y Guardar

Cancelar

Figura E.5: Ventana de registro del nuevo usuario señalizada para el registro (Version 1.0)

También puede pulsar el botón de “Cancelar” si quiere salir cancelar el registro, o la “X” de la esquina de arriba a la derecha si desea cerrar la ventana.

Figura E.6: Ventana de registro del nuevo usuario señalizada para el cierre (Version 1.0)

## INICIAR SESIÓN

Figura E.7: Inicio de sesión detallado (Versión 2.0)

Para iniciar sesión en el programa con su usuario, escriba su DNI sin letra en el apartado llamado “DNI” (el cual será su número de usuario) y la contraseña que ha elegido anteriormente en la casilla denominada “Contraseña”. Cuando ya estén los dos campos rellenos con sus datos de usuario, pulse el botón de **”Inicio”** para iniciar el programa. Además, podrá encontrar el Manual de Usuario del programa en todo momento en la pestaña **“Archivo”**

Sino ha creado aún su usuario, acuda al apartado anterior del manual para seguir las instrucciones y registrarse.

Si desea salir o cerrar el programa, pulse el botón de “Salir” o la “X” de la esquina de arriba a la derecha.

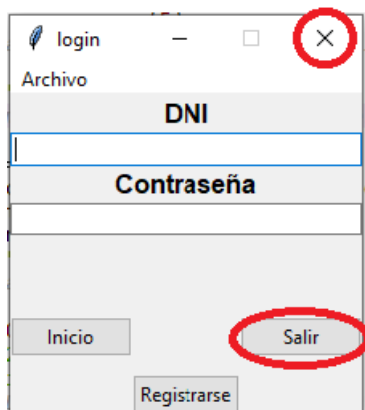


Figura E.8: Imagen detallando el cierre del programa (Version 2.0)

## PÁGINA PRINCIPAL

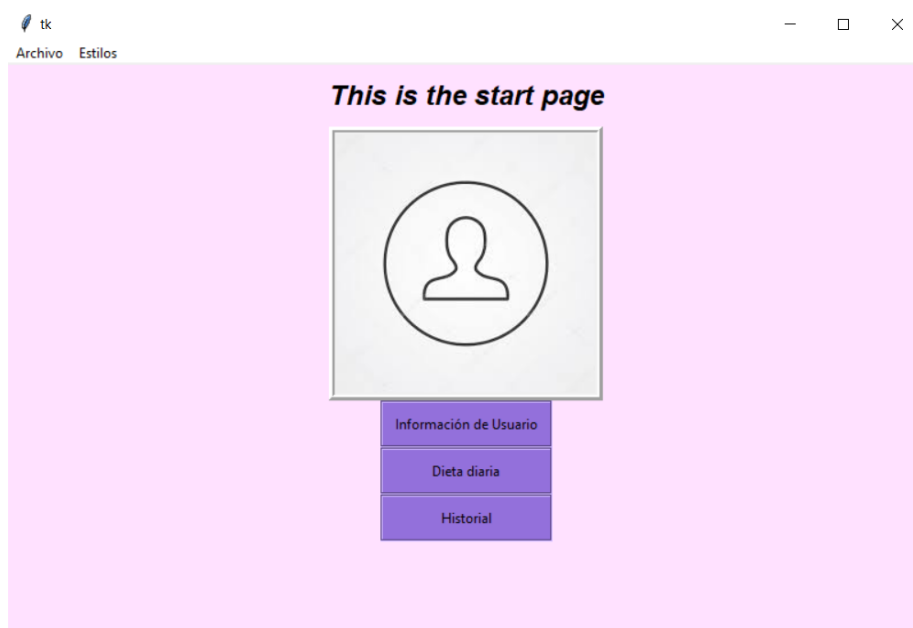


Figura E.9: Página principal de la aplicación (Version 1.0)

## ARCHIVO

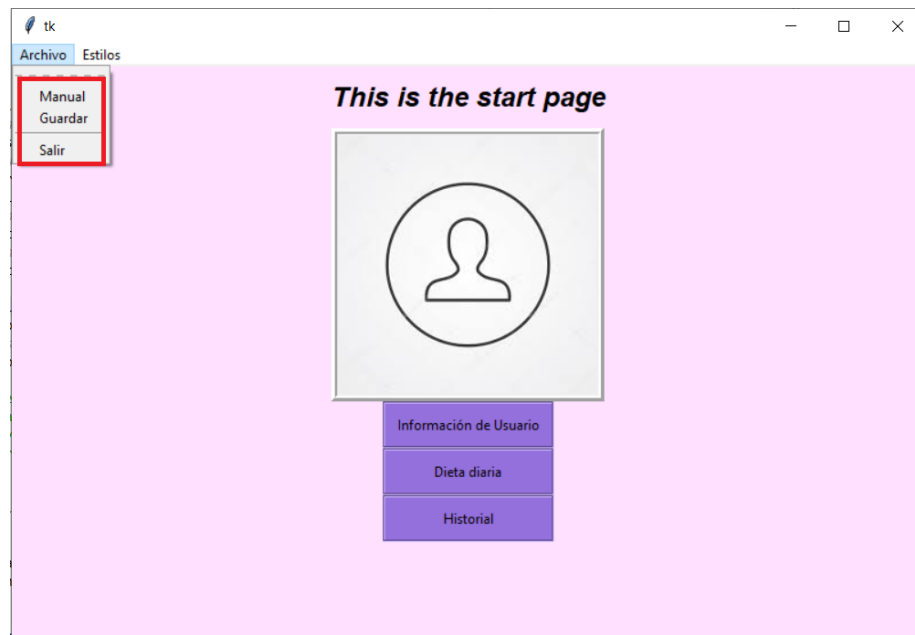


Figura E.10: Pantalla Principal señalizando la opción Archivo (Versión 1.0)

Arriba a la izquierda de la página encontrará el botón “Archivo”, en el que si pulsa se desplegarán otras tres opciones, las cuales se abrirán o ejecutarán haciendo clic sobre ellas.

- **Manual:** contiene el manual de usuario, documento en el que encontrará las instrucciones de cómo utilizar el programa.
- **Guardar:** sirve para guardar la información o los cambios nuevos que haya realizado en el programa o su dieta personal. Recuerde pulsarlo si no quiere perder su progreso o los cambios realizados antes de cerrar el programa.
- **Salir:** sirve para salir y cerrar completamente el programa.

## ESTILOS



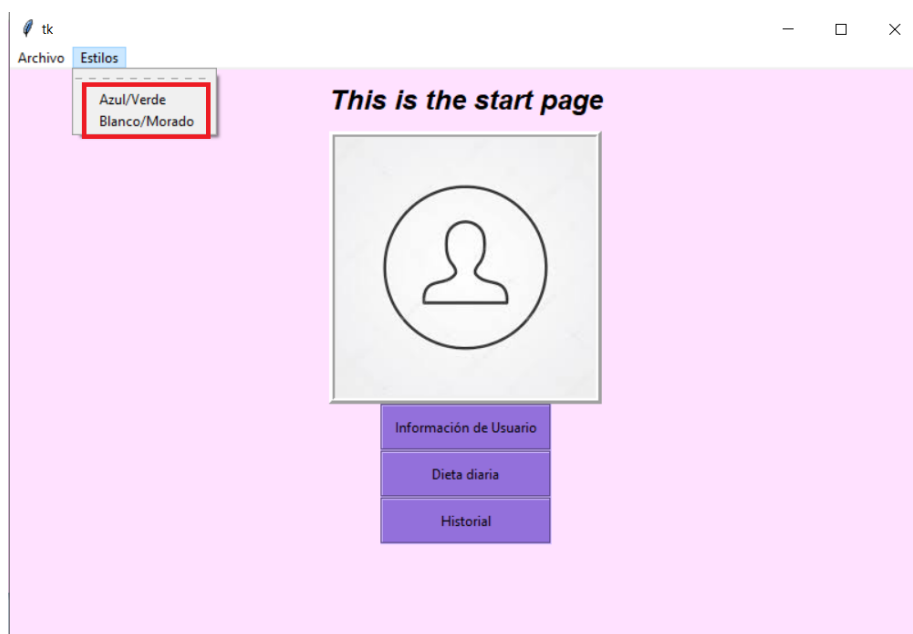


Figura E.11: Pantalla Principal señalizando la opción estilos (Versión 1.0)

En el apartado estilos encontrará un desplegable con las opciones de diseño que tiene el programa, con diferentes combinaciones de colores. Puede elegir el estilo que quiere clicando sobre su favorito.

El estilo se cambia cuando reinicie el programa. Una ventana que aparecerá después de seleccionarlo le informará de ello, pudiendo cerrarla del botón de “Aceptar” o la “X”.

## INFORMACIÓN DE USUARIO

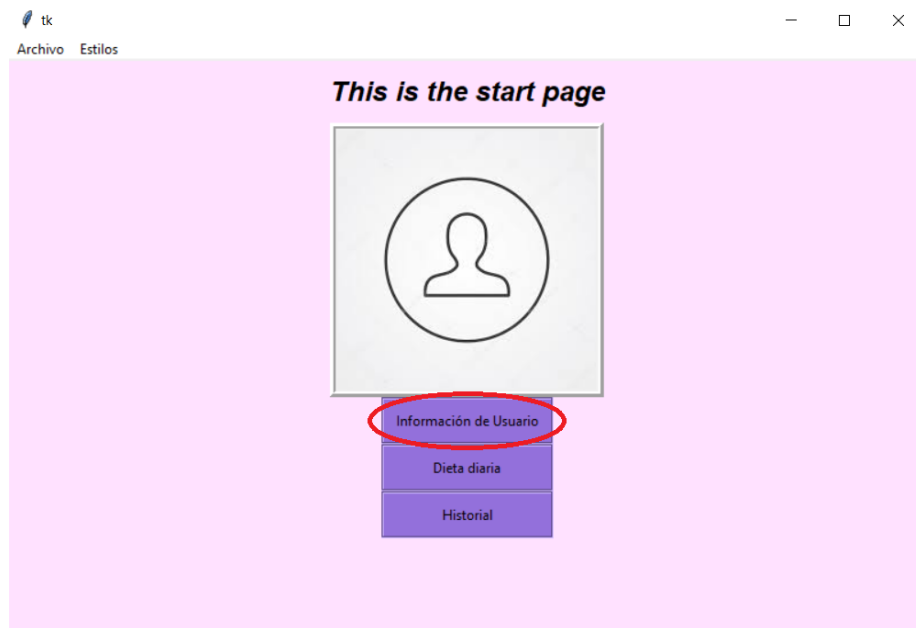


Figura E.12: Botón para pasar e la ventana de la información del usuario (Versión 1.0)

En esta sección podrá consultar y modificar la información personal de su usuario. Si desea modificar sus datos debe pulsar en “Editar Información”, y se abrirá una nueva ventana en la que podrá cambiarlos. Los pasos a seguir son los mismos que durante el procedimiento de registro, ante cualquier duda acuda al punto 1.1. de este manual.

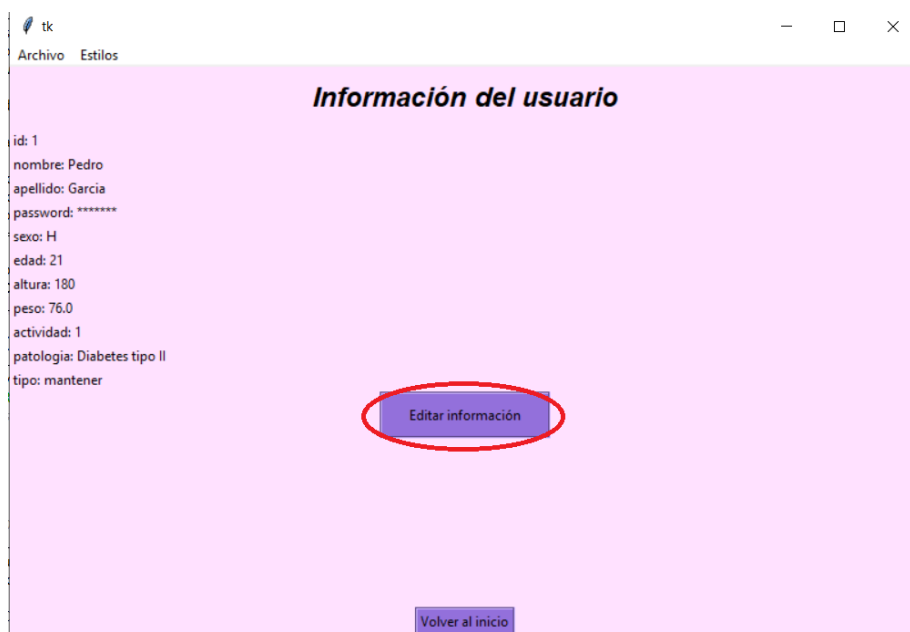


Figura E.13: Botón para editar la información (Versión 1.0)

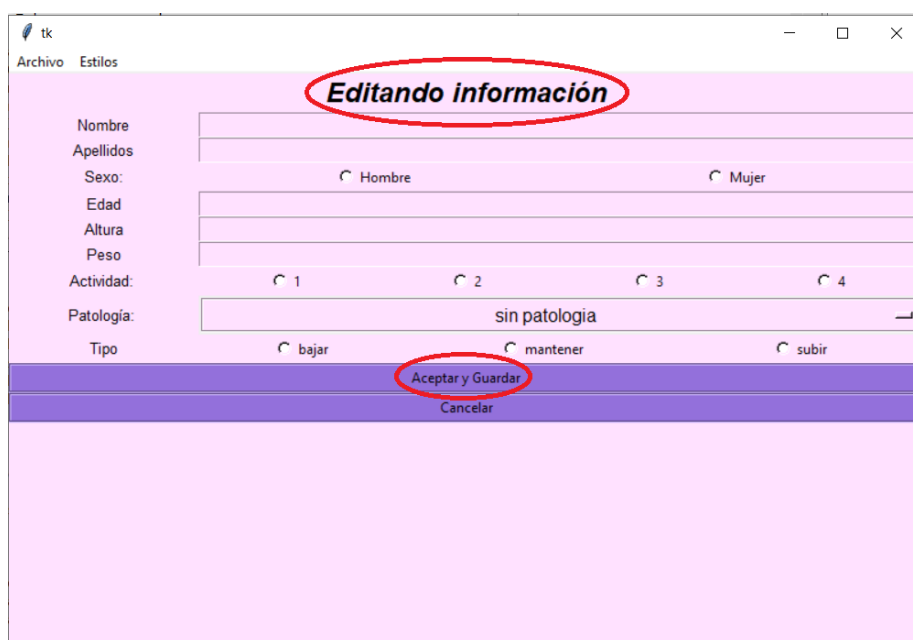


Figura E.14: Pantalla para editar la información del Usuario(Versión 1.0)

Si desea volver a atrás o desechar los cambios realizados para que no

se guarden, Haga Click en la opción de **“Cancelar”**. Y si una vez en su información de usuario quiere volver a la página principal, pulse **“Volver al Inicio”**.

## DIETA DIARIA



Figura E.15: Botón para pasar a la dieta diaria(Versión 1.0)

## COMIDAS DEL DÍA

Una vez haya entrado en la parte de **“Dieta Diaria”**, la ventana cambiará. Arriba aparecerán cinco pestañas con el nombre de cada una de las comidas del día (“Desayuno”, “Almuerzo”, “Comida”, “Merienda” y “Cena”) que podrá abrir pulsando en ellas.



Figura E.16: Pantalla principal de mostrar dieta(Versión 1.0)

Cuando abra cualquiera de las comidas, aparecerán tres posibles opciones de menú, las más saludables para ese momento del día. Si pulsa en ellas, a su derecha se mostrará la descripción de cada plato con sus kilocalorías (“Kcal”), los hidratos de carbono (“Hidratos”), las proteínas, las grasas y la calidad. La calidad se representará con las letras A-B-C-D-E siendo A la opción más saludable y E la peor.

Una vez seleccionado el plato que desea, clique en **“Seleccionar”** para confirmar su elección.

En la parte baja de la pantalla, se irán registrando las cinco comidas que haya ido seleccionando cada día para poder visualizarlas rápidamente.

tk

Archivo Estilos

Desayuno Almuerzo Comida Merienda Cena

### -DESAYUNO-

Objetivo: 2454.0 Kcal // Objetivo Desayuno: 607.0  
Llevo Comido: 1915.0 Kcal

☒ 0) yogurt griego con muesli (439.0Kcal)

☐ 1) leche con avena (381.0Kcal)

☐ 2) leche de almendras con avena (288.0Kcal)

Nombre: yogurt griego con muesli  
Calorias: 439.0  
Grasa: 17.2 (Saturadas: 8.7)  
Hidratos: 71.1 (Azucars: 31.4)  
Proteina: 13.2  
Calidad: 1.0

Seleccionar

Refrescar

Comido hoy:  
desayuno:  
Amuerzo: tostada de atun (50 gr)  
Comida: 150 gramos pure de patata + 100 gramos pollo frito + manzana  
Merienda: manzana  
Cena: 100 GR ARROZ + 200 gr pollo frito + manzana

Volver al inicio Añadir Alimento

Figura E.17: Información del plato y selecciones del día (Version 1.0)

Si el plato que ha seleccionado finalmente no es el que desea, debe pulsar el botón de **“Editar”**. De esta forma, se desmarcará la opción elegida anteriormente y el programa le permitirá elegir de nuevo.

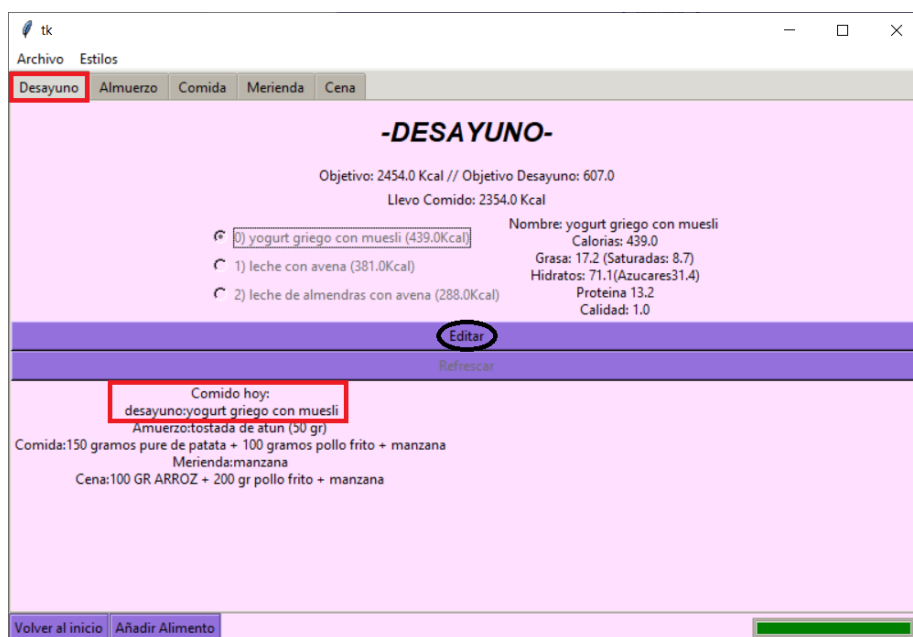


Figura E.18: Selección de comida y guardado de la elección (Version 1.0)

Si desea volver a la página principal, pulse “**Volver al Inicio**”.

## REFRESCAR

La opción de “**Refrescar**” nos permite cambiar las tres opciones de comida sugeridas, por si no fuese de su agrado o no le apeteciese en concreto ninguna de las sugerencias. Pulsando en el botón, aparecerán otras tres diferentes, pudiendo actualizarlas las veces que el usuario desee.

Se debe tener en cuenta que cada vez que se refresque, las nuevas alternativas sugeridas empeorarán en calidad progresivamente, dificultando su objetivo de comer adecuadamente.

En la esquina derecha inferior de la ventana aparece una **barra de progreso**, indicando la calidad de los platos que ha elegido, y por lo tanto lo saludable que ha sido o está siendo su alimentación del día (varía con cada selección). La calidad de sus elecciones estará representada con los colores del semáforo verde-amarillo-naranja-rojo, verde la opción mas adecuada y rojo la peor. Esto le permitirá aprender las comidas que son más saludables

y rectificar si desea mejorar en alguna de ellas.



Figura E.19: Resultados del botón refrescar (Versión 1.0)



Figura E.20: Caliad Nutriscore

## AÑADIR PLATO



tk

Archivo Estilos

Desayuno Almuerzo Comida Merienda Cena

**-DESAYUNO-**

Objetivo: 2454.0 Kcal // Objetivo Desayuno: 607.0

Llevo Comido: 0.0

☒ 0) yogurt griego con muesli (439.0Kcal)  
☐ 1) leche con avena (381.0Kcal) INFORMACIÓN  
☐ 2) leche de soja con copos de maiz (253.0Kcal)

Seleccionar

Refrescar

Comido hoy:  
 desayuno:  
 Almuerzo:  
 Comida:  
 Merienda:  
 Cena:

Volver al inicio Añadir Alimento

Figura E.21: Botón añadir alimento (Versión 1.0)

tk

Archivo Estilos

**Añadiendo Menu**

	Alimento #1	Alimento #2	Alimento #3	Alimento #4
Nombre:				
Gramos:				
A continuación añada la información correspondiente a 100 gramos:				
KiloCalorias				
Grasa				
Saturadas				
Hidratos				
Fibra				
Azucares				
Proteina				
Sodio				

☐ Desayuno
 ☐ Almuerzo
 ☐ Comida
 ☐ Merienda
 ☐ Cena

Se le aconseja la siguiente página web para rellenar la información nutricional: [Bedca](#)

Validar

Cancelar

Figura E.22: Formulario para añadir un nuevo alimento (Versión 2.0)

Si el usuario desea añadir un plato nuevo, debe pulsar en “Añadir Ali-

**mento”** abajo a la izquierda de la ventana. Así, aparece una nueva ventana con hasta cuatro columnas para añadir en cada una de ellas los ingredientes que compongan cada comida. Deberá ir rellenando sus componentes por cada 100 gramos de alimento.

- **Nombre:** solo letras, sin números o símbolos.
- **Gramos:** solo valores numéricos, sin letras o símbolos.
- **Kilocalorías:** solo valores numéricos, sin letras o símbolos.
- **Grasas:** solo valores numéricos, sin letras o símbolos.
- **Grasas saturadas (“Saturadas”):** solo valores numéricos, sin letras o símbolos.
- **Hidratos de carbono (“Hidratos”):** solo valores numéricos, sin letras o símbolos.
- **Fibra:** solo valores numéricos, sin letras o símbolos.
- **Azúcares:** solo valores numéricos, sin letras o símbolos.
- **Proteína:** solo valores numéricos, sin letras o símbolos.
- **Sodio:** solo valores numéricos, sin letras o símbolos.

**Añadiendo Menu**

	Alimento #1	Alimento #2	Alimento #3	Alimento #4
Nombre:	Pollo	Guisantes	Helado	
Gramos:	300	50	70	

A continuación añada la información correspondiente a 100 gramos:

	Alimento #1	Alimento #2	Alimento #3	Alimento #4
KiloCalorias	150	70	230	
Grasa	10	2	13	
Saturadas	2	0	8	
Hidratos	1	1	58	
Fibra	0	0.9	0	
Azucares	0.4	0	57	
Proteína	17	13	7	
Sodio	0.8	0.1	Ejempld	

Tipo

☐ Desayuno
 ☐ Almuerzo
 ☒ Comida
 ☐ Merienda
 ☒ Cena

Se le aconseja la siguiente página web para rellenar la información nutricional: [Bedca](#)

Validar

Cancelar

Figura E.23: Partes del formulario (Versión 2.0)

También deberá elegir el **tipo de comida** que es su plato (desayuno, almuerzo, comida, merienda y/o cena) pulsando sobre ellas, pudiéndose marcar varias opciones si fuese necesario.

Tipo

☐ Desayuno
 ☐ Almuerzo
 ☒ Comida
 ☐ Merienda
 ☒ Cena

Se le aconseja la siguiente página web para rellenar la información nutricional: [Bedca](#)

Figura E.24: Opciones del tipo comida (V.1.0)

Cuando todos los campos mencionados hayan sido completados, pulse **“Validar”** para aceptar o confirmar el plato, o **“Cancelar”** si desea volver a atrás o desechar los cambios realizados para que no se guarden.

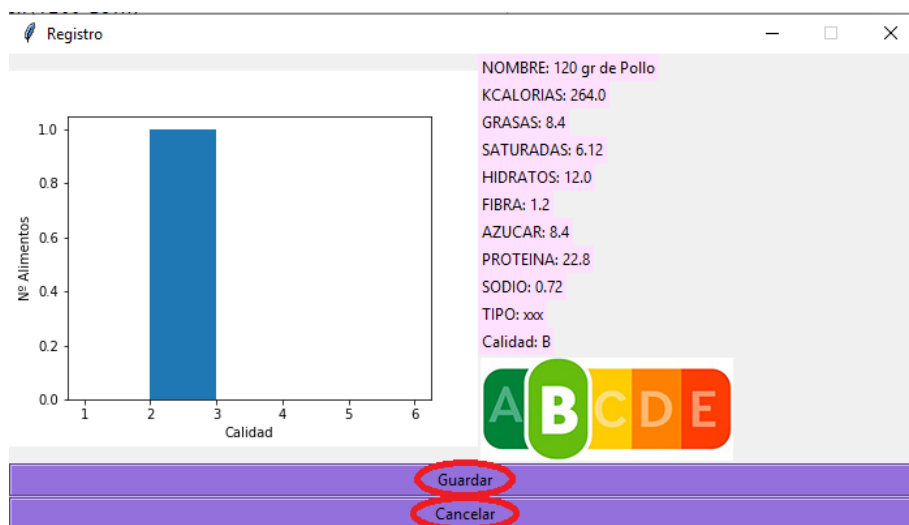


Figura E.25: Pantalla de validación de alimento (Versión 1.2)

Al validar el plato, el programa lo crea y muestra un gráfico, el cual tiene en el eje X u Horizontal la calidad de los alimentos y en el eje Y o Vertical, el número de alimentos que tienen esa calidad, por si el usuario deseara modificar el plato o alguno de sus ingredientes. Para terminar haga clic en **“Guardar”** y el plato quedará registrado definitivamente, o **“Cancelar”** para editarlo o salir porque no desea guardarlo.

Si desea volver a la página de dietas diarias, pulse **“Cancelar”**.

## HISTORIAL



Figura E.26: Botón Historial(Versión 1.0)

En los gráficos que aparecen en este apartado se muestra una línea que representa la calidad de las elecciones que se han realizado en cada comida, clasificándolo de menos saludable (puntos más altos en la representación) a lo más saludable (puntos más bajos en la representación). Los diferentes gráficos que podrá visualizar pulsando en cada uno de ellos serán:

- Gráfico total mensual.
- Gráfico desayuno.
- Gráfico almuerzo.
- Gráfico comida.
- Gráfico merienda.
- Gráfico cena.
- Semana ingerida.

Si desea volver a la página principal, pulse “Volver al Inicio”.

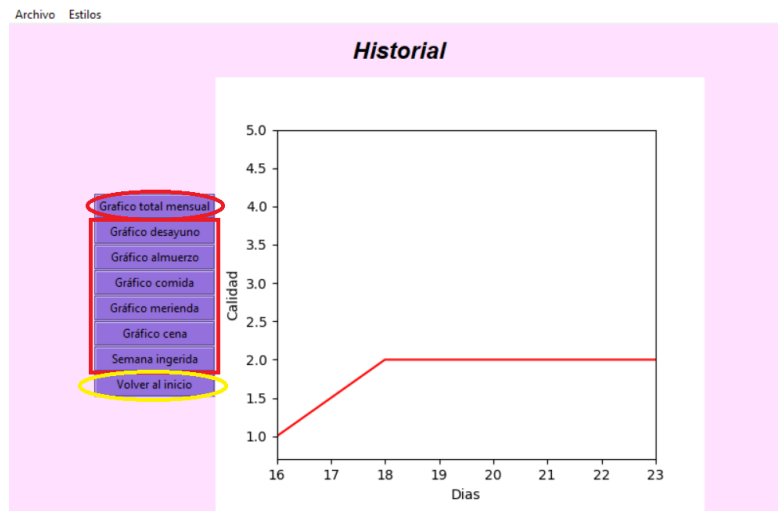


Figura E.27: Pantalla del Historial (V.1.0)

## GRÁFICOS

Existen dos tipos de gráficos:

1. **Gráfico total:** gráfico que muestra la media de la calidad que el usuario toma en un día completo, con todos los alimentos. El eje horizontal o X representa los días, y el eje vertical o Y es la media de la calidad.
2. **Gráfico de desayuno, almuerzo, comida, merienda o cena:** tienen la misma función que el gráfico total, pero de cada comida específica, así podrá observar en cuales come peor o mejor.

## SEMANA INGERIDA

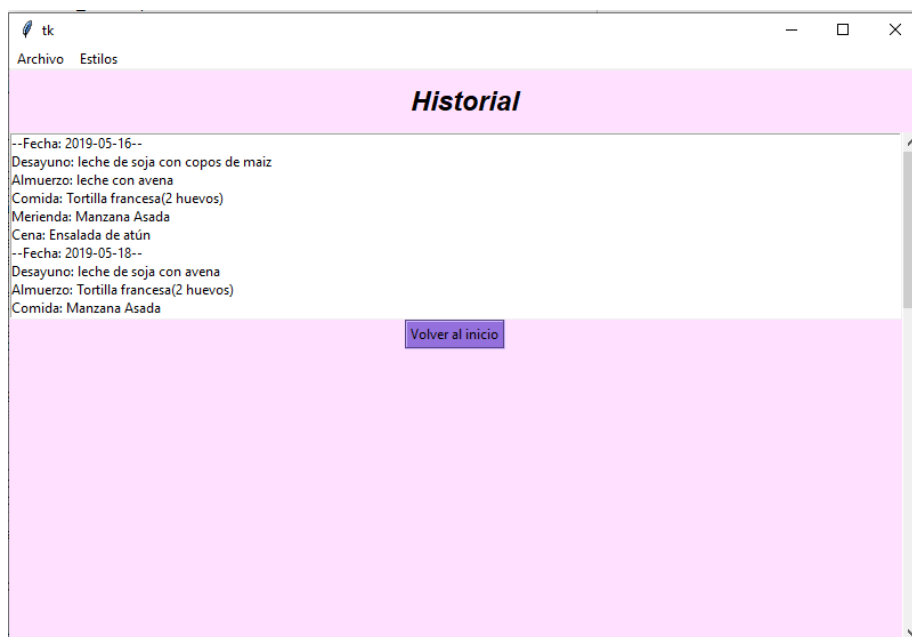


Figura E.28: Pantalla de muestra de la semana ingerida (Versión 1.0)

Cuando se escoge la opción de la semana ingerida, se muestra una lista de lo que el usuario ha comido en los últimos siete días. Sino se escogió nada para esa comida aparece “NaN”.

## E.5. Guía de errores, avisos e informes

A lo largo del uso de la aplicación se pueden encontrar con diferentes ventanas emergentes que informan del estado de la aplicación. Estas ventanas informarán de la situación del programa y del motivo por el que surgen dichas ventanas. Nos podemos encontrar tres tipos de ventanas:

1. **Información** - Ventanas que informan de que se ha guardado correctamente. En la mayoría de ocasiones, se deberá reiniciar el programa para ver los cambios.

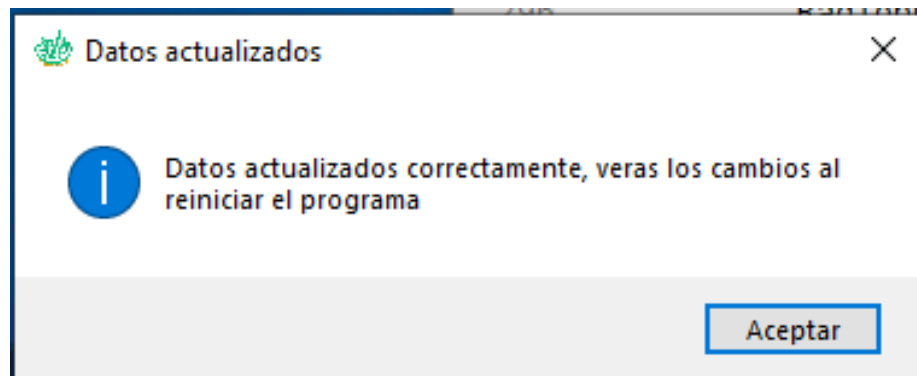


Figura E.29: Ventana emergente de información

2. **Avisos** - Ventana que informa de que no se ha podido llevar a cabo la acción realizada por un fallo menor. Indica cual es el fallo para que se pueda solucionar.

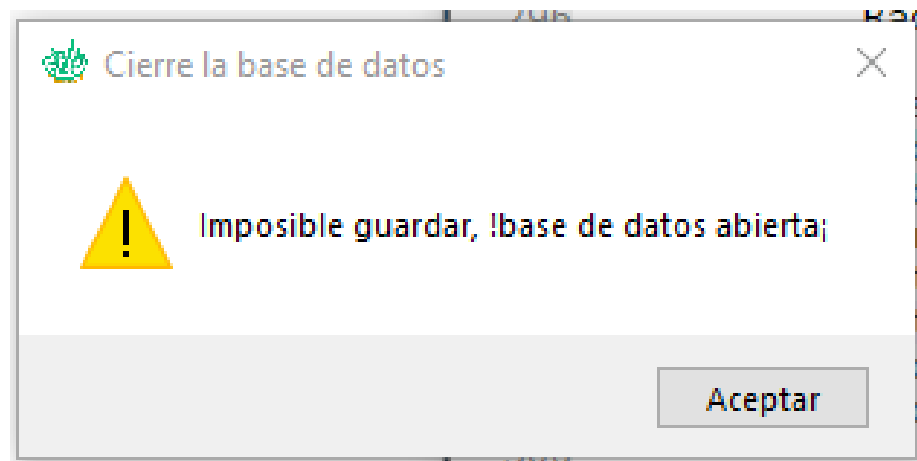


Figura E.30: Ventana emergente de avisos

3. **Error** - Ventana que indica que ha sucedido un error grave, y que le programa a dejado de funcionar. Pruebe a eliminar y reinstalar el programa.



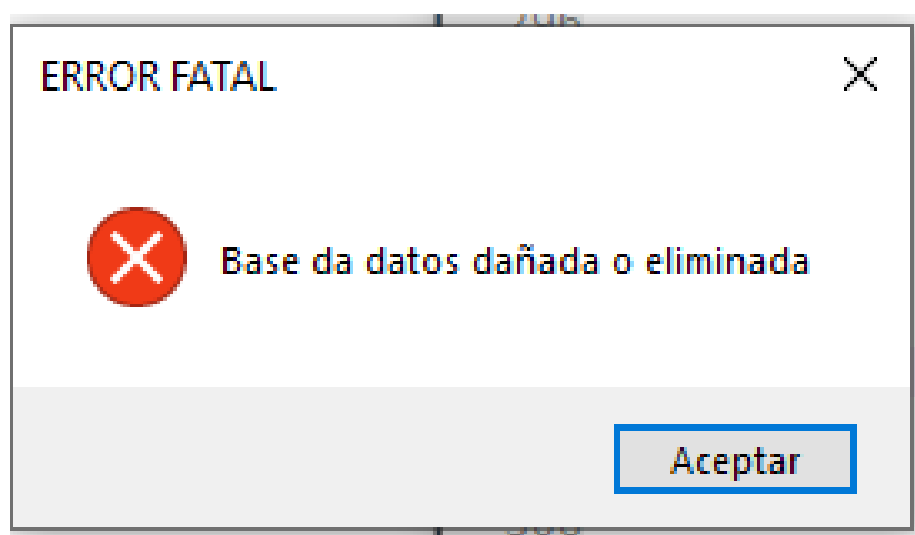


Figura E.31: Ventana de error



---

## Bibliografía

---

- [1] Gestión del ciclo de vida para sistemas de control, 2016. [Internet; consultado 25-junio-2019] [https://www.rockwellautomation.com/es\\_AR/news/blog/detail.page?pagetitle=Vida-%C3%BAtil-del-sistema-de-control-%7C-Blog&content\\_type=blog&docid=30cf16357e57e35dfa880691ddd873c6](https://www.rockwellautomation.com/es_AR/news/blog/detail.page?pagetitle=Vida-%C3%BAtil-del-sistema-de-control-%7C-Blog&content_type=blog&docid=30cf16357e57e35dfa880691ddd873c6).
- [2] Valores nutricionales – reglamento (ue) nº 1169/2011, 2016. [Internet; consultado 29-junio-2019] <https://www.labelfood.es/normativas-de-valores-nutricionales/>.
- [3] ¿qué es scrum?, 2018. [Internet; consultado 29-junio-2019] <https://www.scrum.org/resources/blog/que-es-scrum>.
- [4] The tkinter pack geometry manager, 2018. [Internet; consultado 29-junio-2019] <http://www.effbot.org/tkinterbook/pack.htm>.
- [5] Análisis dafo, 2019. [Internet; consultado 25-junio-2019] [https://es.wikipedia.org/wiki/An%C3%A1lisis\\_DAF0](https://es.wikipedia.org/wiki/An%C3%A1lisis_DAF0).
- [6] Bases y tipos de cotización 2019, 2019. [Internet; consultado 25-junio-2019] <http://www.seg-social.es/wps/portal/wss/internet/Trabajadores/CotizacionRecaudacionTrabajadores/36537>.
- [7] Choosing a license, 2019. [Internet; consultado 28-junio-2019] <https://docs.python-guide.org/writing/license/>.
- [8] Licencia mit, 2019. [Internet; consultado 28-junio-2019] [https://wikis.fdi.ucm.es/ELP/Licencia\\_MIT](https://wikis.fdi.ucm.es/ELP/Licencia_MIT).
- [9] Matriz de mckinsey, 2019. [Internet; consultado 25-junio-2019] [https://es.wikipedia.org/wiki/Matriz\\_de\\_Mckinsey](https://es.wikipedia.org/wiki/Matriz_de_Mckinsey).

- [10] Draw.io. Drawio, 2019. [Internet; usado 20-junio-2019] <https://www.draw.io>.