UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

TFG del Grado en Ingeniería Informática

título del TFG
Documentación Técnica

Presentado por Jesús Carro Tomé
en Universidad de Burgos — 17 de junio de 2019
Tutor: nombre tutor

Índice general

Índice de figuras

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 llegar a la cumbre de su desarrollo. También dentro de este proceso hay que tener en cuenta la viabilidad de llevar a cabo el proyecto fuera de los limites como trabajo de fin de grado que tiene impuesto en cuanto a recursos y tiempo.
Este proyecto ha recibido una planificación bastante estricta como bien se ve a lo largo de las memorias, partiendo en un principio de una idea y un sistema de planificación e intentando en la medida de lo posible que la estructura de dicha plan variase lo mínimo posible. Se siguió de inicio a fin una metodología ágil a través de la herramienta SCRUM.
En verdad es bastante complicado suponer los beneficios y perdidas que puede tener la inversión en una aplicación y su correcto desarrollo, pues esté tiende a un incremento exponencial rápidamente. Pero es muy incierto el saber si crecerá o no a lo largo del ciclo de vida del proyecto.

A.2. Planificación temporal

Como se ha mencionado ampliamente a lo largo de la creación de este proyecto, se fue dividido en las siguientes pautas:

- 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

Esto sería la planificación del proyecto a grandes rasgos, el cual siguió un cronograma organizado, para evitar el atasco de diferentes tareas, e intentar un desarrollo lo mas ágil posible, estructurando la creación de este trabajo de fin de grado, como si de un proyecto empresarial dispuesto a salir al mercado se tratase:

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

Figura A.1: Cronograma del proyecto completo

Fases

Perfeccionamiento de la idea

La primera fase del proyecto fue idea propia, al no escoger ningún proyecto predefinido, se opto por la idea de generar un proyecto de credenciales

propias, haciendo de esta fase una de las más complicadas.
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

Una vez tenida la idea, se planeo una reunión con el tutor del alumno para la coordinación entre alumno/tutor, sobre el alcance de la aplicación, teniendo en cuenta hasta donde se podía llegar en el tiempo del que se disponía y teniendo en cuenta en todo momento donde podríamos llegar si se avanzase más adelante.
Determinar los requisitos
Dentro de la misma reunión se hablo de los posibles requisitos de los cuales se iba a necesitar, ambas facetas se desarrollaron e idearon a la par. Dado el ámbito del proyecto, no requería de grandes requisitos, mas que un trabajo de investigación previo y un ordenador.
Requisitos Hardware
Se tenía que tener en cuenta quien y como iba a ejecutar la aplicación y para ello se iba a tener en cuenta una serie de requisitos hardware, fue una fase de corto desarrollo pues, puesto que se decidió abandonar la idea de plataforma android, los quesitos hardware eran los mínimos posibles. **Interfaz usuario del cliente**
Se desarrollo una interfaz básica dispuesta a modo de shell, con todo funcional, la aplicación hacía todo lo que en base debía hacer, excepto la persistencia de datos que se añadiría con la interfaz gráfica.
Prueba
Se puso a prueba durante varios días probando todos los posibles resultados y opciones, haciendo que en numerosas ocasiones, tuviéramos que modificar el varios métodos dentro del código.
Completar Desarrollo
Etapa más larga, una vez que los algoritmos internos funcionaban a la perfección, se paso a la construcción del programa en el resto de ambitos, 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 diferentes funcionalidades. **Pruebas del sistema**
Se llevaron a cabo numerosas pruebas del sistema, para ver que era solido y aprueba de fallos pese a ser un StartUp de lo que será un gran proyecto,

intentando una presentación impoluta de cara al mercado.

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

Para este apartado se dispondrá de una serie de estrategias clásicas de analisis de viabilidad de proyectos, extensamente utilizados en el mundo laboral, y que ayudar a ver la viabilidad de manera gráfica y clara. Antes de empezar, aclarar que se pensó este proyecto como una idea de negoció para múltiples plataformas y que esto es un startUp de un gran proyecto empresarial, que tiene presente el desarrollo de esto y la traducción a múltiples lenguajes para diferentes plataformas. La idea de negocio se basa principalmente en la que se basan múltiples aplicaciones androis y IOS:

1. Versión Free: Versión gratuita con funcionalidades limitadas que impiden que el usuario pueda hacer uso de todos los complementos de la aplicación, además, de anuncios múltiples para el beneficio de la empresa.

2. Versión Pro: Ganancia por suscripción mensual teniendo acceso a todas las funcionalidades, y atención al cliente para resolver cualquier tipo de duda, libre de anuncios, y con diferentes ofertas de renovación y cambios constantes para la mejora de la aplicación.

3. Versión Enterprise: Versión para los pequeños o grandes negocios, por subscripción mensual basado en packs, donde los negocios podrán dar diferentes claves a los clientes, teniendo una serie de características propias de esta versión, haciendo que le profesional de la salud, o empresa pertinente este en todo momento en contacto con sus clientes y puedan ver sus progresos de manera mas automatizada.

Análisis DAFO

- 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 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

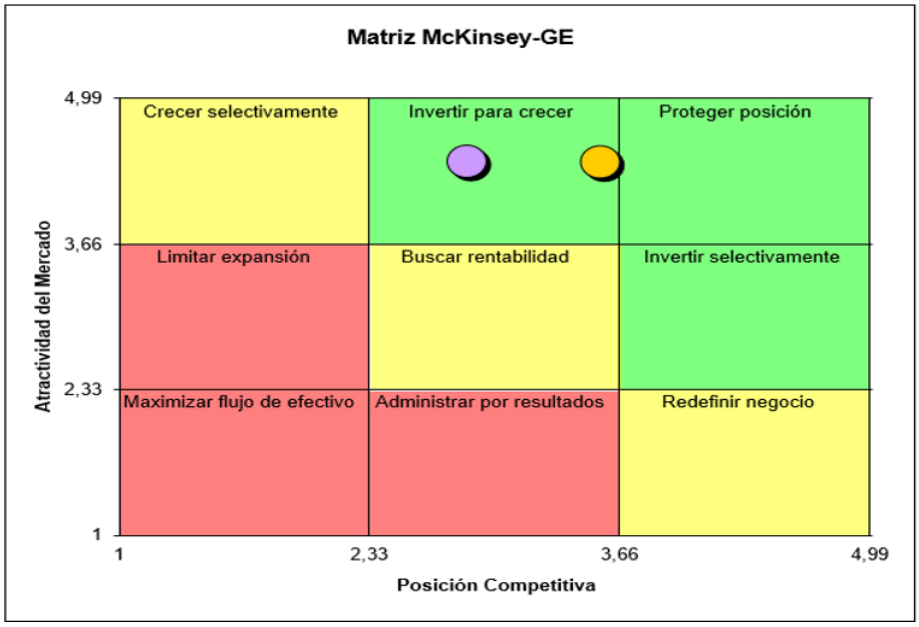

Figura A.2: Matriz mckensey del plan de negocio

Crecimiento propio de una App

Como se menciono anteriormente los ingresos esperados de una aplicación son exponenciales si llega a triunfar. Según el centro de jóvenes emprendedores del Santader Explorer, esto es una estadística real del crecimiento de una App en el mercado:

Figura A.3: Estadística del crecimiento propio de una aplicación

Viabilidad legal

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.

Forma jurídica

Descripción de la forma jurídica

Elección de una sociedad limitada por los motivos que me benefician:
- Capital Social mínimo: 3.000 euros

- Nº 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 tambiéntelemáticamente 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 lalicencia 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 las siguientes apartados de este capitulo se describirá todos los requisitos y objetivos que inicialmente debía cubrir el programa o herramienta 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, que automatice y facilite el autoaprendizaje, haciendo visible al usuario de manera directa, clara y concisa, de cómo influye cada decisión que toma en su día a día.
Es la imagen de un gran proyecto de futuro, y como tal ha de tener las funcionalidades mínimas necesarias para su uso y entendimiento. es decir:

- Posibilidad de iniciar sesión o crear un nuevo usuario.
- Correcto sistema de recomendación y selección por el usuario
- Diferentes elementos indicativos de la calidad de selección del usuario en el programa.
- Estandarización de las unidades métricas de recomendación, calidad y repartición usadas durante la aplicación.
- Objetos visuales de progreso del usuario y elección.

- Guardado de datos y persistencia.

B.3. Catalogo de requisitos

Requisitos funcionales

1. Probar correcto inicio de sesión y registro de nuevos usuarios.
2. Probar que añadir un nuevo alimento funciona correctamente.
3. Probar que según la patología del usuario haga una correcta distribución
4. Probar las diferentes posibilidades de recomendación del menú y que el algoritmo te recomiende la adecuada.
5. Probar que cada vez que seleccionas un alimento el resto de comidas se actualizan en base a tu selección.
6. Calibrar la formula del sistema de recomendación para que sea lo mas precisa posible.
7. Comprobar que conforme el usuario va actualizando las selecciones le puedan aparecer menus de distintas calidades.

Requisitos no funcionales

- Crear diferentes estilos visuales para la ergonómica con el usuario.
- Debido a la lentitud de la interfaz, el software ha de ser lo más rápido posible.
- Para futuras extensiones el software ha de ser modular

B.4. Especificación de requisitos

Requisito 1

Versión: 1.0
Importancia: Media
Descripción:
Para un correcto uso de la aplicación debían poderse añadir diferentes

usuarios además, obviamente de poder iniciar sesión a los usuarios ya existentes en la aplicación. No resulto nada complicado, se crearón las condiciones necesarias y el formulario gráfico, para la correcta inscripción en el programa, haciendo las comprobaciones pertinentes para respetar siempre la consistencia de la base de datos.

Requisito 2

Versión: 2.0
Importancia: Alta
Descripción:
Debido a la limitación de la base de datos, en cuanto a alimentos se refiere, 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 el la calidad que viese pertinente, esto era además de un trabajo tedioso para el usuario, una posibilidad de romper la funcionalidad del programa haciendo que cualquier usuario pudiera poner su comidas favorita como la de mejor calidad.
version 2: El usuario mete el menú por cada alimento y su información nutricional extraída de Bedca, la cual, esta valorada en 100 gramos, este calcula automáticamente el valor del menú y la calidad de dicho menú

Requisito 3

Versión: 1.0
Importancia: Baja
Descripción:
Cada patología viene inscrita con un tipo de dieta, la cual, a la hora de realizar los cálculos de repartición correspondientes, interviene en el porcentaje de dichos cálculos. Esto se resolvió mediante clausulas condicionales que permitía crear esa variedad de forma sencilla.

Requisito 4 y 6

Versión: 3.0
Importancia: Alta
Descripción:
El algoritmo de recomendación es el método o estrategia al que mas vueltas se le dio.
Primero para su versión inicial, se pensó en recomendar solo a través del

LRE y la calidad, lo que provoca un desequilibrio nutricional imperdonable que había que evitar.
En la segunda versión, se añadió el atributo "dif"que hacía referencia a la diferencia calorica con la necesidad, esto provocaba un menor desajuste pero podríamos encontrarnos con una dieta donde todo fuera proteínas y estuviera destrozando al usuario en vez de ayudarle.
Versión 3: Se corresponde con la formula actual, la cual permite que el alimento se ajuste lo máximo posible a las necesidades del usuario basandonse en las características que este busca en ese momento.

Requisito 5

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 o deselecciona un alimento, el programa cambia todas las recomendaciones posibles.

Requisito 7

Versión: 1.0
Importancia: Media
Descripción:
Es un requisito mínimo de toda aplicación que cuando el usuario cierre dicha se guarde su progreso actual, básicamente, eso es lo que cubre este requisito, la necesidad de poder abandonar el programa guardando correctamente y que de esta manera se conserve las elecciones realizadas en el programa.

Apéndice C

Especificación de diseño

C.1. Introducción

En este capitulo veremos como todo lo relacionado con el diseño de la herramienta, por lo general ha llevado un pequeño estudio detrás para buscar la mejor ergonomía posible además del correcto desarrollo.

C.2. Diseño de datos

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

- Diseño visual organizado en Frames y ordenado por packs, sistema proporcionado por Tkinter, que permite subdividir cada pantalla en pequeñas proporciones y estas en sub-proporciones y así sucesivamente, pudiendo colarcarla a grandes rasgos (arriba, abajo,izquierda, derecha, ocupando parte, todo el eje x, etcétera.).

- Alimentos tratados según el estándar de medida para la información nutricional impuesto en todo el mundo.

- Manejo de 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.

Alta o modificación de usuarios

Por lo general 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, que básicamente es:

1. El usuario rellena sus datos y valida
2. El programa comprueba que todo es correcto y si es así lo añade y 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 remplaza.

Se muestra el diagrama de interacción de los diferentes objetos.

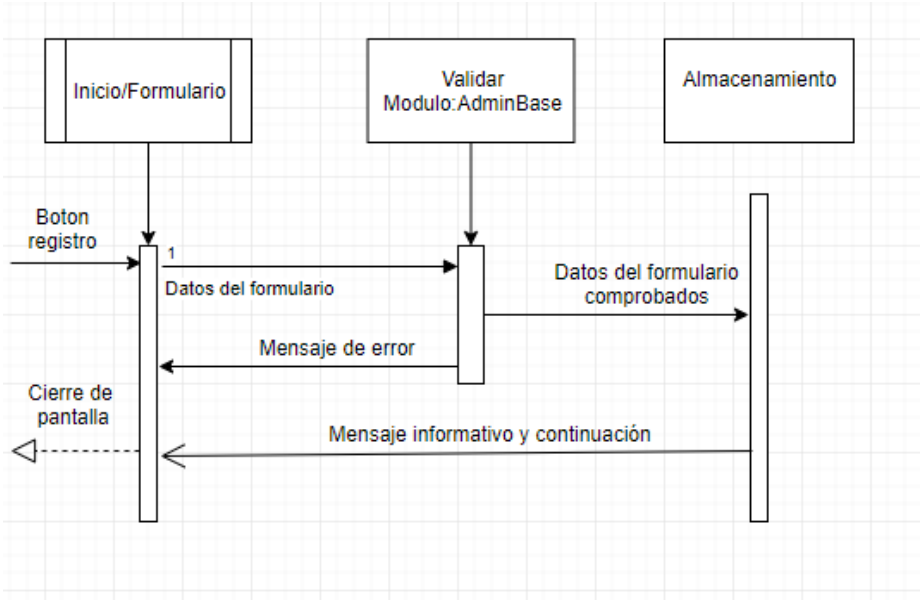

Figura C.1: Diagrama UML sobre la interacción en el registro o modificación de un usuario

Elegir Alimento

En este caso, es algo más complicado que lo anterior, pues hay varias vertientes, puedes refrescar hasta encontrar el alimento que buscas, y cuando lo seleccionas 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:

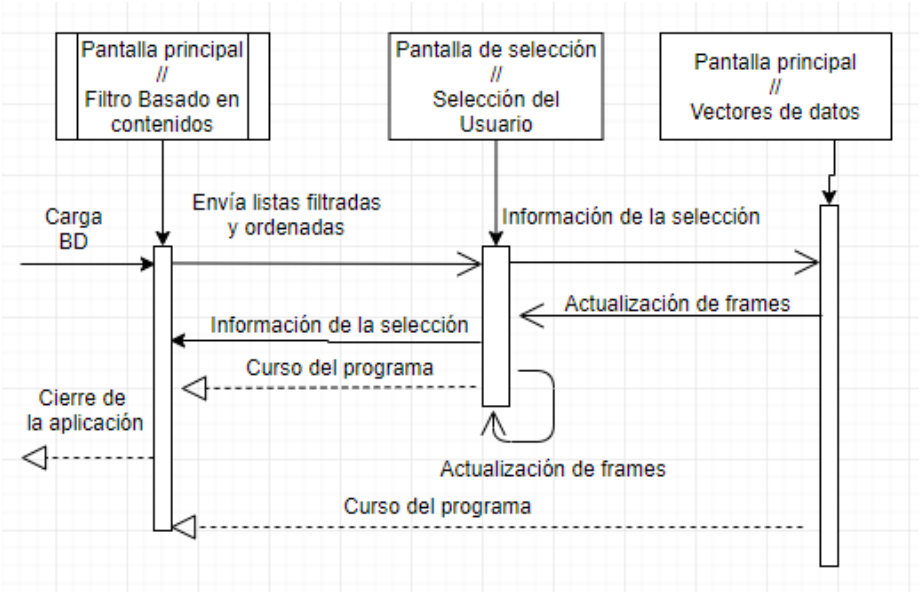

Figura C.2: Diagrama UML que muestra la interacción entre frames y funciones durante la selección

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:

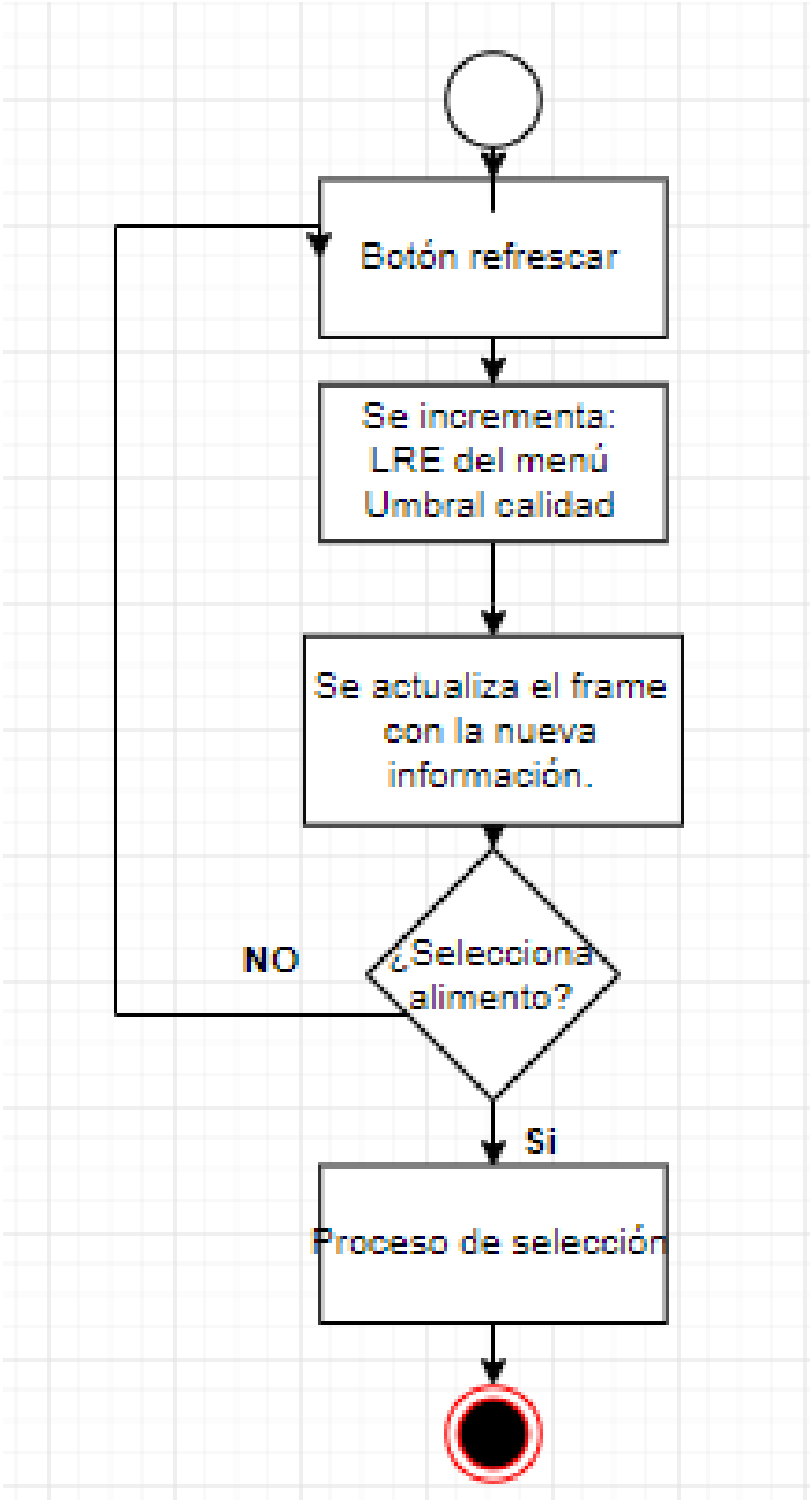

Figura C.3: Diagrama de flujo de la acción de refrescar opciones de menú

Carga Y almacenamiento bases de datos

Se podría hacer un apartado por cada colección de la base de datos que se carga y almacena por separado, pero puesto a que la estructura es similar y existen métodos encargados de unificar estos procesos, partiendo de la atomicidad del programa.

- 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.

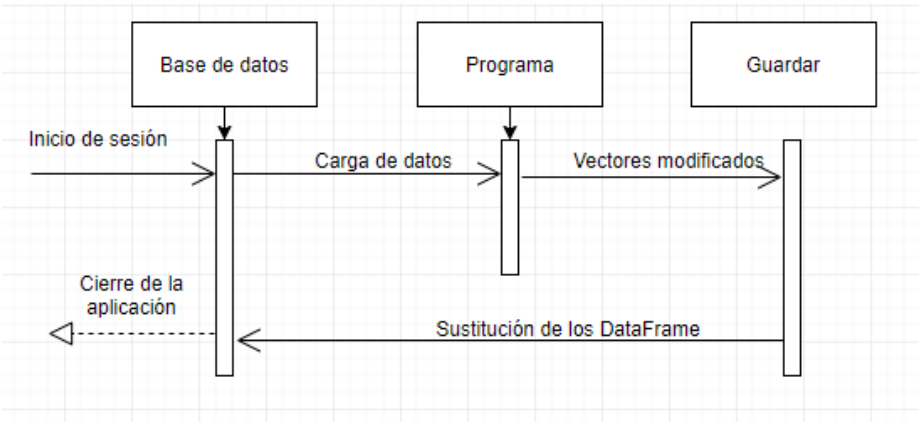

Figura C.4: Diagrama UML de la interacción básica del programa con la base de datos.

Estructura general del programa

Para concluir este pequeño apartado sobre el diseño procedimental, se mostrará el diagrama junto a su explicación de lo que es el funcionamiento e interacción de la estructura del programa por lo general.

- 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 mostrar un diagrama UML básico de la interacción del usuario con el programa en caso de haber iniciado sesión.

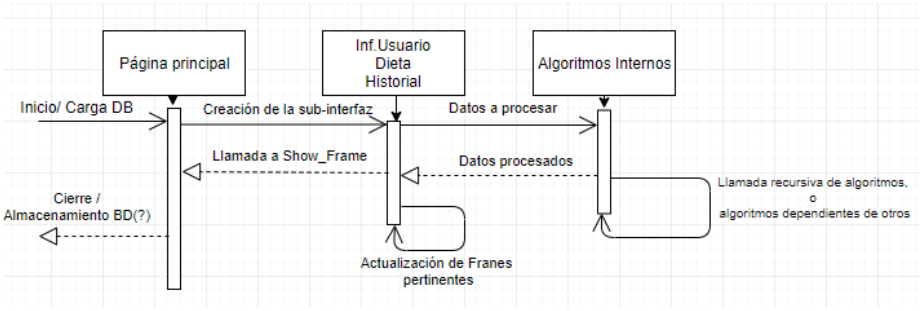

Figura C.5: Diagrama UML general del programa, una vez se ha iniciado sesión.

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 que de alguna manera tiene un sistema de herencia en la navegabilidad y la estructura modular, con las diferentes funciones internas.

Arquitectura visual

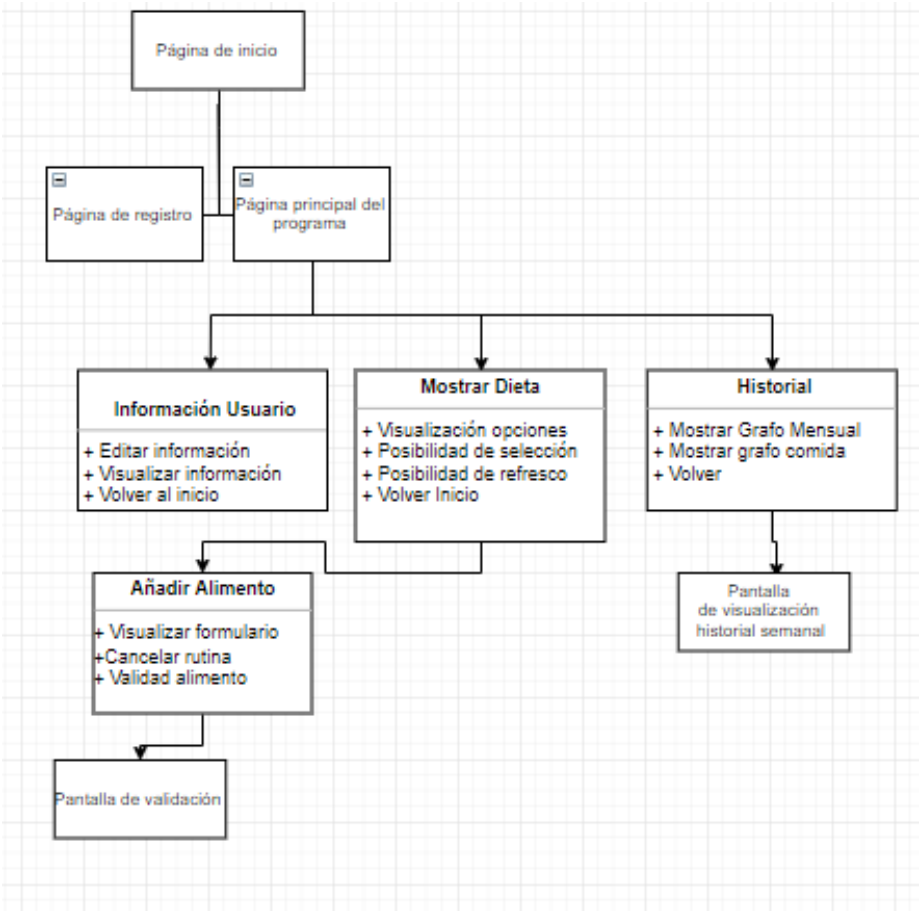

Figura C.6: Imagen de la consiguiente arquitectura visual para la correcta navegación

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).

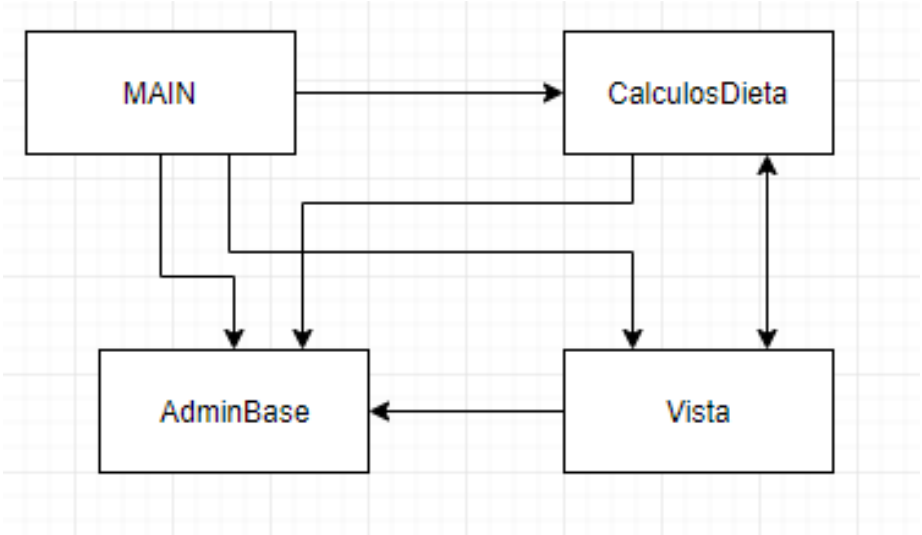

Figura C.7: Diagrama de las relaciones generales entre los distintos módulos

Vista

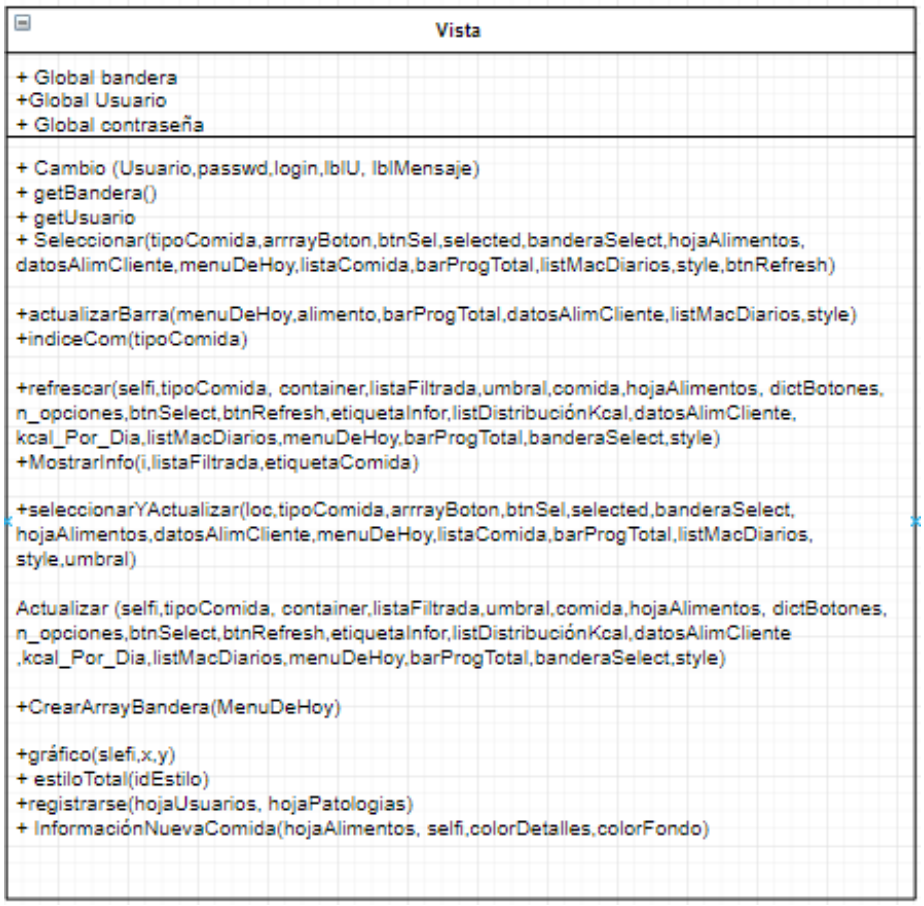

Figura C.8: Diagrama de clases del módulo/clase vista.

AdminBase

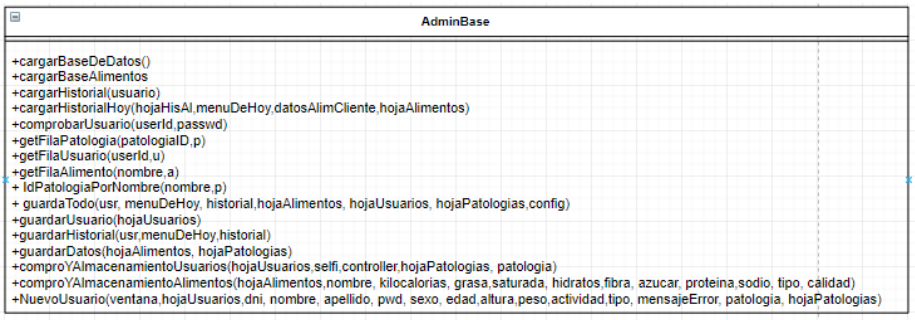

Figura C.9: Diagrama de clases del módulo/clase AdminBase.

CalculosDieta

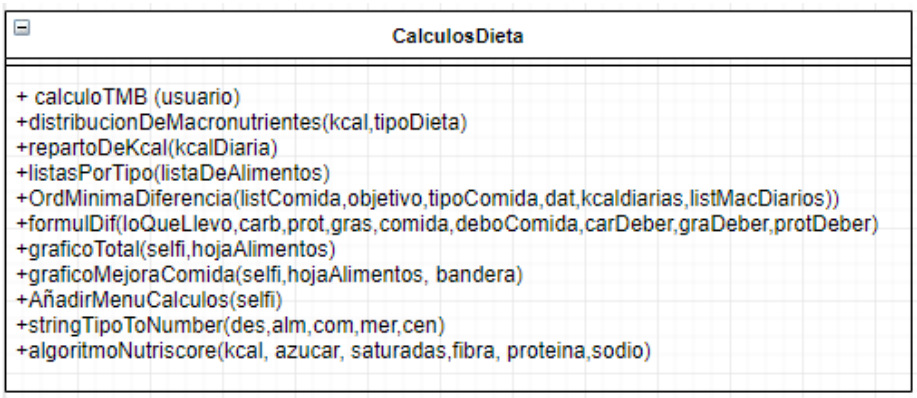

Figura C.10: Diagrama de clases del módulo/clase CalculosDieta.

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.

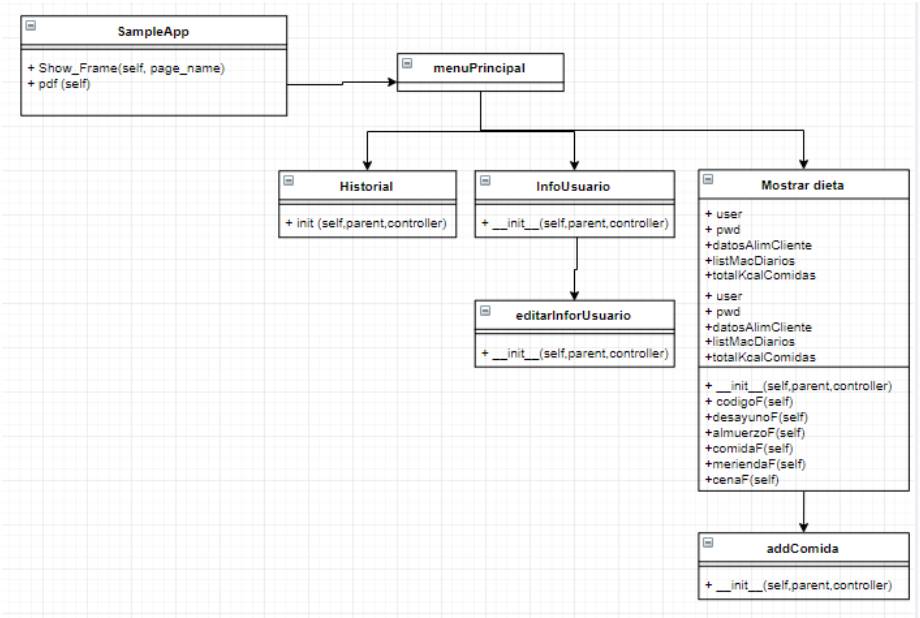

Figura C.11: Diagrama interno del Main

Apéndice D

Documentación técnica de programación

D.1. Introducción

En el siguiente apartado, se mostrarán los pasos a seguir, herramientas, etcétera. Para ponernos a trabajar con este proyecto. Se puede descargar desde: Repositorio GitHub

D.2. Estructura de directorios

Los directorios siguen la siguiente estructura:

- Carpeta principal / Inicial
  - Directorio: Assets
  - Directorio: Memorias
  - AdminBase.py
  - Main.py
  - CalculosDieta.py
  - Vista.py
  - BaseDeDatosDeAlimentos.xlsx
  - BaseDeDatosUsuarios.xlsx
  - Historial

- assets
  - Manual.pdf
  - Logotipo.PNG
- Memorias
  - Anexos.pdf
  - Memorias.pdf
  - Directorio: img (Almacenamiento de imágenes)

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 cual lo podremos hacer desde este enlace Python 3.6.8. Desde la misma página podremos descargar si lo deseamos la versión 3.7.
Paso opcional: A continuación instalaremos Anaconda, el cual nos dará una serie de funciones y programas, además de una powershell propia que nos permitirá el manejo por la aplicación más cómodamente. 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 (Aconsejo que no se use 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
- Ecplipe 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 ya debería cargarse la powershell desde la carpeta actual, acto seguido escriba main.py y el programa se ejecutará para su prueba o test.
Debido a que el desarrollador puede tomar ambos caminos (Instalar o no Anaconda). A continuación se explicar como instalar el resto de librerías a través del comando pip (La funcionalidad o comando pip, viene con la instalación de python y en este caso se deberá escribir pip3 por ser la versión 3.

- pip3 install matplotlib
- pip3 install numpy
- pip3 install pandas
- pip3 install pyinstaller o auto-py-to-exe
- pip3 install tk
- pip3 install webbrowser
- pip3 install os-win
- pip3 install Pillow
- pip3 install functools

Por lo general la mayoría de librerías te ayudará simplemente a que el programa corra correctamente, excepto pyinstaller que ayudará a crear un ejecutable (En caso de auto-py-to-exe, es una librería más visual que pyinstaller, pero que a grandes rasgos hace lo mismo pues lo que hace es usar internamente el comando pyinstaller con las opciones propias en caso de que elijas alguna).
Una vez tenemos preparado el entorno de Python podemos pasar a la instalación del IDE

IDE

En este aparado se hablará de como descargar y preparar el entorno para trabajar, como se realizó durante estos meses. **Spyder** Si se ha realizado la correcta instalación de anaconda, ya tendrá instalado este programa por defecto.

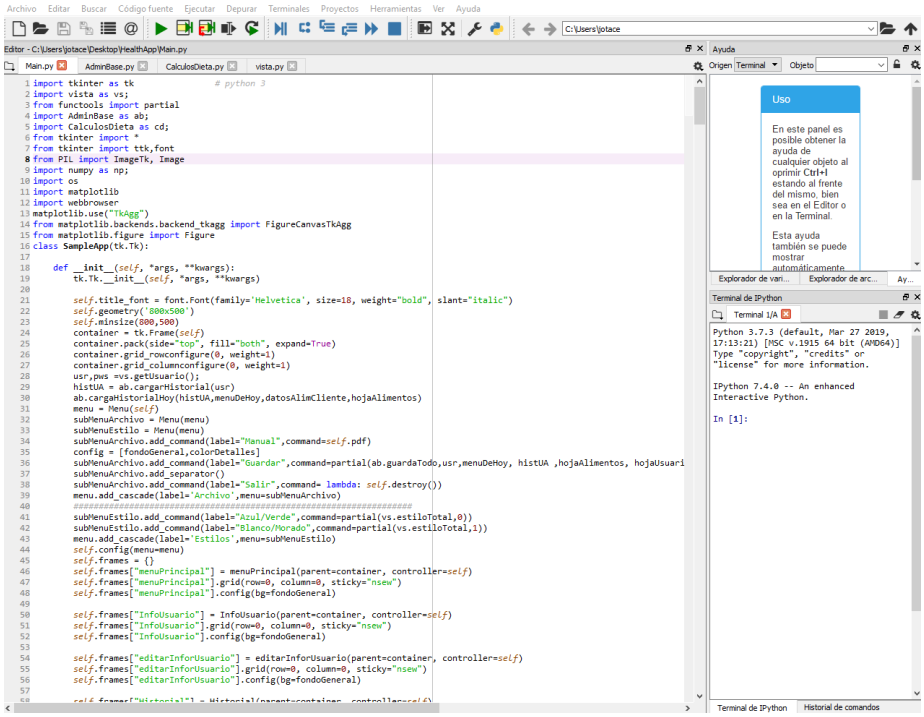

Figura D.1: Interfaz del editor Spyder para Python.

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 porgrama 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.

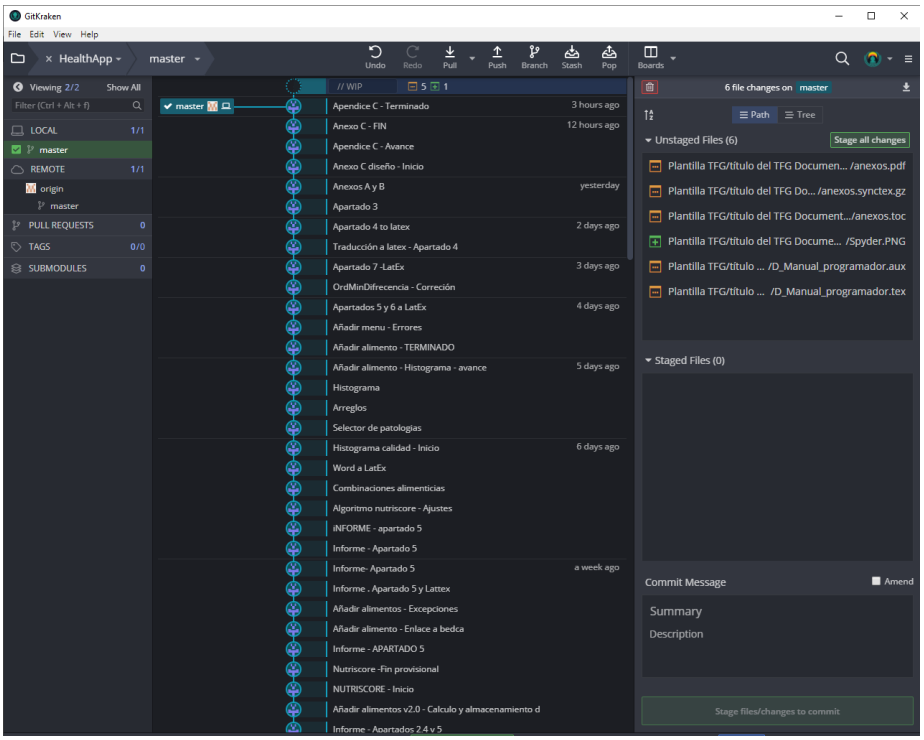

Figura D.2: GitKraken

D.4. Compilación, instalación y ejecución del proyecto

Lo primero que hay que hacer, es abrir nuestro editor, en el caso de este proyecto Spyder. Una vez abierto el editor debemos abrir nuestros archivos *.py. Para ello pulsamos en archivo ->Abrir.
No es necesario abrir todos los archivos, basta con abrir el archivo Main para su ejecución. Entonces nos vamos a la parte superior y pulsamos el boton ejecutar como bien vemos a continuación:

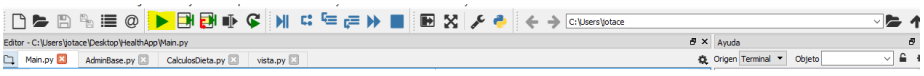

Figura D.3: Boton ejecutar de Spyder.

Una vez pulsemos se abrirá en otra pantalla generada por Tkinter, el programa principal:

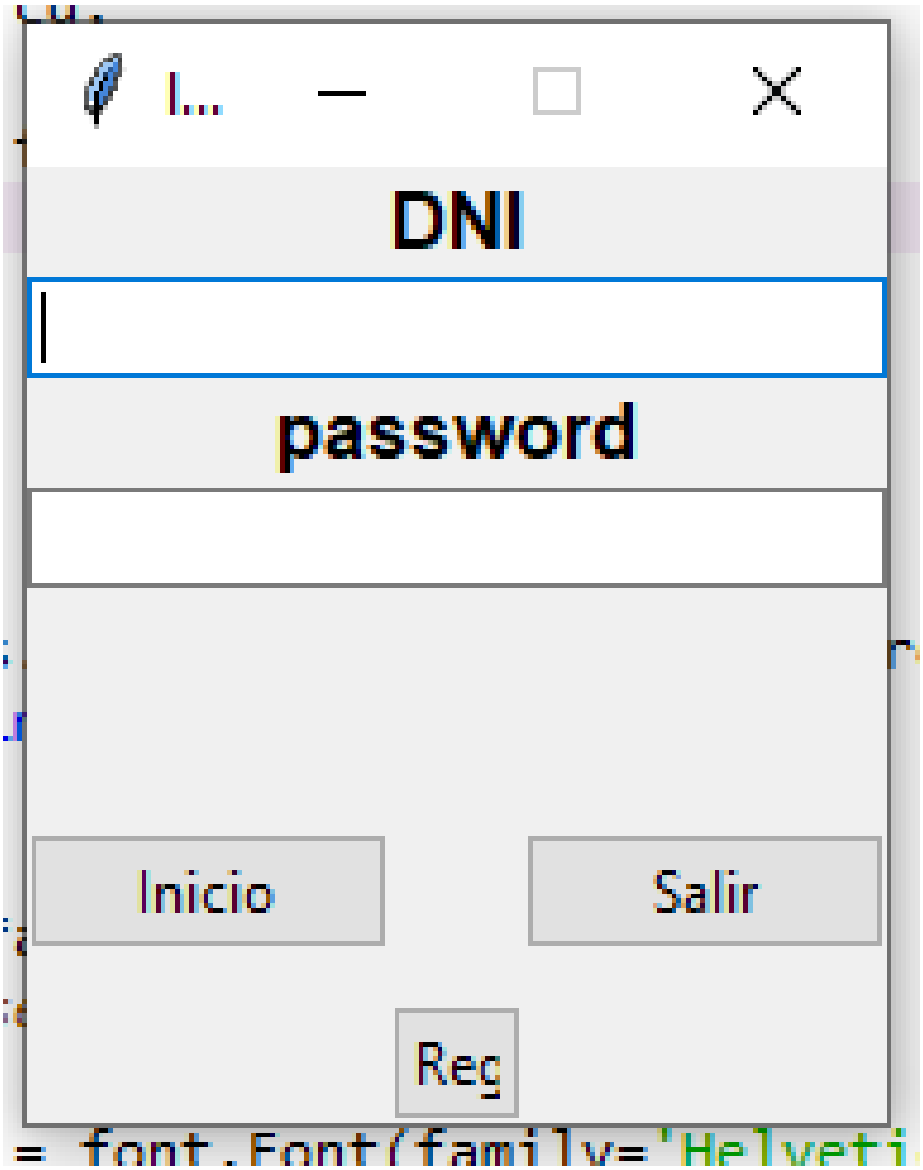

Figura D.4: Pantalla nueva generada por Tkinter.

Ante cualquier prueba que se desee realizar, como comprobar el valor de las variables, la veracidad de los datos, etcétera. Aparecerá en la pantalla de la terminal que por defecto viene posicionada abajo a la derecha:

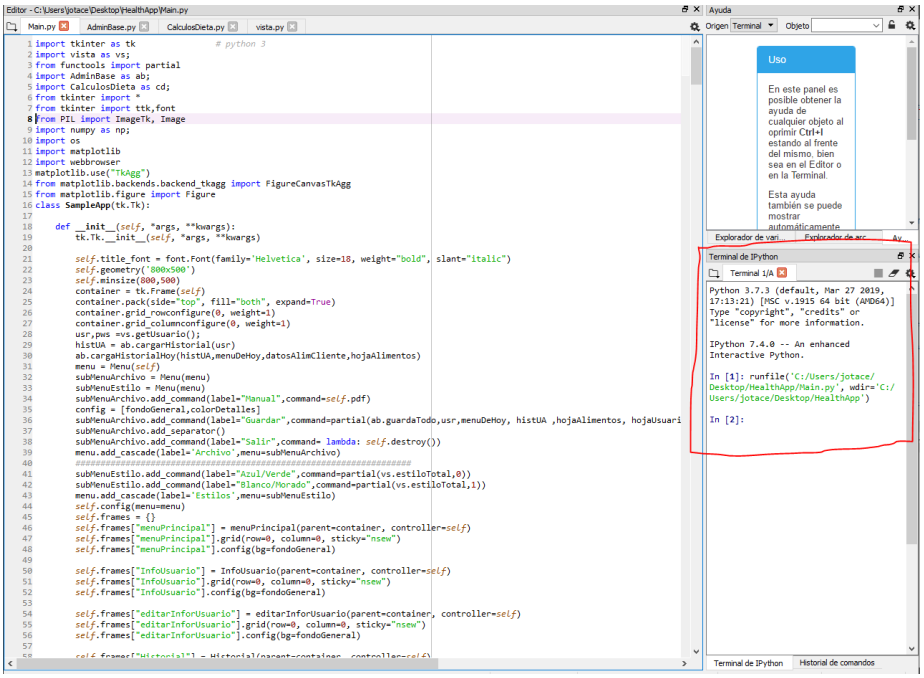

Figura D.5: Señalización de la terminal interna de Spyder

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. Y comprobando acto seguido de que se había guardado correctamente en la base de datos.

Problemas encontrados:
Los DataFrames se ordenaban 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 indices 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 además de que 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.

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.

Apéndice E

Documentación de usuario

E.1. Introducción

E.2. Requisitos de usuarios

E.3. Instalación

E.4. Manual del usuario