

ESCOLA TÈCNICA SUPERIOR D'ENGINYERIA
ELECTRÒNICA I INFORMÀTICA LA SALLE

TREBALL FINAL DE MÀSTER

MÀSTER EN CREACIÓ, DISSENY I ENGINYERIA
MULTIMÈDIA

e

ID-STRESS. PROGRAMACIÓ.

ALUMNE
Leopold Riola

PROFESSOR PONENT
Oscar García
Emiliano Labrador

ACTA DE L'EXAMEN DEL TREBALL FINAL DE MÀSTER

Reunit el Tribunal qualificador en el dia de la data, l'alumne

Leopold Riola Bardají

va exposar el seu Treball Final de Màster, el qual va tractar sobre el tema següent:

iD-Stress. Programación.

Acabada l'exposició i contestades per part de l'alumne les objeccions formulades pels Srs. membres del tribunal, aquest valorà l'esmentat Treball amb la qualificació de

Barcelona,

VOCAL DEL TRIBUNAL

VOCAL DEL TRIBUNAL

PRESIDENT DEL TRIBUNAL

Abstract.

Català

L'objectiu d'aquest projecte ha estat dissenyar i implementar iD-Stress: una aplicació d'iPhone que ajudarà a gestionar l'estrès. S'ha realitzat en col·laboració amb el Centro Enlace. El projecte ha estat desenvolupat en un equip de quatre integrants, motiu pel qual aquest document presenta una part del total.

En aquesta memòria s'explicarà bàsicament la part tècnica de desenvolupament de l'estructura bàsica de l'aplicació, així com els apartats que no contenen animacions ni efectes.

Castellano

El objetivo de este proyecto ha sido diseñar e implementar iD-Stress: una aplicación iPhone que ayudará a gestionar el estrés. Se ha realizado en colaboración con el Centro Enlace. El proyecto ha sido desarrollado en un equipo de cuatro integrantes, por lo que este documento presenta una parte del total.

En esta memoria se explicará básicamente la parte técnica de desarrollo de la estructura básica de la aplicación, así como los apartados que no contienen animaciones ni efectos.

English

The purpose of this Project is the design and implementation of iD-Stress: an iPhone app which will be helping people managing their stress. It has been produced with the collaboration of Centro Enlace. This project has been developed by a team of four members, so this document solely represents a part of the complete project.

This document will mainly explain the technical part of developing the basic structure of the application, and parts that do not contain animations or effects.

Resumen.

Esta memoria es el resultado de un proyecto realizado para una empresa colaboradora del MCDEM (Máster en Creación, Diseño e ingeniería Multimedia), el Centro Enlace.

El proyecto ha sido desarrollado en un equipo multidisciplinar de cuatro integrantes (equipo Relajatech!), formado por un director de proyectos, un diseñador gráfico y dos técnicos. Por esta razón, este documento es una parte de la memoria técnica total.

El objetivo de este proyecto ha sido diseñar e implementar una aplicación iPhone que ayudará a gestionar el estrés mediante tratamientos de audio, proporcionados por Ana Lombard. Aún y así, el trabajo no sólo ha consistido en desarrollar la aplicación, sino que se ha tenido que generar una serie de apartados previos, posteriores y separados de los tratamientos que mejoren el resultado final dándole un resultado más completo.

En esta memoria se explicará básicamente la parte técnica de desarrollo de la estructura básica de la aplicación, así como los apartados que no contienen animaciones ni efectos. También se explicará como se han aplicado las aportaciones de los demás miembros del equipo y los recursos utilizados para todo ello.

Índice.

1. Introducción.	9
1.1. Estado del Arte.	10
1.1.1. Aplicaciones Relacionadas.	10
1.1.2. Conclusiones del Estado del Arte.	21
1.2. Marco Teórico.	22
1.2.1. Sector de la Movilidad.	22
1.2.2. iPhone.	23
1.2.3. Apple App Store.	24
1.2.4. Aplicaciones Móviles.	24
1.3. Descripción del Problema.	26
1.4. Solución.	27
1.5. Perspectiva General del Proyecto.	28
2. Fundamentos Teóricos.	29
2.1. Investigación Técnica Previa.	30
2.2. Recursos de Programación.	30
2.2.1. Objective C.	30
2.2.2. Cocos2D.	31
2.3. Herramientas de Programación.	33
2.3.1. Xcode.	33
2.3.2. Interface Builder.	33
2.3.3. iOS Simulator.	34
3. Parte práctica.	35
3.1. Estructura de la aplicación.	36

3.1.1. Implementación de cocos2D+UIKit.	36
3.1.2. Navegación.	37
3.1.2.1. Navegación entre Clases Principales.	37
3.1.2.2. Diseños Propuestos.	39
3.2. Apartados.	41
3.2.1. Tratamientos.	41
3.2.2. Elementos.	43
3.2.3. Juego.	45
3.2.4. Cuestionario.	46
3.2.5. Alertas.	49
3.2.6. Evolución.	51
3.2.7. Enlace.	52
3.3. Información.	53
3.4. Compartir.	55
3.5. Personalizaciones en la Interfaz Global.	56
3.5.1. Pantalla Inicial.	56
3.5.2. Barra de Navegación.	57
3.5.3. TableView Personalizada.	58
4. Conclusiones.	60
4.1. Problemas Encontrados.	61
4.2. Opiniones Personales.	63
4.3. Líneas de Futuro	64
5. Referencias	65

Figura 1: Captura de pantalla de Hästens MindSpa.	10
Figuras 2 y 3: Capturas de pantalla de la aplicación Hästens MindSpa.	11
Figura 4, 5 y 6: Captura de pantalla de la aplicación Stress Free.	12
Figura 7: Captura de pantalla de Natural Sounds For Me.	13
Figura 8: Captura de pantalla de la aplicación iSleep.	14
Figura 9: Captura de pantalla de Symbioline.	15
Figura 10: Captura de pantalla de la aplicación SymbioFeel.	16
Figuras 11 y 12: Capturas de pantalla de MyCalmBeat.	17
Figura 13: Captura de pantalla de la aplicación Zen Relax.	18
Figuras 14 y 15: Capturas de pantalla de iGuides-Eliminate Stress.	19
Figura 16: Captura de pantalla de 48 Hours Stres Relief.	20
Figura 17: Gráfico ilustrativo. www.allnewsmac.com	22
Figura 18: iPhone 4. www.liberaporimei.com	23
Figura 19: Aplicaciones de iPhone. www.customsolutions.es	25
Figura 20: Logotipo de cocos2D. www.cocos2d-iphone.org	31
Figura 21: Icono de Xcode. www.developer.apple.com	33
Figura 22: Icono de Interface Builder. www.developer.apple.com	33
Figura 23: Icono de iOS Simulator. www.developer.apple.com	34
Figura 24: Tab bar de cuatro pestañas. www.webstuffware.com	38
Figura 25: Vistas en un sistema Navigation. www.developer.apple.com	38
Figura 26: NavigationBar de la aplicación.	39
Figura 27: Vista de la versión con menú.	40
Figura 28: Vista de la pestaña “More”.	41
Figura 29: Vista de la lista de tratamientos.	42
Figura 29: Vista previa a los elementos.	43
Figuras 30 y 31: Vistas que utilizan los elementos guía.	44
Figura 32: Vista previa al juego.	45
Figura 33: Vista del cuestionario en curso.	46
Figura 34: Vista del resultado del cuestionario.	49
Figura 35: Vista de la lista de eventos.	49
Figura 36: Vista de EKEventEditView.	50
Figura 37: Vista de la evolución.	51
Figura 38: Vista de la información sobre el centro.	52
Figura 39: Vista de la información.	54
Figura 40: Vista de la valoración de un tratamiento.	55
Figura 41: Diseño de la pantalla inicial.	56
Figura 42: Diseño de la barra de navegación.	57
Figura 43: Diseño de la lista personalizada.	58

1. Introducción.

1.1. Estado del Arte.

Para conocer el estado del arte lo más común es realizar una búsqueda y testado de las aplicaciones similares o en el ámbito.

A continuación analizaremos las principales aplicaciones que se encuentran disponibles en la Apple Store y que abordan la temática de nuestro proyecto. Por ello hemos elegido las que, dentro de la categoría “Salud y Bienestar” tienen como objetivo principal la realización de tratamientos de relajación, hipnosis, ayuda al insomnio, gestión del estrés, etc.

Además, también mostramos aplicaciones que tienen una finalidad similar y se encuentran en otras plataformas, como web o Android.

De esta manera, podremos ver nuestras oportunidades de éxito y puntos débiles, para tomar decisiones acertadas en nuestro plan de marketing estratégico y posterior desarrollo de la aplicación.

1.1.1. Aplicaciones Relacionadas.

a) Hästens MindSpa iMeditation.

Desarrollador: Neurotech INC

Precio: 3,99€

Valoraciones: 27 valoraciones

“What a waste of time. It’s Awful” por el usuario Grinder70000007

Publicado: 15/01/11

Categoría: Salud y Forma Física

Tamaño: 98.8 Mb

Versión 1.0

Idioma: Inglés

Sitio web del autor:

<http://www.ibandler.com/>

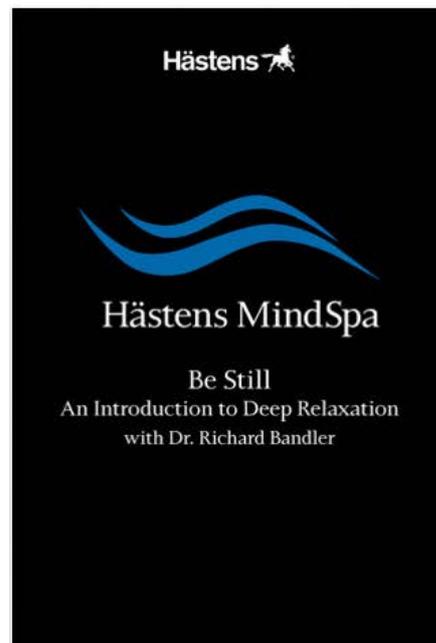


Figura 1: Captura de pantalla de Hästens MindSpa.

Utiliza la estimulación Auditorio-Visual (AVS). El audio y las imágenes están sincronizados de manera que el parpadeo de la luz y el sonido inducen a un estado de relajación. Según sus creadores “La luz y el sonido influyen

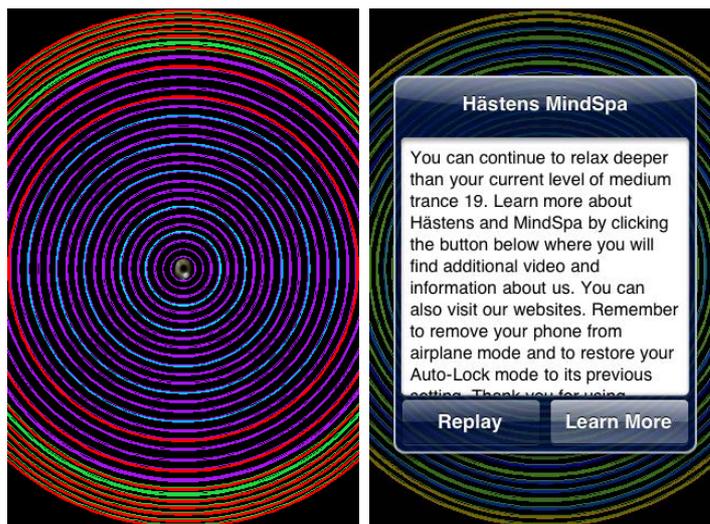
enormemente en la actividad de las ondas cerebrales. De manera natural, la mente encuentra modelos de luz parpadeante relajantes”.

La estimulación actúa sobre las ondas cerebrales beta, alfa, theta y delta.

Esta app se encuentra avalada por la disciplina del Dr. Richard Bandler, autor del libro “Get the life you want” y co-creador de la Programación Neurolingüística.

Hallazgo positivo: Se trata de una app interesante, a pesar de que no tiene ninguna opción configurable ni posibilidad de entender muy bien lo que el usuario está haciendo, aunque esto quizá también forme parte del programa de meditación que ofrece.

Hallazgo negativo: Tras probar la aplicación hemos visto que, durante los ejercicios, no hay posibilidad de detenerlos ni de obtener información acerca de lo que estamos haciendo. Tras unos 10 minutos de locución acompañada de la voz del Dr. Bandler, la aplicación se detiene y nos muestra una advertencia en forma de pop-up y nos ofrece la posibilidad de visitar la web para información sobre el autor.



Figuras 2 y 3: Capturas de pantalla de la aplicación Hästens MindSpa.

b) Stress Free, de Louise L. Hay

Desarrollador: OceanHouse Media, Inc.

Precio: 4,99€

Valoraciones: 1 valoración

“I adore it!” por el usuario Vicvin

Actualizado: 23/09/10

Categoría: Salud y Forma Física

Tamaño: 48.2 Mb

Versión 1.03

Idioma: Inglés

Enlace:

<http://www.oceanhousemedia.com/products/>

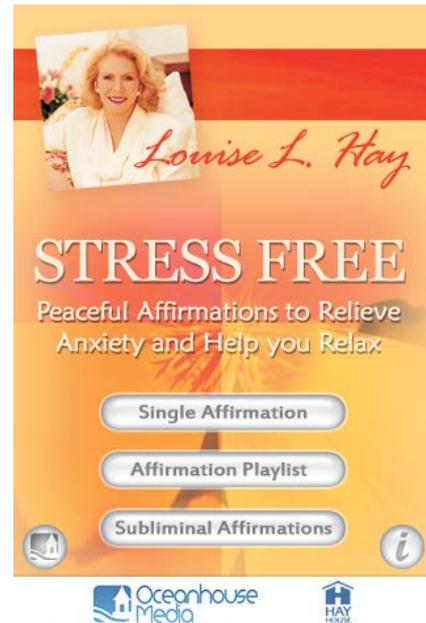


Figura 4: Captura de pantalla de la aplicación Stress Free.

Descripción: Se trata de una colección de 60 autoafirmaciones diseñada para aliviar el estrés y motivar cambios positivos en la vida de los usuarios. Ofrece la posibilidad de seleccionar una autoafirmación al azar o seleccionar una lista de locuciones.

Estas autoafirmaciones duran entre unos 20 segundos y están acompañadas de música. Incluye un audio de 30 minutos con música y afirmaciones subliminales (el usuario sólo escucha una melodía).



Figuras 5 y 6: Capturas de pantalla de Stress Free.

Hallazgo positivo: Instrucciones: Aquí nos ofrece información sobre las diferentes playlists y sobre cómo activar el temporizador para dormir. El temporizador es muy útil y sencillo de activar.

Hallazgo negativo: Durante la locución no se muestra más que el reproductor en la pantalla.

c) Natural Sounds For Me

Licencia: Creative Commons

Desarrollador: The Sound Waves of Nature.

Precio: ninguno

Idioma: Inglés, Polaco y Lativo

Enlace: <http://naturesoundsfor.me/>

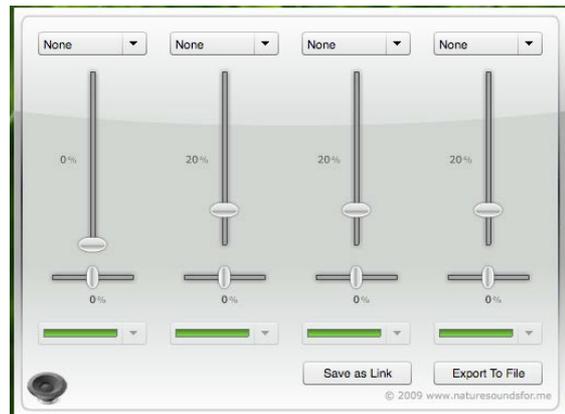


Figura 7: Captura de pantalla de Natural Sounds For Me.

“Natural Sounds for me” es una aplicación Web que funciona como un sencillo gestor de sonidos para ser utilizado cuando el usuario trabaja o estudia en el ordenador. Se aprecian sonidos de la naturaleza como: tormentas, olas, abejas, ríos, fuego, lluvia, entre otros. Incluso puedes reproducir hasta cuatro sonidos a la vez y luego salvarlos en tu ordenador para escucharlos offline.

Hallazgo positivo: Lo interesante de este producto es que intenta ser un banco de sonidos creado por los propios usuarios, es decir, cada usuario puede añadir sonidos que le resulten relajantes y de esta forma aumentar el número de sonidos disponibles.

d) iSleep

Desarrollador: Alejandro Luengo Gómez

Precio: 1,59€

Valoraciones: 8 valoraciones

“Está bien. La uso todas las noches y me va bien” por el usuario Fernando J. Expósito el 16/03/11

Actualizado: 25/01/11

Categoría: Salud y Forma Física

Tamaño: 133 Mb

Versión 1.1

Idioma: Inglés y Español

Enlace: <http://www.atomstudios.es/web/>



Figura 8: Captura de pantalla de la aplicación iSleep.

Descripción: iSleep es una aplicación que se basa en una intensa investigación de neurociencia. Utiliza tonos binaurales, que combinados con sonidos naturales y melodías, ayudan a dormir mejor, concentrarse y relajarse. iSleep Space está formado por 24 melodías o sonidos inspirados en el espacio, 5 sonidos naturales y 5 ondas cerebrales.

El usuario puede escoger varios de esos sonidos y mezclarlos a su gusto, para crear un sonido agradable. La aplicación dispone de un texto explicativo que detalla las propiedades de cada onda cerebral, para que el usuario las utilice según su necesidad. iSleep Space tiene muy buena crítica en la AppStore y es una de las aplicaciones más descargada en este ámbito.

Hallazgo positivo: iSleep tiene la opción de activar un “Modo automático” que modifica las combinaciones de sonidos aleatoriamente para crear ambientes totalmente distintos. También hay la opción de poner una cuenta atrás para desactivar la aplicación según el usuario estime que va a necesitarla.

Hallazgo negativo: Los sonidos son monótonos y se repiten continuamente. El modo aleatorio genera sonidos poco acertados.

e) **SYMBIOLINE**

Desarrollador: SIMBIOFI.

Precio: 399€

Categoría: Salud y Forma Física

Idioma: Inglés/francés

Plataforma: Windows XP/Vista

Contenido: Modulo experto de variabilidad de los latidos del corazón, ejercicios 3D.

E-Learning: Guía interactiva del protocolo ECG Bundle.

Enlace: <http://www.symbiofi.com/>



Figura 9: Captura de pantalla de Symbioline.

Es una aplicación multimedia para ordenador, diseñado para la educación psicológica del estrés y las terapias de autoayuda. Ofrece ejercicios básicos de relajación.

Hallazgo positivo: Permite al usuario controlar su estrés y ansiedad a través de un asistente de biofeedback cardíaco

Hallazgo negativo: Es extremadamente cara.

f) SymbioFeel

Desarrollador: Mental Workout

Precio: 7,99€

Valoraciones: No tiene

Actualizado: 15/12/09

Categoría: Salud y Forma Física

Tamaño: 82.4 Mb

Versión 1.1

Idioma: Francés

Enlace: <http://www.symbiofi.com/fr/produits/symbiofeel>

Enlace del desarrollador: <http://www.mentalworkout.com/>



Figura 10: Captura de pantalla de la aplicación SymbioFeel.

Descripción: Esta aplicación es la adaptación a iPhone de la aplicación web que hemos detallado anteriormente. SymbioFeel es un conjunto de aplicaciones en francés respaldada por el doctor Dominique Servant. La aplicación analizada es la de relajación, pero todas tienen el mismo tipo de funciones.

La aplicación dispone de cuatro apartados:

- En primer lugar, hay videos del doctor Dominique Servant, explicando una breve introducción sobre el tema y dando la bienvenida.
- El segundo apartado son consejos en formato texto sobre la meditación y relajación.
- En tercer lugar se puede encontrar el apartado de ejercicios.
- En el último apartado hay un test de autoevaluación de la relajación, que contiene 10 preguntas y muestra un resultado final y las puntuaciones anteriores.

Hallazgo positivo: Los ejercicios son instrucciones en formato voz. Ésta puede ser mezclada con una música de fondo ambiental y con un tema de imágenes (agua, naturaleza, cielo...), que serán mezclados con la voz durante el ejercicio. MentalWorkout, la empresa desarrolladora, ha ganado premios por sus aplicaciones en éste ámbito, colaborando con varios expertos y organizaciones especializadas.

Hallazgo negativo: Los vídeos tienen mala calidad. Tarda mucho en cargar el sonido de fondo.

g) MyCalmBeat

Desarrollador: e-Faces & Names

Precio: Gratuita

Valoraciones: 4/5 estrellas

Actualizado: 21/07/10

Categoría: Salud y Forma Física

Tamaño: 1,8 Mb

Versión 1.1

Sistema: Android

Idioma: Inglés

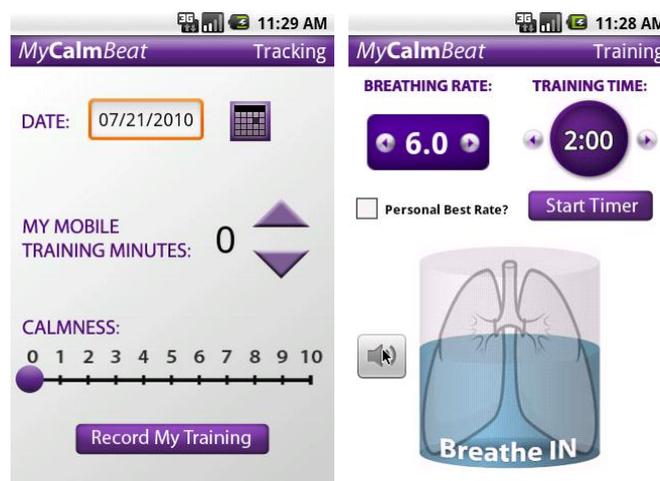
Instalaciones: 10.000 – 50.000

Web: <https://www.mybrainsolutions.com/mycalmbeat>

Es una aplicación móvil para Android que monitoriza la respiración del usuario. Muestra los datos de la tasa de respiración, y el tiempo en que se debe realizar.

Hallazgo positivo: Facilita guardar cada entrenamiento para ser continuado posteriormente. Está avalada científicamente.

Hallazgo negativo: No ofrece ningún extra, por lo que, tras un uso continuado, los usuarios pueden perder el interés.



Figuras 11 y 12: Capturas de pantalla de MyCalmBeat.

h) Zen Relax

Desarrollador: Avant Web Solutions S.L.

Precio: 2,39€

Valoraciones: Sin valoraciones

Publicado: 19/04/10

Categoría: Estilo de vida

Tamaño: 11.7 Mb

Versión 1.0

Idioma: Español

Enlace: <http://www.avantwebsolutions.com/mobileapps/zenrelax.html>

ZEN RELAX



Figura 13: Captura de pantalla de la aplicación Zen Relax.

Descripción: ZEN RELAX es una terapia de relajación guiada por la voz de un maestro Zen, que durante unos 20 minutos te transportará a un nivel de relajación absoluta. Tu mente recorrerá cada una de las partes de tu cuerpo.

La voz del maestro y el sonido de los elementos naturales que la acompañan, te ayudarán a reducir y eliminar tus tensiones, dolores o estrés.

Con Zen Relax descubrirás el poder de tu mente sobre tu propio cuerpo, llevándote a una dimensión hasta ahora desconocida.

Hallazgos Positivos: Antes de comenzar la escucha recomienda acomodarse y cerrar los ojos. La voz es cercana y agradable, natural, creíble.

Hallazgos Negativos: El audio de la voz no suena muy bien, no tiene buena calidad, se notan distorsiones y ecos, la acústica es bastante regular, es un audio como muy comprimido, sucio.

Los sonidos ambientales de fondo no son controlables. Pueden hacerse molestos los trinares de los pájaros tropicales. Solo tiene una terapia en forma de audio que es esencialmente una terapia de relajación que pretende funcionar para eliminar tensión, dolor y estrés. Al terminar vuelve a comenzar.

i) iGuides–Eliminate Stress

Desarrollador: Artic Gerbil Creations

Precio: 5,49€

Valoraciones: Sin valoraciones

Publicado: 04/05/09

Categoría: Educación

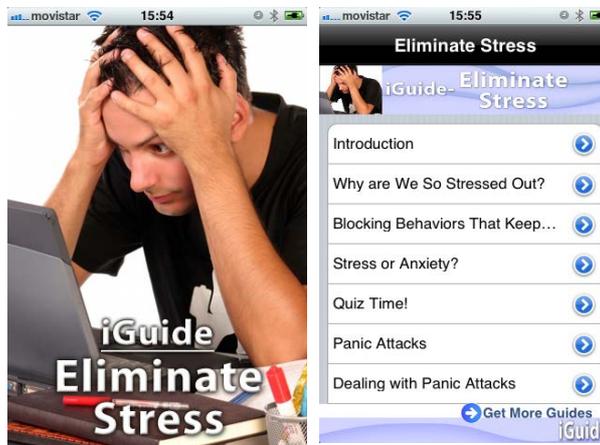
Tamaño: 0.4 Mb

Versión 1.0

Idioma: Inglés

Enlace:

<http://www.techrepublic.com/software/iguides-eliminate-stress-10-mobile/>



Figuras 14 y 15: Capturas de pantalla de iGuides-Eliminate Stress.

Descripción: Pareciera que lo escuchamos de cada persona que conocemos, todos dicen “¡estoy tan estresado!” Se trata de la presión que nos rodea en el mundo de hoy. Esas presiones nos causan estrés y ansiedad.

Uno de cada 8 norte-americanos entre los 18 y los 54 años sufren de desorden de ansiedad. Eso es un total de más de 19 millones de personas.

CARACTERÍSTICAS:

- Introducciones que ayudan a aprender
- Visualización por capítulos para encontrar rápidamente
- Instrucciones paso a paso fáciles de seguir
- Paso entre capítulos con botones de anterior y próximo

DIFERENCIAS

- Precio bajo
- Vínculos fáciles de acceso directo para todas las iGuides
- Fuente de información más completa.

Hallazgos Positivos: Información muy completa.

Hallazgos Negativos: Solo texto. Podría estar descrito antes de comprar que solo se trata de texto, una especie de libro o memoria sobre el estrés

j) **48 Hour Stress Relief**

Desarrollador: Michele Havlik

Precio: 3,99€

Valoraciones: Sin valoraciones

Publicado: 28/06/10

Categoría: Salud y forma física

Tamaño: 17.5 Mb

Versión 1.2

Idioma: Inglés

Enlace:

<http://www.techrepublic.com/software/48-hour-stress-relief-for-ipad-and-iphone-12-mobile/1874975>

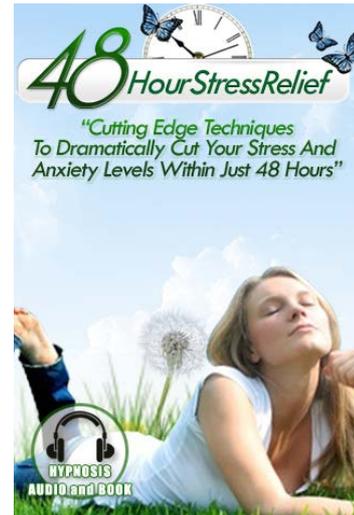


Figura 16: Captura de pantalla de 48 Hours Stres Relief.

Descripción:

Es básicamente un compendio descriptivo en texto sobre el estrés. Como un libro.

Hallazgos Positivos: Información muy completa. Contiene un audio hipnótico

Hallazgos Negativos: En la descripción en el App Store describe el estrés en lugar de la aplicación. Mucho texto. Podría estar descrito antes de comprar que solo se trata principalmente de texto, una especie de libro o memoria sobre el estrés.

1.1.2. Conclusiones del Estado del Arte.

Habiendo visto las aplicaciones más importantes en el ámbito, se ha podido observar que no hay ninguna que ofrezca todo lo que iDStress pretende ofrecer.

Analizando los contenidos de las aplicaciones se ha visto que se pueden encontrar algunas de las partes de nuestra aplicación en algunas aplicaciones ya existentes: tratamientos, test de estrés, interactivos e información, entre otros. Pero no se ha encontrado ninguna aplicación que contenga todos estos aspectos reunidos en una sola y que además ofrezca otros apartados, como la evolución o la alarma.

Algunas de las aplicaciones analizadas, que contienen audio-tratamientos, podrían ser una competencia a tener en cuenta. Creemos que, aunque el núcleo de una aplicación de este tipo debe ser el apartado de los tratamientos, se deben añadir algunos aspectos para que la aplicación destaque. Los apartados del test y información se encuentran en las aplicaciones más completas, mientras que una alarma y un mundo evolutivo hacen que nuestra aplicación innove y ayude a “engancharse” al usuario a nuestra aplicación, punto que las otras aplicaciones no cumplen.

Estos aspectos, son los que consideramos factores claves de éxito para la aplicación y que hacen distinguirla del resto de aplicaciones en el mercado.

Teniendo en cuenta los hallazgos positivos y negativos que se han encontrado en cada aplicación, se podrán evitar errores y anotar puntos fuertes para nuestra aplicación.

1.2. Marco Teórico.

1.2.1. Sector de la Movilidad

El concepto de movilidad en el ámbito de las TIC hace referencia al conjunto de tecnologías que permiten el acceso a la información y a servicios desde dispositivos inalámbricos, en cualquier momento y desde cualquier lugar. [1] [1]

En la actualidad, el sector de la movilidad está avanzando más rápido de lo esperado, tal como reconocía el mismísimo CEO de Google Eric Schmidt a finales de febrero en un encuentro anual del IAB [2], y eso teniendo en cuenta que Google/Android está siendo particularmente favorecido con la eclosión de este mercado cuyos límites se extienden cada año (móviles básicos (featured), smartphones, tablets, e-readers, netbooks...). Los datos de ComScore sobre España de 2009 a 2010 se pueden resumir así: [3]

- más de 13 millones de usuarios de Smartphones (por delante de Francia).
- 37% de penetración de smartphones (10% incremento respecto al año anterior).
- Fabricante líder: Nokia (47%), seguido de Samsung con un 15% .
- Número de líneas móviles respecto al nº de habitantes: 116%.

A continuación podemos ver un gráfico que ilustra las ventas de teléfonos móviles en el primer semestre de 2010 (superior) y los beneficios obtenidos por las diferentes empresas (inferior).

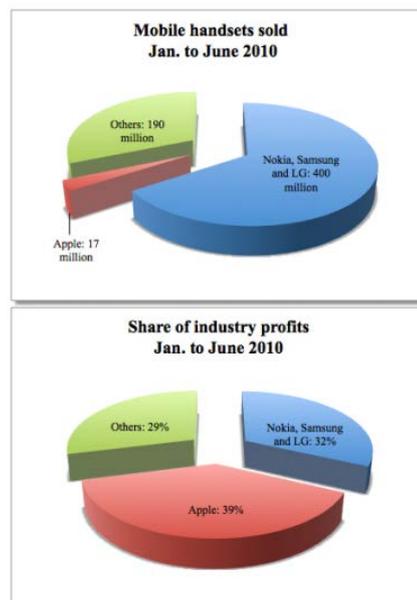


Figura 17: Gráfico ilustrativo.

Podemos observar que el mercado de smartphones cambia rápidamente como podemos leer en el artículo “La venta de smartphones supera a la de ordenadores por primera vez” del ABC del 9 de Febrero de 2011 [4], en donde nos dicen datos como que *“el número de teléfonos inteligentes durante el último cuarto de 2010 alcanzó los 100,9 millones de unidades, mientras que los de PC se quedaron en 92,1 millones, hablando siempre en términos globales.”* O que *“Los usuarios demandaron un 87,2% más de smartphones durante el trimestre analizado con respecto al mismo del año anterior, pero sólo un 5,5% más de ordenadores.”* También nos ofrecen datos interesantes como que *“Nokia volvió a ser el líder en cuanto a los teléfonos distribuidos en 2010, también durante el cuarto trimestre, seguida en este orden de Apple (iPhone), RIM (Blackberry), Samsung y HTC.”*

1.2.2. iPhone

El iPhone es un teléfono móvil GSM con pantalla táctil, pero también es un iPod (reproductor de música), una videocámara y un dispositivo con acceso a internet, al correo electrónico y a los mapas (ofrece posicionamiento y rutas guiadas por GPS).

Además, ofrece acceso a la App Store, con miles de aplicaciones disponibles.



Figura 18: iPhone 4.

1.2.3. Apple App Store

App Store es un servicio para el iPhone, el iPod Touch, el iPad, Mac OS X Snow Leopard y Mac OS X Lion, creado por Apple Inc., que permite a los usuarios buscar y descargar aplicaciones informáticas de iTunes Store o Mac App Store en el caso de Mac OSX, desarrolladas con el iPhone SDK y publicadas por Apple. Estas aplicaciones están disponibles para ser compradas o libres de costo, dependiendo de cada una. Las aplicaciones pueden ser descargadas directamente al iPhone o al iPod Touch por medio de una aplicación del mismo nombre, aunque App Store también está disponible al interior del programa informático iTunes. [5]

En cuanto al sistema de venta a través de iTunes hay un excelente artículo escrito por Eduardo Arcos publicado el 23 de Septiembre de 2010 para la revista electrónica Alt1040 llamado "Apple, iPhone, iTunes Store y por qué la cuota de mercado es irrelevante en estas épocas" [6] en donde nos dice que *"Apple impulsó un ecosistema inmenso alrededor del iPhone y iOS por medio de la venta de música, software, series de TV y películas por medio del iTunes Store y la inmensa extensión de utilidad del dispositivo por medio de aplicaciones que hicieron crecer aún más el ecosistema, dando vía libre servicios nuevos que antes estaban restringidos a la voluntad y decisión de hacerlo funcionar (o no) por las operadoras."*

Para el proyecto iD-Stress es muy importante considerar la App Store y sus "guidelines" [7], de esta forma, la aplicación podrá ser comercializada en este canal de ventas.

1.2.4. Aplicaciones Móviles

El ecosistema de las aplicaciones para móviles se ha convertido en una de las ramas del sector del mundo del software más activas [8]. Apple lanzó en 2008 su hoy afamada y exitosa App Store, tienda a la cual le siguieron otras siendo la segunda más representativa a día de hoy el Android Market.

Las aplicaciones móviles cumplen diversas funcionalidades, como por ejemplo el entretenimiento (juegos), las utilidades (gps, localización, fotografía, etc.), los libros (e-books); y también existe la categoría salud y bienestar (consejos sobre alimentación, salud, etc.). Además, permiten una conexión inmediata con las redes sociales (Facebook, Twitter, etc.)



Figura 19: Aplicaciones de iPhone.

Algunos datos relevantes para mostrar el potencial de este sector son los siguientes [8]:

- Solamente en la App Store se han descargado más de 4 mil millones de aplicaciones.
- Las encuestas muestran que de media un usuario de iPhone se gasta unos 62 euros en comprar aplicaciones.
- Las aplicaciones más rentables son los juegos. En la lista de Apple de las 20 mejores aplicaciones pagadas de todos los tiempos, 14 son juegos (de las 6 restantes 4 estaban dentro de las clasificadas como “Entretenimiento” y las otras dos en “Música”).

Como hemos ido viendo en los anteriores epígrafes que muestran el volumen del mercado del sector Smartphone y de Aplicaciones Móviles, nos encontramos ante un mercado con un altísimo potencial, tanto para las empresas que desarrollen aplicaciones para sus clientes, como para los propios desarrolladores.

1.3. Descripción del Problema.

Más de un 15% de personas (según los estudios más recientes, esta cifra podría alcanzar incluso el 25%) a lo largo de su vida sufrirán algún trastorno de ansiedad, como por ejemplo ataques de pánico y agorafobia, que lo padecen entre un 1'5 y un 3'5% de personas. El estrés laboral puede ser un factor de vulnerabilidad para llegar a sufrir este tipo de trastornos, aunque no es el único.

Dentro de los riesgos laborales de carácter psicosocial, el estrés laboral y el síndrome de quemarse por el trabajo (burnout) ocupan un lugar destacado, pues son una de las principales causas del deterioro de las condiciones de trabajo, y fuente de accidentabilidad y absentismo.

Las personas estresadas acuden generalmente en primera instancia al médico. Sus quejas más habituales suelen ser ansiedad, dolor y depresión. En cuanto a los depresivos, de los tipos que hoy en día más se consumen, en el mismo año se recetaron oficialmente más de 14 millones de envases. [9]

De las personas que acuden al médico de atención primaria, el 21% de los pacientes consume ansiolíticos y/o antidepresivos [10].

El problema social descrito anteriormente nos hace plantear este proyecto como una solución al mismo. Nuestro objetivo será ofrecer un producto destinado a las personas que sufren de estrés laboral, bien como complemento, para los usuarios que ya siguen algún tratamiento; bien como alternativa, para las personas que no quieren o no pueden acudir a un especialista.

Nuestra empresa colaboradora, la Clínica Enlace, nos propusieron un proyecto de desarrollo de una aplicación móvil de ejercicios interactivos con el objetivo de enfrentarse y regular el estrés del usuario hasta eliminarlo. El producto se descompondría inicialmente en varios ejercicios que pudieran escucharse, correspondiendo cada uno a una situación concreta.

La idea nació de la experiencia de Ana Lombard como experta en terapias de tratamiento de gestión del estrés, concentración, relajación, insomnio, etc. Basándose en la Sofrología. [12]

Tras muchos años de estudios y trabajo, ideó un sistema funcional para ayudar a que las personas puedan “auto curarse”. Se trata de que el paciente aprenda a no depender del terapeuta y empiece a realizar diversos ejercicios por su cuenta, de tal forma que aprenda y logre enfrentarse a sus problemas y dificultades mediante técnicas que establecen diversos niveles de compromiso entre las disciplinas de la relajación y la sofrología.

La mejor forma de llevar a cabo dicha labor es mediante una aplicación para móvil, ya que más de 13 millones de habitantes de Españoles poseen un smartphone [3]. Así pues, la aplicación propuesta pondría al servicio de todos el acceso a estas terapias.

1.4. Solución.

Nos proponemos acercar los tratamientos de Ana Lombard del Centro Enlace [11] a las personas interesadas sin necesidad de la presencia de un terapeuta. Queremos satisfacer al cliente externo y a los futuros usuarios, por lo que ha sido necesaria una profunda investigación para la creación de un producto innovador.

Los tratamientos propuestos por la Clínica Enlace son locuciones que dan las instrucciones necesarias para inducir a los pacientes a un estado de relajación. Dependiendo del tipo de terapia, los ejercicios estimulan el desarrollo del potencial del ser humano para generar armonía y sinergia entre los diferentes elementos que constituyen los valores existenciales del ser. Es decir, tratan de ayudar a establecer el equilibrio entre cuerpo y mente.

Para complementar las locuciones y crear una aplicación que ofreciese una experiencia interactiva memorable, se idearon varios extras que complementarían los tratamientos.

Esta aplicación, denominada iD-Stress se ha desarrollado para iPhone, y se descompone en varios módulos que desarrollaremos a lo largo de los siguientes capítulos/puntos. Para su desarrollo se tuvieron que tener en cuenta los requisitos de la Apple Store en cuanto a programación y reglas de publicación.

Por su plataforma de desarrollo y la temática que abarca, esta app va dirigida a un público adulto interesado en mejorar su bienestar y gestionar el estrés.

Este proyecto multimedia, además, debe cumplir las expectativas de creatividad, diseño e innovación uniendo las habilidades de los miembros de este equipo multidisciplinar.

En el caso personal, se debe desarrollar una parte de la aplicación: la estructura general y la navegación. Para poder incorporar los apartados desarrollados en cocos2D [13], aplicar diseños de interfaz y generar contenido estático (sin animaciones ni efectos), se necesita programar la parte llamada UIKit de la aplicación.

1.5. Perspectiva General del Proyecto.

En este apartado se introducirán resumidamente los puntos de la memoria que se desarrollarán a continuación.

El siguiente apartado, **2. Fundamentos Teóricos**, se detallan los aspectos técnicos que se han tenido en cuenta para el desarrollo del proyecto. El punto **2.1. Investigación Técnica Previa**, explica de forma resumida las herramientas y frameworks que se han investigado para decidir cómo programar la aplicación. En el siguiente apartado, **2.2. Recursos de Programación**, se describen los recursos (lenguaje y framework) que se han utilizado finalmente para la programación. Finalmente el punto **2.3. Herramientas de Programación**, comenta el software necesario para llevarla a cabo.

El **3. Parte práctica**, es el destinado a todo el trabajo de programación que se ha realizado. El epígrafe **3.1. Estructura de la aplicación**, trata del “esqueleto” de la aplicación, tanto la base del proyecto en Cocos2D como la navegación general en ella. Las distintas partes básicas de la aplicación se detallan a fondo en **3.2. Apartados**, y los aspectos generales programados son en los puntos **3.3. Información** y **3.4. Compartir**. Por último, los cambios gráficos que se han tenido que programar están en **3.5. Personalizaciones en la Interfaz Global**.

Finalmente el punto **4. Conclusiones**, hace una reflexión sobre el proyecto una vez realizado, analizando los **4.1. Problemas Encontrados** durante el desarrollo, las **4.2. Opiniones Personales** y las posibles **4.3. Líneas de Futuro** de la aplicación.

2. Fundamentos Teóricos.

2.1. Investigación Técnica Previa.

Para poder tener una visión más clara y realista de nuestras capacidades de programación y así poder hacer un brainstorming apropiado, se han investigado las distintas plataformas y librerías que permiten desarrollar aplicaciones para iPhone. También se ha investigado sobre el calendario de iPhone para la definición de la agenda, librerías para poder compartir en redes sociales y los distintos modos de hacer cuestionarios.

Así pues, se han encontrado las siguientes herramientas:

Cocos2D: Framework pensado para juegos en 2D. [13]

ShiVa 3D: Game engine multiplataforma de juegos 3D. [14]

iProcessing: Framework open source para aplicaciones nativas con Processing, utilizando gráficos y animaciones. [15]

Unity3: Motor gráfico para juegos 3D multiplataforma. [16]

Adobe Air: Entorno que utiliza las herramientas de Adobe y permite exportar para varias plataformas. [17]

Analizados los frameworks, se decidió emplear Cocos2D por las ventajas explicadas en el punto 2.2.2 *Cocos2D*.

Para hacer más fácil la expresión de nuestras ideas en el brainstorming con la empresa colaboradora, hemos realizado 5 bocetos utilizando "MockApp"[18], un sistema para crear maquetas de aplicaciones para iPhone. Para ello, se ha investigado sobre los distintos elementos de diseño de interfaz que ofrece apple (tableView, tabBar, navigationBar, entre otros) [20] y se han diseñado distintos modos de navegación.

2.2. Recursos de Programación.

2.2.1. Objective C.

Es un lenguaje de programación orientado a objetos, cuya característica fundamental consiste en su funcionamiento como un pequeño pero potente conjunto de extensiones para el lenguaje estándar C - comparte muchas características con ese lenguaje y por ello puede ser utilizado en conjunto con Objective C sin problemas-. Así, el desarrollador puede elegir en cualquier momento hacer una determinada tarea siguiendo las pautas de la programación orientada a objetos, o mediante técnicas de programación funcional.

Fue utilizado como lenguaje principal por la compañía NeXT, fundada por Steve Jobs, y actualmente es el lenguaje de programación utilizado en Mac OS X y iOS.

Alguna de las características que se pueden encontrar en este lenguaje son: Las interfaces y la declaración de métodos deben hacerse en bloques de códigos separados. No se llama a un método sino que se envía un mensaje. Existen dos tipos de archivos, el header (.h) y la clase (.m) Se deben utilizar signos: positivo (+) si se refiere a los métodos de una clase o métodos que puede ser llamados sin ser instanciados en una clase. Y negativo (-) cuando se denota la instanciación de los métodos que solo pueden ser llamados entre una instancia particular de una clase.

Se distingue un archivo objective C de uno escrito en C por la extensión del fichero (.mm). Muchas decisiones que en otros lenguajes se toman en tiempo de compilación, Objective-C las deja para tiempo de ejecución Las variables siempre se crean en memoria dinámica, evitando así el desbordamiento de pila. [19]

2.2.2. Cocos2D.

Cocos2D for iPhone es un framework creado para diseñar y construir aplicaciones interactivas gráficas, demos y juegos 2D para iPhone. Desarrollado por el argentino Ricardo Quesada basándose en el diseño Cocos2D de python para ofrecer una herramienta con las siguientes características:

Fácil uso: pues utiliza una API (Application Programming Interface) familiar (Xcode) y posee una gran cantidad de ejemplos que resultan de gran utilidad para principiantes.

Rápida: pues emplea OpenGL ES (variante de la API OpenGL) que permite la generación y control de gráficos en dispositivos móviles y estructuras de datos optimizadas.



Figura 20: Logotipo de cocos2D.

Flexible: permite ser integrada facilmente en librerías de terceros. **Gratis:** es una herramienta open source, compatible tanto con fuentes de juegos abiertos o cerrados.

Comunidad: de gran tamaño, amistosa y activa que darán soporte a las dudas de programación que puedan surgir.

Aprobada por la AppStore: el punto más importante de todos pues existen actualmente 2500 juegos en la AppStore que han sido desarrollados con Cocos2D for iPhone. Entre las prestaciones que encontramos en Cocos2D tenemos:

- Transiciones de escenas
- Sprites (imágenes)
- Efectos (Ripple, Lens, Waves, etc)
- Sistema de Partículas
- Sistemas físicos (gravedad, fricción, rebotes)
- Renderizado de texto
- Soporte de Audio
- Soporte Touch/Acelerómetro – Teclado/Mouse

Puesto que utiliza como lenguaje de programación Objective C se pueden utilizar otras librerías ya desarrolladas en este lenguaje o importar librerías implementadas en lenguaje C, ya que también es soportado por el IDE Xcode.[13]

2.3. Herramientas de Programación.

Todas las herramientas de programación utilizadas para la realización de una aplicación para iPhone se encuentran en el iPhone SDK [21] de Apple.

El iPhone SDK es un Kit de desarrollo de software para la plataforma iOS de Apple Inc., con el objetivo de permitir a terceros desarrollar aplicaciones nativas para el iOS. Fue liberado en Febrero del 2008, ha sido objeto de críticas al estar solo disponible para Mac OS X.

2.3.1. Xcode.

Es un entorno de desarrollo o IDE (siglas en inglés), para Mac OS X que es utilizado para el desarrollo de aplicaciones de dicho sistema operativo y para el sistema operativo móvil iOS, trabaja conjuntamente con Interface Builder y el iPhone Simulator que genera las interfaces gráficas de usuario y permite trabajar con proyectos escritos en C, C++ , Java y Applescript.

Características: Crear y administrar proyectos, incluyendo plataforma, dependencias y requisitos del dispositivo. Escritura de código en un editor con opciones para la sintáxis e indentación. Depuración del proyecto, bien sea en el simulador, en el sistema operativo o en el dispositivo móvil. [21]



Figura 21: Icono de Xcode.

2.3.2. Interface Builder.

Interface builder es una aplicación que permite editar manualmente las vistas de interfaz creadas en Xcode. Contiene una librería con los elementos gráficos principales y permite relacionar estos elementos con propiedades y acciones programadas. El archivo resultante es de tipo “.xib”. [21]



Figura 22: Icono de Interface Builder.

2.3.3. iOS Simulator.

iOS Simulator es un simulador de iPhone que permite ejecutar aplicaciones programadas para iPhone en el mismo ordenador en el que se programa. Actúa de forma muy parecida al iPhone aunque se conocen algunas diferencias entre la ejecución de aplicaciones en el simulador y el dispositivo real.

Junto con la herramienta “console” de Xcode, es la base para testear aplicaciones y debugarlas. [21]



Figura 23: Icono de iOS Simulator.

3. Parte práctica.

3.1. Estructura de la aplicación.

3.1.1. Implementación de cocos2D+UIKit.

Para poder implementar las funciones de cocos2D [13] sin perder las facilidades y posibilidades que ofrece UIKit [20], y dado que cocos2D es una plataforma pensada para ser la base de las aplicaciones que lo implementen, se ha tenido que ejecutar la aplicación como si ésta fuera principalmente en cocos2D. Una vez ejecutada la aplicación, ésta esconde directamente la capa de cocos2D para mostrar la vista general, realizada en UIKit.

El siguiente código muestra como, después de ejecutar la aplicación en base a Cocos2D (parte que viene escrita por defecto en cada proyecto de cocos2D), se muestra la parte realizada en UIKit:

```
// make the OpenGLView a child of the main window
[window addSubview:glView];

// make main window visible
[window makeKeyAndVisible];

GameScene *gs = [GameScene node];

[[CCDirector sharedDirector] runWithScene:gs];
General *principal= [[General alloc] init];
self.general = principal;
[principal release];

[self showUIViewController:general];
```

Cocos2D seguirá en segundo plano hasta que la aplicación mande un mensaje de tipo NSNotification al appDelegate (clase principal de toda aplicación) y éste haga aparecer una escena de cocos2D según el mensaje recibido. Ésto se aplicará, de forma inversa, cuando se quiera volver a mostrar la vista general.

Un ejemplo de código de la función `trackNotifications`, donde se muestra como gestiona el mensaje que recibe de Panic Button:

```
- (void) trackNotifications: (NSNotification *)
notification
{
    id nname = [notification name];

    if([nname isEqual:@"empezarPanicButton"]){
        [self hideUIViewController:general];

        [[CCDirector sharedDirector] pushScene:
        [CCTransitionMoveInB transitionWithDuration:2.0f
        scene:[HelloWorld scene]]];
    }
}
```

3.1.2. Navegación.

3.1.2.1. Navegación entre Clases Principales.

La aplicación consta de 7 apartados básicos:

- Tratamientos: Ejercicios en audio proporcionados por Enlace.
- Cuestionario: Test para saber el nivel de estrés.
- 4 elementos: Interactivos con los elementos de la naturaleza.
- Juego: Juego corto y rápido.
- Evolución: Imágenes que muestran la evolución del usuario.
- Alerta: Agenda de eventos con alarmas.
- Centro Enlace: Información sobre el centro.

Para navegar entre ellos, se escogió utilizar como sistema principal el llamado Tab Bar [22] por ser el más común en las aplicaciones con numerosos apartados y por permitir una navegación mucho más rápida, intuitiva y ordenada por la aplicación. Una tab bar es una barra con pestañas o iconos que se encuentra en la parte inferior de la pantalla permaneciendo siempre visible. En ella se pueden encontrar los distintos apartados de la aplicación, para poder navegar entre ellos y saber en qué apartado se encuentra en todo momento.



Figura 24: Tab bar de cuatro pestañas.

Cada pestaña muestra una vista de uno de los apartados. Estas vistas pueden contener otros tipos de navegación como Navigation [20], una barra de navegación situada en la parte superior de la pantalla y que contiene un botón de regresar a la vista anterior y además puede contener otros botones y un título.



Figura 25: Imagen explicativa de las vistas y barras en un sistema tipo Navigation.

En esta aplicación, todos los apartados son Navigation Views, lo que significa que el usuario puede volver atrás en todo momento utilizando la barra superior de la pantalla. Ésta contiene además un botón de información o ayuda que sirve para orientar y dar información extra a los usuarios acerca de algunas operaciones que estén realizando.

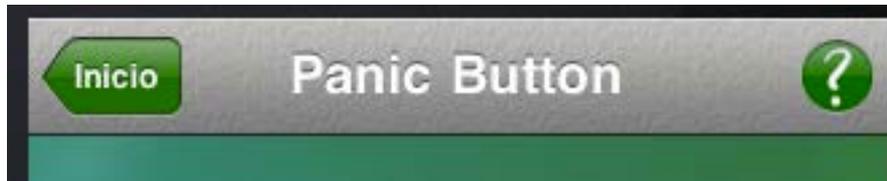


Figura 26: NavigationBar de la aplicación.

3.1.2.2. Diseños Propuestos.

Para el primer boceto de la aplicación, se propuso una estructura de navegación basada en una pantalla “inicio”, que contiene un menú con los cuatro apartados más importantes de la aplicación. Este menú y los demás apartados se encuentran en la barra inferior de la pantalla (tab bar [20]), del siguiente modo:

- Menú inicio
- Evolución
- Enlace
- Alerta
- Juego

Así, los apartados que se encuentran en dicho menú son:

- Tratamientos
- Cuestionario
- 4 elementos
- Panic button (uno de los tratamientos que, a petición de la empresa colaboradora, tenía que ser muy fácilmente accesible).

Esta estructura permite tener menos pestañas en el tab bar y que queden más destacados los apartados importantes de la aplicación. Su mayor desventaja es que, si el usuario navega varias vistas en el menú inicio, podría tener dificultades en encontrar los apartados principales ya que tendría que usar la barra de navegación varias veces.



Figura 27: Vista de la versión con menú.

Sin embargo, después de la investigación de las otras aplicaciones para iPhone en el mercado, pudimos ver que las aplicaciones que contienen una tab bar, rara vez disponen de un menú “inicio”. A diferencia de nuestra primera propuesta, las aplicaciones suelen tener todos los apartados ubicados en la tab bar, agrupando los menos importantes en una pestaña que xCode [21] crea por defecto llamada “More”. En ella, los usuarios pueden encontrar los apartados “escondidos” y editar cuáles son los que quiere que sean siempre visible y cuáles no.

La versión con navegación estándar contiene de forma visible por defecto, en su tab bar:

- Tratamientos
- Cuestionario
- 4 elementos
- Juego.

Los otros tres apartados se encuentran en la pestaña “More”, aunque esta distribución podrá ser reordenada por el usuario en todo momento.

De este modo ningún apartado pierde importancia y está siempre accesible directamente desde la tab bar. Por otra parte, en el caso que el usuario no sea experto en iPhone, puede suponer un problema encontrar los apartados en la pestaña “More”, y la posibilidad de edición es desconocida por la mayoría.



Figura 28: Vista de la pestaña “More”.

Así pues, manteniendo la primera versión, se realizó una estructura de navegación más estándar, siguiendo las guías de las otras aplicaciones que se encontraron en la appStore. A partir de estos dos modelos, se pudieron hacer los tests con usuarios utilizando ambas versiones, para poder comparar sus distintas opiniones y comprobar en qué entorno el usuario se sentiría más cómodo.

Tras realizar varios tests y hablar con la empresa colaboradora, se decidió utilizar la primera versión ya que resultó ser más intuitiva y agradable visualmente.

3.2. Apartados.

3.2.1. Tratamientos.

El apartado de los tratamientos se inicia con una lista (llamada table view [20]) de todos los tratamientos. En ésta aparece el nombre, descripción, icono y duración de cada tratamiento (detalles sobre la lista en 3.5.3. *TableView Personalizada*).

Una vez el usuario ha seleccionado algún tratamiento se muestra, mediante la función `pushViewController`, una ventana propia de ese tratamiento. Dicha vista contiene una breve texto acerca del tratamiento y un botón para empezar el tratamiento.

Ya que la reproducción de los tratamientos se ejecuta en Cocos2D, la llamada a éste se realiza mediante un mensaje `NSNotification` (como se ha comentado anteriormente) y se carga seguidamente una nueva ventana que se mostrará al finalizar el tratamiento.

Un ejemplo del código que se ejecuta cuando se va a reproducir un tratamiento:

```
- (IBAction) empezar: (id)sender{
    NSLog(@"empezar");
    [[NSNotificationCenter defaultCenter]
postNotificationName:@"empezarPanicButton" object:@""];
    PostTratamientos *post =[[PostTratamientos alloc]
initWithNibName:@"PostTratamientos" bundle:nil];
    [self.navigationController pushViewController:post
animated:YES];
    [post release];
}
```

La última vista en todo este proceso, sirve para que el usuario pueda dar una valoración a la eficacia del tratamiento. Ésto se realiza mediante una barra deslizante (UISlider [20]) que utiliza el valor dado para actualizar el “nivel” de la vista Evolución (ver 3.2.6 *Evolución*). Además, hay dos botón que permite compartir la valoración en twitter i Facebook (detallado en el punto 3.4. *Compartir*).



Figura 29: Vista de la lista de tratamientos.

3.2.2. Elementos.

Este apartado permite al usuario escoger, mediante cuatro botones con imágenes de piedras dibujadas, uno de los cuatro elementos. Como éstos son interactivos programados en cocos2D [13], las acciones de los botones únicamente generan un mensaje que indica qué elemento ejecutar y preparan, como en el caso de los tratamientos, una ventana posterior a la interacción.



Figura 29: Vista previa a los elementos.

El siguiente código muestra la primera parte de la acción que realiza el botón de agua: manda el mensaje y llama a la vista siguiente. Los demás elementos siguen el mismo formato.

```
- (IBAction) agua: (id)sender{  
    [[NSNotificationCenter defaultCenter]  
    postNotificationName:@"empezarEfectoAgua" object:@""];  
    PostElementos *post =[[PostElementos alloc]  
    initWithNibName:@"PostElementos" bundle:nil];  
    [self.navigationController pushViewController:post  
    animated:YES];  
    [post release];  
}
```

Una vez el usuario ha terminado de interactuar con el elemento, este elemento será guardado para así adaptar todos los iconos del avatar guía de la aplicación. Éste proceso se efectúa mediante `NSUserDefaults`, el método para guardar información sencilla.

`NSUserDefaults` guarda cualquier tipo de información en un apartado interno de la aplicación dentro de iPhone. Esto permite que no se pierdan estos datos cuando la aplicación es cerrada y habilita un fácil acceso desde todas las clases a dichos datos. En el caso de los elementos, se ha guardado un identificador numérico (del 1 al 4) para simplificar código.

El siguiente código muestra la segunda parte de la acción de los botones de los elementos, donde guarda un entero para la clave "elemento" en `NSUserDefaults`:

```
NSUserDefaults *prefs = [NSUserDefaults
standardUserDefaults];

[prefs setInteger:1 forKey:@"elemento"];
```

Una vez guardada esta información, ésta afectará a todas las vistas que contengan un avatar. Para que esto suceda, se tiene que consultar el `NSUserDefaults` en cada método `viewWillAppear` (método que se ejecuta cuando se va a mostrar una vista [22]) y cambiar la imagen acorde con esta información (el último elemento con el que se ha interactuado).



Figuras 30 y 31: Vistas que utilizan los elementos guía.

Un ejemplo del código que se incluye en todas las funciones `viewWillAppear [22]` de las ventanas con elementos:

```
NSUserDefaults *prefs = [NSUserDefaults
standardUserDefaults];

NSString *imagen;

switch ([prefs integerForKey:@"elemento"]) {
    case 1:imagen = @"Avatar_Agua-e.png";break;
    case 2:imagen = @"Avatar_Fuego-e.png";break;
    case 3:imagen = @"Avatar_Tierra-e.png";break;
    case 4:imagen = @"Avatar_Aire-e.png";break;
    default:break;
}

guia.image = [UIImage imageNamed:imagen];
```

3.2.3. Juego.

Al igual que en el punto anterior, este apartado está basado en Cocos2D. En el caso del juego, únicamente aparece un botón para empezar a jugar que, como en los casos anteriores, crea un mensaje `NSNotification`.



Figura 32: Vista previa al juego.

3.2.4. Cuestionario.

El cuestionario, especificado por Enlace, es una sola vista que cambia los textos y su interpretación para cada pregunta. Además, la vista contiene un `UIProgressView` y un `UILabel` que muestran el progreso del cuestionario.

Los botones “Sí” y “No” tienen acciones distintas pero, ya que no todas las preguntas tienen la misma valoración, estas actúan según el número de la pregunta actual. Además, si la pregunta está relacionada con el insomnio, esto será guardado en un booleano que se interpretará en la siguiente ventana.

A continuación el código que ejecutan los botones “Sí” y “No”:

```
- (IBAction) siTouched: (id) sender{
    if (contador==1) {
        puntos +=0;
        insomnio = NO;
    }

    else if (contador==3 || contador==8 || contador==10 ||
contador==11) puntos += 3;

    else if (contador==4 || contador==12) puntos += 4;
    else puntos += 2;
    [self siguiente];
}

- (IBAction) noTouched: (id) sender{
    if (contador==1) {
        puntos +=3;
        insomnio = YES;
    }

    else puntos += 0;
    [self siguiente];
}
```



Figura 33: Vista del cuestionario en curso.

Como se puede observar en el código anterior, estas funciones llaman a otra llamada "siguiente" que se encarga de actualizar la vista con la pregunta adecuada. Al acabar las preguntas éste envía la información a la siguiente clase "ResultadoTest" para ser interpretada allí y resetea el cuestionario para permitir la repetición del mismo en el caso de volver atrás.

El código mostrado a continuación es la función interna "siguiente", obviando las preguntas 2-21:

```
- (void) siguiente {
    contador ++;
    if (contador <= total) {
        NSMutableString *text = [[NSMutableString alloc]
initWithString:@""] autorelease];
        [text appendString:[NSString stringWithFormat:@"%i", contador]];
        [text appendString:[NSString stringWithFormat:@"%i", total]];
        progresoTexto.text = text;
        [progresoBarra setProgress:(float)contador/total];
        switch (contador) {
            default: break;
            case 1: preguntaTexto.text = @"¿Consigues dormir toda la
noche de corrido?"; break;
            ...
            case 22: preguntaTexto.text = @"¿Has tenido varias
enfermedades de poca importancia recientemente?"; break;
        }
    }
    else {
        ResultadoTest *resultadoTest = [[ResultadoTest alloc]
initWithNibName:@"ResultadoTest" bundle:nil];
        resultadoTest.puntos = puntos;
        resultadoTest.insomnio = insomnio;
        [self.navigationController pushViewController:resultadoTest
animated:YES];
        [resultadoTest release];
        puntos = 0;
        contador = 0;
        [self siguiente];
    }
}
```

La siguiente vista, controlada por ResultadoTest, muestra un texto explicativo acorde con el resultado obtenido. Hay tres resultados distintos y se clasifican según la cantidad de “puntos” acumulados anteriormente.

Cada resultado tiene asociados dos tratamientos recomendados que se muestran en dos botones. Si el usuario padece de insomnio según el test, uno de los tratamientos recomendados será el de insomnio.

Por último, el estado de “evolución” (ver 3.2.6. *Evolución*). será modificado acorde con los puntos obtenidos, incrementando o disminuyendo la evolución.

El siguiente código es el que se ejecuta al cargar la clase ResultadoTest, obviando la parte repetitiva:

```
- (void)viewDidLoad {
    [super viewDidLoad];
    scrollView.frame = self.view.frame;
    scrollView.contentSize = CGSizeMake(self.view.frame.size.width,
670);
    [self.view addSubview:scrollView];
    [self.view setBackgroundColor: [UIColor
colorWithPatternImage:[UIImage imageNamed:@"background.png"]]];
    if (puntos<10) {
        evalTexto1.text = @"La tensión que sufres es de";
        evalTexto2.text = @"Moderada Intensidad";
        evalTexto3.text = @"Te encuentras cerca de los límites de estrés
aceptables.";
        evalTexto4.text = @"Debes dar prioridad a lo urgente sobre lo
importante para que la tensión no te debilite mucho más.";
        [tratamiento1 setTitle:@"Relajación Base"
 forState:UIControlStateNormal];
        [tratamiento2 setTitle:@"Concentración"
 forState:UIControlStateNormal];
    }

    else if (puntos>=10&&puntos<25){
        evalTexto1.text = @"Estás soportando una ";
    ...
    if (insomnio) {
        [tratamiento2 setTitle:@"Insomnio"
 forState:UIControlStateNormal];
    }
}
```

Para que estos botones dirijan al tratamiento correspondiente, se realiza una lectura del texto del botón dentro de la acción propia, pudiendo así relacionar el botón con el tratamiento.

El usuario dispone también de los botones para compartir en Facebook i Twitter sus resultados (ver 3.4. *Compartir*).



Figura 34: Vista del resultado del cuestionario.

3.2.5. Alertas.

Utilizando los ejemplos de Apple, se ha implementado una vista que permite añadir alertas para la aplicación y sincronizarlas con el calendario de iPhone. Esta vista muestra inicialmente una lista de las alertas programadas y permite modificarlas o añadir nuevas.

Las alarmas se cargan, añaden y modifican de EKEEventStore, que contiene los eventos guardados en iPhone.

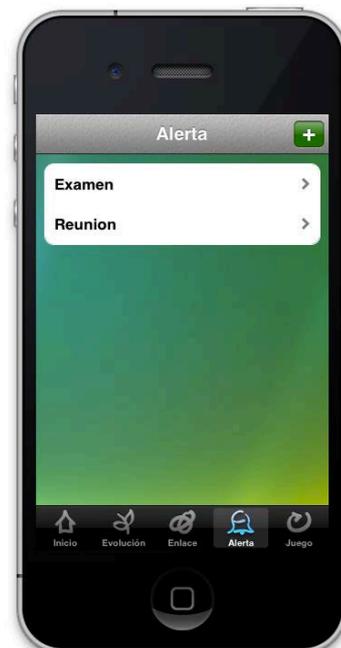


Figura 35: Vista de la lista de eventos.



Figura 36: Vista de EKEEventEditView.

El botón de añadir alertas muestra un formulario de creación de eventos con el formato de iOS, utilizando la clase `EKEEventEditViewController`. Para ahorrar trabajo al usuario y permitir identificar estas alarmas con `iDStress`, se ha dado un título y nota por defecto.

```
- (void)addEvent:(id)sender {  
    EKEEventEditViewController *addController =  
    [[EKEEventEditViewController alloc] initWithNibName:nil  
    bundle:nil];  
  
    EKEEvent *event = [EKEEvent  
    eventWithEventStore:self.eventStore];  
  
    event.location = @"iDStress";  
    event.notes = @"Recuerda utilizar iDStress";  
    addController.eventStore = self.eventStore;  
    addController.event = event;  
  
    [self presentViewController:addController  
    animated:YES];  
  
    addController.editViewDelegate = self;  
    [addController release];  
}
```

3.2.6. Evolución.

Evolución es la pantalla que muestra cómo el usuario va progresando entorno al uso de la aplicación. Contiene un entero llamado “nivel” que recibe actualizaciones desde los apartados tratamientos (una vez realizado un tratamiento y valorado su experiencia) y desde el cuestionario, el cual utiliza el resultado del mismo. Este entero varía de 0 a 1000, y puede incrementar o decrementar dependiendo de si las valoraciones recibidas son positivas o negativas.

La interfaz de evolución consta de una imagen (UIImageView [20]), una barra de progreso (UIProgressView [20]) y un botón de compartir en redes sociales (ver 3.4. *Compartir*).



Figura 37: Vista de la evolución.

La imagen, que ocupa la mayor parte de la vista, muestra una ilustración que representa el estado de evolución actual. Ésta se actualiza cada vez que el valor varía entre X fracciones de 1000 (la cantidad de fracciones depende de la cantidad de imágenes que se incluyan).

La barra de progreso muestra gráficamente el porcentaje de evolución, contemplando los valores de 0 a 1000, mientras que el botón de compartir (se detalla más adelante), permite hacer una actualización en las redes sociales explicando el estado actual.

Para poder actualizar los valores no se pueden utilizar las funciones `viewDidLoad` ni `viewWillAppear` [22] ya que la vista forma parte del `tabBar`, lo que significa que estas funciones sólo serán llamadas cuando se accede a Evolución por primera vez, permaneciendo oculta pero ya inicializada a partir de entonces.

3.2.7. Enlace.

La clase de Enlace se implementa en una scroll view [20] (vista que, por tener dimensiones mayores que la pantalla de los iPhone, permite al usuario deslizar la vista como en los navegadores de ordenador). Para aplicar dicho tipo de vista, se debe configurar qué parte va a ser la visible inicialmente y cuál es el tamaño real de la vista.

Para configurar la vista scroll se utiliza el siguiente código:

```
scrollView.contentSize =  
CGSizeMake(self.view.frame.size.width, 1070);  
  
scrollView.frame = self.view.frame;  
  
[self.view addSubview:scrollView];
```

Esta clase solo contiene información del centro, así que se ha configurado el número de teléfono y página web de la empresa de forma que el usuario pueda acceder directamente con un simple “tap”.



Figura 38: Vista de la información sobre el centro.

Además, hay un mapa real que muestra dónde se encuentra el centro. Éste se ha implementado utilizando el MapKit de xCode, configurando las coordenadas y impidiendo que el usuario pueda hacer scroll para que no pierda el centro de vista (puede alejarse y acercarse, pero no desplazarse horizontal o verticalmente) y no tenga problemas con el resto del scroll.

El siguiente código es el utilizado para configurar el mapa de la clase `MKMapView` [20]:

```
MKMapView *mapView = [[MKMapView alloc]
initWithFrame:self.mapa.frame];

mapView.mapType = MKMapTypeStandard;

CLLocationCoordinate2D coord = {latitude: 41.394518748,
longitude: 2.1489263333};

MKCoordinateSpan span= {latitudeDelta: 0.0003,
longitudeDelta: 0.003};

MKCoordinateRegion region = {coord, span};

[mapView setRegion:region];

[mapView setScrollEnabled:NO];

[scrollView addSubview:mapView];
```

3.3. Información.

Las ventanas de información aparecen en algunos de los apartados anteriores y dan información extra sobre el apartado por el que se está navegando. Los botones se encuentran en la barra de navegación y ellos hacen que se muestre una nueva vista con texto explicativo.

El código escrito a continuación se ejecuta en cada clase que requiera el botón de información, en la función `viewDidLoad` [22]:

```
UIButton *infoButton = [UIButton
buttonWithType:UIButtonTypeCustom];

[infoButton setImage:[UIImage imageNamed:@"btn_help.png"]
 forState:UIControlStateNormal];

[infoButton addTarget:self action:@selector(info:)
forControlEvents:UIControlEventTouchUpInside];

infoButton.frame = CGRectMake(0.0, 0.0, 28.0, 28.0);

self.navigationItem.rightBarButtonItem = [[UIBarButtonItem
alloc] initWithCustomView:infoButton];
```

Para programarlo óptimamente, se ha creado una sola clase que contiene dos campos de texto (uno sirve de “titular” y el otro de “especificaciones”). Para que los dos textos muestren el texto adecuado en relación a la vista que lo llama, éstos se especifican en la propia llamada.



Figura 39: Vista de la información.

El siguiente código muestra la acción “info” que contiene Panic Button. Cada clase que contenga este botón tendrá una función “info” con una estructura exactamente igual:

```

-(IBAction) info: (id)sender {
    Info *info = [[Info alloc] initWithNibName:@"Info"
bundle:nil];
    info.texto1 = [[NSMutableString alloc] initWithString:
@"El Panic Button es un ejercicio de relajación concentrada, en
pocos minutos recupera la serenidad."];
    info.texto2 = [[NSMutableString alloc] initWithString:
@"Elaborado para anticipar un momento de nerviosismo extremo, o
para recuperar sensaciones y sentimientos positivos después de
una situación conflictual, de enfado o de contrariedad. \n\n Por
su mínima duración, permite aislarse durante unos minutos para
regenerarse."];
    [self.navigationController pushViewController:info
animated:YES];
    [info release];
}

```

3.4. Compartir.

Para dar un valor añadido al programa y que al mismo tiempo crear una forma de publicidad ejecutada por los mismos usuarios, se decidió añadir la opción de compartir resultados de tratamientos y del test en las redes sociales. Por su gran público en España decidimos que estas fueran Facebook y Twitter.

Se ha utilizado una librería para iOS llamada ShareKit que permite añadir las principales redes sociales a una aplicación. Ésta requiere una configuración previa que consiste en registrar la aplicación en Facebook y Twitter (ésta última requiere que la aplicación esté en la AppStore) e introducir los datos del registro.



Figura 40: Vista de la valoración de un tratamiento.

Los campos que aparecen a continuación son los utilizados para definir los parámetros de la aplicación en Facebook:

```
// Facebook – http://www.facebook.com/developers
```

```
// If SHKFacebookUseSessionProxy is enabled then  
SHKFacebookSecret is ignored and should be left blank
```

```
#define SHKFacebookUseSessionProxy NO
```

```
#define SHKFacebookKey @""
```

```
#define SHKFacebookSecret @""
```

```
#define SHKFacebookSessionProxyURL @""
```

En las clases que contienen la opción de compartir se han añadido dos botones que llaman a la función `ShareItem` de la librería, indicando qué red social es y qué texto se quiere compartir. En el caso de Twitter el código es:

```
- (void)compartirTwitter:(id)sender{  
    SHKItem *item = [SHKItem text:@"Acabo de realizar un  
tratamiento en iDStress"];  
    [SHKTwitter shareItem:item];  
}
```

3.5. Personalizaciones en la Interfaz Global.

Como se trata de un proyecto realizado en un grupo multidisciplinar, se ha tenido que trabajar conjuntamente con el diseñador gráfico para que pudiera aplicar sus diseños en la aplicación. Esto se ha programado, además de lo que se ha detallado en los apartados anteriores, en los siguientes puntos principales:

3.5.1. Pantalla Inicial.

Ésta es la pantalla que se muestra al arrancar la aplicación y es una ventana de bienvenida que se ejecuta únicamente al inicio. Esta vista substituye la configuración por defecto de mostrar una de las pestañas de la `TabBar` [20] y fuerza así al usuario a fijarse en ellas.

Para configurar esta personalización se ha llamado a esta vista desde el método `viewDidLoad` de la clase "General", que gestiona la `tabBar`.

La ventana contiene el logotipo de la aplicación y una "Frase del día" que muestra, de entre una lista de frases proporcionadas por Enlace, una frase aleatoria distinta cada vez que se ejecuta la aplicación.



Figura 41: Diseño de la pantalla inicial.

El código que se encuentra en la función `viewDidLoad` [22] de la vista general para mostrar esta vista es:

```
Intro *intro=[[Intro alloc] initWithNibName:@"Intro"
bundle:nil];

[rootController setSelectedViewController:intro];
```

3.5.2. Barra de Navegación.

La barra superior de la aplicación está diseñada acorde con la estética de ésta, por lo que se ha tenido que programar que esa imagen con textura de piedra sea el fondo de la barra.



Figura 42: Diseño de la barra de navegación.

Por otro lado, los botones de navegación y por defecto (en el caso del “+” de la ventana de alertas) son, por defecto, azules. Para que tuvieran un color acorde con la estética general, se tiene que definir un tinte que el propio `UINavigationController` se encarga de aplicar en todos los botones que contenga la barra.

Para aplicar el mismo formato a todas las ventanas se ha modificado la implementación de `UINavigationController` en el `AppDelegate` o “main” de la aplicación. De este modo, cada vez que se cree una barra de navegación, se ejecutará la función creada. Se ha adaptado la función `drawRect` de ésta para que muestre la imagen de fondo y aplique el tinte verde:

```
@implementation UINavigationController (CustomImage)
- (void)drawRect:(CGRect)rect{

    UIImage *image = [UIImage imageNamed: @"barra.png"];

    [image drawInRect:CGRectMake(0, 0, self.frame.size.width,
self.frame.size.height)];

    self.tintColor =[UIColor colorWithRed:45.0/255.0
green:144.0/255.0 blue:24.0/255.0 alpha:1.0];
}

@end
```

3.5.3. TableView Personalizada.

Otro de los puntos característicos en la interfaz de esta aplicación es la tableView donde se muestran los tratamientos. Se quería tener una lista con iconos, títulos, descripciones y duraciones, por lo que la tableView por defecto no podía ser aplicada. Es por esta razón que se ha diseñado una celda nueva (UITableViewCell), llamada CustomCell, que contiene el formato deseado.

Los cambios programados en la clase Tratamientos han sido:

- Aplicar el diseño de CustomCell a la TableView.
- Modificar el color de las líneas separadoras.
- Ampliar ligeramente el alto de las celdas.
- Convertir la TableView en "grouped".
- Definir una imagen de fondo.
- Modificar el color de la celda al estar seleccionada.



Figura 43: Diseño de la lista personalizada.

El código se divide en varias funciones, siendo la más importante la llamada cellForRowAtIndexPath:

```
- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath {

    static NSString *CellIdentifier = @"CustomCell";

    CustomCell *cell = (CustomCell *) [tableView
    dequeueReusableCellWithIdentifier:CellIdentifier];

    if (cell == nil) {

        NSArray *topLevelObjects = [[NSBundle mainBundle]
        loadNibNamed:@"CustomCell" owner:self options:nil];

        for (id currentObject in topLevelObjects){

            if ([currentObject isKindOfClass:[UITableViewCell
            class]]){

                cell = (CustomCell *) currentObject;

                break;

            }

        }

    }

}
```

Para determinar el fondo de la vista, código que se ha utilizado en muchos apartados, se ha hecho con las siguientes líneas:

```
UIImageView* imageView = [[[UIImageView alloc]
initWithFrame:cell.frame] autorelease];

imageView.backgroundColor = [UIColor whiteColor];

cell.selectedBackgroundView = imageView;
```

4. Conclusiones.

4.1. Problemas Encontrados.

Los problemas encontrados durante la realización de este proyecto se pueden dividir en dos niveles:

Nivel humano:

Los problemas que se han dado en relación al proyecto en global han sido los comunes en el mundo laboral, dado que el formato del mismo es el mismo que un proyecto real.

Entre ellos, cabe destacar los retrasos que se han dado tanto por parte del equipo como por parte de la empresa colaboradora. En ambos casos, se trata de material que debía ser utilizado para la programación y en algún caso no ha sido recibido a tiempo.

Esto incluye hardware como un iPhone para testear la aplicación, así como software: una licencia de desarrollo para iPhone, interfaces y los tratamientos de audio que, por motivos administrativos, de horarios o de planificación, se recibieron más tarde de lo previsto.

Por último hemos sufrido cambios en la planificación debido a otras asignaturas o eventos que han requerido nuestra dedicación con urgencia. Al no dedicar el tiempo completo al proyecto únicamente, se han tenido que prestar atención a otras materias.

Estos retrasos en general han afectado varios aspectos: reprogramación de algunos apartados, la reorganización de las tareas (por dependencia de material) y breves intervalos de inactividad.

Durante el transcurso del proyecto se han añadido aspectos que han condicionado el desarrollo del mismo: animaciones durante los tratamientos, elección de lugares y objetos mediante una lista de imágenes y diseño de dos modelos de navegación, entre otros.

Además, no siempre se ha comprendido el diseño propuesto por el diseñador gráfico, por lo que se ha tenido que programar junto a él en algunos casos.

Encontrar horarios y fechas para sincronizar tareas y realizar reuniones no siempre ha sido fácil ya que los miembros del proyecto trabajamos y tenemos otras obligaciones.

Nivel técnico:

A nivel técnico, el aspecto principal ha sido la incorporación de cocos2D con UIKit [20], ya que no se encontraron ejemplos en los que la base de la aplicación fuera en UIKit y se tubo que empezar el proyecto de nuevo. Aunque se reaprovecharon la mayoría de clases, se tubo que modificar las llamadas a cocos2D y el tiempo empleado para la investigación fue muy superior al previsto.

Otro aspecto que ha causado problemas han sido algunas modificaciones en la interfaz, ya que estos apartados son poco comunes en las aplicaciones de ejemplos y muy distintas entre todas las aplicaciones. A demás, algunos aspectos pedidos (como cambiar el color de los botones de la tab bar o modificar el idioma de la ventana de eventos) requerían modificar las librerías privadas de Apple, por lo que no sería aceptada en su App Store.

Como es habitual, se han encontrado dificultades en algunas partes de programación que parecían sencillas pero no se encontraba el problema o ejemplos o librerías (alerta y mapa son los dos puntos a destacar).

Por último, el apartado del cuestionario ha requerido una programación lógica ligeramente complicada, y se ha resuelto los problemas mediante métodos poco convencionales pero efectivos (cambio entre preguntas y sus respectivas valoraciones, dirigir a un tratamiento distinto dependiendo del texto del botón o enviar datos como si el usuario sufre de insomnio).

4.2. Opiniones Personales.

Durante el transcurso de este proyecto, se ha demostrado un desarrollo de la capacidad de encontrar soluciones a problemas complejos, tanto técnicos como comunicativos. Esto implica aprender a organizar el tiempo correctamente y adaptarse a los retrasos o cambios imprevistos sobre el proyecto.

Por otro lado, se ha apreciado mucho el sistema SCRUM para la planificación del proyecto y se ha aprendido a trabajar con un producer. El desarrollo del proyecto hubiera sido inviable sin un sistema de planificación y un producer al cargo de todo.

Trabajar en un equipo multidisciplinar de manera muy satisfactoria ha hecho que el proyecto haya obtenido un resultado tan positivo. También se han recibido y hecho críticas constructivas entre los miembros del grupo por lo que se ha aprendido a aceptarlas y comunicarse mejor entre perfiles técnicos y no-técnicos.

Se ha realizado un brainstorming sobre una aplicación que parecía muy simple, dándole mucho valor añadido y mejorando la capacidad de creación de ideas y de perfeccionar una aplicación.

Obviamente, se ha profundizado en la programación en Objective-C y se han aprendido muchas utilidades de Cocos2D, hasta entonces desconocida. La colaboración entre ambos técnicos ayuda a resolver problemas ofreciendo un punto de vista nuevo a pesar de no ser el experto en el tema.

Durante el proceso de investigación, se han descubierto muchos frameworks y metodologías de programación para aplicaciones en iPhone que se desconocían.

El hecho de trabajar en equipo ha motivado a aprender más cosas ya que, aún y resultar difíciles, se ha exigido la programación de algunos aspectos por petición de los otros miembros del equipo, ya sea diseñador, técnico o producer.

El resultado del proyecto ha sido una aplicación para iPhone casi lista para ser publicada en la tienda de Apple y con un resultado incluso superior a las aplicaciones ya existentes del género.

Como conclusión final del proyecto, se debe decir que estoy muy satisfecho con el transcurso de todo el proyecto y de su resultado.

4.3. Líneas de Futuro.

El primer paso a medio plazo a dar tras la finalización de la aplicación es subir esta aplicación a la App Store. Una vez esté a la venta, se deberá poner en marcha el Plan de Márketing Estratégico y hacer una fuerte campaña de comunicación en la web 2.0 para que iD-Stress adquiriera notoriedad y se comience a hablar de la aplicación.

En segundo lugar y a largo plazo, el proyecto iD-Stress consta de varias líneas de futuro en tres niveles:

Nivel de diseño: Se estudiará la mejora de la aplicación en diversos aspectos: añadir un sonido del lugar favorito del usuario durante la reproducción del tratamiento, incorporar una lista extensa de imágenes relacionadas con los gustos del usuario, ampliar y mejorar el test y añadir nuevos tratamientos.

Todo ello dará un valor añadido superior y un acabado aún más completo a la aplicación, alejándonos aún más de las aplicaciones de la competencia.

Nivel lingüístico: Por un lado, se propone su extensión a otros idiomas siendo el primero y más importante el inglés. Por ello, en la plataforma iPhone se haría una traducción íntegra de todos los textos e instrucciones, siendo necesario además grabar de nuevo las locuciones en este idioma. El vídeo, por otra parte, podrá ser doblado o subtulado.

De esta forma la aplicación adquirirá el carácter universal que buscaban nuestra empresa colaboradora en un primer momento, y de esta forma se podrá comercializar en cualquier país.

Nivel técnico: Por otra parte, iD-Stress también estará presente en otras plataformas, siendo la más importante la adaptación al sistema operativo Android. Tras esta plataforma, le seguiría Blackberry. La elección de este orden viene determinada por la demanda en el mercado de aplicaciones así como también por petición de Enlace.

La adaptación a los diferentes lenguajes de programación supondrá un trabajo adicional sobre todo a nivel técnico, ya que será necesario aprender de nuevo cómo funciona la estructura para cada plataforma. No obstante, consideramos que algunas partes de la aplicación podrán ser reutilizadas, como por ejemplo, la parte gráfica, el audio y el vídeo.

Por último, la aplicación necesitará actualizaciones regularmente, por lo que esto es algo que también debemos tener en cuenta para el futuro más próximo.

5. Referencias.

- [1] Web ITS duero. Soluciones de software (en línea). Disponible en la web: <http://www.itsduero.es/opcion-6-software-especifico-por-sector-movilidad> (última consulta: mayo 2011)
- [2] Márketing & it (en línea) [blog] (2011, marzo) disponible en la web: <http://iturjc.blogspot.com/2011/03/espana-lider-en-adopcion-de-smartphones.html> (última consulta: mayo 2011)
- [3] Comscore (en línea). Datos de españa (de dec 2009 a dec 2010) pendiente de revisión. Disponible en la web: <http://www.comscore.com/> (última consulta: junio de 2011)
- [4] Abc. Artículo publicado el 9 de febrero de 2011. Abc (en línea) disponible en la web: <http://www.abc.es/20110209/tecnologia/rww-abci-venta-smartphones-supera-ordenadores-201102091234.html> (última consulta: junio de 2011)
- [5] Wikipedia. Enciclopedia en línea. App store. Disponible en la web: http://es.wikipedia.org/wiki/app_store (última consulta: junio de 2011)
- [6] Alt 1040 [blog] escrito por: eduardo arcos. Disponible en la web: <http://alt1040.com/2010/09/apple-iphone-itunes-store-y-por-que-la-cuota-de-mercado-es-irrelevante-en-estas-epocas> (última consulta: junio de 2011)
- [7] Apple store. Guías para la revisión de aplicaciones móviles para desarrolladores (en línea). Disponible en la web: <http://developer.apple.com/appstore/guidelines.html> (última consulta: junio de 2011)
- [8] Alt 1040 [blog]. "El ecosistema de las aplicaciones móviles en cifras y datos". Escrito por elías notario. Disponible en la web: <http://alt1040.com/2010/07/el-ecosistema-de-las-aplicaciones-moviles-en-cifras-y-datos> (última consulta: junio de 2011)
- [9] Gil-monte, p. R. (2000). Aproximaciones psicosociales y estudios diagnósticos sobre el síndrome de quemarse por el trabajo (burnout). Monográfico, revista de psicología del trabajo y de las organizaciones, págs: 101-102
- [10] Dr. Antonio cano vindel, presidente de la sociedad española para el estudio de la ansiedad y el estrés "epidemiología y costes del estrés laboral" 2002 (en línea). Disponible en

http://www.ucm.es/info/seas/estres_lab/epidem_y_costes.htm (última consulta: junio de 2011)

[11] Ana lombard. Página web del centro enlace. Disponible en www.enlacebcn.com (última consulta: junio de 2011)

[12] Definición de la sofrología. Ana Lombard (en línea). Disponible en: <http://www.enlacebcn.com/lasofrologia.html> (última consulta: junio de 2011)

[13] Página oficial de cocos2d: www.cocos2d-iphone.org/ (última consulta: junio de 2011)

[14] Página oficial de Shiva 3D: www.stonetrip.com/ (última consulta: marzo de 2011)

[15] Página oficial de iProcessing: www.luckybite.com/iprocessing (última consulta: marzo de 2011)

[16] Página oficial de Unity3: www.unity3d.com/ (última consulta: marzo de 2011)

[17] Página oficial de Adobe Air: www.adobe.com/es/products/air/ (última consulta: marzo de 2011)

[18] Página oficial de MockApp: www.mockapp.com/ (última consulta: febrero de 2011)

[19] Descripción de Objective C: <http://es.wikipedia.org/wiki/objective-c> (última consulta: junio de 2011)

[20] Referencia oficial de UIKit y sus componentes: http://developer.apple.com/library/ios/#documentation/uikit/reference/uikit_frame_work/_index.html (última consulta: junio de 2011)

[21] Guía de las herramientas que ofrece iPhone SDK: http://developer.apple.com/library/ios/#referencelibrary/gettingstarted/url_tools_for_iphone_os_development/_index.html#apple_ref/doc/uid/tp40007593 (última consulta: junio de 2011)

[23] Documentación de la clase UIViewController http://developer.apple.com/library/ios/#documentation/uikit/reference/uiviewcontroller_class/reference/reference.html (última consulta: junio de 2011)