



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

título del TFG



Presentado por Jesús Carro Tomé
en Universidad de Burgos — 14 de junio
de 2019

Tutor: nombre tutor



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. nombre tutor, profesor del departamento de nombre departamento, área de nombre área.

Expone:

Que el alumno D. Jesús Carro Tomé, con DNI 71308125Y, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado título de TFG.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 14 de junio de 2019

Vº. Bº. del Tutor:

D. nombre tutor

Vº. Bº. del Tutor:

D. nombre tutor

Resumen

En este proyecto se llevará acabo una aplicación de escritorio, desarrollada en python, que propone en base a las características físicas del usuario, una serie de opciones, enseñándole a comer saludablemente, y mostrándole sus progresos.

Descriptores

Aplicación de escritorio, dietoterapia, análisis y administración de datos...

Abstract

Se aborda la **dietoterapia** mezclada con las **nuevas tecnologías** para un sencillo aprendizaje de estas.

Keywords

dietoterapia, salud, python, analisis de datos, mejor condicion fisica, ...

Índice general

| | |
|---|------------|
| Índice general | III |
| Índice de figuras | V |
| Índice de tablas | VI |
| Introducción | 1 |
| Objetivos del proyecto | 5 |
| 2.1. Objetivos generales | 5 |
| 2.2. Objetivos específicos como finalidad | 6 |
| 2.3. Objetivos incrementales del desarrollo | 7 |
| Conceptos teóricos | 9 |
| 3.1. Secciones | 9 |
| 3.2. Referencias | 9 |
| 3.3. Imágenes | 10 |
| 3.4. Listas de items | 10 |
| 3.5. Tablas | 11 |
| Técnicas y herramientas | 13 |
| 4.1. ¿Por qué Python? | 13 |
| 4.2. Metodología | 14 |
| 4.3. Aprender a aprender | 15 |
| 4.4. Estructura del programa | 17 |
| 4.5. Librerías | 19 |
| 4.6. Módulos | 21 |

| | |
|--|-----------|
| Aspectos relevantes del desarrollo del proyecto | 29 |
| 5.1. La idea | 29 |
| 5.2. Formato de trabajo | 30 |
| 5.3. Algoritmos del programa | 31 |
| 5.4. Conocimientos aprendidos en la carrera | 34 |
| 5.5. Conocimientos externos a la carrera | 35 |
| 5.6. Fases del proyecto | 38 |
| Trabajos relacionados | 39 |
| 6.1. Estudios: | 39 |
| 6.2. Aplicaciones Similares | 42 |
| Conclusiones y Líneas de trabajo futuras | 45 |
| 7.1. Conclusiones | 45 |
| 7.2. Líneas de trabajo futuras | 47 |

Índice de figuras

| | |
|--|----|
| 3.1. Autómata para una expresión vacía | 10 |
| 4.2. Estructura base de datos alimentos | 24 |
| 4.3. Estructura base de datos Usuarios | 25 |
| 4.4. Estructura base de datos Patologías | 26 |

Índice de tablas

| | |
|---|----|
| 3.1. Herramientas y tecnologías utilizadas en cada parte del proyecto | 12 |
| 4.2. Valor de la actividad física en TMB | 17 |
| 4.3. Porcentajes totales de cada comida | 21 |

Introducción

Este proyecto, desarrollado en Python, nace de la idea de una aplicación nueva de enseñanza dieto-terapéutica. Este trabajo, en cierto modo, hace honor al propio nombre de informática ", llevando a los usuarios una información, que como se demuestran en diversos estudios citados y desarrollados mas adelante en esta memoria, la mayoría de gente desconoce; y automatizando de manera ágil y transparente al usuario el aprendizaje exhaustivo sobre la dieto-terapia, para conseguir que el usuario consiga un habito de vida saludable y no reciba una estricta dieta imposible de seguir.

Actualmente se estima que alrededor del 81 por ciento de los españoles, que se proponen cambiar de hábitos y seguir una dieta acaba desistiendo. Eso en su mayoría nace de la dificultad, de mantener un ritmo constante de rigor a la hora de seguir una serie de instrucciones que de una manera u otra te condicionan, impidiéndote hacer una vida 100 por ciento, más allá de lo que en el ámbito alimenticio se refiere, como es en el ámbito social.

Por ello, Combinando el análisis y tratamiento de datos, mediante sistemas informáticos; con los últimos estudios relacionados con la dieto-terapia. Se busca crear una aplicación, que más que inculcar una dieta al usuario, sirva de guía para percibir un nuevo estilo de vida como propio. Gracias a esto, se puede llegar a conseguir un gran avance medico/informático, debido a la importancia que la dieto-terapia tiene en la salud de las personas.

Se calcula que una de cada cinco muertes en el mundo esta relacionada con la mala alimentación, por supuesto, dejando fuera las muertes por desnutrición. Por lo tanto, el proyecto es sustentado, de la constante necesidad y la creciente demanda de sistemas que ayuden y faciliten a una adecuada alimentación o dieto-terapia para el cuidado personal. Se pensó en un modelo de auto-aprendizaje, debido a que se ha considerado que pese a ser más lento es más sencillo para el ser humano adecuar un estilo de vida, que no seguir una dieta estricta. La idea principal es que el usuario, en base a sus necesidades,

elija de manera libre el menú que vaya a consumir en cada plato a lo largo del día, teniendo en cuenta en todo momento, como de bueno o mala es esa decisión para él.

El estilo de aprendizaje auto-aprendizaje es mas efectivo cuando reconoces que es lo mas importante para ti ” [?]

Tras estudiar varios métodos de aprendizaje, se consideró, que debido a la relación directa: dieta/salud, el auto-aprendizaje sería el mas oportuno, de esta manera a un ritmo lento y constante, el usuario aprende a comer bien, y no se pierde entre dietas. Se pensaron en una serie de pasos imprescindibles para inculcar un nuevo estilo de vida:

1. Identificar que se quiere cambiar
2. Metas específicas, realistas y constantes
3. Creación de un plan
4. Recordatorios que seguir
5. Mide tus avances

Se ha buscado que la herramienta desarrollada durante todos estos meses, de una sensación de falsa simplicidad, es decir, se ha buscado hacer lo mas simple, para abordar cuestiones complejas, de manera común, sin rizar el rizo, pero evitando dejar el mínimo de cabos sueltos, se ha buscado cumplir de manera pasiva cada uno de estos puntos dentro de la aplicación, puesto que dentro de sus objetivos y funcionalidades esta:

1. Su función principal, aprender, y cambiar el estilo de vida a través de la dietoterapia, cumpliendo el objetivo uno, de identificar que parte de tu estilo de vida, deseas cambiar.
2. Posiblemente, una de las partes mas complicadas e invisibles del proyecto. El proyecto siempre va a recomendar lo que considera la mejor opción, y siempre va a ser en base a los datos del cliente, consiguiendo, que las metas sean constantes, y que poco a poco se vayan volviendo mas estrictas, y pase de manera imperceptible al usuario.

3. Creación de un plan, la mas fácil e importante de los cinco pasos. Si se piensa detenidamente, es el usuario el que se crea su propio plan, pero siempre teniendo una “mano invisible”, que le ayudo a redirigir sus opciones, se decir, cada cosa que decide el usuario dentro de la aplicación queda registrada, y se ve en los gráficos sobre la calidad, como en verdad ha comido nuestro cliente.
4. Igual es la menos intuitiva para el usuario, pero no nos olvidamos de ella, se encuentra en el historial, esos gráficos que permiten ver al usuario como avanza y en la barra que le dice como esta comiendo en el día. Dentro de estas opciones aparecen reflejados los puntos 4 y 5.

A lo largo de esta memoria será plausible el como se ha desarrollado cada parte del proyecto, tanto en el ámbito informático como en el nutricional. Detallando cada estudio, desarrollo y opción que se ha tenido a lo largo de este, para que toda decisión tomada a lo largo del periodo de desarrollo quede clara, y se entienda el ¿por qué?, de cada decisión. Con este proyecto se consigue proyectar la unión de dos mundos complejos, de la manera más simple posible, haciendo alarde de que en la sencillez se oculta la complejidad de todo esto.

Objetivos del proyecto

En este apartado hablaremos sobre los objetivos generales y específicos de este proyecto, aclarando las dudas, sobre las intenciones de cada parte de este proyecto.

2.1. 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. Se da la misma importancia a cada comida para evitar el “por una vez no pasa nada”.

Este proyecto, es hijo de la unión en constante crecimiento, entre la informática y las ciencias de la salud, dando un paso adelante y haciendo énfasis en la importancia que tiene una correcta dietoterapia para la salud de las personas, y aprovechando el constante crecimiento de la informática, para llevar a todos los usuarios, una aplicación cuyo reto es conseguir crear un nuevo estilo de vida, y hacer que las necesidades del usuario sean cumplimentadas de manera gradual.

Se consigue una interfaz sencilla, apta para cualquier usuario, además de una programación simplista y modularidad haciendo que cualquier programador con los conocimientos básicos.

2.2. Objetivos específicos como finalidad

En este subapartado se expondrán los diferentes objetivos que fueron pautando el proyecto final. Toda aquella característica, opción o detalle que se ha llevado a cabo a lo largo del proyecto ha sido para cumplimentar alguno de los puntos expuestos a continuación.

- El Usuario podrá iniciar sesión a través de su DNI y la contraseña que él mismo haya escogido
- Un nuevo usuario podrá registrarse, para hacer uso de la aplicación, siguiendo un formulario básico.
- El usuario podrá navegar libremente por la interfaz, de manera ergonómica y sencilla.
- Un usuario podrá ver su información y editarla, siempre que lo desee.
- Se le recomendará al usuario el menú más adecuado en base a sus características.
- Las recomendaciones variaran en base a la selección del usuario, manteniendo siempre la mayor coherencia en cuanto a la recomendación, con las necesidades del usuario, creando una experiencia adaptativa a este y completa.
- El usuario podrá consultar el manual de uso en todo momento, por si alguna duda le surge.
- Se cargará de manera automática las elecciones del usuario.
- El usuario podrá añadir nuevos alimentos a la base de datos, cuya calidad se hallará en base al computo: NUTRISCORE.
- El usuario podrá refrescar en caso de que ninguna opción le gusta desatando opciones de peor calidad.
- Se mantendrá un registro semanal para su consulta, además de una serie de gráficos para el aprendizaje del usuario sobre el proyecto
- Se buscará que el proyecto, mecanice una enseñanza del tipo: aprender a aprender, para llegar al usuario de manera que el mismo vaya enderezando su camino.

- Se podrá elegir entre diferentes estilos de diseño para la comodidad del usuario
- Se podrá guardar las elecciones del usuario, de esta manera se llevara un registro.

2.3. Objetivos incrementales del desarrollo

El proyecto fue fragmentado en pequeños y diversos objetivos específicos para conseguir llegar al gran objetivo final. Se pensaron objetivos complementarios, para conseguir un desarrollo incremental y mantenido, haciendo que todos los objetivos específicos encajaran de manera secuencial y facilitando así su constante desarrollo, evitando al máximo posible, parches que retrasaran la creación del proyecto.

- La idea: Lo primero de todo ha desarrollar fue la idea, había que conseguir una idea original, para una herramienta original, algo que cubriera alguna necesidad del mundo actual, y que fuera original. Entonces decidí unir los conocimientos adquiridos estos años de estudio, con un tema de interés actual y en auge. De ahí surgió la idea, mas adelante, se pensó en el plus, de tener en cuenta distintas patologías del usuario a la hora de planificar la dieta.
- Dado que la idea ya estaba desarrollada, se paso a estructurar el proyecto, teniendo en cuenta que tipo de personas iban a usar esta herramienta, se pensó en como poner a disposición del usuario la información, que el programa iba a mostrar al usuario. Se planificó una metodología de autoaprendizaje, e interfaz básica para evitar toda distracción y facilitar tanto el desarrollo como el uso.
- Una vez aclarado todo concepto teórico sobre el desarrollo, se paso a pensar los objetivos a nivel de desarrollo software, y se llevo a cabo la creación y estructura de la base de datos, se crearon dos colecciones los alimentos y los usuarios, con sus diferentes características para el posterior tratamiento de los datos.
- Una vez pensada la mejor disposición y estructura de las bases de datos, se empezó a crear el esqueleto del programa, se desarrollo una aplicación de línea de comandos, que, de manera básica y lineal, te pedía comprobación de usuarios y te mostraba la comida.

- Una vez hecho el esqueleto se empezaron a desarrollar los cálculos necesarios, para redistribuir la información y llevar a cabo los cálculos de la manera más precisa posible.
- Una vez creados los módulos y hechos los cálculos se empezó a trabajar sobre la interfaz gráfica, en este paso, hubo un pequeño parche, pues hubo que adaptar el esqueleto para encajarlo con las competencias de Tkinter para la interfaz gráfica.
- En este punto ya habíamos conseguido una aplicación sencilla y básica y había que empezar a cubrir las necesidades del usuario. Era necesario que el usuario pudiera cambiar su elección, que tuviera a la vista la información de cada comida, y la información sobre su día a día. Aquí se creo el tronco del programa, que se basa en actualizar todas las comidas cada vez que se actualiza los datos del actual día del cliente. Para esto se creó el módulo vista.
- Una vez la aplicación funcionando, y todos los algoritmos haciendo su correspondiente trabajo toco perfilar detalles, a simple vista, de menor importancia, pero que en verdad juega un gran papel en el objetivo final del proyecto, se empezó a crear el sistema de gráficos e historiales, para que los usuarios pudieran aprender de sus errores, y tener un seguimiento de su avance desde el inicio del programa.
- Cuando por fin se crearon los gráficos, y teníamos el programa operativo, se decidió dar la posibilidad al usuario de añadir nuevos menús a la aplicación, permitiendo un mayor crecimiento y rompiendo uno de los grandes limites que presentaba el proyecto. Junto con este punto se creó la posibilidad de que un nuevo usuario se registrará en el programa.

Conceptos teóricos

En aquellos proyectos que necesiten para su comprensión y desarrollo de unos conceptos teóricos de una determinada materia o de un determinado dominio de conocimiento, debe existir un apartado que sintetice dichos conceptos.

Algunos conceptos teóricos de L^AT_EX¹.

3.1. Secciones

Las secciones se incluyen con el comando `section`.

Subsecciones

Además de secciones tenemos subsecciones.

Subsubsecciones

Y subsecciones.

3.2. Referencias

Las referencias se incluyen en el texto usando `cite [?]`. Para citar webs, artículos o libros `[?]`.

¹Créditos a los proyectos de Álvaro López Cantero: Configurador de Presupuestos y Roberto Izquierdo Amo: PLQuiz

3.3. Imágenes

Se pueden incluir imágenes con los comandos standard de \LaTeX , pero esta plantilla dispone de comandos propios como por ejemplo el siguiente:



Figura 3.1: Autómata para una expresión vacía

3.4. Listas de items

Existen tres posibilidades:

- primer item.
- segundo item.

1. primer item.
2. segundo item.

Primer item más información sobre el primer item.

Segundo item más información sobre el segundo item.

▪

3.5. Tablas

Igualmente se pueden usar los comandos específicos de \LaTeX o bien usar alguno de los comandos de la plantilla.

| Herramientas | App | AngularJS | API REST | BD | Memoria |
|------------------|-----|-----------|----------|----|---------|
| HTML5 | | X | | | |
| CSS3 | | X | | | |
| BOOTSTRAP | | X | | | |
| JavaScript | | X | | | |
| AngularJS | | X | | | |
| Bower | | X | | | |
| PHP | | | X | | |
| Karma + Jasmine | | X | | | |
| Slim framework | | | X | | |
| Idiorm | | | X | | |
| Composer | | | X | | |
| JSON | | X | X | | |
| PhpStorm | | X | X | | |
| MySQL | | | | X | |
| PhpMyAdmin | | | | X | |
| Git + BitBucket | | X | X | X | X |
| MikTeX | | | | | X |
| TeXMaker | | | | | X |
| Astah | | | | | X |
| Balsamiq Mockups | | X | | | |
| VersionOne | | X | X | X | X |

Tabla 3.1: Herramientas y tecnologías utilizadas en cada parte del proyecto

Técnicas y herramientas

Esta parte de la memoria tiene como objetivo presentar las técnicas metodológicas y las herramientas de desarrollo que se han utilizado para llevar a cabo el proyecto. Si se han estudiado diferentes alternativas de metodologías, herramientas, bibliotecas se puede hacer un resumen de los aspectos más destacados de cada alternativa, incluyendo comparativas entre las distintas opciones y una justificación de las elecciones realizadas. No se pretende que este apartado se convierta en un capítulo de un libro dedicado a cada una de las alternativas, sino comentar los aspectos más destacados de cada opción, con un repaso somero a los fundamentos esenciales y referencias bibliográficas para que el lector pueda ampliar su conocimiento sobre el tema.

4.1. ¿Por qué Python?

La razón es la sencillez y capacidad de este lenguaje para el análisis y tratamiento de los datos gracias a librerías como Pandas o Numpy. Base gran parte de la metodología de análisis usada en este proyecto en el libro: Python for Data Analysis. [?]

Las distintas posibilidades que se barajaron antes de empezar fueron Java, Android, R y Python. Al final me decanté por Python por diferentes motivos, el primero en ser descartado fue Java, tras un estudio inicial sobre lo que quería hacer y el como hacerlo, me di cuenta que necesitaba un lenguaje potente que me dejase tratar los datos con claridad para su análisis, y después de leer varios artículos como: REFERENCIA ARTICULOS, me percaté de que Java no era la mejor opción, estos artículos siempre te orientaban hacia

Python y R. Entonces, ¿Por qué Android?, por la sencilla razón de que este trabajo está desarrollado para unir el mundo de la dietoterapia, y las ciencias de la Salud con la informática para hacer que el usuario tenga un fácil aprendizaje de dicha metodología. A mi parecer la forma más clara y rápida de llevar al usuario dicha tecnología es a través de su SmartPhone, pero se acabó descartando debido a la falta de conocimientos sobre sistemas Android. A estas alturas ya solo me quedaba elegir entre Python y R, tras indagar superficialmente sobre ambos lenguajes para el análisis y el tratamiento de datos, se llegó a la conclusión que ambos lenguajes tienen una forma de trabajar muy similares, y entonces la decisión fue clara, debido a que he trabajado en numerosas ocasiones con Python y la sintaxis de R para mi era desconocida, al final, me decante por Python.

4.2. Metodología

Introducción

En este subapartado explicaremos el cómo y porque se ha tratado los datos en este TFG además de los diferentes cálculos internos que se realizan para el sistema de recomendaciones, cálculos, etc.

Excell y Pandas

Se ha usado la herramienta de Microsoft, Excel, para trabajar como una base de datos, y poder tratarlos en forma de DataFrame, esta particularidad, nos la da la librería de Pandas. Hay un Excel para la base de datos genérica, y otro para la base de históricos. Se leen los datos automáticamente en cuanto el usuario entra en la aplicación, pero solo se guarda si el usuario así lo desea.

DataFrame

Se usan los DataFrame, para llevar un registro de todos los datos que el programa necesita, se tiene en cuenta tanto las bases de datos de los alimentos, usuarios y comidas como lo que el usuario lleva en el día. Los alimentos se tratan en el momento en el que se registra el usuario, de manera que, se separan las comidas de la lista principal, creando 5 listas (Desayuno, Merienda, Comida, Almuerzo y Cena), se tratan por separado y se filtran por el umbral, el umbral es un medidor de calidad de la comida, para qué solo te muestre lo más sano en un principio, y se van actualizando

sensibles a los cambios de la aplicación. Si se refresca la página aumenta el umbral para que sea mas amplia las posibilidades de comida, aunque esto signifique una peor alimentación. Con cada elección se actualizan el resto de las comidas, para que se ajuste al máximo la dieta a sus necesidades, se ordena por la formula que se hablará mas adelante, se crea la variable “dif” que sale de esta fórmula, y este es el eje principal de las recomendaciones, cada vez que se elige una comida, la comida se almacena para poder tratarla mas tarde, en cuanto se selecciona los botones se desactivan, y se cambia el botón seleccionar a editar, para que se pueda editar en cualquier momento. Cada vez que se elige una comida se reparte las calorías sobrantes/que faltan, y se recalculan las posibilidades.

Técnicas

Introducción

Cada subapartado de este apartado fue descrito con mayor rigor, en el apartado tres sobre las técnicas a aprender para la realización de este proyecto. No obstante, teniendo en cuenta que son, además, métricas que han sido realizadas durante este proyecto, haciendo que el proyecto este estructurado, tal y como esta, se comentarán de manera breve para que el lector, tenga un breve recordatorio de estas técnicas.

4.3. Aprender a aprender

Como ya vimos, tras encuestar y hablar con varios graduados en Magisterio y psicología y tener respuestas diferentes, pero no obstantes similares, uno de los mejores métodos de enseñanza ausente del enseñante, es la técnica de autoaprendizaje, más comúnmente conocida como: aprender a aprender, como bien dice la página del Centro del Profesorado de Córdoba:

“Aprender a aprender supone disponer de habilidades para iniciarse en el aprendizaje y ser capaz de continuar aprendiendo de manera cada vez más eficaz y autónoma de acuerdo a los propios objetivos y necesidades.

Esta competencia tiene dos dimensiones fundamentales. Por un lado, la adquisición de la conciencia de las propias capacidades (intelectuales, emocionales, físicas), del proceso y las estrategias necesarias para desarrollarlas, así como de lo que se puede hacer

por uno mismo y de lo que se puede hacer con ayuda de otras personas o recursos. Por otro lado, disponer de un sentimiento de competencia personal, que redunde en la motivación, la confianza en uno mismo y el gusto por aprender. Significa ser consciente de lo que se sabe y de lo que es necesario aprender”. [?]

Es decir, el auto-aprendizaje no solo ayuda al usuario a aprender de manera autónoma, sino que le ayuda, a que todo aquello que ha conseguido, para el sea, de alguna forma, una meta, algo que lograr, algo suyo, que no se le va a olvidar tan fácilmente, como información memorizada, sino que será el resultado de pequeños logros personales que integrarán al usuario, una nueva información, como si fuese suya.

TMB

Se traduce como el calculo o tasa del metabolismo basal, por si no lo recuerdan, el metabolismo basal, es el gasto calórico que llevamos a cabo en reposo, por el mero hecho de respirar, a este calculo se le suman una serie de variables relacionadas con la actividad física del usuario y como resultado tenemos las kilocalorías diarias que el usuario gasta al día o que es equivalente que el usuario debe tomar para mantenerse en su peso, obviamente, este calculo es genérico, es muy utilizado por endocrinos, cuya finalidad no es hacer un seguimiento estricto de una dieta, sino un calculo aproximado de esta, para la correcta alimentación del paciente.

Tras horas de estudio, se aproximó que la forma mas correcta de bajar o subir peso en base al cálculo del metabolismo basal, es dar un margen de 500 kilocalorías, arriba o abajo, según los propósitos de nuestro usuario. Se recalca, que es claro que es un cálculo genérico pero hasta el perfeccionamiento del proyecto, es el calculo mas adecuado que podemos realizar. Dicho calculo se sustenta a través de la siguiente formula: **Hombres:**

$$TMB = ((10*Peso(kg))+(6,25*Altura(cm))-(5*edad)+5)*ActividadFisica \quad (4.1)$$

Mujeres:

$$TMB = ((10*Peso(kg))+(6,25*Altura(cm))-(5*edad)-161)*ActividadFisica \quad (4.2)$$

Donde actividad física se corresponde con los siguientes valores:

| Ejercicio | Valor de ActividadFisica |
|---|--------------------------|
| Poco ejercicio | 1,2 |
| Ejercicio ligero(1-3 dias/semana) | 1,35 |
| Ejercicio Moderado (3-5 días/semana) | 1,55 |
| Ejercicio fuerte (6-7 dias/semana) | 1,725 |
| Ejercicio muy fuerte (dos veces al día) | 1,9 |

Tabla 4.2: Valor de la actividad física en TMB

Nutriscore

tras varios meses de desarrollo, se discutió sobre como otorgar el valor a las comidas. Debía ser un valor calculado, y estandarizado para que todo el mundo pudiera entenderlo, y buscar información sobre dicho algoritmo en caso de quedar alguna duda. Después de varios días, se propuso implantar el algoritmo Nutriscore, o método del semáforo, pues eventualmente se iba a implantar en España. A lo largo de estas memorias, se explica con detenimiento el algoritmo y su funcionamiento.

4.4. Estructura del programa

Introducción

En un principio se pensó en hacer una estructura MVC clásica pero conforme el proyecto fue creciendo, y prácticamente de manera imperceptible se convirtió en una versión un tanto modificada de este modelo. La estructura se basa en cuatro módulos principales, además de las librerías pertinentes, obviamente.

1. **Main** – Tronco del programa
2. **Vistas** – Relacionado con la parte visual
3. **CalculosDieta** – Todo calculo usado en el programa
4. **AdminBase** – Todo proceso relacionado con la base de datos

En los apartados siguientes se hablará de manera extensa sobre las librerías utilizadas y los módulos creados para el desarrollo de esta aplicación, por el momento, este apartado será explicando en que deriva y se basa dicha estructura, conforme el MVC clásico y cuales son las consecuencias de esto.

Calculos, Administración y vista

Podríamos llamar propiamente a este modelo el CAV, y se basa en separar toda función que se vaya a crear en el programa en estos tres pilares. Todo surgió de la necesidad de separar la administración de la base de datos con cualquier otro tipo de función en el programa, esto sin darme cuenta creaba una cuarta pata en el trípode que supone el MVC. Este fue uno de los motivos, pero el motivo principal fue menos precipitado, se iba creando un módulo cada vez que la lógica de una función necesitará un hueco donde encajar y ninguno de los módulos creados lo sostenía. Debido a mi forma de trabajar, me sentí cómodo de dividirlo así, creando inicialmente una estructura básica y modulada, pero no en exceso. Por un lado, todo elemento relacionado con la base de datos, por otro lado todo cálculo matemático realizado en el programa y por ultimo toda función que intervenga en la interacción usuario-aplicación.

Esto tuvo en su momento sus inconvenientes, surgían funciones que encajaban en diferentes módulos debido a que igual era puramente visual, pero interactuaba con la base de datos, o llevaba a cabo unos cálculos para dibujártelos visualmente, por ello se llevo a cabo una serie de reglas sencillas:

- Si su finalidad es puramente visual – Vista
- Si realiza cálculos u operaciones matemáticas, indistintamente de si sea sobre la dieta -CalculosDieta
- Si interacciona y guarda datos en la base de datos -AdminBase
- Si mezcla alguna de estas funcionalidades, por ejemplo, guardar en la base de datos y mostrar por pantalla algo, se tendrá en cuenta cual es su principal finalidad, por ejemplo, se guardan estilos, que se cargaran automáticamente para el diseño del programa, obviamente su finalidad es puramente visual, y guardar en la base de datos una transacción necesaria.
- Cualquier Objeto, ventana, tabulación, etc. Que siempre vaya estar ahí al Main, pero sus modificaciones se repartirán entre los diferentes módulos.

Estas reglas, a día de hoy, han cumplido con todas las funciones que se han creado en el programa dando el resultado previsto.

4.5. Librerías

Pandas

La librería desarrollada por Wes McKinney, Pandas, es una librería usada para el tratamiento de los datos como estructuras. No fue la primera opción para este proyecto, debido a que este proyecto se enfoca en el análisis de datos, y no en el tratamiento de bases de datos, se decidió crear una base de datos local en un archivo Excel, la primera opción fue la librería “openpyxl”, la cual, es una librería de código abierto, que permite la carga y manejo de datos XLS, el problema vino que te devolvía objetos desestructurados difíciles de tratar, y complicaba el objetivo principal del proyecto, acto seguido basándome en el libro: LIBRO, decidí probar con Pandas, esta librería me permitía leer los datos de los archivos .XLS, en forma de dataframes. Esto simplificaba el análisis de los datos, permitiendo tratarlos tanto como `dataFrame` como en forma de `numpy.array`, dicha librería será analizada a continuación.

Numpy

Como ya he nombrado en varias ocasiones en esta memoria, el libro: LIBROO, se basa principalmente en esta librería, como añadido, es la librería sobre la que he trabajado a lo largo de la carrera en cuanto a análisis de datos en Python, y la más recomendada por los usuarios en la web, sobre esta librería se sustenta principalmente este proyecto, siendo la encargada, del tratamiento, procesamiento y cálculo, de todos los datos que internamente realiza el programa.

Interfaz Gráfica

La librería utilizada para realizar y diseñar la interfaz gráfica fue Tkinter, tras buscar en diferentes páginas que encontraremos en la bibliografía, me decante por Tkinter debido a la falta de conocimientos en interfaces gráficas con Python, me decante por la más fácil de aprender. Entre las opciones que baraje se encontraban: Tkinter, WxPython, PyQt y PyGTK. Tkinter traía una serie de ventajas, entre ellas que viene preinstalada con Python, era fácil de aprender, y hay una documentación amplia y extensa. Pero también tiene sus desventajas, incluye pocos elementos gráficos, tiene un limitado control y la navegabilidad más sencilla se hace complicada, y sobre todo lo que más noto es su lentitud, cuanto más elementos añadas, más lento va la interfaz, no tiene alguna especie de cache o memoria que guarde lo que has “dibujado”, sino que los dibuja cada vez que salen en pantalla, lo hace sobre

cada botón etiqueta, etc.

La segunda opción que se me paso por la cabeza, fue WxPython, tenia grandes ventajas, como la rapidez, la flexibilidad que este ofrecía y sobre todo su mayor ventaja son todas las opciones que tiene para crear una interfaz gráfica compleja y “profesional”, tras barajar, estas opciones y compararlas con las de Tkinter, me di cuenta que todo lo que te ofrecía WxPython, era innecesario para la interfaz tan simplista que necesitaba para mi proyecto, y el aprendizaje era más complejo, además de ser mas complicado encontrar información o documentación sobre esta librería. El mayor problema de esta librería era que tiene una comunidad muy activa, la cual, esta constantemente insertando cambios, esto era un problema para proyectos largos, pues creaba problemas de compatibilidad, pero para este proyecto, no era algo que me resultará un inconveniente.

El resto de librerías fueron descartadas, al poco de buscar información sobre ellas debido a que daban las mismas ventajas o similares que la WxPython, pero tenían mas inconvenientes, al menos para el tema que aborda este proyecto.

Diseño

Se decidió realizar un diseño simple, sin complicaciones, ni laberintos internos de pura navegabilidad que haga del manual de usuario un mapa para guiarse a través del programa. Esta decisión trivial se llevó a cabo debido al principal objetivo del proyecto, el aprendizaje de la dietoterapia, esta aplicación esta, básicamente, orientada a todos los públicos, por lo que si se construye una aplicación tediosa, solo complicaría el uso y aprendizaje del usuario.

Se basa en un menú principal ramificado en tres vertientes: Usuario, Dieta y Registro (en el programa aparecen con otros nombres, pues aquí se explica la lógica del diseño). En la vertiente usuario, esta la información del usuario y la posibilidad de cambiar dichos datos para un avance del programa. En la rama de la dieta, se encuentra el tronco principal de la aplicación, se muestra las recomendaciones alimenticias, a parte de darte libertad a la hora de escoger y decidir que deseas comer en el día de hoy. Por último, estaría el registro o historial, es decir todo aquello que necesites para llevar un registro de tu progreso y concienciar de esta manera al usuario.

Respecto a los colores, se decido un tema básico que no agote la vista del usuario, ni tenga múltiples colores deslumbrantes, da la posibilidad al usuario de elegir entre una serie de estilos predefinidos para que escoja el que mejor se ajuste a sus gustos y necesidades.

| Comida | Porcentaje(%) | Descripción |
|----------|----------------|--|
| Desayuno | 24,75 | Alta carga calórica, rica en hidratos |
| Almuerzo | 13,5 | Baja carga calórica. |
| Comida | 30,5 | Mayor carga calórica, eje central. |
| Merienda | 11,5 | comida de paso y casi prescindible. |
| Cena | 19,75 | Alta carga calórica pero con moderación. |

Tabla 4.3: Porcentajes totales de cada comida

4.6. Módulos

Cálculos Dieta

Es el modulo principal sobre el que se sustenta el programa, hace todos los cálculos de recomendación, kilocalorías, repartos, etc.

En el manual del programador se extenderá mas sobre cada función, a continuación, se hará un breve comentario sobre cada función en base a la metodología.

CalculoTMB, repartoDeKcal y distribuciónDeMacronutrientes

Estas tres funciones trabajan de manera paralela, el **calculoTMB**, lo que hace es calcular el gasto calórico basal, de aquí se saca el objetivo diario, una vez calculada la cantidad de Kilocalorías diarias, se reparten las kcal (**repartoDeKcal**) según sea desayuno, almuerzo, merienda o cena, se reparten en base a unos porcentajes calculados en base a distintas fuentes de información encontrados a lo largo del estudio sobre el que se sustenta este proyecto, no son porcentajes exactos, Se distribuyen de la siguiente manera: A la par se calcula el número de Kilocalorías que se tiene que tomar de hidratos, grasas y proteínas, en base al tipo de dieta recomendable para tu patología en la función distribuciónDeMacronutrientes, Esta función toma como parámetros las kilocalorías diarias calculadas previamente por la función calculoTMB, y el tipo de dieta del usuario en base a su patología, y te devuelve una lista de los macronutrientes diarios, repartidos en gramos y las kilocalorías totales en el siguiente orden: ListMacDiarios [Kilocalorías, Hidratos, Proteínas, Grasas].

OrdMinimaDiferencia y formulDif

Antiguamente eran dos funciones, pero se prescindió de una tras comprobar que la función **OrdMinimaDiferencia**, internamente y de manera indirecta ya lo implementaba. Estas dos funciones son usadas juntas estando formulDif dentro de OrdMinimaDiferencia.

son el pilar base de esta aplicación, en cuanto al calculo de comidas se refiere, OrdMinimaDiferencia, coge como parámetros la lista de las comidas, el objetivo, el tipo de comida, lo que ya llevo comido y las kilocalorías diarias. En base a esto hace un calculo de lo que debería llevar de cada macronutriente concreto en cada comida concreta en el total del día (Siendo esto el sumatorio de las comidas anteriores), lo que debería llevar de cada macronutriente específico en esa comida específica, luego se recorre todas las comidas de las bases de datos calculando cual es la que mas se ajusta a lo que necesitamos, una vez cogemos la comida le pasamos todo lo calculado anteriormente, y se lo pasamos a formulDif, para llevar a cabo la fórmula, que consiste en:

$$DiferenciaMacronutriente = (KMCD - A) / ((KCDT - KMM) + (KMTC - KTM)) \quad (4.3)$$

Donde:

KMCD: Número de Kcalorias del macronutriente (Hidratos, proteínas o grasas) de la Comida (Desayuno, almuerzo...) que debería llevar en esa comida

A: Actual número de kcalorias de ese macronutriente que llevo en todo el día (Si estoy en Comida: desayuno+almuerzo+comida).

KCDT: Kilocalorias que debería comer en esta comida concreta para este macronutriente concreto.

KMM: Kilocalorias del macronutriente concreto del menú.

KMTC: Kilocalorias totales que debería comer en esta comida.

ktm: kilocalorias totales que tiene el menú a recomendar.

De esta manera cuanto mayor sea la diferencia entre lo que debo llevar y lo que llevo, y menor sea la diferencia entre lo que deseo y las características del alimento, mayor será la puntuación de ese alimento para su recomendación (**Nota: Se ordena de mayor diferencia a menor**).

Por ultimo se realiza la media entre los tres macronutrientes principales:

$$DiferenciaTotal = (DiferenciaHidratos + DiferenciaProteina + DiferenciaGrasas) / 3 \quad (4.4)$$

En el caso de que en algún macronutriente nos hallamos pasado y la diferencia de negativa, ese macronutriente no se tendrá en cuenta.

Gráficos

Para comenzar, he de transmitir que no estoy conforme al completo respecto a que al calculo de los gráficos este dentro de este módulo, pero debido al estado avanzado del proyecto, y la existencia de tres módulos, que siguen un patrón, base de datos, vista y cálculos, se decidió no romper esta estructura, y añadir estos cálculos a calculo dieta, y dividir los grafos en su parte visual (Que veremos a posteriori en el modulo VISTA) y su parte de cálculo1.

Hay dos tipos de gráficos por lo general, que buscan mostrar al usuario de manera clara y transparente su progreso a lo largo del uso del programa, dicho progreso se verá a través de la calidad, por ello, dividirlo en dos tipos de gráficos Gráfico total, que hace una media de la calidad, es decir del como has comido en todo el día, del ultimo mes, y te lo muestra por pantalla, permitiéndote ver si hay mejora o no.

Los otros tipos de gráficos son por comida, una vez el usuario, vea su progreso general, este puede mirar los gráficos del desayuno, comida, merienda. . . Así poder ver que comida le cuesta mas o menos de manera sencilla y clara.

AdminBase y estructura de la base de datos

En este apartado hablaremos de como esta estructurada la base de datos, y de como se cargan, se guardan y se trabajan con los datos desde el módulo AdminBase, el cual, es un modulo bastante simple, necesario para el trato de datos.

CargaBaseDeDatos y guardaDatos

Estas dos funciones cargan y guardan los datos de/en la base de datos respectivamente, como se decidió simular una base de datos con un Excel, se usa la librería Pandas, mencionada antes en este apartado para cargar las bases de datos en los respectivos arrays, nada más arrancar el programa, se guarda las tres bases de datos principales, Alimentos, usuarios, patologías e historial en cuatro vectores independientes, que se trabajaran con cada uno de ellos sin tener en cuenta al resto y se guarda el resultado de los cambios producidos durante la ejecución del programa. Recordemos que estos cuatro valores se sacan de tres Excel diferentes, un Excel nombrado BaseDeDatosDeAlimentos que contiene la lista de alimentos y patologías, otro llamado BaseDeDatosUsuarios, en un principio estos dos estaban juntos, pero separarlos en dos Excel diferentes nos permite la mejor trata de datos

a la hora del almacenamiento, y el historial, el cual lleva consigo un listado de las fechas y los usuarios como clave primaria, junto a todo lo que han comido ese día.

La base de datos se estructura en dos archivos Excel independientes, donde cada hoja sería lo equivalente a una colección de una base de datos, y cada fila sería un objeto dentro de dicha colección. A distinguir las siguientes hojas:

ALIMENTOS

La hoja de alimentos, en verdad, sería una hoja de menús ya contruidos los cuales tienen 10 campos, sobre los que actualmente se trabajan 8, pero se mantiene su estructura para posibles ampliaciones.

La clave primaria de esta colección sería el nombre, en un futuro está pensado añadir un id, o depende la metodología de desarrollo de la base de datos, ese id se generaría de manera automática teniendo en cuenta que no sigue ningún tipo de normal en relación con el alimento. Los campos son:

| Nombre | Calorias | Grasa | Saturadas | Hidratos | Fibra | Azucares | Proteína | Sodio | Tipo | LRE | Calidad |
|----------------|----------|-------|-----------|----------|-------|----------|----------|-------|------|-----|---------|
| Tarta de queso | 547 | 18,9 | 9,4 | 75,9 | 1 | 41,6 | 17,5 | 0,7 | 10 | 0 | 4 |
| Torrijas | 566 | 39,6 | 8,4 | 38,9 | 1 | 12 | 12,4 | 0,7 | 26 | 0 | 4 |
| Flan | 244 | 7,4 | 3,1 | 37,4 | 1 | 33,3 | 7 | 0,7 | 26 | 0 | 4 |

Figura 4.2: Estructura base de datos alimentos

Nombre (PK), Calorías, Grasa, Saturadas, Hidratos, Fibra, Azucares, Proteína, Sodio, Tipo, LRE, Calidad De los primeros siete campos no hay mucho más que añadir, que lo que el propio nombre indica, no obstante, se hará una pequeña aclaración de los últimos tres.

El tipo, marca el tipo de comida que es, se pensó en que metodología usar, teniendo en cuenta que un mismo menú puede ser a la vez, desayuno, almuerzo y merienda, cena y comida, y un largo etcétera. Por ello se pensaron dos posibilidades, la primera poner la inicial, letra o nombre de la comida para la que valga ese menú, tratarlo como un array y descomponerlo en su debido momento, pero en el futuro caso de que se diera la posibilidad al usuario de editar un alimento de manera que no le apareciera entre una comida específica se complicaba algo más, y la segunda opción era tratarlo como una cadena de bits: desayuno-almuerzo-comida-merienda-cena, donde el valor 1 sería si es válido para esa comida y 0 si no lo es. Por ejemplo:

1-1-0-1-0

Significa que es desayuno, almuerzo y merienda, pero no comida o cena, luego esto se traduce en el número decimal, que en este caso sería 26.

El LRE, recibe ese nombre por un juego de palabras con la gestión de procesos de un sistema operativo LRU (last recently used), pues lo que hace

es llevar una métrica de la frecuencia con la que el usuario toma esa opción, literalmente significa “last recently eat”.

La Calidad, como el propio nombre indica, es la calidad del alimento, es la variable de indicar la diferencia entre 100 kilocalorías de ensalada y 100 kilocalorías de azúcar, tiene un rango de valores entre el uno y el cuatro, siendo cuatro la peor calidad de todos, y uno por el contrario la mejor, la calidad va estrechamente relacionada con el umbral del momento y la comida dentro del programa, esta variable, empieza con el valor uno (siendo este el mas bajo posible) y según vas refrescando va aumentando para que las posibilidades de menú aumenten con el número de veces que refresques la comida. El umbral sirve para cribar del DataFrame de la comida que estemos trabajando, todos los alimentos cuya calidad sea superior al valor del umbral en el momento de selección, de esta manera se le complicará al usuario la posibilidad de hacer una mala elección a la hora de organizar su menú, además que se verá reflejado en el momento en la barra de progresión.

Usuarios

La hoja de usuarios alberga los datos de todos los usuarios que usan la aplicación la estructura de esta colección en si es muy básica y contiene toda la información necesaria para el calculo de la dieta, del usuario.

| id | nombre | apellido | password | sexo | edad | altura | peso | actividad | patología | tipo |
|----------|--------|----------|----------|------|------|--------|------|-----------|-----------|----------|
| 71308125 | jesus | carro | 1234 | H | 24 | 178 | 73,4 | 4 | 0 | subir |
| 1234 | Guille | Calvo | 1234 | H | 21 | 177 | 69 | 3 | 0 | subir |
| 1 | Pedro | Garcia | 1234 | H | 21 | 180 | 76 | 1 | 1 | mantener |
| 3 | Ana | Perez | 3 | M | 21 | 168 | 55 | 2 | 1 | mantener |

Figura 4.3: Estructura base de datos Usuarios

La clave primaria sería el DNI del usuario, el cual, se encuentra dentro del campo ID. La hoja se estructura de la siguiente manera:

Id (PK)-nombre-apellido-password-sexo-edad-altura-peso-actividad-patología (FK de la tabla Patologías) y tipo.

Tanto id, como nombre, apellido y password, son los datos privados del usuario, los cuales, son simplemente informativos y no tienen ningún valor adicional en el calculo de la dieta y los resultados. En cambio, el sexo, la edad, altura, peso, actividad, patología y tipo, influyen de manera directa con el calculo de la dieta, como pasaba en la hoja de alimentos en los primeros campos su nombre, explica su significado. A tener en cuenta, que el valor del campo patología es numérico para mayor privacidad y conexión con la base de datos de Patologías, pues es el id de cada patología y en base a su id

se busca la información sobre dicha fila. En el caso de valer -1, significa que el usuario no tiene ningún tipo de patología y su uso es exclusivamente para el aprendizaje y seguimiento de la dietoterapia adecuada. La actividad tiene valores entre 0 y 4, donde cero significa el máximo nivel de sedentarismo y 4 el máximo nivel de actividad. Por último el tipo, hace referencia, a los objetivos que el usuario tiene respecto a si mismo, si quiere mantenerse, subir de peso, o bajar.

Patologías

En esta hoja, la cual se encuentra dentro de las BaseDeDatosDeAlimentos, se encuentra la información básica de las patologías nombre, id, y el tipo de dieta que lleva alto o bajo en carbohidratos, proteínas o grasas. Esto se carga al inicio del programa, se comprueba si el usuario padece alguna patología y se selecciona el tipo de dieta correspondiente.

| id | nombre | dieta |
|----|------------|------------|
| 0 | sin patolo | normal |
| 1 | Diabetes t | bajo azuca |

Figura 4.4: Estructura base de datos Patologías

Guardar Todo

Función que se encarga de juntar en una única función el almacenamiento de todos los datos con los que se trabaja en esta herramienta. Dentro de esta función encontramos:

- **guardarHistorial:** Como el propio nombre indica, se encarga de almacenar las selecciones que hicimos en el mismo día en la base de datos. A través de la librería `dateTime` de Python sacamos el día de hoy, con la fecha en forma de cadena de caracteres y el DNI del usuario creamos la clave primaria, no se puede repetir dos veces esta combinación, si a la hora de guardar ya existe, se actualiza pero no se añade; se crea un objeto tipo `DataFrame` segregado del array `menuDeHoy`, y se escribe en el Excel “historial”.

- **guardarUsuario:** Mencionada en el apartado anterior, se encarga de guardar el grosor de la aplicación, que es todos los cambios que hayas hechos en la aplicación respecto a la comida permitiendo que el programa poco a poco se vaya adecuando a los gustos y lo que mas come el usuario.
- **guardarDatos:** Guarda los datos del usuario. Se llama en dos ocasiones, cuando se guarda todo y cuando se edita la información del usuario. La primera vez, en la ocasión que se guarda todo, es redundante, se hace para tener un respaldo en caso de que al editar usuario se de algún fallo y no se guarde, en un principio, en el momento que edites la información del usuario y se pulse en “Aceptar y guardar”, se guardarían los datos.

getXXXX

En este apartado se respaldaran las tres funciones que se encargan de dar la información de una fila concreta de una de las colecciones de la base de datos.

Vista

Para empezar, recordar que se ha seguido una adaptación de la estructura típico MVC, donde se sustituye el modelo y el controlador propio y se tiene una vista adaptada, esto ya fue explicado al inicio de este apartado, a continuación me centrare en la vista.

En el módulo vista se muestran, o llevan a cabo toda función que tendrá influencia en lo que el usuario ve, es decir, actualización de pantallas, muestra de datos, transacciones entre las dos ventanas principales, el cambio de los gráficos, etc.

El modulo vista contiene tres variables calves globales, que son el DNI del usuario (usr), la contraseña del usuario (contraseña) y la bandera. De los dos primeros poco hay que explicar, pero del ultimo sí, esto se hará en el siguiente apartado, junto a la debida explicación de las funciones que hacen uso de ello.

Comprobación usuario

Empezando por la función “cambio”, que se nombro de esta manera porque indica un cambio en la funcionalidad del programa, explicamos esto: El programa sigue su curso, cuando se cierra la ventana de log-in, o acceso, según como se haya procesado la información en cambio cuando la ventana

se cierra, el programa continúa con su normal ejecución o se cierra completamente.

Cambio comprueba que la combinación usuario y contraseña sea válida en caso de ser así cambia el valor de bandera a verdadero (True), indicando así que el acceso a sido un éxito, si no, la bandera seguirá a False, y se mostrara por pantalla un mensaje de error. Cuando la tupla usuario y contraseña es valida cierra la pantalla de “log-in”, lo que hace que en el “main” se siga con su normal ejecución, lo cual lleva, de manera directa a una condición que comprueba que la bandera es verdadera, si lo es relanza la aplicación principal, sino cierra el programa, y se termina toda ejecución.

Como el propio nombre indica getBandera devuelve este valor al main.

Seleccionar y refrescar

Sin duda alguna esto presento uno de los mayores retos de la aplicación debido a las deficiencias que presenta Python para construir interfaces gráficas, y dentro de dichas deficiencias generales, se añadía la dificultad que presentaba el simplismo de la librería tkinter para dichas interfaces.

Todo surgió de la necesidad de actualizar lo que se mostraba por pantalla cada vez que hacía una elección, pues si se quería sacar de este proyecto la mayor precisión posible del sistema de recomendación, cada elección, cada cambio de variable, debía suponer nuevas recomendaciones, esto en un principio parecía sencillo, pero Tkinter no permite la recarga de los datos, sino que al inicio del programa lo “dibuja”, todo de manera simultánea, impidiendo una actualización del frame cada vez que haya un cambio, por ello se crearon una serie de funciones, necesarias las cuales se engloban en una mayor, para que de una llamada realice todas las funciones necesarias. **seleccionarYActualizarResto**, esta función se encarga principalmente de actualizar todos los datos en pantalla, el menú que hemos comido hoy, además de llamar a la función **seleccionar**, la cual guarda la opción escogida y bloquea las posibles opciones, y acto seguido recorre los otros cuatro frames recalculando todas las listas, todos los resultados y sus respectivos LRE, etc.

Se ha decidido que para evitar errores de coherencia el programa solo actualizará aquellos frames no bloqueados por el método seleccionar, sacrificando un pequeño porcentaje de la precisión, para conseguir un funcionamiento mas fluido sin errores de compatibilidad o múltiples selecciones.

Main

Es la columna vertebral del programa, el cual, se encarga de cargar y procesar toda la información relevante que más adelante se va a ir editando en el programa, el main tiene una única función y el resto se divide en clases que se instancian en esta función. La función principal hace de flujo de entrada y salida, donde un if condicional hace la función de interruptor permitiendo que la aplicación original, distribuida en clases se lance, o se cierre, como se ha explicado anteriormente.

Todo se basa en una especie de jerarquía de objetos, en el que el objeto principal (Menú principal), contiene otros tres objetos que estos a su vez contienen otros tantos, creando una simple estructura de árbol, entendible por cualquier programador.

De entre las tres subclases, instanciadas dentro del Menú principal (Usuario, Dieta, Historial), cabe destacar la de Dieta, pues esta a su vez tiene una clase por cada comida, estas clases son actualizadas constantemente en cuanto se hace algún cambio en el programa, esto es una abstracción del **patrón de diseño observador**, en el que las clases observan esperando el cambio para actualizar sus recomendaciones.

Se quiso evitar un main lioso con cosas que no son propias del dicho, por ello se separo en lo que es el main, y las clases las cuales, se encargan del funcionamiento principal del programa. De alguna forma, podríamos ver el main como los cimientos del programa completo, y los diferentes módulos, son los encargados de terminar de construir el programa.

4.7. Herramientas

Una vez explicadas el por que de cada decisión que se ha tomado a lo largo del desarrollo del programa, y expuesto cada decisión meticulosamente, procurando que cualquier duda del lector respecto a como se ha trabajado quede resuelta; Nos centraremos en mencionar de manera breve, dando una ligera explicación del por qué, se han usado las herramientas que se han usado para el desarrollo del programa.

Anaconda

Distribución de código abierto, la cual, esta indicada para el análisis, procesamiento y computo de datos, de Python y R, trae consigo una serie de programas y características entre los que destacaremos: Spyder y VisualCode. Sobre la razón, por la cual, se decidió usar anaconda es básica, la propia definición lo dice, a fin de cuentas, es una distribución para el análisis de

datos con Python y R, lo cual, es de manera encubierta, el principal trabajo en este proyecto.

VisualCode

Al igual que Spyder, es un entorno de desarrollo incluido en el paquete de anaconda, con la diferencia de que VisualCode, es ampliamente utilizado por la comunidad de programadores para múltiples lenguajes, y frameworks, como angular2, javascript, c3, Python, ect.

Se pensó largo y tendido sobre que entorno utilizar entre Spyder y VisualCode, y durante los primeros periodos de la aplicación, se turnaron ambos sin ninguna razón aparente, con el tiempo, me sentí más cómodo trabajando con Spyder, y lo fui adoptando como entorno principal, delegando en VisualCode la tarea de hacer debug, debido a la notable mejora que este presenta sobre Spyder.

Indicar que antes de empezar se probaron varios entornos como NoteBook, PyCharm, y eclipse añadiendo el API de Python, pero se acabo decantando por estas dos herramientas nombradas anteriormente.

GitHub

Proyección de la metodología GIT para el almacenamiento y control de versiones, sin duda GIT es actualmente la estructura más usada en el mundo de la programación, y GitHub, una de las herramientas más conocidas y valoradas que hay. Con el tiempo, fue mi mayor aliado, permitiéndome recuperar antiguas versiones con el programa era corrompido por algún error lógico, y dando una seguridad de soporte sobre el desarrollo del proyecto.

GitKraken

Potente interfaz gráfica multiplataforma desarrollada por Electron, que nos permite controlar de forma sencilla y cómoda nuestro repositorio GitHub, permitiendo que el uso y control del GitHub fuera más ameno y rápido, sin duda es una de las mejores aplicaciones que he probado, y de las que mas me han ayudado a lo largo del proyecto.

Aspectos relevantes del desarrollo del proyecto

5.1. La idea

Uno de los aspectos a destacar en el desarrollo de este proyecto, es el nacimiento de dicho proyecto. Pues se trata de un trabajo de fin de grado, el cual, es un trabajo puro del alumno, donde tanto idea como desarrollo son realizados por él, y gran parte del tiempo gastado en el desarrollo fue en la ocurrencia de varias ideas, el pulimiento de dichas ideas, y la conversión de todo esto en lo que se convirtió en la idea final. Aunque dicha idea fuera cambiando a lo largo de su desarrollo.

Todo llegó de la necesidad de crear un trabajo interesante, algo diferente y que fuera de actualidad, a esto se le sumó, que para un desarrollo más ágil debía ser un tema conocido e interesante para su desarrollo. Entonces se llevó a cabo un periodo de investigación de las necesidades que azotan a las personas hoy en día, con las últimas corrientes de moda y que sobre todo pudiera resultar útil. Tras varios días se llegó a la conclusión de que actualmente la gente tiene una constante necesidad por cuidarse, por la vida sana, y que, además de ser una necesidad básica para cualquier persona, se estaba tornando en una moda explotable en cualquier página web, red social, etc.

Una vez encontrado el grueso del proyecto tocaba perfilar detalles, para ver a través de qué medio se podía llevar “una mejor calidad de vida” al usuario. De aquí surgieron varias vertientes:

- Deporte: descartada, por la inmensa cantidad de aplicaciones de plani-

ficación de entrenamiento que existes a día de hoy.

- **Nutrición:** Inicialmente descartada, debido a las mismas razones que lo anterior, existen miles de aplicaciones tanto que te proponen una dieta como que te ayudan a seguirla.
- **Recordatorios:** Con esto me refiero, a recordatorios de cuando hacer deporte, de cuando tomarse los medicamentos, etc. Se descarto rápidamente por ser una más, a la par que un proyecto banal.
- **Nutrición con otro enfoque:** Aquí es donde llego la idea, una vez descartada estas tres, se pensó que estaba todo inventado y el proyecto se convertiría en otra forma de “reinventar la rueda”, hasta que nos percatamos, que no existe una aplicación que te enseñe a comer, y que tenga en cuenta las patologías aunque sean en menor medida. Ahí es cuando nació la idea, y se empezó a formar la idea de la estructura de lo que iba a ser el proyecto.

De este modo se empezó con el proyecto, partiendo totalmente de cero, hasta el punto de conseguir algo perfectamente funcional, cumplimentando los objetivos propuestos para dicho proyecto.

5.2. Formato de trabajo

Se desarrollo una metodología ágil de forma incremental, se empezó formalizando las bases hasta llegar a la etapa final. No se pasaba a la siguiente etapa sin haber terminado la anterior. Las tres principales etapas del proyecto fueron:

1. Funcionamiento por interfaz Shell
2. Funcionamiento de manera gráfica
3. Finalización de cada detalle dejado indicado durante el proyecto.

De esta forma, se fueron creando todas las características de manera incremental, de forma que se dio mas importancia al claro desarrollo del proyecto, que a la velocidad de dicho desarrollo. Pese a esta importante decisión, en ningún momento el proyecto se atasco, y se pudo llevar a cabo de manera ágil, gracias a la reutilización de código entre capas, permitiendo una clara transparencia a la hora del desarrollo.

Al no lanzarnos directamente con el entorno gráfico, el cual, nunca había sido visto a lo largo de la carrera, permitió que fueran más obvios los pequeños errores de implementación, de cálculos, etc. Para que cuando se pasará a la etapa de desarrollo gráfico, nos pudiéramos centrar en los errores, o correcciones que el entorno gráfico suponía, sin amontar pequeños fallos, que mas tarde dieran lugar a grandes errores.

5.3. Algoritmos del programa

Se quiere hacer énfasis en dichos algoritmos debido a que el programa utiliza una serie de algoritmos propios, ideados por el autor, que permite que el sistema de recomendación funcione como es debido. En este subapartado no se va a extender con la explicación de cada uno de los algoritmos utilizados, o métodos similares, creados por terceros y desarrollados por el alumno, sino aquellos que han sido ideados por él, y son un aspecto relevante en el desarrollo por el tiempo de ideación que han llevado dichos.

Algoritmos de recomendación

Es el algoritmo que estructura todo el sistema de programación, nace de la necesidad de que la recomendación se ajuste al máximo a las necesidades del usuario. Si se ordenaba única y exclusivamente por la calidad, nunca se verían los menús de peor calidad, además de que podría haber una desigualdad inmensa entre las calorías ingeridas y las necesitadas. Mas tarde, se pensó en hacer una formula basada en la diferencia calórica, cosa que se adecuaba mas a las necesidades del usuario, pero creaba un desajuste en los macronutrientes, pues el usuario podía llevar 300 kcal exclusivamente de grasa, y que le recomendase otras 300 también solo de grasa, rompiendo la idea de dieta equilibrada. Por ello, se ideo otro prototipo de formula.

La formula tiene en cuenta la proporción de las necesidades calóricas del usuario, partiendo de cada macronutriente, se calcula la diferencia entre la diferencia de macronutrientes que debería llevar y los que llevo, todo llevo partido por la suma de la diferencia entre: Las kilocalorías que debo comer y las que tiene el alimento y la diferencia entre el macronutriente específico y lo que el alimento tiene.

FORMULAAAAAAAAAAAAAAAAAAAAA

Persistencia de los datos

Esto en verdad, no es implementado como tal por el alumno, sino más bien, es un recurso para romper una de las limitaciones que presentó utilizar Excel como simulación de la base de datos.

Para evitar datos sobre escritos y problemas con los datos, se procuró durante todo el programa atomizar lo máximo posible cada método sobre la base de datos, haciendo que los cambios sean lo mas inconexos y manejables para el usuario.

En el método añadir alimento se rompe esta norma, añadiendo un alimento en la base de datos que hay que guardar, pero sin guardar todos los cambios en el resto de los alimentos, por si el usuario no quisiera hacerlo. Para evitar esta falta de control por parte del usuario sobre la persistencia de sus datos, a la hora de almacenarlo, se carga el array de nuevo, y se añade esta nueva línea tanto al array usado durante el programa, como al nuevo array de alimentos, este nuevo array se almacena en la base de datos, mientras que el otro sigue con su propósito en el programa. Si el usuario hace algún cambio y lo guarda, el nuevo alimento estará a salvo, si, por el contrario, el usuario decide salir sin guardar su progreso en el día, el nuevo alimento ya estará guardado en la base de datos.

Distinción entre tipos de comida

Uno de los retos a afrontar a lo largo de este proyecto fue simplemente, el como, se iba a indicar que tipo de comida/menú era. Sin duda no fue uno de los retos más complejos abordados a lo largo del proyecto, pero si uno de los más curiosos. Recordamos que un mismo menú, puede pertenecer a varias comidas del día, entonces se pensaron varios métodos hasta llegar al óptimo, los cuales explicaremos con brevedad a continuación:

En un principio se planteo hacer una cadena de Strings separadas por algún delimitador cualquiera, y cargarlo en forma de String y separarlo por algún método (Por ejemplo Slice()), pero dio ciertos problemas de carga y almacenamiento, problemas no demasiados complejos, pero que aumentaban la complejidad algoritmo al buscar a través de un bucle todas las posibilidades complicando esto, y se pensó en una concatenación de bits donde 1 es que era esa comida y 0 que no, es decir: 11010, Significa que no es ni comida ni cena. Esto presentaba el mismo problema que la cadena de

caracteres y era quizás más confuso para su comprensión, Además del pequeño inconveniente de que si un menú no era desayuno (Es decir empezaba en 0), en algunas ocasiones se almacenaba sin ese dígito creando una desigualdad en el programa. Por estas razones se decidió pasar al siguiente método, una mezcla de ambas, se iba a usar una cadena para calcular un único valor que cargar. Finalmente se paso esa cadena de bits a un valor numérico, de esta manera se asegura una única carga, y el tratamiento a través de cláusulas condicionales simples.

Ejemplo:

Un alimento que sea comida y cena: 00101 -> Se representará con el número 5.

Esto permite la carga de un único elemento, sin posibilidad de fallo además de un calculo sencillo, cuyo mayor inconveniente es llevar a cabo todas las clausulas condicionales, para hacer la criba por tipo de comida.

SCRUM

Quizás uno de los aspectos más destacados a la hora de realizar el proyecto, fue la organización por parte del alumno, a través de la metodología SCRUM, que permitía al alumno mantenerse centrado en cada nueva tarea, no pasando a la siguiente hasta dejar esta zanjada del todo.

Para la realización de esta metodología se tuvo en posesión una pizarra, en la cual se iba pegando post-it con una breve descripción de la tarea y una fecha máxima para la realización de esta. La pizarra era dividida en cuatro partes (pequeña adaptación del SCRUM).

1. TODO – Por hacer, eran todas las tareas, pautas, e ideas que iban surgiendo a lo largo de la semana o tras las reuniones semanales. Había un máximo de cinco, para evitar que el proyecto se atascase por una carga excesiva de trabajo. De TODO se pasaba a in progress.
2. IN-PROGRESS: En proceso, máximo dos cosas, una principal y una secundaria, la principal, consistía en el desarrollo de grandes métodos, cálculos, etc. Mientras que los secundarios, podían ser trabajos de investigación, rellenar datos o cualquier detalle que faltase de completas. De esta manera se evitaba la bifurcación del trabajo, haciendo del camino, pequeños objetivos, claros y directos a completar. Cuando el alumno pensaba que había terminado la tarea, la movía a “TO-VERIFY”.

3. TO-VERIFY: A verificar, es decir, falta la aprobación de los cambios realizados, si eran cambios propuestos en las reuniones semanales, eran zanjados en la siguiente reunión semanal; De esta forma se creaban distintos modelos y versiones de métodos, perfilando al máximo cada detalle en cada método. Las pequeñas tareas, eran resueltas, ya fuera en las reuniones semanales, o preguntando a personas, sobre como sería para ellos el correcto funcionamiento del programa.
4. DONE: Una vez todas las partes se ponían de acuerdo en que la tarea estaba terminada, pasaban a DONE, este apartado existía básicamente, para hacer mella en el trabajo que el alumno realizó durante la semana, haciendo palpable las semanas donde se tuvo una mayor o menor carga de trabajo, pudiendo regular esto, y teniendo de esta manera un avance constante.

En progreso solo se podía encontrar una tarea, y si el tutor desechara una tarea como incompleta pasaba a todo de manera que, de esta forma, semanalmente se cumplían con una serie de propósitos de forma clara y concisa, y no se apelotonaban todas las ideas y propuestas dejando pequeños cabos sueltos que arruinaran la representación final del proyecto.

Este método fue estudiado en la asignatura de gestión de proyectos por el alumno, y culminado en su periodo de practicas con el instituto de castilla y león, el cual, usaban de manera muy similar a la adaptación del alumno; La metodología SCRUM para gestionar todos sus proyectos.

5.4. Conocimientos aprendidos en la carrera

INTRODUCCIÓN

Apartado donde se hablará de forma breve sobre todo conocimiento aprendido durante los años de carrera, relacionado directamente con el desarrollo del proyecto.

Conocimientos sobre python

A lo largo de la carrera, se han estudiado diferentes lenguajes de programación (Ej: Java, C, CSHARP...), entre los cuales se encontraba Python. Por ejemplo en asignaturas como Sistemas Inteligentes, Nuevas tecnologías,

etc., o en mi paso por la politécnica de Varsovia, en asignaturas como “Algoritmia y computación matemática”, la cual, toda la asignatura consiste en un gran proyecto, donde se ponía a prueba los conocimientos algorítmicos en el lenguaje Python.

Además, multitud de las librerías usadas durante este proyecto fueron utilizadas en dichas asignaturas, haciendo que el proyecto, partiera desde un punto avanzado facilitándome el aprendizaje.

Conocimientos sobre la gestión de proyectos

A fin de cuentas, la metodología que se ha usado para idear, planificar y desarrollar el proyecto, es el conjunto y suma de una serie de actitudes adquiridas a lo largo de los últimos años. Ya sea con asignaturas dedicadas específicamente para ello como fue Gestión de proyectos, como con el prueba y error, que ha hecho que se fuese aprendiendo la mejor manera de trabajar a nivel personal. Permitiendo que este proyecto, fuese con diferencia el mas amplio y hasta la fecha, mas cómodo, a la hora de ser elaborado.

Metodologías como el SCRUM, Las pruebas de errores, la organización interna del scrum, las reuniones semanales sobre el estado del proyecto, etc.

Investigación

Básico, pero de vital importancia, es el nivel de capacidad de investigación y búsqueda de la información no es el mismo que al inicio del grado. La velocidad y capacidad, de saber buscar, es uno de los conocimientos más importantes que se adquieren a lo largo de la carrera. Conocimiento indirecto, pero presente en todas y cada una de las asignaturas que se han cursado, dicho nivel de investigación ha permitido que errores que antiguamente hubieran durado días, durasen horas, buscando en diferentes páginas y libros y contrastando cada información recibida a lo largo de este proceso de investigación

5.5. Conocimientos externos a la carrera

INTRODUCCIÓN

En este apartado, se explicarán los aspectos relevantes del desarrollo del proyecto externos a los conocimientos iniciales del alumno. Todo aquello que ha tenido que aprender para poder llegar al resultado final.

Psicología

Siendo sinceros, sería una parte de psicología, intercalada con otra parte de educación primaria. Para el desarrollo del proyecto, y el moldeado de lo que es la idea genérica de este; Ha sido necesario el estudio de ciertos campos para entender cual podría ser el mejor camino para el entendimiento con el usuario. Este proyecto no buscaba una forma rápida de llegar al cliente/usuario, sino constante, que el cliente convierta los cimientos de la aplicación, en su propio estilo de vida. Por ello se busco consejo a través de varias entrevistas a alumnos y graduados de psicología y magisterio, tras las cuales se llego al método: Aprender a aprender.

Aprender a aprender

Resulta ser uno de los mejores métodos de aprendizaje ausente que hay. Donde el aprendizaje ausente, como su propio nombre indica, consiste en el aprendizaje ausente de figura dedicada a la enseñanza, para poder llevar a cabo ese proceso evolutivo. Aquí el profesor, sería la aplicación, pero al no tener conciencia propia, ni libre capacidad de elección, el método de enseñanza habitual queda descartado.

La idea principal nace de las ventajas que tiene implementar un estilo de vida, sobre el seguir ordenes estrictas. Tras varias entrevistas, se decanto por un estilo incremental, en el que se intenta concienciar al usuario de cada decisión que toma, haciendo que el aprendizaje se base en pequeñas metas personales, y en la concienciación del usuario para que el mismo se de cuenta de que es lo mejor para él y aunque de manera lenta pero segura, llegue a la meta que tenga como objetivo.

Nutrición

Se tenían conocimientos básicos sobre el mundo nutricional y el movimiento “real-fooding” o comida real, que consiste en comer no solo las calorías necesarias para subir o bajar de peso, sino comida saludable; A efectos prácticos, esa es la idea principal sobre la que se fundamente el proyecto. Buscar la manera, de que el usuario, adquiera una mejor calidad de vida, a través de comida saludable, y que ya de paso, consiga sus objetivos principales.

Calidad"de los alimentos

Es una parte fundamental del proyecto, la calidad marca el umbral por el cual se cribaran los alimentos y se aconsejarán principalmente los alimentos

de mejor calidad, y esto irá variando conforme el usuario busque nuevos alimentos, u opciones.

Nutriscore

En un principio para que la calidad fuera lo mas precisa posible, y pensando en esta aplicación como proyecto de negocio, se pensó que, un nutricionista, experto en la materia, fuera quien añadiera los menús y la calidad de dichos. Esto a nivel de negocio, era la idea más optima, pero a nivel de proyecto, dejaba un vacío en la clara objetividad en la selección de calidad y en la limitación de la base de datos. Por esta razón se eligió el algoritmo del semáforo o NUTRISCORE, algoritmo que se basa en una serie de características buenas y malas de los alimentos, y según el resultado de la diferencia, la cual, se acaba cribando en uno de los cinco posibles valores: A,B,C,D Y E, ordenador de mejor a peor.

IMAGEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEN

De esta manera conseguimos que el usuario pueda añadir sus propios alimentos y no haga “trampa” con la calidad de la comida. Manteniendo la integridad de la aplicación lo máximo posible.

Imperfecciones

Este algoritmo no es perfecto, pues se basa única y exclusivamente en cifras, en las cifras de azúcar, de proteína, de sal, etc. Por lo cual, cosas saludables como puede ser el aceite de oliva virgen extra, tienen como resultado una pésima calidad, mientras que, quizás, ultraprocesados como cereales bajos en azúcares, bebidas edulcoradas, y cosas similares. Esto se debe por ejemplo, a que por cada 100 gramos, el aceite de oliva tiene un alto contenido en kilojulios y en grasas, por lo cual lo coloca en la categoría E, cuando debería ser un A.

Interfaces Gráficas

Pese a tener los conocimientos básicos sobre programación en Python, se desconocía cualquier tipo de implementación de interfaz gráfica. Debido a esto, gran parte del inicio del desarrollo del proceso, fue la selección de la librería que se utilizaría durante el resto del proyecto para la implementación de este.

Finalmente se escogió Tkinter, por su sencillez, pues permitió dar grandes avances y adaptarse a este nuevo sistema rápidamente. Permitiendo que la

interfaz se complicará de manera exponencial, aplicando todo aquello que se iba aprendiendo de manera colateral a lo ya existente.

5.6. Fases del proyecto

Las fases de este proyecto pueden ser divididas de diferentes modos, según que criterio se utilice, de por si podemos dividir todo el proyecto en tres grandes fases: Idea, desarrollo y testing. Teniendo en cuenta esto, hay que tener en cuenta que el desarrollo a su vez, se dividió en otras tres fases principales que fueron: creación de estructuras iniciales, desarrollo en interfaz de comandos y desarrollo en interfaz gráfica. Por lo cual un breve esquema del desarrollo podría ser:

- 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

Cronograma:

AÑADIR IMAGEEEEN DEL CRONOGRAMA

Trabajos relacionados

El mundo de la dieta como terapia para el tratamiento de distintas patologías se investiga desde hace años, tomando día a día mas fuerza e importancia. Al principio se pensaba que la dieta solo era efectiva para las patologías cuya causa estaba estrictamente vinculada a la alimentación del paciente como es el caso de la diabetes. Con los años se fue fundamentando que se podrían prevenir otros tipos de enfermedades con una buena alimentación, como era el caso de los problemas cardiovasculares, la Osteoporosis e incluso el Cancer. En la actualidad, numerosos estudios defienden que una adecuada alimentación no solo ayuda a prevenir y tratar enfermedades relacionadas con la alimentación, sino que es una parte fundamental del tratamiento y/o recuperación de enfermedades tanto livianas como importantes, como son la Obesidad, el Cancer, y la esclerosis multiple.

6.1. Estudios:

Estudios DASH

Los estudios DASH (Dietary Approaches to Stop Hypertension), son una serie de estudios relacionados para frenar la hipertensión a través una correcta alimentación, el primero de estos estudios se realizó en 1997. Realizado por la asociación del corazón americano (American Heart Association), el colegio americano de cardiología (American College of Cardiology), la sociedad contra el cancer americano y muchas otros colegios y universidades. A grandes rasgos se basa en la relación directamente proporcional entre comer una dieta rica en frutas y verduras y baja en azucares, grasas y carnes rojas reduce terriblemente la hipertensión y las enfermedades relacionadas

con el corazón.

Las pruebas de grasas saturadas

Son una serie de estudios, que se basan en demostrar la correlación entre las grasas saturadas y el incremento de factores de riesgo en enfermedades relacionadas con el corazón. Pronto se observó en estudios como: <https://www.sevencountriesstudy.com/> del que hablaremos más adelante, que la calidad de vida de las personas cuya alimentación era alta en grasas saturadas era bastante peor, teniendo una tendencia superior al padecimiento de enfermedades cardiovasculares.

By 1965 at the latest, it was beyond a reasonable doubt that if you replace saturated fats with polyunsaturated fats, you get a substantial lowering of total cholesterol. por Katan

(Traducción: En 1965 se probó que si se reemplazan las grasas saturadas por polinsaturadas se obtiene un decremento del colesterol total.)

El programa de prevención de diabetes (DPP)

Actualmente uno de cada tres adultos tiene diabetes, lo más significativo es que 9 de cada 10 no lo saben. Los resultados del DPP, se basaron en una prueba asignada de manera aleatoria a 3234 personas con diabetes que tomaron placebos, metformina (Una droga que baja el azúcar en sangre), o un “estilo de vida”.

La meta del estilo de vida consistía en perder peso y hacer ejercicio por lo menos dos horas y media a la semana. Para el momento en el que se realizaron fue un resultado increíble ver que el medicamento aumentaba en un 31 por ciento el riesgo de diabetes, mientras que el estilo de vida lo reducía en un 27 por ciento. Un estilo de vida había superado con creces los resultados del medicamento.

Pounds Lost

Estudio basado en la pérdida de peso, este estudio se realizó en 2004, y es uno de los estudios mas largos sobre la dietoterapia realizados hasta la

fecha. Consistía en observar entre 800 personas, que macronutriente habría que reducir para mejorar el estado físico y la salud, si grasas, proteínas o carbohidratos.

Fue un estudio que se realizó durante dos años, y se observó que los sujetos a observación habían adelgazado prácticamente lo mismo, y ninguno sufría ninguna deficiencia notable que indicase que un macronutriente era más necesario que otro. Se esperaba que las personas con una dieta baja en carbohidratos y con resistencia a la insulina obtuvieran mejores resultados, pero no fue así.

Entonces se llegó a la conclusión de que era porque todos estaban comiendo saludable, sin azúcares añadidos o hidratos refinados. Por lo que se demuestra que es increíble los avances que puede hacer la humanidad con una buena alimentación.

Seven countries Study

Fue el primer gran estudio que se realizó para relacionar la calidad de vida con la dieta y los posibles factores de riesgo que podría tener una mala alimentación sobre la salud de las personas. Nació en dos partes, primero se realizaron estudios en Italia, España, Sudáfrica y Japón de 1952 a 1956 y se sugirió, que los niveles de colesterol, las tasas de ataques cardíacos y las dietas variaban ampliamente y esto había que estandarizarlo de alguna manera, por ello en la “segunda parte”, se realizó un estudio piloto más forma entre 1956 y 57 en Finlandia, Italia y Grecia que indicaban que sin duda había una correlación entre estos tres factores y que debía ser analizados de manera efectiva. Esto fue solo el inicio del estudio, el posterior estudio, ya con las metas fijadas duró de 1958 a 1999, y también se dividió en dos partes:

1. Se realizaron encuestas sobre el estilo de vida y los factores de riesgo iniciales, y después de 5 y 10 años se realizó un seguimiento en 16 hombres de mediana edad de siete países.
2. La segunda fase se realizaron encuestas sobre la salud cardiovascular en ancianos de nueve cohortes europeas.

Este estudio proporciona evidencia de:

- El concepto de poblaciones sanas y enfermas
- Que los principales factores de riesgo cardiovasculares son Universales

- La hipótesis de la relación dieta-corazón
- Que la enfermedad cardiovascular es prevenible
- Que un estilo de vida saludable mejora varios ámbitos de la salud

6.2. Aplicaciones Similares

FatSecret

Aplicación Móvil y API de búsqueda, con una extensa base de datos, con toda la información nutricional de alimentos. Es muy usada en España para el seguimiento de dietas y el calculo de macronutrientes diarios ayudando a miles de usuarios. Pero no te indica que has de comer, ni que es bueno ni nada, es útil exclusivamente para aquellas personas con un conocimiento básico sobre alimentación saludable.

DietPro

Posiblemente lo más parecido al proyecto que se esta detallando en este documento. Es una aplicación de pago, que crea un seguimiento de dietas con mas de 3100 platos, y una gran variedad de alimentos, la cual, cambia en función a los objetivos y estados fisiológicos y patologías del paciente. Actualmente es una de las aplicaciones más potentes del mercado. El inconveniente es que se basa en dietas pautadas por exportes, cortando la versatilidad para las patologías y teniendo el principal problema de toda aplicación existente relacionada con la nutrición, y es que, esta basada en una dieta estricta, el usuario se puede hacer una idea de que es bueno o malo, pero sin saber muy bien porque, complicando el aprendizaje, creando necesidad de la aplicación, y el posible abandono al ser estricto.

MyFitnessPal

Aplicación bastante completa que además de ayudarte con tu alimentación con una base de datos, te ofrece una herramienta de gestión para tu actividad física, pero finalmente tiene el mismo problema que FatSecret, y es que necesitas de unos conocimientos básicos para que la aplicación sea realmente eficiente.

LifeSum

Planificador de comidas muy completo, que tiene una amplia gama de opciones entre las diferentes dietas y tipos, permitiendo llevar un gran seguimiento y ofreciéndote varias recetas entre las de la propia aplicación y su comunidad. El inconveniente es el mismo hablado hasta la fecha, y es que es un planificador, que no fomenta el aprendizaje y adquisición de un buen hábito de vida.

Conclusiones y Líneas de trabajo futuras

A continuación, se desarrollarán las diferentes conclusiones surgidas a lo largo de la creación del proyecto, a la par, de las líneas futuras de desarrollo que el proyecto a de seguir para la explotación a nivel tanto empresarial como informático.

7.1. Conclusiones

A continuación, se desarrollarán la conclusión que el proyecto ha ido dejando a lo largo de su creación:

- Aunque pueda pasar desapercibido, es la dificultad de inculcar, enseñar o mostrar un camino a un estilo de vida mas saludable a través de una aplicación para ordenador. En principio no se tienen ningún tipo de información, ni los recursos necesarios en cuanto a métodos pedagógicos, ni bases de datos que lo contemplen, por lo que se partió de cero, basándonos en diseños, parcialmente similares, pero sin asemejarse demasiado a lo necesitado. Resulto un reto conseguir encontrar la mejor forma de hacer que el usuario vaya aprendiendo a través de sus propias elecciones y errores.
- El algoritmo de recomendación, pese a todo lo aprendido a lo largo de la carrera, incluyendo entre las asignaturas, algunas donde se estudiaron los sistemas de recomendación, siempre se basaban en productos u usuarios con similitud entre ellos, dando unos pesos, y valorando cada

opción para poder hacer la recomendación. Este proyecto, al nacer desde cero, no se podía realizar ninguno de esos tipos de recomendación pues era necesario una gran cantidad de registros en la base de datos de la que no se disponía, además, no se buscaba una recomendación por similitud, sino, que al usuario se le recomendará cual era la mejor opción para el dependiendo de cada momento del día y de lo que llevará comido. Esto tan trivial, trajo consigo un inmenso trabajo de investigación para poder hallar una puntuación, que serviría como ranking de comida frente a las necesidades momentáneas del usuario.

- Algo que parece irónico, es que, al inicio del proyecto se decidió usar un Excel para simular la base de datos, y que así fuese más sencillo el tratamiento de los datos. De este modo, no se perdería tiempo en un ámbito ajeno al objetivo principal del proyecto que era el análisis y distribución de los datos para la enseñanza de dietoterapia. Pero lo que me encontré, es que trajo más problemas que soluciones, teniendo que tener yo en cuenta la integridad de la base de datos simulada, y haciendo al final, pensar más líneas sobre como simularla lo mejor posible, en vez de crear una entera, lo cual, durante el periodo de practicas del alumno, este aprendió a manejarlas con libertad en Python presentando bastante menos problemas, que el Excel.
- Fue complicado la idea de desarrollar un proyecto desde cero, a lo largo de los estudios del alumno, se han realizado varios programas de distinta índole, pero siempre se partía de una base dada por el profesor y un guion a seguir. La idea de crear un proyecto, y tener que ir desarrollando y creando una serie de objetivos, fue complicado, pues muchas veces eran objetivos absurdos, imposibles, o que se desviaban totalmente del sino del proyecto.
- La idea de crear por primera vez un diseño, parecía sencillo, pero cumplimentar todos los detalles concretos de una buena ergonomía, para evitar que el usuario se pierda en la aplicación resulto difícil.
- La recursividad, como se ha tratado anteriormente en estas memorias, debido a la interfaz gráfica de Python, se trabajó con una serie de funciones recursivas que iban variando ligeramente cada vez que se las llamaba, para poder ofrecer al usuario la mejor recomendación según su elección. Esto resulto un reto mas arduo de lo que se esperaba inicialmente, siendo por un momento, un factor que puso en riesgo el correcto funcionamiento del programa.

- El trabajo de investigación también fue complejo y extenso, sobre todo en el ámbito mas médico-alimenticio donde existen mil fuentes de información de las cuales muy pocas son fiables. Por ello se opto por una vía mas personal como es la búsqueda de profesionales en la materia que pudiera apoyarme.

7.2. Lineas de trabajo futuras

Al ser un proyecto ambicioso, que cubre dos mundos cada vez mas conexos, existen varios métodos de expansión de la aplicación para su correcta explotación.

Base de datos Online

Evidentemente, era uno de los principales puntos de desarrollo si se quería seguir explotando la aplicación, la necesidad de crear una base de datos, o servidor, que te permita no solo mantener los archivos del sistema a salvos, sino que además te permite el inicio de sesión del usuario desde cualquier ordenador el cual tenga descargado. Esto crearía una mayor versatilidad a la hora de su expansión y seria uno de los puntos clave de su avance.

Perfeccionamiento del algoritmo de recomendación

Actualmente es un algoritmo que tiene en cuenta bastante bien las necesidades de cada usuario en cada comida, pero que no está culminado del todo. ¿Qué pasa si para mi es más conveniente un macronutriente concreto?, o ¿sí, he comido mucho un alimento? La fórmula debería tener en cuenta el LRE y la calidad de la comida en pequeñas cantidades, y los nutrientes como las grasas saturas y el azúcar para una mayor precisión.

Aumento de la base de datos

Obviamente se parte de una pequeña muestra, de todos los posibles menús y patología que se podrían añadir a la base de datos. No deja de ser un Start-Up de un gran desarrollo, con los ejemplos básicos para la muestra de su funcionamiento.

Autoaprendizaje del programa

Uno de los campos que se quedó sin desarrollar por la falta de tiempo, es de una pequeña inteligencia artificial, o variable adaptativa vinculada al

usuario, la cual fuese aprendiendo para hacer que el usuario tuviera una experiencia más personal con el prolongado uso del programa. Se pensó en una variable almacenada en la base de datos (Como solución rápida), que si en un determinado periodo de tiempo el usuario no había evolucionado en su progreso hacia la meta del peso que quería esta variase en “X” calorías para ajustarse más a las necesidades del usuario, pues todo calculo existente hasta la fecha es demasiado genérico y no funciona para todos los usuarios por igual.

Adaptación a diferentes plataformas

Obviamente, es una aplicación de autoaprendizaje que intenta llegar al mayor número de personas, por lo que se debería expandir a diferentes plataformas, y no exclusivamente a una aplicación de escritorio. Principalmente, a su desarrollo para SmartPhones.

Desarrollo empresarial

Este subapartado es meramente explicativo, y solo propone posibles métodos para una futura explotación, no esta estrechamente relacionado con el desarrollo puramente informático, sino con su posible explotación como producto.

Si este proyecto quisiera verse en un ámbito empresarial para su desarrollo y explotación, habría que tener en cuenta una serie de medidas y mercados:

1. Como se habló antes la expansión de la aplicación a distintas plataformas multimedia (como Smartphones).
2. La constitución de versiones diferentes para su explotación
 - Free: Sustentada a través de anuncios y con una funcionalidad limitada
 - Pro: De pago mensual y con todas las funcionalidades posibles.
 - Enterprise: Funciones específicas, entregados a empresas dedicadas a la salud y nutrición y su posible clientela, venta por packs.
3. Creación de un sistema de atención al cliente para dudas, y contratación de expertos, para valorar la calidad de los platos, y no llevarlo a cabo a través del algoritmo Nutriscore evitando lagunas como el que este tiene.

4. Campaña de marketing
5. Contratación de personal, oficina, recursos, asesoramiento, para el lanzamiento del producto al mercado.