

PROYECTO:



Departamentos y empresas relacionados:



ALUMNO: Jesús Reseco Gago

TUTOR: Vicente Traver Salcedo



UNIVERSIDAD
POLITECNICA
DE VALENCIA

E.T.S.I.T. (Universidad Politécnica de Valencia)

RESUMEN:

“NUTRI-TRAINER” - Definición, desarrollo y validación de una solución para el aprendizaje gradual de temáticas específicas de salud

El objetivo del presente proyecto es la creación de una aplicación que mejore el aprendizaje de aspectos específicos relacionados con la salud de manera sencilla y práctica a través de la motivación, basándose en el ejemplo de soluciones de tipo ‘Brain Trainer’ o ‘Wii Fit’.

Estos productos comerciales han conseguido dos éxitos fundamentales, por un lado que segmentos de la población tradicionalmente alejados de la tecnología comiencen a utilizar plataformas TIC no habituales (i.e. consolas de videojuegos) o que utilicen las actuales para desarrollar actividades no habituales (i.e. móviles) y por otro lado, han permitido promocionar conceptos y actividades saludables a través de juegos interactivos de base sencilla. La idea de este PFC es utilizar estos ejemplos para construir nuestro propio ‘**Nutri-Trainer**’ que nos permita promocionar y evaluar el aprendizaje de diferentes grupos poblacionales en relación a hábitos saludables, recomendaciones de salud o tratamientos de enfermedades crónicas, sobre todo en aspectos relacionados con el estilo de vida (nutrición, actividad física, estrés, etc). El proyecto incluiría tanto la definición del concepto a través de la adquisición de los requisitos de usuario, el diseño de la interacción y los interfaces visuales, el desarrollo de la aplicación y su validación con usuarios reales utilizando las instalaciones del “Living Lab” del CIAMI (Centro de Conocimiento de Inteligencia Ambiental) que está disponible en el Instituto ITACA.

ÍNDICE:

Introducción	4
Objetivos del PFC	5
Metodología de trabajo del PFC	6 -13
● WPF	6
● Diseño y desarrollo: dos conceptos separados	6-7
● Microsoft Expressions	8-10
● Esquema de trabajo	10
● Plataforma	11
● Programación	12
● Base médica	13
● Juegos	13
Resultados previos del PFC	14
Conclusiones	15
Bibliografía	16

INTRODUCCIÓN:

A día de hoy, muchas personas intentan seguir una dieta equilibrada, médicamente comer de todo en su justa cantidad es lo ideal, sin embargo en la sociedad aún falta consciencia de lo que es una buena alimentación y, sobre todo, cómo llevarla a cabo sin recurrir a dietas, régimen o mitos sobre los alimentos que están tan instaurados que no nos damos cuenta de su falsedad.

Citando lo anterior, podemos añadir unos claros ejemplos:

¿Evitar hidratos de carbono para adelgazar? De ninguna manera esto es recomendable, ya que se corre el riesgo de presentar alteración de neurotransmisores cerebrales como serotonina, y desencadenar, especialmente al final de la tarde, crisis de “comer compulsivo” de pan, galletas, golosinas u otros alimentos ricos en hidratos de carbono y grasas. Los hidratos de carbono son algo que el cuerpo necesita y que, al no ser aportados en su justa medida, es fácil que al final del día acabemos comiendo algo que, aparte de carbohidratos, contenga otro tipo de nutriente como grasas o grasas saturadas.

¿El huevo crudo es útil para los deportistas? La práctica de agregar un par de huevos crudos al jugo del desayuno no sirve para nada. Para que el huevo sea nutritivo, requiere siempre cocción. Consumir huevos crudos equivale arrojarlos a la basura en prácticamente su totalidad.

Además de “desmitificar” algunas de estas afirmaciones y leyendas, Nutri-Trainer nos va a permitir aprender sobre la cantidad de nutrientes y el tipo que contiene cada alimento, permitiendo gráficamente compararlo con otros y establecer la relación existente entre varios tipos de comida.

Así pues, por ejemplo, comer pollo proporciona fundamentalmente proteínas, es una carne con poca grasa y bajo contenido calórico. Sin embargo la forma de prepararlo es también importante, por ejemplo, el pollo hervido puede llegar a contener un 29% de proteínas, mientras que frito un 18%.

Este tipo de consciencia es la que pretende conseguir la aplicación. Al ser un tema con cierto nivel de complejidad, Nutri-Trainer se presenta como un juego interactivo, con gran cantidad de movimientos, acciones, sonidos y color, con una presentación llamativa y algunos elementos o “gadgets” útiles, como reloj de cocina, marco de fotos, etc. Para que el proyecto sea más que un juego, para que el usuario lo vea como una herramienta útil con la que aprende a la vez que se entretiene e incluso compite con otros usuarios mediante un sistema de puntuación y logros que intentan crear una adicción sana a la utilización de Nutri-Trainer.

OBJETIVOS DEL PFC:

Los objetivos del PFC en cuanto al proyectista son los siguientes:

- Aprendizaje en la utilización de nuevas tecnologías.
- Documentación y relación con profesionales del campo de la nutrición, para la definición de la base médica de los juegos propuestos.
- Innovación en técnicas de aprendizaje.
- Interacción con potenciales usuarios finales para la definición del modelo de interacción de usuario.
- Diseño de la aplicación a todos los niveles: arquitectura, modelo de datos, interfaces de usuario, conectividad con los dispositivos externos del CIAMI (sensores, actuadores, interfaces).
- Desarrollo de la aplicación.
- Validación de la aplicación en el Living Lab: usuarios profesionales y usuarios finales.
- Propuesta de mejoras y líneas de investigación futuras.
- Soltura en el manejo y creación de bases de datos.
- Creación de una interfaz atractiva al usuario, que proporcione cierto interés y utilidad en el manejo de la herramienta.
- Elaboración de un tutorial para el departamento TSB, sobre la tecnología utilizada. Acompañado de un seminario a realizar por el proyectista.

METODOLOGÍA DE TRABAJO DEL PFC:

Para realizar este proyecto hemos empleado las ventajas que nos ofrecen las aplicaciones realizadas en .NET. Concretamente nos hemos centrado en una nueva tecnología y lenguaje, que combina la potencia del lenguaje C# con la facilidad de crear objetos de forma descriptiva como en Flash o utilizando .XML.

Todo ello deriva en la utilización de la emergente metodología de la programación que nos ofrece **Windows Presentation Foundation (WPF)** y **Microsoft Expression Services (ME)**.

WPF :

WPF Es una de las tecnologías Microsoft para el desarrollo de aplicaciones. Forma parte de lo que se espera de .NET framework 3.0 o superior. WPF combina aplicaciones, interfaces de usuario (UIs), gráficos 2D, 3D, documentos, multimedia, etc. Todo incluido en una única framework o entorno. Vamos a poder realizar UI de forma mucho más rápida, escalables y con una resolución independiente.

Los controles en WPF son de fácil composición. Se puede definir prácticamente cualquier tipo de control como contenido de otro, lo cual es una poderosa característica si se consigue usar de forma apropiada. Por ejemplo, poner *una imagen en un botón* o poner *una lista de vídeos en un combobox* para elegir un archivo de vídeo.

La resolución independiente en WPF es otra de las características mencionadas. Todos los tamaños en WPF son especificados en unidades “logical”, no en píxels. Un “logical” es aproximadamente 1/96 de una pulgada. Si incrementamos la resolución de la pantalla, la interfaz de usuario tendría el mismo tamaño proporcional y sin degradación de la calidad de imagen. WPF construye sus aplicaciones basándose en un motor de **renderizado vectorial**, por lo que los interfaces de usuario que realicemos van a resultar perfectamente ampliables sin prácticamente degradación de la imagen y manteniendo la proporción de los elementos.

Diseño y desarrollo, dos conceptos separados:

Una de las principales características es que WPF separa la parte de **apariencia** de un interfaz de usuario de su **comportamiento**. La apariencia normalmente vendrá especificada en XAML (Extensible Application Markup Language), el comportamiento (behaviour) es implementado en lenguajes como C# o Visual Basic. *C# será el que utilicemos para la realización de este*

proyecto, debido a la sencillez que proporcionan los lenguajes .NET y su capacidad multiplataforma (Windows), lo que nos permitirá migrar a una posible posterior solución a sistemas operativos como Vista o Windows 7. Las ventajas que la separación apariencia/comportamiento trae son:

- Ambas van a ser finalmente acopladas en el desarrollo.
- Permite que los desarrolladores y los diseñadores trabajen en módulos separados.
- Las herramientas de diseño gráfico pueden trabajar en documentos XML en lugar de realizar un análisis de código.

La separación apariencia/comportamiento establece una clara diferenciación entre el trabajo realizado por un **diseñador** y el de un **desarrollador**. El desarrollador simplemente tiene que ir añadiendo su código sin tener que tocar nada de la interfaz y el diseñador lo mismo pero en su terreno. Las herramientas que ambos usan son, por el lado del desarrollador [Visual Studio](#), en alguno de los lenguajes que soporta (como VB o C#), y por el lado del diseñador [Expression Blend](#) y Expression Design.

Anteriormente la forma de comunicación entre ellos era una imagen que creaba el diseñador de cómo debía más o menos aparecer la interfaz del programa y el desarrollador intentaba llevarla a cabo con algún lenguaje de alto nivel.

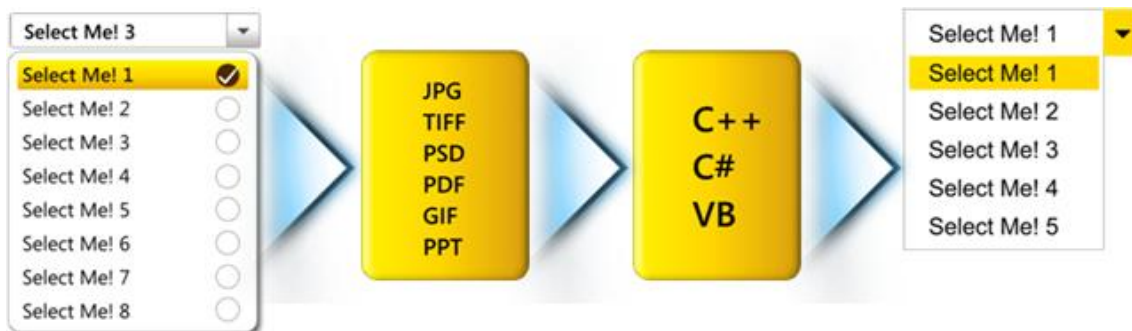


Ilustración 1: Esquema metodología

Ahora el diseñador, al diseñar la interfaz, la crea directamente siendo capaz de visualizar cómo va a quedar el proyecto final. La separación implica que el desarrollador va a poder coger esa interfaz, ya diseñada, y trabajar a partir de la misma para darle funcionalidad mediante un lenguaje de programación (comportamiento). En todo esto, existe un elemento de unión llamado lenguaje XAML.



Ilustración 2: Esquema nuevo metodología

XAML proporciona una forma declarativa de expresar objetos en un texto con sintaxis XML. Este lenguaje ha ido asociado a la tecnología WPF, aunque realmente ni WPF necesita XAML ni XAML está concebido solo para WPF. Es decir, podemos desarrollar una aplicación WPF en código C#, declarando los elementos y los manejadores de eventos de la interfaz directamente en código, aunque sería contraproducente no utilizar XAML para ello, ya que es una de las principales ventajas que ofrece WPF, proporcionando un entorno “Wysiwyg” para acercarnos a una solución final.

En WPF el lenguaje XAML se utiliza para representar los controles (objetos) que componen la interfaz de usuario permitiendo utilizar herramientas más relacionadas con aspectos de diseño, como **Microsoft Expression Blend**.

Microsoft Expressions

WPF consta de varias herramientas que van a poder permitir diseñar y desarrollar numerosas imágenes y eventos para las mismas. Dichas herramientas se encuentran en un paquete que, junto con “Visual Studio”, nos permitirán realizar animaciones, diseños y eventos. Este paquete es conocido como “**Microsoft Expressions**”, con el cual vamos a trabajar sobre todo en la parte de diseño de imágenes, botones, figuras y toda clase de elementos que nos servirán para establecer una UI atractiva. Microsoft Expressions está compuesto básicamente por 5 programas, entre los que destacamos **Design** y **Blend**, ya que son los que más hemos utilizado.

1. Microsoft Expression Design

Design, como su nombre indica, es una gran herramienta de diseño. Se focaliza sobre todo en el diseño y tratamiento de imágenes, ya sea desde cero o alguna que hayamos creado previamente con otra herramienta. Permite importar ficheros de imagen en todos los formatos: .psd, .ai, .jpg, .bmp, .png, .gif, .tif, .ico y .wdp (Windows Media Photo) y podemos retocarla para obtener resultados más atractivos de cara al usuario. Con el

diseño creado, podremos exportar en los formatos: .psd, .jpg, .bmp, .png, .gif, .tif, .wdp, .pdf y **.xaml**, lo cual supone una característica interesante para llevar los diseños a otros entornos (como Blend) o personalizar una aplicación en creación, añadiendo contenido enriquecido (código .xaml), con Visual Studio y WPF.

Los eficaces controles de exportación y segmentación (podemos segmentar trozos de imagen importada o creada para realizar un diseño más centrado) crean código XAML que funciona perfectamente con Expression Blend al crear experiencias de próxima generación, así como una amplia variedad de otros formatos, incluidos PDF y PSD. Expression Design es un eficaz portal de gráficos donde se concentran todos los recursos XAML.

Puede aprovechar las imágenes existentes incorporando gráficos de otras aplicaciones mediante formatos de archivo estándar (como JPEG, PNG y GIF), imágenes vectoriales abiertas de otras herramientas de gráficos conocidas o crear recursos vectoriales sofisticados a gran velocidad con herramientas de diseño intuitivas y flexibles, como son el pincel, pluma, cuentagotas de color,... cada uno con sus propias características y utilidades en cuanto al color, forma, opacidad, etc.

2. Microsoft Expression Blend

Blend, es la herramienta de diseño profesional para la creación de elementos e interfaces de usuario atractivas y llamativas, de forma relativamente sencilla. Es perfectamente compatible con Web y Silverlight (otros elementos pertenecientes a ME), además es la herramienta que mejor acompaña a nuestras imágenes diseñadas con Design, ya que podremos programar movimientos, desplazamientos y numerosos efectos a realizar en tiempo de ejecución. Expression Blend 2 permite trabajar con .NET Framework 3.5, sin embargo utilizaremos Expression Blend 3 debido a su similitud con la versión anterior además del añadido de mejoras sustanciales.

Es una herramienta intuitiva y sencilla. Consta de varias ventanas, las cuales podemos filtrar, para la personalización de controles o una UI completa de forma general. Permite combinar imágenes vectoriales y de mapa de bits, vídeo, audio, texto enriquecido, controles y contenedores, y crear elementos vectoriales personalizados con herramientas sofisticadas de colocación y ajuste con comentarios en tiempo real. Podremos tomar decisiones de creación inmediatas tras estudiar las vistas previas dinámicas de los elementos, incluidos los que están enlazados dinámicamente entre sí o a orígenes de datos externos. Posee **más de 30 controles y contenedores**, lo cual nos deja un gran potencial en la personalización de la interfaz en cuanto a la cantidad de componentes que podremos incluir y que están a nuestra plena disposición. La personalización de botones y otros controles es bastante fácil de realizar y da un aspecto muy dinámico y atractivo a las posibles aplicaciones.

Expression Blend presenta una escala de tiempo de máxima calidad basada en un eje temporal en lugar de en fotogramas. Con ella podremos **animar las propiedades individuales de cualquier objeto** a lo largo de una o varias escalas de tiempo configurando fotogramas clave de propiedades, y podremos utilizar la edición de trazados de movimiento en pantalla para controlar la trayectoria con toda precisión.

Al igual que con Design, tendremos nuestro código XAML para poder integrar la aplicación con Visual Studio, añadiendo el comportamiento o funcionalidad a los diversos componentes mediante lenguajes como C# o Visual Basic, en un proyecto WPF o Silverlight. De este modo, los diseñadores y programadores pueden crear y probar rápidamente revisiones de la funcionalidad y el diseño de la aplicación.

Esquema de trabajo

Para probar todo esto, actualmente en el proyecto estamos realizando dicha separación actuando por un lado la labor del **desarrollador**, que se encarga de elaborar una base de datos relacional con los usuarios posibles del programa además de dar funcionalidad a los objetos, y por otro lado el **diseñador**, que se encarga de dibujar todos los elementos de que constará la aplicación final.

Así, de esta manera, el proyecto va avanzando por dos frentes paralelos, diseño y desarrollo, juntándose bajo el lenguaje XAML para dar las animaciones y efectos necesarios para hacer una interfaz de mayor atractivo.

Todo es posible gracias a la gran interacción entre **Microsoft Expression Blend** y **Visual Studio 2008**, que permiten tener un mismo proyecto abierto en dos frentes distintos y actualizándose continua y automáticamente.

Para ir avanzando de forma segura y con vistas hacia posibles ampliaciones se ha basado el desarrollo en las capas básicas de la metodología de la programación clásica, como son:

- INTERFAZ DE USUARIO.
- CAPA DE NEGOCIO.
- BASE DE DATOS.

De esta forma, nos será posible si fuese conveniente conectar cualquier base de datos al proyecto sin modificar seriamente el contenido, o modificar la interfaz de usuario sin que ello afecte a la funcionalidad o capa de negocio en su mayoría.

Se baraja la posibilidad de conectar la base de datos de OASIS, proyecto que está siendo llevado a cabo por el mismo departamento y con la importancia que suscita la financiación por parte de la Unión Europea. Con ello el programa crecería exponencialmente en cuanto a riqueza en contenido y podríamos realizar llamadas numerosos datos y avisos importantes al usuario basándonos en esto.

Plataforma

Antes de empezar con el proyecto ha sido llevado a cabo un estudio de impacto visual en cuanto al modelo de interfaz gráfica. Para ello, diseñador y desarrollador nos hemos reunido en numerosas ocasiones para sugerir y debatir las distintas pantallas que se encontrará el usuario a lo largo del manejo de Nutri-Trainer. Varios han sido los intentos de conseguir una interfaz lo suficientemente atractiva hasta que se ha conseguido, no siendo un diseño definitivo pero sí acercándose bastante a lo que será la solución final.

La plataforma para la que se está desarrollando el proyecto es el dispositivo **Eee-Top** de Acer®, una ventajosa forma de dar dinamismo al programa, consiguiendo interacción total y natural con el usuario al tratarse de un ordenador con pantalla táctil integrada.

Se ha querido aprovechar su tamaño discreto para que sea un elemento posible de colocar en cualquier lugar de la casa, con la ventaja de que pueda estar situado en zonas comunes y visibles como la cocina o el salón y que de forma llamativa suscite la utilización del mismo, no cayendo el proyecto en el olvido y consiguiendo una continuidad, objetivo importante si lo que se pretende es conseguir un cierto aprendizaje en nutrición.



Ilustración 3: Dispositivo Eee-Top

Además de nuestra aplicación y sus “gadgets”, **Eee-Top** consta de numerosas aplicaciones interesantes que no solo lo convierten en un ordenador, si no en una plataforma dinámica. Intentaremos aprovechar estas características para dar más funcionalidad a Nutri-Trainer.

Sin embargo, durante el desarrollo se nos afrontan problemas, que aunque de sentido común, han sido un pequeña dificultad en el trabajo de diseño. Su pantalla “super-panorámica” ha necesitado de un especial cuidado en la dimensión de los objetos e imágenes que contendrá el proyecto, ya que siempre es interesante que el proyecto funcione también bajo otros dispositivos. El redimensionado entre pantallas de muy diversas características ha sido estudiado, evaluado y efectuado a grandes rasgos, y aunque hemos conseguido una solución satisfactoria debemos estudiar más en este ámbito con otro tipo de plataformas.

Programación:

En cuanto a programación, el lenguaje utilizado ha sido **C#** un lenguaje sencillo de utilizar bajo la tutela de **Microsoft Visual Studio 2008**. Durante la realización del proyecto se ha partido de cero en este lenguaje por parte del desarrollador, sin embargo se ha realizado en poco tiempo un proceso de aprendizaje, consiguiendo soltura en el manejo de la programación orientada a objetos y el manejo de bases de datos desde código, además de animaciones, propiedades en controles de usuario, hilos y efectos.

Siempre bajo el diseño se encuentra el nivel de funcionalidad, llevado por el desarrollador, haciendo que el juego presente una estructura fuerte y sostenible respaldada por un código potente. Si bien, algunos métodos y clases de WPF son distintos a los que nos podemos encontrar en Windows Forms, ha sido necesario recurrir en ocasiones al ingenio para solventar algunos problemas y encontrar las clases y/o eventos que buscábamos.

Tratándose de efectos, es especialmente interesante, como hemos remarcado, que la interfaz realice movimientos dinámicos para ser suficientemente atractiva. Para ello, en código C# se ha conseguido programar **hilos musicales**, controlables desde la pantalla, **teclado alfanumérico virtual**, ya que el dispositivo no cuenta con periféricos (cuenta con ellos pero su uso no es interesante para nuestro objetivo), funcionalidad de **reloj con la hora del sistema**, eventos de **Drag'n'Drop** (arrastrar y soltar) muy llamativos en algunos juegos, sobre todo tratándose de una pantalla táctil. Todo esto además de la funcionalidad de los distintos botones personalizados de los que consta la aplicación, el acceso a la base de datos, la carga de elementos como las imágenes y avatares, el resultado de los juegos en cuanto a la puntuación y la programación de los mismos se han realizado desde código C#.

El lenguaje **XAML**, antes mencionado, ha sido utilizado ampliamente para las labores de diseño y para algunos efectos dada la facilidad que presenta a la hora de crear y establecer animaciones con un eje temporal. Con **XAML** y el manejo de **ME Blend 3**, hemos conseguido realizar numerosos controles de usuario únicos en nuestra aplicación y muy llamativos, con movimiento, efectos y funcionalidad. El diseño de las pantallas y los distintos objetos de los que consta en su mayoría son creados e instanciados en código XAML, siendo posible el acceso a los mismos en el código C# asociado.

Cuando se crea una ventana o un elemento WPF en Visual Studio, automáticamente se crea un archivo **.XAML** y otro **.cs** asociado para código funcional, de ahí que la fuerte separación entre código y diseño sea establecida directamente desde esta tecnología.

No ha habido ningún problema con esto, es más, es una grandísima ventaja y XAML nos ha permitido realizar una primera versión de la aplicación en poco tiempo sin recurrir demasiado al código como hubiese ocurrido usando **Windows Forms**. Tras haber aprendido la metodología que supone WPF, nos hemos ido dando cuenta de lo útil que puede resultar a la hora de realizar cualquier tipo de aplicación, desde las más serias y formales, hasta las más divertidas informales, siendo un claro ejemplo de híbrido nuestro proyecto **Nutri-Trainier**.

Base médica

El proyecto no podría llegar muy lejos sin una base médica que lo respaldase fuertemente, ya que es importantísimo que la aplicación sirva de utilidad en cuanto a la salud nutricional. Para ello contamos con varios libros de nutrición y algunos nutricionistas profesionales a los que se les ha realizado entrevistas y charlas sobre nutrición, mitos, cantidades, dietas y más aspectos de la alimentación normal de la ciudadanía.

En la bibliografía se han añadido algunos documentos, páginas web y libros de los que se ha extraído la información, consciente de la importancia de no cometer ningún error que pueda inducir a una malformación nutricional, o crear nuevos mitos o leyendas lo cuales se intentan precisamente desmitificar.

El éxito del proyecto depende de que el aprendizaje sea correcto, despacio y sobre paso seguro, conduciendo al usuario a llevar una vida sana y recordando las propiedades nutricionales de los alimentos gracias a las técnicas asociativas que plantea Nutri-Trainer. Aprender jugando y de forma amena.

Juegos

Para poder presentar un prototipo funcional en Noviembre y luego realizar una posible ampliación se ha pensado en los siguientes cuatro juegos de aprendizaje.

- **Grupos alimentarios:** En este juego se intentará establecer una catalogación correcta de los tipos de alimentos que consumimos diariamente, estableciendo la relación mediante imágenes y objetos. Con eventos de Drag'n'Drop seleccionaremos el alimento correspondiente y lo colocaremos en el grupo alimentario al que creemos que corresponde. Será uno de los primeros juegos a implementar de hecho ya se ha empezado con su desarrollo.
- **Cantidades:** Intentaremos comparar la cantidad en cierto tipo de nutrientes como grasas o proteínas que tiene un alimento respecto a otro. Este programa permitirá al usuario recordar la proporción y la equivalencia entre distintos tipos de alimentos en cuando a llevar una dieta equilibrada.
- **Mitos y leyendas:** Juego sencillo de preguntas y respuestas sobre mitos y leyendas sobre la nutrición y temas alimentarios. Es muy interesante e irá acompañado de una explicación médica y científica para cada pregunta desarrollada.
- **Nutrientes:** Juego que consiste en seleccionar el alimento con mayor contenido en un determinado tipo de nutriente como vitamina C, B, hidratos, grasas, etc.

Estos son los juegos que en un principio se han pensado, la idea es implementar cierto nivel de dificultad incremental para que no resulten monótonos y que siempre supongan un reto para el usuario. Aún falta determinar la puntuación y los logros que se pueden conseguir con cada uno de ellos (actualmente en estudio).

En una futura ampliación en caso de que el proyecto siga adelante, se añadirán más juegos o más funcionalidad a los mismos.

RESULTADOS PREVIOS del PFC:

En principio, a día de hoy, se continúa en fase de desarrollo y diseño de algunos elementos. Si bien, hemos podido probar las pantallas de menú con personas del departamento y se han conseguido buenas impresiones. Sin embargo aún falta por desarrollar la parte de diseño y funcionalidad de los juegos. Se están desarrollando los juegos “Grupos Alimentarios” y “Mitos y Leyendas” ya que pueden ser los que generen mayor expectación.

Con objeto de tener un prototipo listo en Noviembre, se desarrollarán dos juegos completos, quedando los otros dos en una versión Beta o de prueba. Si el tiempo disponible lo permite, se desarrollarán más juegos y se añadirán mayor cantidad de extras y “gadgets” al programa.

Con un diseño aproximado se pretende ir en pocas semanas al “**Living Lab**” del TSB para efectuar una prueba con usuarios, poniendo en boca de los mismos posibles mejoras o añadidos. Además fotografiaremos lugares donde el dispositivo puede tener cabida dentro de un hogar familiar.

En cuanto a la base médica, falta aún mucho por documentar para que el proyecto esté lo más completo posible y sea un auténtico reto para el jugador. Siempre poniendo un límite, ya que dada la libertad que nos ofrece el programa puede ser tan extenso como queramos y como aprendamos sobre nutrición.

Con todo y algo más, la salida del prototipo no debería extenderse más allá de finales de Noviembre, quedando un diseño funcional aunque con alguna posible falta de mejora o ampliación.

CONCLUSIONES:

En los orígenes del proyecto, cuando meramente era una idea, la elaboración resultaba en cierto modo difícil y tediosa, sobre todo teniendo en cuenta que se partía desde cero y contando únicamente con los conocimientos de programación adquiridos en la carrera. Sin embargo, el acierto de utilizar la tecnología WPF ha dado sus frutos.

Se decidió estudiar detenidamente este tipo de tecnología, transcurriendo más de un mes hasta alcanzar un nivel suficiente. Transcurrido este tiempo y con la elaboración de un tutorial para poder afrontar futuros proyectos en el departamento, se han podido realizar pequeños “gadgets” y pruebas, algunos de los cuales se han incluido en el proyecto final. Tras esto, y un par de intentos de dar vida a un proyecto que partía de nuestras ganas e imaginación se consiguió una idea estable, con un fuerte apoyo y avanzando paso a paso hacia lo que sería y es Nutri-Trainer.

La aplicación intenta convertirse en algo más que un juego, ya que desde aquí pensamos que la nutrición hoy en día es un tema importante, social y de actualidad. Nutri-Trainer es una plataforma con la que aprender sobre alimentación a la vez que te diviertes, lo cual dado el éxito actual de aplicaciones y juegos análogos (“Wii Fit”, “Brain Trainer”), es una gran forma de aprender, mantenerte en forma y tener una vida saludable.

El dispositivo es una gran ventaja, ya que es de reducido tamaño, manejable y atractivo. Se valora implementar objetos como un reloj de cocina digital o un marco de fotos en el mismo para que no sólo se recurra a él para usar la aplicación, si no conseguir que acabe formando parte de la vida cotidiana, lo cual no deja beneficiar el uso diario de **Nutri-Trainer**.

Como valoración final, el proyecto **Nutri-Trainer** es un proyecto atractivo, destinado cuidar a la salud del ciudadano y la sociedad, concienciar a las personas sobre temas de alimentación y sobre todo hacerlo de forma divertida para que tanto adultos como jóvenes tengan esa curiosidad sana y el conocimiento de cómo llevar una alimentación equilibrada.

BIBLIOGRAFÍA:

- *Nutrición y alimentación:*

www.alimentacion-sana.org

www.isftic.mepsyd.es

<http://nutrimundo.blogspot.com>

<http://www.diabetesjuvenil.com>

TABLA DE CALORÍAS Y GRASAS

AUTOR/ES: Santos Berrocal Domínguez; Ulrich Klever
(Año 2004)

DESAYUNO Y EQUILIBRIO ALIMENTARIO. Estudio enKid. Vol. 1

AUTOR/ES: Serra Majem, Ll. / Aranceta, J.
(Año 2000)

- *Programación WPF y C#*

www.msdn.com

<http://expression.microsoft.com>

INTERFACES DE USUARIO AVANZADAS EN .NET

AUTOR: Manuel Llavador, Facultad de Informática, UPV
(Año 2008)

C# AND THE .NET 3.5 PLATFORM (4º Edición)

AUTOR: Andrew Troelsen
(Año 2007)

PROGRAMACIÓN CON VISUAL C# .NET

AUTOR: Francisco Charre Ojeda
(Año 2002)