ESCOLA UNIVERSITÀRIA D'ENGINYERIA
TÈCNICA DE TELECOMUNICACIÓ LA SALLE

TREBALL FINAL DE MÀSTER

MÀSTER EN CREACIÓ, DISSENY I ENGINYERIA MULTIMÈDIA

ZOO XXI:

Arquitectura y Desarrollo

ALUMNE

Berta Izquierdo Farré

PROFESSOR PONENT

Oscar García Pañella

Emiliano Labrador Ruiz de la Hermosa

ACTA DE L'EXAMEN
DEL TREBALL FINAL DE MÀSTER

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

Berta Izquierdo Farré

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

ZOO XXI: Arquitectura y Desarrollo

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

ABSTRACTO

Este trabajo es un estudio sobre la evolución de la relación hombre-animal a través del concepto de zoológico y su evolución histórica. El estudio se centra en la apuesta por un concepto de zoológico innovador, *ZOO XXI: mucho más que un zoo*, que se diferencía por sus objetivos que van más allá del entretenimiento. Un zoológico inmersivo que permite al público zambullirse literalmente en el mundo de los animales.

La tecnología permitirá no solamente que nos sintamos inmersos en la fauna y la flora de una región en particular, sino que a la vez nos hará vivir la experiencia como real, permitiendo al público interactuar con los animales virtuales.

ABSTRACTE

Aquest treball és un estudi sobre l'evolució de la relació home-animal a través del concepte de zoològic i la seva evolució històrica. L'estudi es centra en l'aposta per un concepte de zoològic innovador, *ZOO XXI: mucho más que un zoo,* que es diferencia pels seus objectius que van més enllà de l'entreteniment. Un zoològic immersiu que permet al públic submergir-se literalment en el món dels animals.

La tecnologia permetrà no solament que ens sentim immersos en la faula i la flora d'una regió en particular, sinó que a la vegada ens farà viure l'experiència como a real, permetent al públic interactuar amb els animals virtuals.

ABSTRACT

This work is a study on the evolution of man-animal relationship through the concept of zoo and its historical evolution. This study focuses on the bet on a concept of innovative zoo, *ZOO XXI: mucho más que un zoo,* that differs in its goals that go beyond entertainment. An immersive zoo that allows the public to literally dive into the world of animals.

The technology will not only make us feel immersed in the fauna and flora of a particular region, but at the same time it will let us live the experience as real, allowing the public to interact with virtual animals.

AGRADECIMIENTOS

Dar las gracias a la empresa Emotique, en especial a Joan Coll por habernos abierto las puertas a ZOO XXI y dejarnos aportar nuestro granito de arena. Del mismo modo, agradecer a los ponentes, Oscar García y Emiliano Labrador, su tarea de coordinación, y asesoramiento en todo lo que ha estado en sus manos.

Agradecer también, a mis compañeros del máster MCDM el apoyo durante las muchas horas invertidas en la universidad, trabajando codo a codo. Pero especialmente a mis compañeros de equipo, cada uno de ellos también forma parte de este proyecto.

Y, especialmente quiero agradecérselo a mis padres, por ofrecerme la oportunidad de llegar donde he llegado, y por apoyarme siempre de manera incondicional.

RESUMEN

La relación del hombre con los animales ha evolucionado y forma parte de la propia historia de la humanidad. Esta relación puede estudiarse mediante los cambios producidos en el concepto de zoológico en el transcurso del tiempo, desde la simple posesión y exhibición, hasta más modernos conceptos de conservación de especies, investigación, y educación.

No obstante, los zoológicos no se han acomodado suficientemente a la evolución socio-cultural de nuestra sociedad, y muchos de ellos mantienen las mismas infraestructuras del siglo XIX. Como consecuencia, estos zoológicos urbanos van perdiendo el interés del público, según se desprende de la información publicada por los propios zoológicos. Aunque, a pesar de ello, se mantiene la atracción ancestral que los animales salvajes ejercen en el ser humano.

En la actualidad se está adquiriendo mayor concienciación ciudadana sobre la importancia de la acción humana respecto a la vida animal que lleva a riesgo de extinción a numerosas especies, y a la necesidad de salvaguardar la biodiversidad.

Muchos pensadores han escrito e investigado la relación humano-animal, y como dijo el escritor John Berger: *"Los humanos tenemos una relación discorde con los animales. Nos gusta mirarlos en los parques zoológicos, ver documentales sobre sus hábitos de vida y películas de dibujos animados donde los protagonistas son animales con las preocupaciones propias de seres humanos, los adoptamos como mascotas, los cuidamos y apreciamos el cariño que ellos nos profesan. Por otro lado los tratamos como un producto más que manufacturamos en granjas, utilizamos su carne y su piel para nuestro beneficio, y permitimos su exhibición y tortura para la diversión de unos cuantos."*

El proyecto ZOO XXI incorpora todos aquellos adelantos y desarrollos tecnológicos que, por únicos y vanguardistas, darán vida a animales virtuales. Un espacio en el que concienciar sobre la importancia de la conservación de especies, el respeto a los animales, sus hábitats y ecosistemas. Un espacio educativo con programas adaptados a los distintos públicos dónde se promueva la colaboración con las organizaciones que desarrollan programas de conservación de especies en sus propios lugares de origen.

Sobre esta idea innovadora del concepto *ZOO XXI: mucho más que un zoo*, se desarrollan diversos prototipos multimedia para dar a conocer esta nueva conceptualización, cuyo desarrollo es objeto de este trabajo.

Índice

1. Introducción

1.1 Marco

A los animales
que hemos vuelto
nuestros esclavos,
no nos gusta
considerarlos
nuestros iguales.

Charles Darwin (1809-1882)

Guru Advertainment Agency, empresa de marketing promocional y publicidad con sede en Barcelona, sabe que el concepto de zoológico está cambiando. Junto con **Emotique**, empresa especializada en tecnología software para audiovisuales interactivo, han creado el **proyecto ZOO XXI**.

El proyecto ZOO XXI incorpora todos aquellos adelantos y desarrollos tecnológicos que, por únicos y vanguardistas, darán vida a animales virtuales inmersos en recreaciones de sus hábitats naturales y a los que podremos ver actuando de forma absolutamente natural. La tecnología permitirá no solamente que nos sintamos inmersos en el hábitat de cada especie animal y vegetal, sino que a la vez nos hará vivir la experiencia como real, permitiendo al público interactuar con los animales virtuales.

En definitiva, el proyecto ZOO XXI presenta un concepto de zoo innovador que se diferencía por sus objetivos que van más allá del entretenimiento. Un espacio en el que concienciar sobre la importancia de la **conservación** de especies, el respeto a los animales, sus hábitats y ecosistemas. Un espacio **educativo** con programas adaptados a los distintos públicos. Un espacio dónde se promueva la **colaboración** con las organizaciones que desarrollan programas de conservación de especies en sus propios lugares de origen. Un espacio desde donde se impulse la **investigación científica**.

Desde nuestro equipo de trabajo multidisciplinar del MCDM de Ingeniería y Arquitectura La Salle, Universitat Ramón Llull, pretendemos dar soporte a esta iniciativa. Un equipo formado por Marta Bosch (gestora del proyecto), Eladio Gómez (diseñador gráfico), Iván Muñiz (técnico de usabilidad), Luís López (programador) y Berta Izquierdo (programadora).

1.2 Estado del arte

1.2.1 Recorrido histórico

Los primeros zoológicos de los que se tiene testimonio surgieron en la antigüedad. Se trataba de colecciones particulares, pertenecientes a reyes y nobles, que consideraban a los animales exóticos como tesoros y símbolos de poder, y a los que sólo tenían acceso invitados distinguidos.

En el año **1500 a.C.**, la reina Hatsheptuf de Egipto estableció el primer zoológico de la historia.

Hace 3000 años, el emperador chino Wen Wang, fundador de la dinastía Zohu, mandó construir el Ling-Yu o "Jardín de la inteligencia", un gran parque de más de 1500 acres, donde exhibía peces, aves, serpientes, anfibios y mamíferos como tigres, ciervos, antílopes y rinocerontes, entre otros animales. Sólo los visitantes distinguidos del imperio podían conocer tal lugar.

Los sucesores de Wen Wang conservaron la tradición de construir **jardines zoológicos** hasta el siglo XIII de nuestra era. Cuando el mongol Kublai Khan asumió el trono de Pekín en **1260**, amplió los jardines y mandó hacer, en el centro de Mongolia, el más imponente jardín imperial.

El viajero veneciano Marco Polo conoció ese magnífico sitio y describió diversos animales desconocidos en Occidente, como el tapir malayo y el panda.

Figura 1. Marco Polo en la corte de Kublai Kan

En Europa, los griegos establecieron los primeros **zoológicos públicos** después que en el **siglo IV a.C.**, las expediciones de Alejandro Magno llevaron animales de numerosas especies a Grecia.

Los romanos continuaron con la costumbre de mantener **colecciones zoológicas**, pero con el objetivo de proveer animales a espectáculos circenses.

Durante la **Edad Media**, los monarcas y señores feudales de Europa reunieron colecciones privadas de animales, como signo de poder. Una de las principales fue la *Ménagerie* de Chantilly, en Francia, que persistió por dos siglos y fue destruida durante la Revolución Francesa.

A partir del **siglo XVI**, con la ocupación de las colonias por parte de los países europeos, llegaron al viejo continente muchos animales exóticos, la mayoría de los cuales moría en el viaje. Los pocos que sobrevivían iban destinados, en su mayor parte, a colecciones privadas de animales.

El rey Nezahualcóyotl fue el creador del **primer jardín botánico** y el **primer zoológico de América** en Tezcutzingo, un pequeño cerro al Oriente del reino de Texcoco.

A Tenochtitlán, el emperador azteca Moctezuma Xocoyotzin mandó construir la Casa de las Fieras, con plantas y animales traídos desde todos los rincones de su imperio. El mantenimiento y la limpieza que se le daba regularmente a los estanques y las jaulas, así como la alimentación y los cuidados de los animales de la Casa de las Fieras eran muy parecidos a los de un zoológico de nuestros días.

Figura 2. La Gran Tenochtitlán

En la Casa de las Aves, que se ubicaba en Chapultepec, había desde águilas reales y otras muchas aves de grandes cuerpos, hasta pajaritos pequeños de diversos colores, como quetzales, papagayos y patos.

Hacia fines del **siglo XVI**, hubo en la India un gobernador ilustre de nombre Akbar, quien decía compartía la tradición nómada del amor hacia los animales y estableció **zoológicos abiertos al público** en varias ciudades de ese país. Con fuertes jaulas para tigres, leones y rinocerontes, contaba con personal entrenado y estaba prohibido molestar a los animales.

Sin embargo, no fue hasta la segunda mitad del **siglo XVIII** cuando comenzaron a establecerse los **zoológicos modernos** en Europa. El primero fue **La Casa Imperial de Fieras en Viena,** Austria, cuya construcción inició en **1752** y se abrió al público trece años después, en **1765**.

Creado en 1635, el *Jardin du roi* se renombra en la **Revolución Francesa** como *Le Jardin des Plantes de Paris*, y se abre al público parisino en **1793** considerándolo el **primer zoológico popular**.

A principios del **siglo XIX** se creó la Sociedad Zoológica de Londres**.** Ésta creó el primer zoológico científico del mundo, el ***Regent's Park***, inaugurado en **1828**. No sólo se pretendía la exhibición de distintas especies, sino que sus objetivos también incluían el estudio e investigación del comportamiento animal.

Figura 3. Zoológico de Londres, 1835 [1]

En **1880** se creó el **Jardín Zoológico Imperial de Berlín**, con la más impresionante y elaborada arquitectura que jamás ha existido en **un zoológico**. Incluía la Casa del Elefante, una exhibición diseñada simulando un templo hindú, ricamente decorada con artísticos murales y construida con los materiales más costosos para impresionar al visitante. Paralelamente a semejante ostentación, el área destinada para el animal era tan reducida, que éste apenas lograba darse vuelta.

Estos episodios claramente antropocéntricos eran muy comunes en cuanto a la exhibición de animales en aquella época y, por desgracia, este tipo de zoológicos fue el que se extendió por el continente europeo.

Mientras tanto, el zoológico más antiguo de los Estados Unidos fue inaugurado en el Central Park de Nueva York en **1864**. A éste le siguieron los de Chicago, Filadelfia, Washington y el Zoo del Bronx, fundado por la Sociedad Zoológica de Nueva York (hoy Sociedad para la Conservación de la Vida Salvaje), que abrió sus puertas en **1899** y actualmente cuenta con una de las mayores colecciones de animales del mundo

En los primeros **zoológicos modernos**, los animales vivían en jaulas. A principios del **siglo XX** la situación cambió gracias a Carl Hägenbach, un visionario y propulsor de hábitats naturales para la ubicación de animales que introdujo un concepto

revolucionario en cuanto al diseño de exhibiciones al inaugurarse en **1907** el primer **zoológico sin barrotes**, el Parque Stellingen de Hamburgo, Alemania.

Sin embargo, estas ideas no se popularizaron y en muchos zoológicos se seguían dejando a un lado las necesidades sicológicas de los animales.

En **1916** se fundó en los Estados Unidos el zoológico de San Diego, uno de los más grandes del mundo, que cuenta con una colección de animales completísima.

En México, por iniciativa del ilustre Biólogo Alfonso Luis Herrera y con apoyo de la entonces Secretaría de Agricultura y Fomento, el 6 de julio de **1923**, se colocó la primera piedra del Zoológico de Chapultepec, que fue abierto al público un año más tarde.

Otro gran cambio en la forma de concebir los zoológicos, se produjo después de la II Guerra Mundial cuando, además de la búsqueda de entretenimiento, cobraron importancia la investigación y la educación.

Con la aparición de los postulados del llamado padre de la biología de los zoológicos, Heini Hediger, se llamó la atención sobre las necesidades biológicas y conductuales de los animales.

A mediados de los **años 70**, con el desarrollo del plan maestro del Woodland Park Zoo de Seattle, en los Estados Unidos, por parte de la firma de arquitectos paisajistas Jones & Jones, se implementó por primera vez a gran escala el nuevo concepto de inmersión en el diseño de exhibiciones zoológicas. Los zoológicos continuaron cambiando, con variaciones en sus instalaciones, arquitectura, organización y hasta su señalización. De ser un sitio de simple entretenimiento, su concepción fue evolucionando hasta llegar a la actual, que otorga a los animales mejores condiciones de vida, tiende a eliminar de modo progresivo las jaulas, les concede más espacio y busca crear ambientes que se asemejen a sus hábitats naturales.

En el **siglo XXI**, los zoológicos están cambiando de prioridades, considerando el entretenimiento, la educación, la investigación científica y la conservación de las especies animales como objetivos principales.

Los conceptos de exhibición han variado radicalmente a través de la historia. Cuando las primeras colecciones exhibían de modo itinerante animales salvajes en pequeñas jaulas, la relación del hombre respecto de la naturaleza era de conquista y sometimiento, aunque también de ignorancia y curiosidad. Hoy, cuando el conocimiento nos ha permitido comprender mejor el mundo que nos rodea, los

conceptos de la exhibición han evolucionado y lo seguirán haciendo, generando nuevas soluciones para los zoológicos.

Actualmente existen más de **500 zoológicos** alrededor del mundo, que son visitados cada año por más de 100 millones de personas. Lugares como Missouri, Bombay, Calcuta, El Cairo, Tokio, Berlín, Múnich, Madrid, Barcelona y Roma albergan grandes colecciones de especial importancia. En Latinoamérica destacan los zoológicos de Buenos Aires y Mendoza, en Argentina; el de Pará en Brasil, el de Santiago de Chile y el de Chapultepec en la Ciudad de México.

1.2.2 Panorama actual

Parque Zoológico de Barcelona (1892), Barcelona. Parque zoológico situado en medio de la ciudad. Los primeros animales provenían de una colección privada que fue donada al Ayuntamiento de Barcelona. El zoológico afirma tener tres propósitos, y así se muestra en su balance anual, sostenibilidad, investigación y conservación, y educación.

Figura 4. Parque Zoológico de Barcelona [2]

Parque de la Naturaleza de Cabárceno (1989), Cantabria. Es un espacio natural que acoge un centenar de especies animales de los cinco continentes. El parque está situado sobre las 750 Has. de una antigua explotación minera a cielo abierto. El Parque de la Naturaleza de Cabárceno está concebido con fines educativos, culturales, científicos y recreativos. Salvo la alimentación que se les facilita a los animales, el resto de las actividades están marcadas por su casi total libertad e instinto.

Figura 5. Parque de la Naturaleza de Cabárceno [3]

Faunia (2001), Madrid. Es un parque biológico, es decir, jardín botánico y parque zoológico al mismo tiempo. Ofrece diferentes ecosistemas como el de los Polos, el de la Jungla, el del Bosque Africano, entre otros. Faunia promueve la investigación y la conservación de los ecosistemas y sus especies animales y vegetales más amenazados como consecuencia de las actividades humanas.

Figura 6. Faunia [4]

CosmoCaixa (2005), Barcelona. El Museo de la Ciencia de la Obra Social "La Caixa" ocupa las instalaciones del que fue el primer Museo de la Ciencia interactivo de España (1981). Es un espacio que incluye nueve plantas, seis subterráneos pero con luz natural, y una gran plaza pública. Entre sus espacios permanentes destacan El Muro Geológico, El Bosque Inundado, La Sala de la Materia, el Planetario, y La Plaza de la Ciencia.

Figura 7. El Bosque Inundado [5]

Bioparc Valencia (2008), Valencia. Es un zoológico de nueva generación que ha sido creado basándose en el concepto de zoo-inmersión, sumergiendo totalmente a los visitantes en los hábitats salvajes. Bioparc Valencia está comprometido con la conservación de los animales, así como con la educación y la concienciación sobre la necesidad de preservar los ecosistemas.

Figura 8. Bioparc Valencia [6]

1.3 Descripción del problema

Los zoológicos han experimentado una rápida evolución, muy especialmente en las últimas décadas. Las organizaciones profesionales como la Asociación Europea de Zoológicos y de Acuarios (EAZA) [7] y la Asociación americana de Zoológicos y de Acuarios (AZA) [8] exigen a sus miembros que incluyan entre sus funciones aspectos como educación, investigación y conservación, y acrediten su implicación en los mismos. Muchos zoológicos disponen ya de un equipo de profesionales altamente cualificados y concienciados respecto a las funciones que deben acometer. Sin embargo, todavía quedan demasiadas instalaciones obsoletas, muchas veces demasiado pequeñas.

La tendencia actual y futura viene condicionada de una parte, por la actual normativa legal (Directiva 1999/22/CE) y Ley 31/2003 en materia de investigación científica, divulgación y conservación, y de otra parte por las preferencias del público-usuario.

En este sentido se han realizado encuestas para conocer su opinión respecto a las instalaciones y actividades de los zoológicos.

Se ha constatado que los encuestados desearían que los zoológicos dispusieran de medios interactivos, videojuegos o aplicaciones tecnológicas. En cuanto a las preferencias respecto a las alternativas, dentro del sector, un 55% de los encuestados elige el parque natural, un 36% el acuario, y solamente un 9% el zoológico.

También se han valorado las cifras que ofrece el estudio del sector de *parques temáticos* de la consultora DBK [9], especializada en el estudio de sectores económicos, que pone en evidencia que las previsiones de facturación del sector para el 2010 caerá en un 10%, si bien en el caso de parques zoológicos, acuarios y parques de la naturaleza podría aumentar un 1,5%, con motivo de apertura de nuevas instalaciones y a las importantes inversiones públicas realizadas.

Se ponen pues en evidencia que es preciso incorporar nuevas ideas que permitan ofrecer a un público más amplio un conjunto de actividades, tanto en las instalaciones, como en la experiencia del usuario acordes con su misión y objetivos, enfocadas a afianzar su público actual y atraer un nuevo público objetivo mucho más amplio en franja de edad, nivel cultural, y procedencia.

1.4 Solución propuesta

La evolución reside ya no solamente en mostrar los animales individualmente (sXIX), ni al animal en su hábitat (sXX), sino en mostrar ecosistemas completos. Con ZOO XXI aparece un nuevo concepto en cuanto a la forma de exhibición, un zoológico inmersivo que permite al público zambullirse literalmente en el mundo de los animales.

Un mundo donde poder introducirnos y descubrir todo aquello que no nos cuentan los documentales, el poder vivir a través de los ojos de un animal, poder comunicarte con el resto de la manada y vivir como ellos.

Google Earth lanzó en marzo del 2010 una aplicación para recorrer Barcelona en imágenes tridimensionales diseñadas por los usuarios. ZOO XXI propone un zoológico virtual distribuido por toda la ciudad. Un proyecto escalable, y extrapolable a cualquier ciudad. Un zoológico que esté al alcance de cualquier ciudadano, sin necesidad de desplazamiento y en el que puedan contribuir tanto particulares como instituciones.

En definitiva, un zoológico de todos al alcance de cualquiera.

1.5 Perspectiva general del proyecto

Los primeros apartados de este trabajo están dedicados a la evolución histórica producida en el concepto de zoológico en el transcurso del tiempo y panorama actual. (1.2) A continuación se introduce el sistema operativo iPhone OS (2.2), se realiza un estudio de las herramientas de desarrollo (2.3), y de todos los principios fundamentales (2.4) que permiten el desarrollo de aplicaciones para iPhone (lenguaje de programación, gestión de la memoria, delegación y notificaciones). A continuación se presenta el modelo de datos (2.5), basado en el Modelo-Vista-Controlador, estilo de arquitectura de software analizado y elegido (2.6). Se hace una breve referencia a la tecnología key-value (2.7), así como a los frameworks utilizados (2.8) y se analizan las distintas opciones de servidores (2.9).

La parte práctica del trabajo presenta los resultados obtenidos en la aplicación de los conceptos teóricos previamente citados con las funcionalidades adecuadas, dividiéndose en una primera fase de generación de ideas (3.1), una segunda fase de desarrollo (3.2) dónde se analiza la elección del terminal móvil, la evolución de los juegos interactivos, las instalaciones y proyecciones interactivas para presentar seguidamente los prototipos desarrollados específicamente para este proyecto; y concluye con una fase de ejecución (3.3).

2. Fundamentos teóricos

2.1 Requerimientos

Para desarrollar aplicaciones para iPhone OS, se necesita un equipo Mac OS X, así como las herramientas de desarrollo que proporciona Apple. Éstas proporcionan apoyo a la gestión de proyectos, edición de código, la creación de ejecutables, la depuración a nivel fuente, la gestión de código fuente del repositorio, el ajuste del rendimiento, y mucho más.

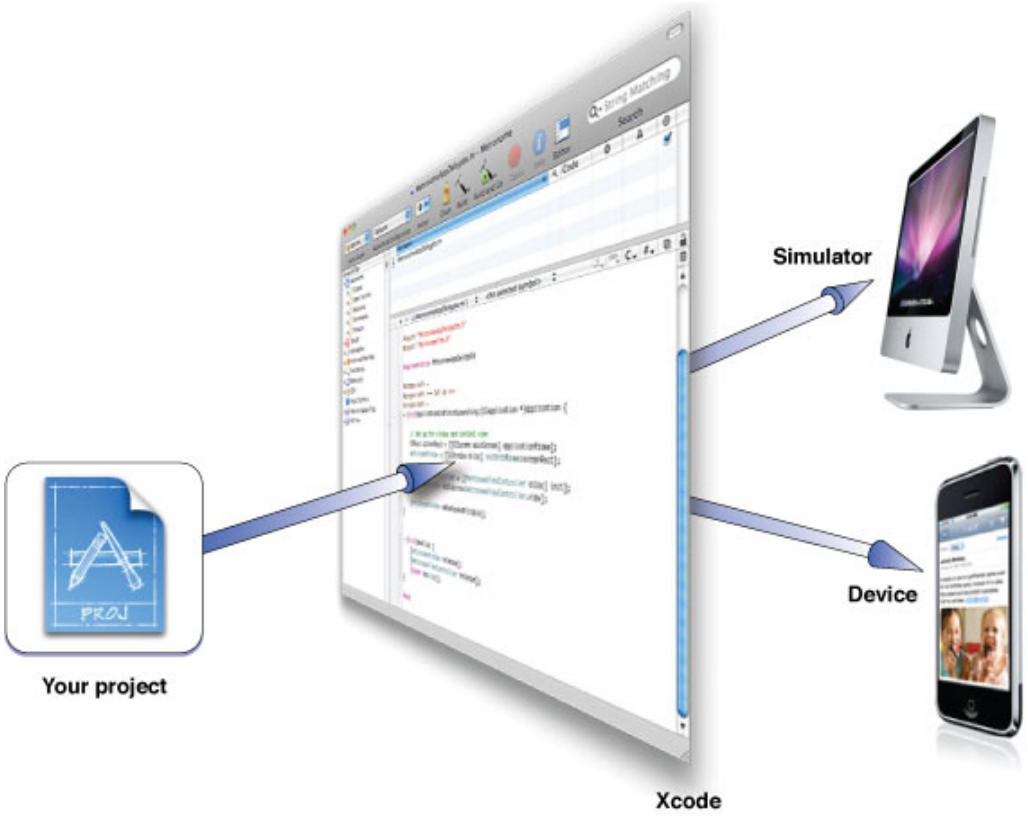


Figura 9. Ejecución de un proyecto desde Xcode [10]

2.2 iPhone OS

iPhone OS es el sistema operativo que utiliza el iPhone, desarrollado, al igual que el dispositivo, por Apple Inc. Está basado en una variante del Mach kernel que se encuentra en Mac OS X. El sistema operativo, localizado en la partición /root del dispositivo, ocupa menos de 500MB. Apple pone a disposición de los usuarios actualizaciones gratuitas del sistema operativo del iPhone por iTunes. El iPhone OS está compuesto por cuatro capas de abstracción.

Cocoa Touch: API que permiten el desarrollo de programas para iPhone. Cocoa Touch se basa en el conjunto de herramientas que facilita la API de Cocoa para crear aplicaciones sobre la plataforma Mac OS X. UIKit y Foundation son sus frameworks más populares.

Media: La capa que provee la parte gráfica y multimedia. Open GL ES, Core Animation, Core Audio y Video Playback son algunos de los frameworks de esta capa.

Core Services: La capa de servicios principales. Destacamos SQLite y Core Location.

Core OS: La capa que contiene el núcleo del sistema operativo, el sistema de ficheros, la seguridad, las licencias y la gestión de energía.

Figura 10. iPhone OS

2.3 Herramientas de desarrollo

2.3.1 XCode IDE

Xcode es el entorno de desarrollo integrado (IDE) de Apple Inc. La versión actualmente disponible, la 3.2.2, viene con el SDK para iPhone 3.2.

Xcode proporciona todas las herramientas necesarias para crear y gestionar proyectos para iPhone; generar el código fuente, crear un ejecutable, y ejecutar y depurar el código, ya sea en un simulador o en el propio dispositivo. En definitiva, se encarga de los detalles desde su inicio hasta la implementación.

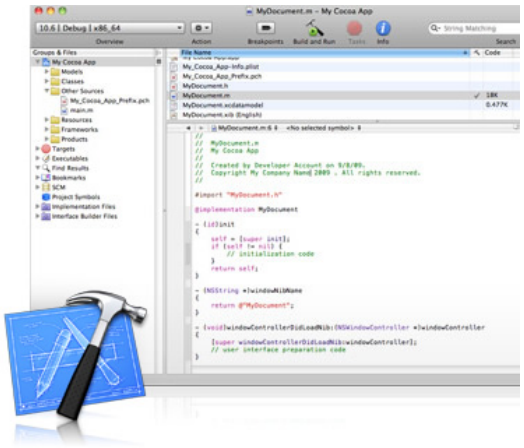


Figura 11. Xcode IDE [11]

Xcode construye proyectos a partir del código fuente, escrito en C, C + +, Objective-C y Objective-C + +. Genera ejecutables de aplicaciones para iPhone OS, aunque para Mac OS X incluye muchas más opciones de ejecutables. Xcode permite un elevado grado de personalización de todas sus herramientas También es compatible con varios sistemas de administración de código fuente, lo que le permite añadir archivos a un repositorio, confirmar los cambios, acceder a la versión actualizada, y comparar versiones.

2.3.2 Interface Builder

Figura 12. Interface Builder [11]

Interface Builder es un editor gráfico. Permite el diseño del más mínimo aspecto relacionado con la interfaz gráfica de la aplicación. Interface Builder almacena el diseño de la interfaz en un o más archivos, cómo un conjunto de objetos con sus respectivas relaciones. Los cambios realizados en la interfaz se sincronizan automáticamente con Xcode.

Interface Builder se centra alrededor de cuatro elementos principales:

Archivos Nib: Un archivo nib es un contenedor de archivos que contiene los objetos que aparecen en una interfaz de usuario de forma archivada.

Librería de objetos: La librería de Interface Builder contiene todos los objetos que se pueden colocar a la interfaz de usuario.

Inspector: El inspector tiene una serie de paneles seleccionables para establecer la configuración inicial de los objetos.

Panel de conexiones: El panel de conexiones es una pantalla que muestra las conexiones del outlet actual y la acción para el objeto seleccionado y permite la gestión de dichas conexiones.

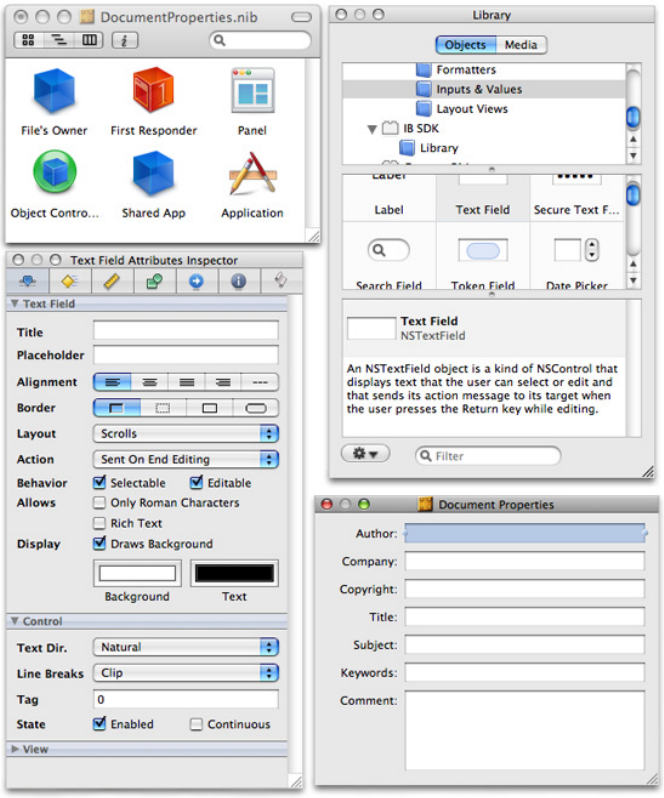


Figura 13. Elementos IB [11]

2.3.3 iPhone Simulator

El simulador ejecuta la aplicación casi del mismo modo en que lo haría un dispositivo iPhone. La rapidez para lanzar un proyecto y depurarlo en el simulador, lo convierte en una herramienta perfecta de testeo para asegurarnos que la interfaz del usuario hace exactamente lo que se supone debe hacer. El simulador supone un ahorro de tiempo a la hora de desarrollar.

Figura 14. iPhone Simulator [11]

Aún así, siempre debe realizarse la fase final de la depuración en el dispositivo. Hay que tener en cuenta que ciertas funcionalidades, así como el rendimiento de la aplicación pueden verse alteradas en el dispositivo.

2.4 Cocoa Fundamentals

2.4.1 Objective C

El lenguaje Objective-C es un lenguaje de programación sencillo, pero a la vez poderoso, diseñado para permitir la programación orientado a objetos. Objective-C está construido sobre el estándar ANSI C, proporcionando sintaxis para definir clases y métodos.

```
@interface classname : superclassname {
    // instance variables
}
+classMethod1;
+(return_type)classMethod2;
+(return_type)classMethod3:(param1_type)parameter_varName;

-(return_type)instanceMethod1:(param1_type)param1_varName :(param2_type)param2_varName;
-(return_type)instanceMethod2WithParameter:(param1_type)param1_varName andOtherParameter:(param2_type)param2_varName;
@end
```

Figura 15. Objective C [12]

Muchos de los conceptos tradicionales orientados a objetos, tales como encapsulación, herencia y polimorfismo, están todos presentes en Objective-C, aunque hay algunas diferencias. El uso de objetos y patrones de diseño orientado a objetos es fundamental para el diseño de aplicaciones Cocoa, y comprender cómo interactúan es fundamental para desarrollar aplicaciones.

2.4.2 Gestión de la memoria

Para mantener el consumo de memoria lo más bajo posible en una aplicación, hay que deshacerse de objetos que no están siendo utilizados, pero hay que asegurarse. Por lo tanto, necesitamos un mecanismo que permita marcar un objeto mientras siga siendo útil. La gestión de la memoria se entiende en términos de propiedad de objetos. Básicamente los conceptos son cuatro. Un objeto puede tener uno o más propietarios. Cuando un objeto no tiene dueños, se destruye. Para asegurarnos que un objeto que nos interesa no sea destruido, deberemos convertirnos en su propietario. Y, para permitir que un objeto en el que ya no se está interesado sea destruido, se debe renunciar a la propiedad [13].

Para apoyar este modelo, Cocoa proporciona un mecanismo llamado "recuento de referencias" o "mantener el conteo." Un objeto se crea con un recuento de 1. Cuando el recuento llega a 0, se cancela la asignación de un objeto (se destruye). Hay diferentes métodos para manipular el conteo:

`alloc`

- Asigna memoria para un objeto, y lo devuelve con el recuento de 1.
- Quien crea un objeto con cualquier método que comienza con la palabra *alloc* o *new* se convierte automáticamente en su propietario.

`copy`

- Hace una copia de un objeto, y lo devuelve con el recuento de 1.
- Si se copia un objeto, éste pertenece a quien ha realizado la copia. Esto se aplica a cualquier método que contiene la palabra *copy* cuando ésta se refiere al objeto que se devuelve.

`retain`

- Aumenta el recuento en 1.
- Se queda con la propiedad del objeto.

`release`

- Disminuye el recuento en 1.
- Renuncia a la propiedad de un objeto.

`autorelease`

- Disminuye el recuento en 1 en algún momento en el futuro.
- Renuncia a la propiedad de un objeto en algún momento en el futuro.

2.4.3 Delegación

Delegación, en el sentido más simple consiste en delegar una tarea de un objeto a otro, es una técnica común en la programación orientada a objetos.

Es un mecanismo en el que un objeto referencía a otro, su delegado, y le llama cuando le requiere para una tarea. El delegado es un objeto que actúa en nombre de, o en coordinación con otro objeto, cuando ese objeto se encuentra con un evento. Por lo que en realidad tenemos un conjunto de funciones sin implementar que son adquiridas por cualquier objeto y que será éste el que escriba el código particular para esas funciones.

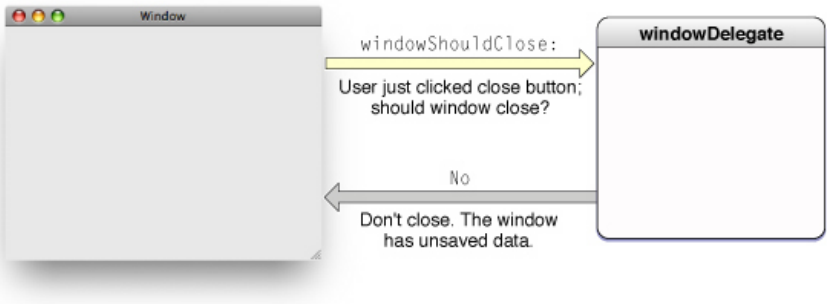


Figura 16. Mecanismo de delegación [14]

El objeto que delega es a menudo un objeto que hereda de UIResponder en UIKit, que está respondiendo a un evento de usuario. El delegado es un objeto que se encarga del control de la interfaz de usuario para ese evento.

Para apreciar mejor el valor de la delegación, ayuda el hecho de considerar una objeto Cocoa como una ventana (una instancia de UIWindow) o una vista de tabla (una instancia de UITableView). Estos objetos están diseñados para cumplir una función específica de un modo genérico. Este comportamiento restringido y genérico limita necesariamente lo que el objeto puede saber acerca de cómo un evento afecta qué en la aplicación.

El mecanismo de programación de la delegación da una oportunidad a los objetos para coordinar su aspecto y su estado con los cambios que ocurren en otras partes de la aplicación, por lo general los cambios provocados por las acciones del usuario. Más importante aún, la delegación hace posible que un objeto modifique el comportamiento de otro objeto sin la necesidad de heredar de él. El delegado es casi siempre un objeto

personalizado y, por definición, incorpora la lógica específica de la aplicación que el objeto que delega no tiene posibilidad de conocer por sí mismo.

El diseño del mecanismo de delegación es simple. La clase que delega declara, sin implementar, uno o más métodos que constituyen un protocolo formal o informal. Un protocolo formal que utiliza métodos opcionales es el enfoque por excelencia, pero los dos tipos de protocolos son utilizados por el framework de Cocoa. En el protocolo informal, la clase que delega declara métodos en una categoría de NSObject, y el delegado implementa sólo aquellos métodos en los que tenga interés. Si, por el contrario, la clase que delega declara un protocolo formal, el delegado podrá decidir si implementa los métodos marcados como opcionales, pero deberá implementar los requeridos.

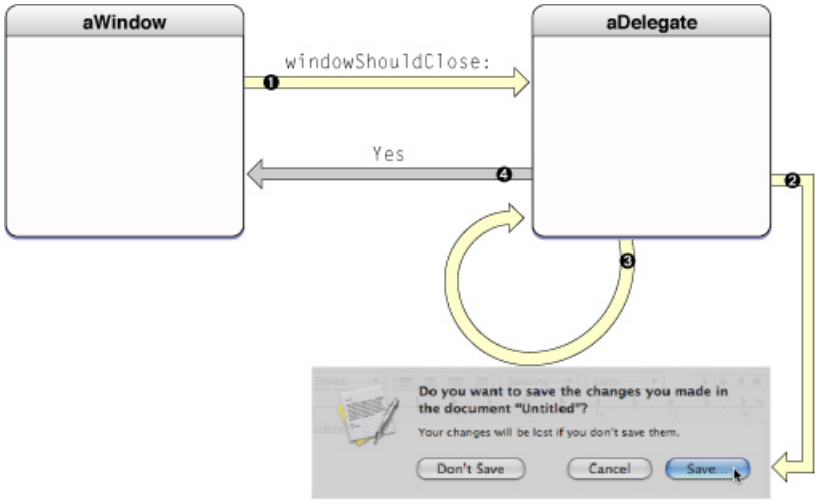


Figura 17. Respuesta a un evento con el método del delegado implementado [14]

2.4.4 Notificaciones

La manera estándar para pasar información entre los objetos es el paso de mensajes, un objeto invoca al método de otro. Sin embargo, el paso de mensajes requiere que el objeto que envía el mensaje sepa quién es el receptor y cuál es el mensaje de respuesta. A veces, esta estrecha conexión entre dos no es deseable, sobre todo porque sería unir lo que podría ser de otro modo dos subsistemas independientes. Y no es práctico, porque requeriría conexiones entre objetos muy dispares en una aplicación.

Para los casos en que el paso de mensajes estándar simplemente no se hace, Cocoa ofrece el modelo de difusión de la notificación. Al utilizar el mecanismo de notificación, un objeto puede tener otros objetos informados de lo que está haciendo. En este sentido, es similar a la delegación, pero las diferencias son importantes. La distinción fundamental entre la delegación y la notificación es que la primera es una ruta de comunicación uno-a-uno (entre el objeto que delega y su delegado). Sin embargo, la notificación es una forma de comunicación por difusión uno-a-muchos. Un objeto sólo puede tener un delegado, pero puede tener muchos observadores, que cómo los destinatarios de las notificaciones son conocidos. Y el objeto no tiene por qué saber quiénes son los observadores. Cualquier objeto puede observar un evento indirectamente a través de la notificación y ajustar su propia apariencia, comportamiento, y estado en respuesta al evento. La notificación es un mecanismo poderoso para alcanzar la coordinación y la cohesión en un programa.

Cómo funciona el mecanismo de notificación es conceptualmente simple. Un proceso tiene un objeto llamado centro de notificación, que actúa como un centro de selección y distribución de notificaciones. Los objetos que necesita saber acerca de un evento se registran al centro de notificaciones para comunicar a éste que desea ser notificado cuando este evento en concreto ocurra. Cuando el evento ocurre, el objeto que se encarga de la gestión del evento, envía una notificación al centro de notificaciones, que luego se distribuye a todos sus observadores.

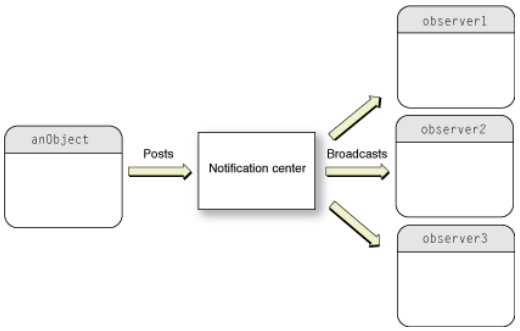


Figura 18. Envío y difusión de una notificación [15]

Cualquier objeto puede enviar una notificación y cualquier objeto puede registrarse en el centro de notificaciones en calidad de observador de la notificación. El objeto que envía la notificación, el objeto que se incluye en la notificación, y el observador de la notificación pueden ser todos objetos diferentes o el mismo objeto. Los objetos que envían las notificaciones no necesitan saber nada acerca de los observadores. Por otra parte, los observadores deben saber al menos el nombre de notificación y las claves para cualquier diccionario encapsulado por el objeto de notificación.

2.5 Modelado de Datos

Cuando se utiliza el framework Core Data, se necesita una manera de describir los objetos que no dependa de vistas ni controladores. En un diseño reutilizable, tanto las vistas como los controladores necesitan una forma de acceder a las propiedades del modelo sin que existan dependencias entre ellos. Core Data resuelve este problema basándose en conceptos básicos de las bases de datos tradicionales, especialmente el modelo entidad-relación.

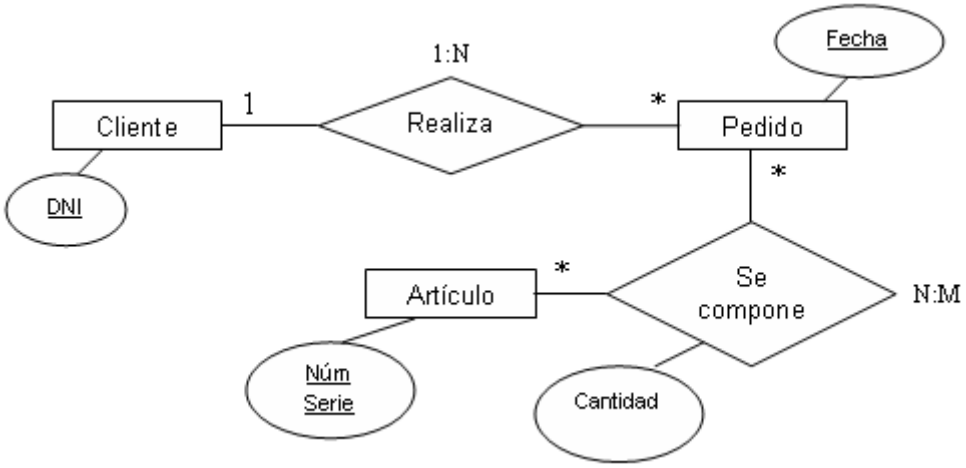


Figura 19. Diagrama E-R

El modelo entidad-relación es una manera de representar objetos típicamente usado para permitir que determinadas estructuras de datos puedan ser mapeadas a objetos. Es una representación que facilita el almacenamiento y recuperación de objetos de una fuente de información. Ésta puede ser una base de datos, un archivo, un Web Service, o cualquier otro tipo de almacenamiento persistente. Este modelo puede utilizarse para representar cualquier tipo de objeto y su relación con otros objetos, puesto que no depende de ninguna fuente de información en concreto.

El modelado de objetos es particularmente útil en la representación de los objetos en un patrón de diseño Modelo-Vista-Controlador (MVC).

2.5.1 Entidades

En el patrón de diseño MVC, los modelos son los objetos de la aplicación que encapsulan datos específicos y proporcionan métodos que operan sobre esos datos. Los modelos son por lo general persistentes, pero más importante aún, los modelos no dependen de cómo se muestran los datos al usuario.

En el modelo de entidad-relación, los modelos se denominan entidades, los componentes de una entidad se denominan atributos, y las referencias a otros modelos se llaman relaciones. Juntos, los atributos y las relaciones se conocen como propiedades. Con estos tres componentes (entidades, atributos y relaciones), se puede modelar de manera arbitraria sistemas realmente complejos. Si las partes de un sistema pueden ser identificadas, el sistema puede ser expresado como un modelo de objetos.

Figura 20. Diagrama de objetos [11]

2.5.2 Atributos

Los atributos representan las estructuras que contienen datos. Un atributo de un objeto puede ser un valor simple, como un escalar, pero también puede ser una estructura o una instancia de una clase primitiva. En Cocoa, un atributo corresponde por lo general a una variable de una instancia o a un método de acceso.

2.5.3 Relationships

No todas las propiedades de un modelo son atributos, algunas propiedades son las relaciones con otros objetos. Una aplicación suele estar modelada por varias clases. En tiempo de ejecución, el modelo de objetos es una colección de objetos relacionados que constituyen un gráfico de objetos. Éstos son los objetos persistentes que los usuarios crean, manipulan y guardan en un contenedor de datos antes de terminar la aplicación (como en una aplicación basada en el documento). Se puede acceder a las relaciones entre estos objetos del modelo en tiempo de ejecución para acceder a las propiedades de los objetos relacionados.

Las relaciones son inherentemente bidireccionales. Sin embargo, las relaciones pueden ser navegables en una sola dirección para que no exista relación inversa.

Todas las relaciones tienen una cardinalidad, ésta indica el número de objetos de destino que puede devolver la relación. Es posible especificar un rango para la cardinalidad que especifica un mínimo y un máximo.

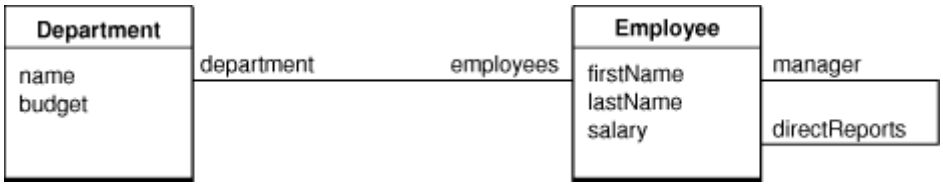


Figura 21. Relación [11]

2.5.4 Propiedades de acceso

Para que los modelos, vistas y controladores puedan ser independientes los unos de los otros, es necesario acceder a las propiedades de una manera que sea independiente de la implementación del modelo. Esto se logra mediante el uso de la tecnología key-value.

2.5.4.1 Claves

Las propiedades se especifican con una clave, habitualmente una cadena de caracteres. La vista correspondiente o el controlador utiliza la clave para buscar el valor del atributo asociado.

La codificación key-value utilizada para realizar esta búsqueda es un mecanismo para acceder a las propiedades de un objeto indirectamente y, en ciertos contextos, de forma automática. Utilizando como clave, el nombre de la propiedad del objeto de la que queremos saber el valor.

2.5.4.2 Valores

Todos los valores para un atributo de una entidad son del mismo tipo. El tipo de dato se especifica en la declaración de su instancia de variable correspondiente o en el valor de retorno de su método de acceso.

La codificación key-value solo devuelve objetos. Si se detecta un no objeto, se envolverá en una instancia NSNumber o NSValue. Del mismo modo, si cuando se fija un valor, el tipo de dato requerido para la clave específica no es un objeto, entonces el valor se extrae del objeto enviado utilizando el método apropiado.

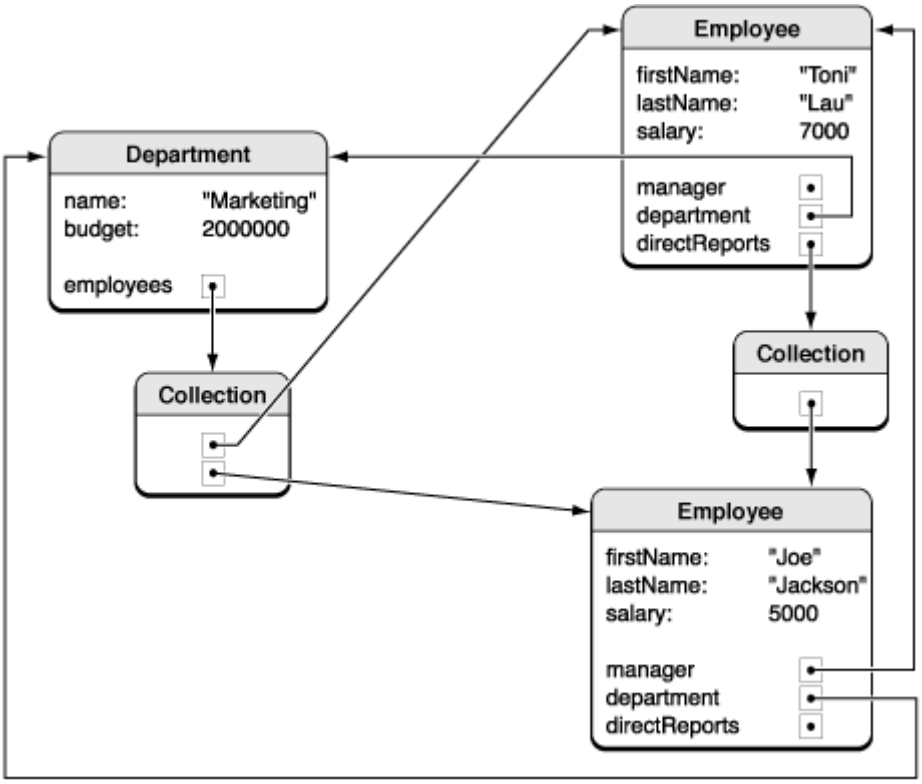


Figura 22. Objeto gráfico [11]

2.5.4.3 Key Paths

La ruta de la clave es una cadena de claves separadas por puntos que especifican una secuencia de propiedades a recorrer. La primera clave sirve de referencia, ya que todas las que siguen son relativas a la anterior. La ruta permite especificar las propiedades de objetos de una manera que es independiente del modelo.

2.6 Patrón de diseño Modelo-Vista-Controlador

El Modelo-Vista-Controlador (MVC) es un estilo de arquitectura de software que fue descrito por primera vez en 1979 por Trygve Reenskaug, entonces trabajando en Smaltalk en laboratorios de investigación de Xerox.

Este patrón de diseño separa los tres componentes de los que está formado, los datos de la aplicación, la interfaz de usuario y la lógica de control, siendo cada componente independiente del resto.

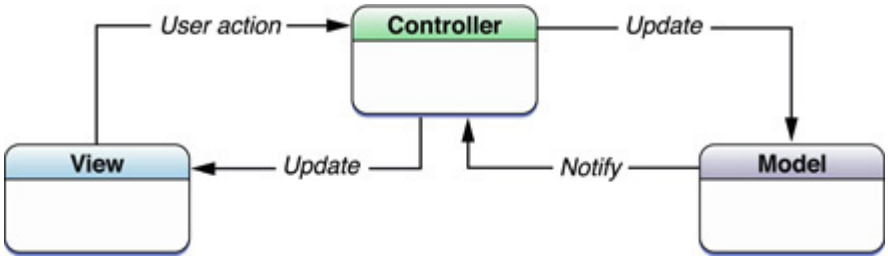


Figura 23. Modelo-Vista-Controlador [16]

Para desarrollar una aplicación Cocoa es fundamental un buen diseño del MVC. Más allá de los beneficios que esto conlleva respecto reutilización y escalabilidad, muchas de las tecnologías y arquitecturas de Cocoa están basadas en el MVC y requieren que nuestros objetos jueguen uno de los roles del patrón de diseño.

2.6.1 Modelo

Esta es la representación de la información con la que el sistema opera. La lógica que gestiona y procesa los datos también viene definida en el modelo. En éste deben residir los datos que forman parte del estado persistente. Idealmente, los objetos del modelo no tienen ninguna conexión explícita con los objetos de la vista.

2.6.2 Vista

Los objetos de la vista son aquellos que residen en la aplicación del usuario y que éste puede ver. Un objeto de la vista debe saber cómo presentar los datos al usuario y cómo responder a sus acciones.

Los objetos de la vista garantizan la consistencia entre aplicaciones, ya que generalmente son objetos que se reutilizan. Existen diversos frameworks que ofrecen colecciones de clases vista, y Interface Builder ofrece también objetos para la vista en su librería.

2.6.3 Controlador

El controlador responde a eventos, habitualmente acciones del usuario, e invoca peticiones al modelo y a la vista.

Un objeto controlador actúa de intermediario entre los objetos de la vista y los del modelo. Por consiguiente, los objetos del controlador responden a eventos, habitualmente acciones del usuario, e invocan peticiones tanto a los objetos del modelo cómo a los de la vista. Los objetos del controlador gestionan la coordinación de tareas para una aplicación.

2.6.4 Flujo del MVC

Las acciones del usuario en la vista se comunican mediante el controlador al modelo. Del mismo modo, cuando los datos del modelo se ven modificados, el modelo notifica al objeto del controlador correspondiente, el cual actualiza el objeto de la vista adecuado.

Así pues, el controlador será clave para que exista una buena comunicación entre modelo y vista. Los objetos del controlador tiene la función de interpretar las acciones del usuario hechas en los objetos de la vista y comunicarlas al modelo.

2.7 Tecnología key-value

2.7.1 Codificación Key-Value

La codificación key-value es un mecanismo de acceso a las propiedades de un objeto indirectamente, utilizando cadenas para identificar las propiedades, en lugar de a través de la invocación de un método de acceso o accediendo a ellas directamente a través de instancias de variable. En esencia, la codificación key-value define los patrones y los métodos que la aplicación implementa [17].

Los métodos de acceso proporcionan acceso a los valores de las propiedades del modelo de datos de la aplicación. Hay dos formas básicas de acceso *get* y *set*. *Get*, devuelve el valor de una propiedad. *Set*, establece el valor de una propiedad. Existen variantes para ambos, con la finalidad de gestionar tanto los atributos de los objetos, como las relaciones a varios.

Implementar la codificación key-value compatible para los métodos de acceso de la aplicación es un principio de diseño básico. Los métodos de acceso obligan a encapsular los datos adecuadamente, aislar la gestión de memoria en lugares centralizados, y facilitar la integración con otras tecnologías. Los métodos de codificación key-value pueden, en muchos casos, utilizarse también para simplificar el código de la aplicación.

Los métodos principales para la codificación de key-value están declarados en el protocolo NSKeyValueCoding de Objective-C, aunque las implementaciones por defecto vienen proporcionadas por NSObject.

2.7.2 Observación Key-Value

La observación key-value facilita un mecanismo que permite a los objetos ser notificados de los cambios en las propiedades de otros objetos.

La tecnología vinculante con la capa del controlador cuenta principalmente con la observación key-value.para ser notificada de los cambios. Para las aplicaciones que no cuentan con esta capa, la observación key-value les provee de métodos simplificados para implementar los inspectores y actualizar los valores de la interfaz de usuario [18].

A diferencia de la NSNotification, no hay ningún objeto central que envíe directamente una notificación a los observadores cuando se realizan cambios. NSObject

proporciona esta implementación base de la observación key-value, y raramente hay que reescribir estos métodos.

Se pueden observar las propiedades de cualquier objeto incluidos los atributos simples, en cualquier tipo de relación. La observación key-value proporciona una capacidad automática de observación a todos los objetos.

2.8 Frameworks

2.8.1 UIKit

El framework UIKit para iPhone OS es el framework equivalente al Application Kit para Mac OS X. Su objetivo es esencialmente el mismo: proporcionar todas las clases que una aplicación necesita para construir y gestionar su interfaz de usuario. Sin embargo, existen diferencias significativas en cómo los marcos realizar esta finalidad.

Una de las mayores diferencias es que, en el iPhone OS, los objetos que aparecen a la interfaz de usuario de una aplicación (text views, table views, y buttons) se comportan de manera diferente de los objetos de una aplicación para Mac OS X. Además, el control de eventos y la elaboración de los modelos en las dos plataformas son muy diferentes.

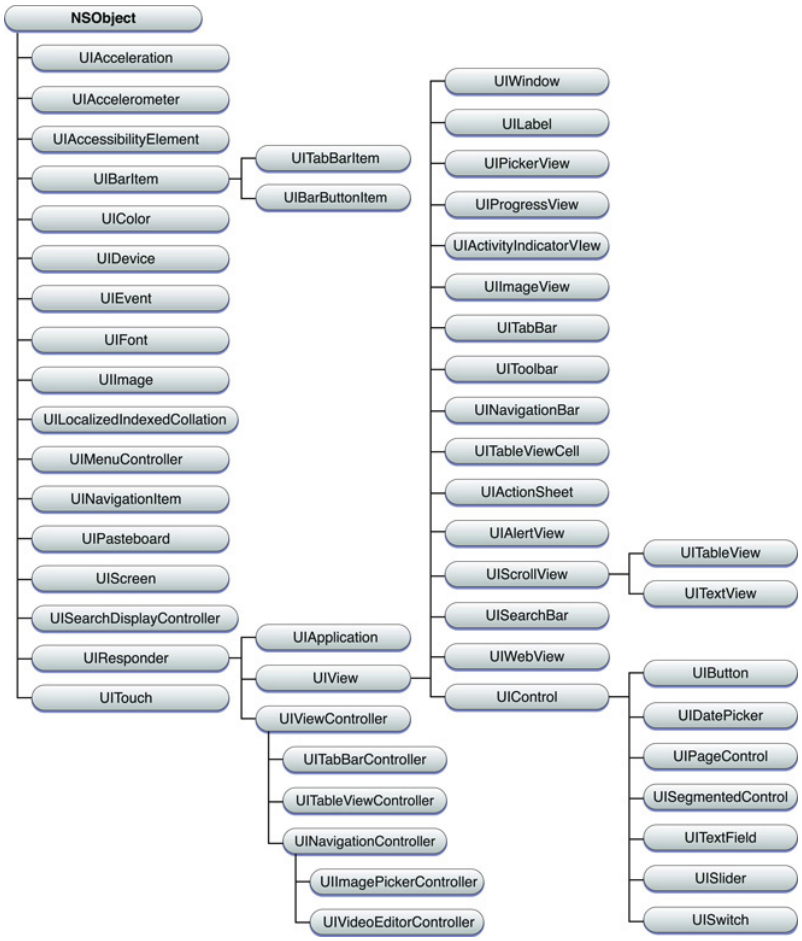


Figura 24. Visión general de las clases de UIKit [11]

2.8.2 Foundation

Este framework define una capa base. El criterio que separa las clases de Foundation de las de UIKit es la interfaz de usuario. Si un objeto no aparece ni en la interfaz de usuario ni es exclusivamente utilizado para dar soporte a la interfaz de usuario, esta clase pertenecerá a Foundation. Así pues, este framework define básicamente el comportamiento de objetos básicos y proporciona un cierto grado de independencia del sistema operativo para apoyar la portabilidad.

2.8.3 Map Kit

El framework MapKit permite integrar un mapa en la propia aplicación. Este framework incorpora una serie de clases y métodos que permiten variar los atributos del mapa en cuestión, indispensable para el desarrollo de los prototipos que se incluyen en el proyecto. Necesitaremos acceder a la posición actual donde se encuentra el dispositivo, y personalizar la información a mostrar.

2.8.4 Core Location

El framework Core Location permite determinar la latitud y la longitud exacta de la ubicación donde se encuentra el dispositivo. El framework utiliza el hardware disponible para triangular la posición del usuario. Se basa en señales de información del GPS, cobertura telefónica o WIFI.

2.9 Core Data

2.9.1 Arquitectura básica

En la mayoría de las aplicaciones, se necesita un medio para abrir un archivo que contiene, a su vez, un archivo de objetos, y una referencia por lo menos a un objeto raíz. Se necesita también ser capaz de guarda todos los objetos en un archivo y, si se quiere poder deshacer, tener constancia de los cambios en los objetos. Por ejemplo, en una aplicación de gestión de empleados, se necesita abrir un archivo que contiene, a su vez, un archivo de objetos de los empleados y departamentos, y una referencia, por lo menos, a un objeto raíz. También es necesario poder guardar en un archivo todos los empleados y todos los departamentos.

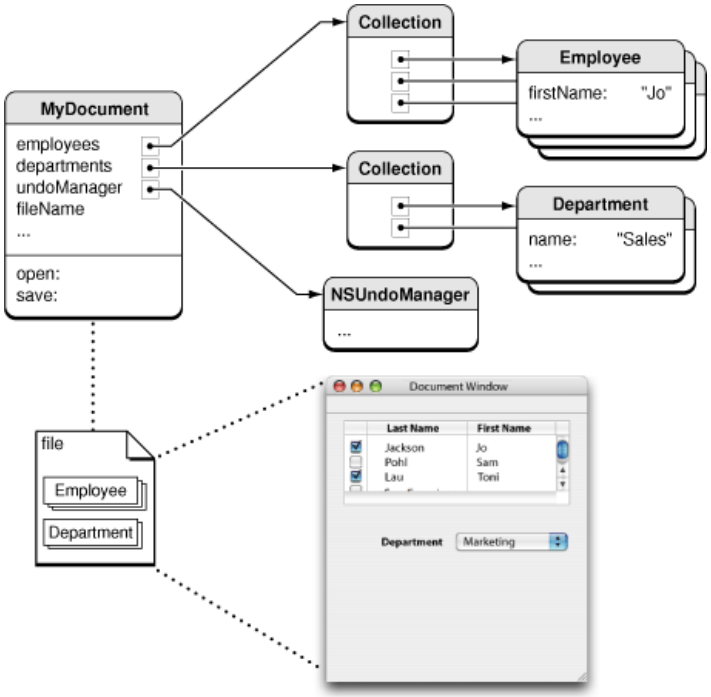


Figura 25. Gestión de documentos utilizando la arquitectura básica de Cocoa [19]

Utilizando el framework Core Data, la mayoría de estas funcionalidades se nos proporcionan de forma automática, principalmente a través de un objeto conocido como contexto. El contexto sirve como puerta de acceso a una colección subyacente de objetos, conocidos en conjunto como pila de persistencia, que media entre los objetos de la aplicación y los almacenes de datos externos. En la parte inferior de la pila se encuentran los objetos persistentes.

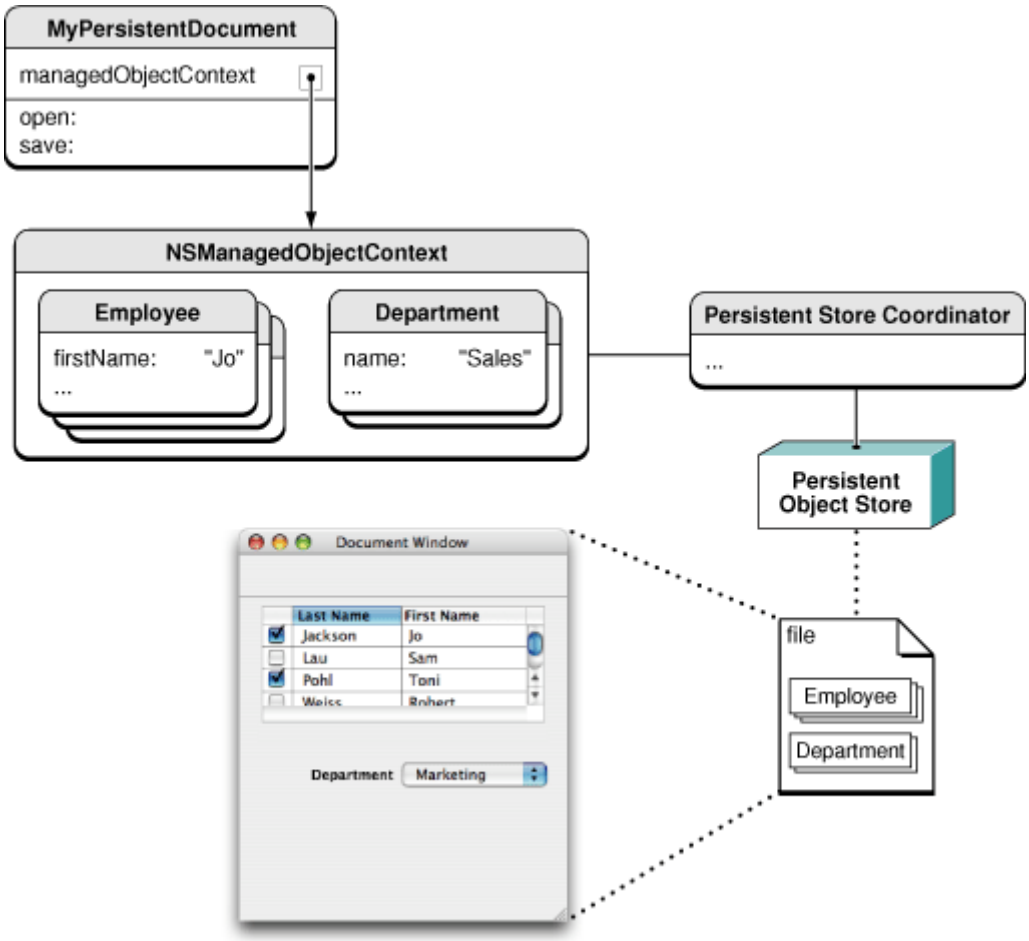


Figura 26. Gestión de documentos utilizando Core Data [19]

2.9.1.1 Objetos de gestión y Contextos

Se puede asociar un contexto a un bloc de notas inteligentes. Cuando se buscan objetos en un almacén persistente, se tendrán que traer copias temporales en el bloc de notas donde formaran un gráfico de objetos. A continuación, se pueden modificar los objetos y, a menos que se guarden los cambios, el almacén persistente permanece inalterado.

Todos los objetos para el modelo deben estar registrados en un contexto. Se agregan y se quitan objetos al gráfico utilizando el contexto. El contexto sigue los cambios que se hacen, tanto a los distintos atributos de los objetos como a las relaciones entre ellos. Mediante el seguimiento de cambios, el contexto es capaz de deshacer y rehacer. También asegura que la integridad del gráfico de objetos se mantiene si cambian las relaciones entre objetos.

Si se decide guardar los cambios que se han hecho, el contexto se asegura de que los objetos se encuentran en un estado válido. Si es así, entonces los cambios se escriben en el almacén persistente.

Es posible que haya más de un contexto en la aplicación. Por cada objeto en un almacén persistente puede haber como máximo un objeto asociado al contexto. Para considerar esto desde una perspectiva diferente, un objeto dado en un almacén persistente puede ser editado en más de un contexto al mismo tiempo. Cada contexto, sin embargo, tiene su propio objeto que se corresponde con el objeto de origen, y cada objeto puede ser editado de forma independiente. Esto puede dar lugar a incoherencias de datos al guardarlo. Core Data proporciona una serie de maneras para solventar esta cuestión.

2.9.1.2 Peticiones de búsqueda

Para recuperar los datos mediante los objetos del contexto, se crea una petición de búsqueda. Ésta no es más que un objeto que especifica los datos que desea. La petición consta de tres partes. Cómo mínimo debe especificar el nombre de una entidad (por implicación, sólo puede buscar un tipo de entidad a la vez). También puede contener un predicado que especifique las condiciones que deben cumplir los objetos y un vector que puede contener objetos que especifiquen el orden en que deben aparecer los objetos, por ejemplo.

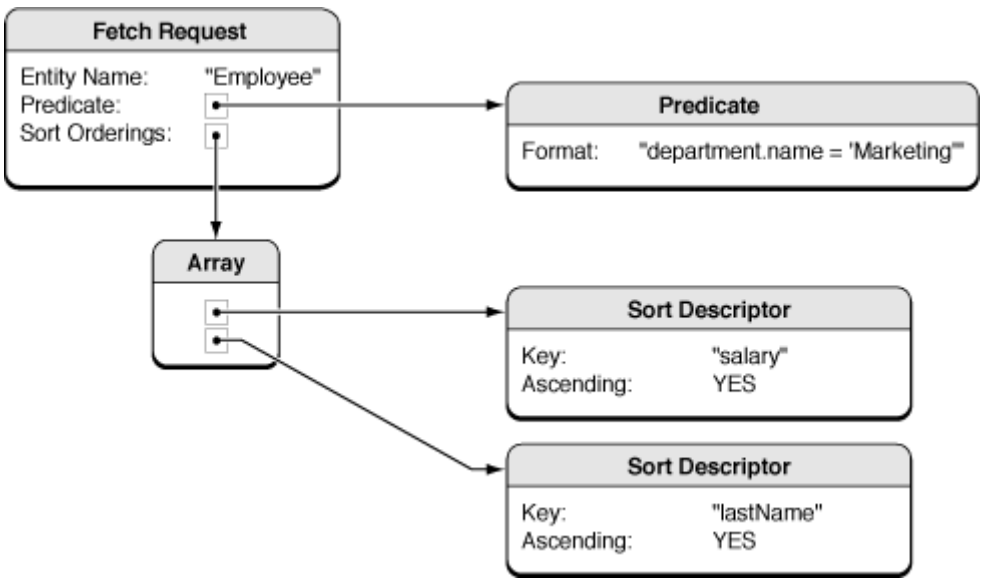


Figura 27. Petición de búsqueda [19]

Se envía la petición a un contexto, que devuelve los objetos que coinciden con la petición. Dado que todos los objetos deben estar registrados en un contexto, los objetos retornados de una petición se registran automáticamente en el contexto utilizado para la petición. Si un contexto ya contiene un objeto, entonces el objeto existente se devuelve cómo resultado de la petición.

El framework pretende ser lo más eficiente posible. Con Core Data no hay necesidad de crear más objetos de los que realmente se necesitan. El gráfico no tiene que representar a todos los objetos en el almacén persistente. El hecho de buscar en el almacén de datos no implica que éstos estén en el contexto. La petición solamente devuelve los objetos solicitados.

2.9.1.3 Coordinador de almacén persistente

Como se señaló anteriormente, la colección de objetos que median entre los objetos de una aplicación y almacenes de datos externos se conoce colectivamente como la pila de persistencia. En la parte superior de la pila están los contextos, en la parte inferior de la pila se almacenan los objetos persistentes. Entre los contextos y los objetos persistentes hay un coordinador.

El coordinador está diseñado para presentar una fachada a los contextos, de manera que un grupo de almacenes persistentes aparece como un único almacén global. Un contexto puede crear un gráfico de objetos sobre la base de la unión de todos los almacenes de datos que el coordinador cubre. Un coordinador sólo se puede asociar con un modelo. Si se quieres poner diferentes entidades en diferentes almacenes, se deben particionar las entidades definiendo configuraciones en los modelos.

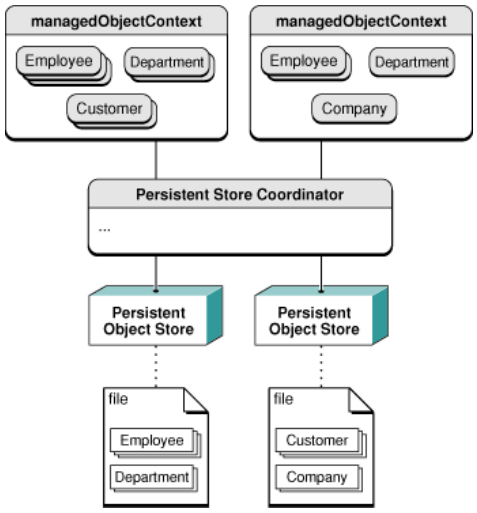


Figura 28. Pila de persistencia [19]

2.9.1.4 Almacén persistente

Un almacén de objetos persistentes se asocia con un único archivo o un almacén de datos externo y es en última instancia responsable de realizar un mapeo entre los datos en el almacén y los objetos correspondientes en el contexto. Normalmente, la única interacción que se tiene con un almacén de objetos persistentes es cuando se especifica la ubicación para asociarlo con la aplicación (por ejemplo, cuando el usuario abre o guarda un documento). La mayoría de las otras interacciones con el framework Core Data se realizan a través del contexto.

El código, y en particular la lógica de la aplicación asociada con la existencia de objetos, no debe hacer ninguna suposición sobre el almacén persistente en que los datos pueden residir. Core Data proporciona soporte nativo para varios formatos de archivo. Se puede elegir cuál utilizar en función de las necesidades de la aplicación. Si en algún momento se decide elegir un formato de archivo diferente, la arquitectura de la aplicación se mantiene sin cambios. Por otra parte, incluso si la aplicación inicial es capaz de traer sólo los registros del sistema de archivos local, si la aplicación no hace suposiciones acerca de dónde obtiene los datos, y posteriormente, se añade soporte para un tipo de almacén de datos remoto, debe ser capaz de utilizar este nuevo tipo sin revisiones de código.

2.9.1.5 Documentos persistentes

Se puede crear y configurar la pila mediante programación. En muchos casos, sin embargo, lo único que se desea crear es una aplicación basada en un documento capaz de leer y escribir archivos. La clase NSPersistentDocument es una subclase de NSDocument. De forma predeterminada, una instancia de NSPersistentDocument crea su propia pila de persistencia, incluido un contexto y un solo almacén de objetos persistentes. Hay en este caso un mapeo, uno a uno, entre un documento y un almacén de datos externo.

La clase NSPersistentDocument proporciona métodos para acceder al contexto del documento y proporciona implementaciones de los métodos estándar de NSDocument para leer y escribir archivos que utilizan el framework Core Data. Por defecto, no hay que escribir código adicional para manejar la persistencia de objetos. La funcionalidad de deshacer de un documento está integrada en el contexto.

2.9.2 Managed Objects and Managed Object Model

Con el fin de gestionar el gráfico de objetos y apoyar su persistencia, Core Data necesita una amplia descripción de los objetos con los que opera. Un modelo es un esquema que proporciona una descripción de los objetos gestionados que utiliza la aplicación. Normalmente, se crea el modelo gráficamente utilizando Xcode.

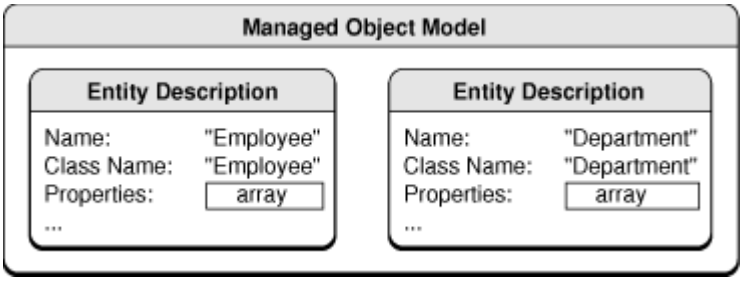


Figura 29. Modelo de objetos [20]

El modelo se compone de una colección de objetos de descripción de entidades dónde cada uno proporciona metadatos sobre una entidad concreta, incluyendo el nombre de la entidad, el nombre de la clase que la representa en la aplicación, y sus atributos y relaciones. Los atributos y las relaciones a su vez están representados por objetos de descripción de atributos y relaciones.

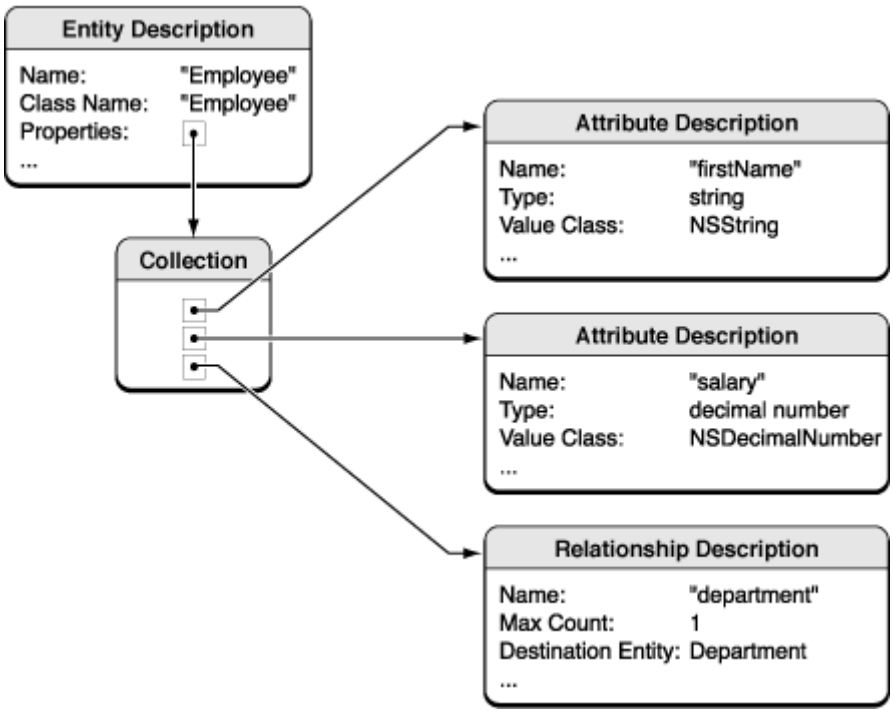


Figura 30. Descripción de una entidad [20]

Los objetos deben de ser instancias de NSManagedObject o de una subclase de éste. NSManagedObject es capaz de representar cualquier entidad. Utiliza un almacén privado para mantener sus propiedades e implementa todo el comportamiento básico necesario de un objeto. El objeto tiene una referencia a la descripción de la entidad para la entidad de la cual es una instancia. Se refiere a la descripción de la entidad para descubrir los metadatos acerca de sí mismo, incluyendo el nombre de la entidad que representa y la información sobre sus atributos y relaciones. También se pueden crear subclases de NSManagedObject para implementar un comportamiento adicional.

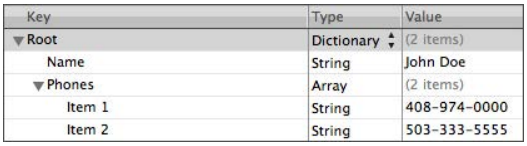
2.10 Servidores

Técnicamente existen dos opciones para realizar la conexión entre una aplicación para iPhone y una base de datos. Ésta se puede realizar directamente, o bien a través de un servidor web.

2.10.1 Conexión directa

2.10.1.1 Property Lists

Una property list es una representación de datos estructurados, utilizada como una forma de almacenar, organizar, y acceder a tipos de datos estándares. Coloquialmente referenciada como plist. Dado que las propety lists se basan en la abstracción para expresar jerarquías simples de datos, el formato de archivos subyacente se puede implementar de muchas maneras.



Key	Type	Value
▼ Root	Dictionary	(2 items)
Name	String	John Doe
▼ Phones	Array	(2 items)
Item 1	String	408-974-0000
Item 2	String	503-333-5555

Figura 31. Property List (Editor Xcode)

Procedimiento

Trabajar con property lists comporta, básicamente, tener que leer los datos de un archivo plist, convertir los datos en objetos y mostrar los datos del objeto en la interfaz de usuario. Configurarlo de manera que cuando los objetos son modificados, el plist también se modifique.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>Name</key>
    <string>John Doe</string>
    <key>Phones</key>
    <array>
        <string>408-974-0000</string>
        <string>503-333-5555</string>
    </array>
</dict>
</plist>
```

Figura 32. Property List (Editor de texto)

2.10.1.2 SQLite

SQLite es un sistema de gestión de bases de datos relacional de dominio público creado por D. Richard Hipp, compatible con las propiedades ACID, y que está contenido en una relativamente pequeña biblioteca en C.

La biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. El conjunto de la base de datos son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción.

```
$ sqlite3 test.db
SQLite version 3.0.8
Enter ".help" for instructions
sqlite> .quit
$
```

Figura 33. SQLite

Procedimiento

Deberemos configurar la base de datos SQLite, configurar la conexión, realizar las consultas pertinentes, convertir las consultas en objetos de la aplicación y mostrar en la interfaz de usuario los datos de los objetos. A partir de este momento, ya podemos trabajar con ellos.

2.10.1.3 Core Data

Core Data es un framework para la construcción de los componentes de la arquitectura modelo-vista-controlador (MVC), consistente en un conjunto de clases que implementan un mapeo objeto-relacional.

El framework Core Data permite una definición abstracta de los objetos del modelo, expresado en términos de entidades y sus relaciones. Los datos pueden manipularse a nivel de objeto, sin necesidad de preocuparse por el almacenamiento y la recuperación. El controlador de objetos disponible en el Interface Builder puede recuperar y manipular las entidades directamente.

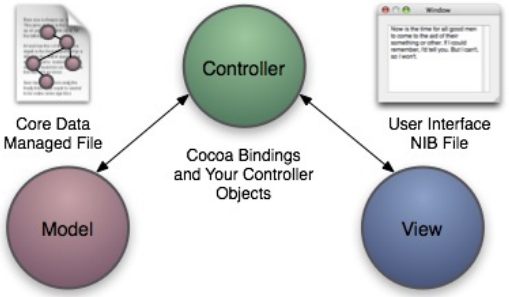


Figura 34. Modelo MVC

Procedimiento

Crear el esquema de datos en Xcode, arrastrarlo en una ventana del Interface Builder para crear una interfaz gráfica que nos permita el acceso y escribir el código estándar Objective-C necesario para leer/escribir archivos de/a nuestro almacén de datos (XML, binario o SQLite).

2.10.1.4 Comparativa

	Property List	SQLite	Core Data
Simple de entender y utilizar	X		
Queries complejas		X	X
En memoria, sólo los objetos necesarios		X	X
Cuenta con facilidades para manejar la migración cuando cambia el modelo subyacente			X
Gran cantidad de documentación y ejemplos			X
Control de la persistencia de forma automática			X
Control del *Deshacer*, por defecto			X

2.10.2 Web Service

Un Web Service es un tipo de aplicación emitida a través de HTTP (Hyper Text Transport Protocol), y distribuida, cuyos componentes se pueden implementar y ejecutar en dispositivos distintos.

Los Web Services están configurados para usar estándares de la industria, como HTTP y XML, que son omnipresentes y ampliamente conocidos. Por un lado, los Web Services pueden aprovecharse de las redes, formatos de datos, seguridad y otras infraestructuras ya en vigor, lo que reduce los costes iniciales y promueve la interoperabilidad entre servicios. Por otro lado, los Web Services y sus clientes pueden interoperar incluso si han estado escritos con diferentes lenguajes de programación. El diseño modular de los Web Services permite que los nuevos servicios puedan ser generados a partir de la integración y la estratificación de los servicios existentes.

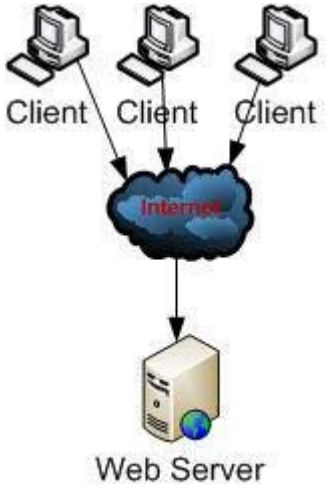


Figura 35. Web Server [21]

Los Web Services se pueden dividir a grandes rasgos en dos grupos, basados en SOAP (Simple Object Access Protocol) y REST (Representation State Transfer), aunque un servicio basado en SOAP emitido a través de HTTP es un caso especial de un servicio REST.

2.10.2.1 XML Web Services (JAX-WS)

Basado originariamente en SOAP, protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML (Extensible Markup Language).

JAX-WS (Java Api for XML Web Services) es el API Java que se utiliza para la creación de Web Service. JAX-WS forma parte del estándar Java EE de Sun

Microsystems. La implementación de referencia de JAX-WS es parte del proyecto GlassFish, y es de calidad productiva [22].

2.10.2.2 RESTful Web Services (JAX-RS)

Se trata de una técnica de arquitectura de software para sistemas hipermedia distribuidos como la World Wide Web.

Un concepto importante en REST es la existencia de recursos, a los que se puede acceder utilizando un identificador global. Para manipular estos recursos, los clientes y servidores se comunican a través de un interfaz estándar (HTTP) e intercambian representaciones de estos recursos [22].

3. Parte práctica

El desarrollo de un proyecto de estas características conlleva un seguido de etapas. Deberemos entender cada una de ellas para llegar a obtener una visión global del proceso que se ha llevado a cabo.

3.1 Fase de generación de ideas

Es importante saber la conexión entre la idea, la audiencia, el espacio y la tecnología, pero es básico establecer un punto de partida, necesitamos definir el concepto, la base que sustentará el proyecto.

3.1.1 Generación de ideas

Figura 36. Zoológico de Memphis

La primera fase de un proyecto de estas características nos obliga a preguntarnos cuál es el problema que queremos solventar, o lo que es lo mismo, cuál es nuestra aportación al mundo. Debemos observar qué es lo que nos rodea, cuáles son las necesidades de la gente para saber dónde está la oportunidad.

Del mismo modo, deberemos ser muy críticos con nuestras limitaciones, sean cuales sean. Deberemos ser conscientes de qué somos capaces de hacer en el tiempo estipulado. Disponer de un equipo multidisciplinar viene a suplir esta deficiencia, sumando conocimientos y experiencias en campos diferentes, enriqueciendo al conjunto. Pero aún así, el alcance deberá estar claramente definido.

En esta etapa todas las ideas son bienvenidas, cuantas más mejor, quedando excluida cualquier crítica al respecto de una idea en concreto, nunca se sabe de dónde va a surgir una buena idea. Toda nueva contribución puede derivar en una.

Figura 37. Generación de ideas

3.1.2 Organizar, evaluar y sintetizar posibilidades

Con todas las ideas en mano, el primer paso reside en organizarlas. Lo hicimos en función de sobre qué queríamos concienciar, resultando tres grandes grupos. Sostenibilidad, libertad de los animales y acción/impacto del hombre daban cabida al resto de ideas.

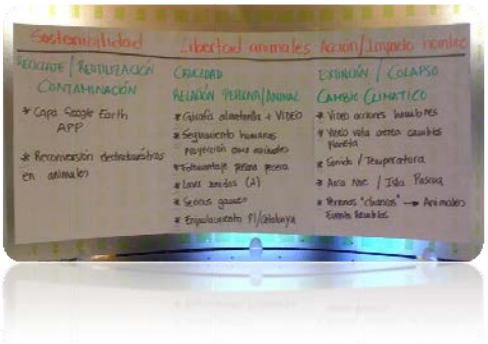


Figura 38. Ideas clasificadas

Seguidamente, evaluamos la importancia de cada una de las ideas teniendo en cuenta nuestros objetivos, y sintetizamos las posibilidades, eliminando aquellas que quedaban fuera de nuestro alcance, y englobando aquellas que coincidían en esencia.

Entre todos valoramos la viabilidad de cada una de ellas, a la vez que íbamos definiendo progresivamente la dirección del proyecto.

3.1.3 Conclusiones

Basándonos en todo momento en las necesidades del cliente, y con todo lo expuesto, conseguir **ponernos en la piel de un animal** resultaba una de las ideas más atractivas. Si a esto le sumábamos la importancia de la bidireccionalidad y la interactividad, cautivando la atención de las personas, parecía que teníamos el punto de partida.

3.2 Fase de desarrollo

Apoyándonos en las conclusiones de la primera fase, iniciamos tres propuestas en paralelo, las cuales empezamos a bocetar independientemente.

3.2.1 Aplicación de entretenimiento

Fomentando el entretenimiento, a la vez que la concienciación y la divulgación, valoramos el desarrollo de una aplicación de entretenimiento cómo una de las posibilidades para vestir nuestro concepto.

3.2.1.1 Investigaciones previas

Mercado de telefonía móvil

Consideramos el dispositivo móvil como una opción para ejecutar la aplicación, por lo que realmente nos interesa saber quién lidera el mercado.

Con la intención de ser el más duro competidor del iPhone, llega al mercado el **HTC Desire**, abanderado por el sistema operativo Android, Open Handset Alliance. Pero aún así, el **iPhone 3G S** es el móvil con más ventas del mundo.

Figura 39. HTC Desire [23]

Figura 40. iPhone 3G S [24]

Evolución de los juegos

Con el tiempo el concepto de juego ha evolucionado, sobre todo a partir de la creación del primer videojuego en 1947, éxito que se ha extendido hasta hoy día y que posee un futuro prometedor, tanto para la industria que hay detrás como para el propio consumidor.

Tamagotchi (1996), Bandai. Una mascota virtual a la que se podía dar de comer, llevar al baño, regañarla, curarla, entre otras actividades.

Figura 41. Tamagotchi, Bandai [25]

Zoo Tycon (2001), Microsoft Game Studios. Se trata de un juego de simulación económica donde se debe administrar un zoológico. Combina el entretenimiento y el aprendizaje ya que permite obtener información adicional de los animales con los que se está jugando. El objetivo es la de crear un zoológico, creando los hábitats de los animales y manteniéndolo de manera óptima para llegar al estado máximo de bienestar para el animal.

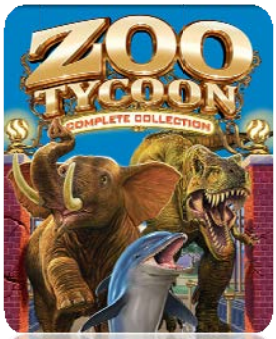


Figura 42. Zoo Tycon, Microsoft [26]

Nintendogs (2005), Nintendo. Es un videojuego que tiene como objetivo cuidar a un perro a lo largo de su vida. Un juego que deja atrás las mascotas virtuales como el Tamagotchi, evolucionando en términos de interacción entre el jugador y el personaje virtual.

Figura 43. Nintendogs, Nintendo [27]

EyePet (2009), SCEE. Es un juego para PlayStation 3 dónde la cámara permite a la mascota virtual interactuar con la gente y los objetos que le rodean mediante realidad aumentada.

Figura 44. EyePet, SCEE [28]

Figura 45. FarmVille, Zynga [29]

Farmville (2009), Zynga. Es un videojuego a tiempo real que permite manejar una granja virtual. Desde el lanzamiento del juego, éste se ha convertido en el más popular de la red social Facebook.

Facebook Developer

Existen tres opciones para desarrollar para Facebook. La metodología de trabajo vendrá definida por el destino de nuestra aplicación (Facebook, web o móvil). Aún así, todas las formas de desarrollo comparten aspectos generales y se basan en la vertiente social que proporciona Facebook.

Facebook facilita ciertas herramientas a los desarrolladores. El núcleo de la plataforma es la **API gráfica**, la cual permite acceder a las personas y a las relaciones entre ellas. La **autenticación** también es un aspecto potente de Facebook, y permite además interactuar con la API gráfica. Pero cabe destacar también los **plugins sociales**, extensiones de Facebook que permiten saber cómo se comportan los usuarios. Así como, el **protocolo Open Graph** que permite integrar nuestras páginas en la gráfica social.

Figura 46. Facebook Developer [30]

Google Earth

Google Earth permite visualizar imágenes en 3D del planeta, combinando imágenes de satélite, mapas y el motor de búsqueda de Google. La forma de moverse en la pantalla es fácil e intuitiva, con cuadros de mando sencillos y manejables.

Además, es posible compartir con otros usuarios enlaces, medir distancias geográficas, ver la altura de las montañas, y también dispone de conexión con Sistema de Posicionamiento Global (GPS), alimentación de datos desde fichero y base de datos en sus versiones de pago.

Figura 47. Google Earth [31]

3.2.1.2 Diseño conceptual

Partiendo de la investigación realizada, valoramos las opciones que teníamos sobre la mesa. Apostamos por conceptualizar una aplicación de entretenimiento para el iPhone. De manera aleatoria cada usuario se descarga un animal en concreto y a través de retos, el usuario es capaz de devolver al animal las habilidades que le han sido arrebatadas. Así pues, deberemos ayudarle a recuperar la memoria, así como a recuperar la velocidad, su agudeza visual y el arte de comer.

Figura 48. Diseño conceptual

3.2.2 Instalación interactiva

3.2.2.1 Investigaciones previas

Superfícies esféricas

Sphere (2008), Microsoft. Es un producto que consiste en una pantalla multitáctil interactiva esférica que utiliza hardware óptico personalizado, así como la visión por ordenador y software personalizado para gráficos.

Figura 49. Microsoft Sphere [32]

Magic Planet (2009), Global Imagination. Se trata de una pantalla esférica que ofrece interactividad, y una línea de visión de 360º. Se puede integrar con pantallas táctiles, así como con videojuegos. Se puede fijar al suelo, en la pared o colgarla del techo. El contenido puede ejecutarse desde un lector de DVD, desde un servidor, en streaming desde una fuente de vídeo o desde la aplicación de un ordenador.

Figura 50. Magic Planet [33]

Figura 51. Glooo [34]

Glooo (2004), Igor Polykov. Es un prototipo de un nuevo formato de navegador. Este dispositivo da el punto de vista del estado actual del planeta, las condiciones meteorológicas, titulares de noticias personalizadas y contactos en línea. El diseño también incluye el uso de un teclado láser para comunicarse con el resto del mundo.

Multi-touch 360 Sphere (2009), Thorsten Blum y Johann Korndorfer. Es un proyecto basado en un proyecto previo que rinde homenaje a Space Invaders.

Figura 52. Space Invaders 360 [35]

Proyecciones

Media Markt Interactive Floor, Múnich. Utiliza la tecnología Displax™ Moovit para hacer una proyección en el suelo que permite la interacción con el usuario.

Figura 53. Media Market Floor [36]

Bugs, Berlín. Instalación interactiva dónde se proyectan hormigas en un hormiguero con multicaras, donde los usuarios con sus propias sombras pueden interferir en el movimiento y guiar a las hormigas en la dirección que deseen.

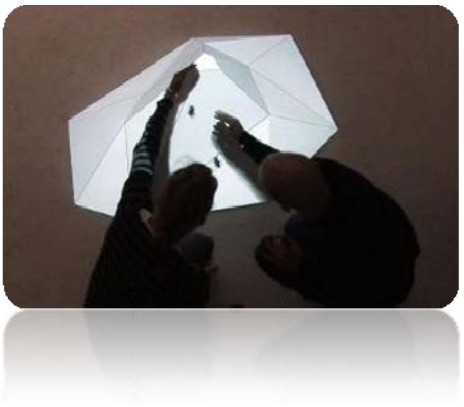


Figura 54. Bugs [37]

3.2.2.2 Diseño conceptual

La finalidad de promover una instalación de estas características reside en el hecho de fomentar un aprendizaje de manera interactiva, ya sea creando una experiencia individual, o ofreciendo una experiencia colectiva.

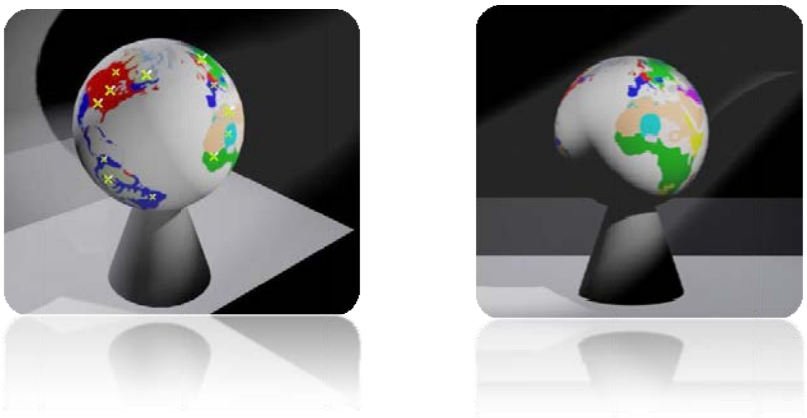


Figura 55. Diseño de la esfera interactiva

El logo del proyecto, creado a partir de figuras geométricas básicas, ajustándose a la idea de vincular la tecnología con la naturaleza, se convierte en un elemento básico de la esfera interactiva.

Figura 56. Logo ZOO XXI

Cada una de las aspas del logo sirve para representar puntos localizados en la esfera interactiva, y nos permiten acceder a material audiovisual de una forma atractiva para el espectador.

Para combinar el eje óptico de la cámara y el proyector a través de una sola lente, utilizamos un espejo frío (refleja la luz visible y transmite la luz infrarroja). La proyección de la luz llega a la superficie difusa y se dispersa, cuando el usuario toca la superficie se refleja luz infrarroja que es capturada por la cámara.

Figura 57. Funcionamiento esfera

3.2.3 Conexión de escenarios

Las ideas desarrolladas en paralelo eran muy atractivas independientemente, pero ¿Como de potente podía ser un proyecto que las integrara? Debíamos idear cómo podíamos hacerlo posible.

Figura 58. Conexión de escenarios

Creamos un sitio web donde al conectarse el usuario accediera automáticamente a una introducción. La finalidad de ésta es intentar poner en situación al usuario. La introducción inicial sitúa al usuario en el mundo animal, concretamente en la sabana, pudiendo apreciar los distintos animales que allí habitan. Pero de forma repentina aparece la figura de un personaje, que tiene el rol de llevarse cuantos más animales mejor en su mundo digital, creando una analogía con la realidad, ya que del mismo modo hay quién se dedica a llevarse los animales de su hábitat para transportarlos a lugares artificiales. Se intenta pues, hacer partícipe al usuario de esta situación enseñándole la índole de la acción que se acaba de realizar. Se le muestra toda la información relacionada, fecha, ubicación, número de especies y número de individuos. En este momento se presenta una isla. Una isla que el personaje anteriormente mencionado utiliza de cárcel para los animales. Queremos que el usuario se implique y le pedimos su ayuda. Si acepta la misión, el siguiente paso será descargarse la aplicación.

Figura 59. Storyboard de la introducción

Figura 60. Descarga de la aplicación [38]

El mismo sitio web nos facilita el enlace a la descarga de la aplicación. Una vez ejecutada en el dispositivo móvil, el usuario deberá entrenar a su animal para que recupere las habilidades perdidas en su cautiverio, para que cuando llegue el día, asista a un evento, y allí pueda liberar al animal otra vez en su hábitat natural, siendo capaz de sobrevivir en él.

La liberación del animal se realiza en la esfera interactiva, el logo del proyecto nos sirve como representación de los animales que han sido liberados. Cada vez que una persona libera un animal vía bluetooth, el logo se coloca en la esfera para simbolizar la liberación. De esta manera, se consigue aportar un feedback al usuario. Si éste ve representada gráficamente su aportación al proyecto, su participación habrá sido más merecedora y habrá más posibilidades que vuelva a participar.

El evento es un punto de encuentro, con gente real compartiendo una experiencia virtual. Una experiencia que culminará el proceso de sensibilización del usuario respecto a la realidad de los animales. Y todo ello, abanderado por una campaña de promoción y marketing de guerrilla, con acciones en la calle y con publicidad del proyecto. Una estrategia ideada para llegar a cuánto más público mejor.

Figura 61. Etiquetas publicitarias

Para promocionar el proyecto, se desarrollaron un par de carteles que pretendían transmitir una imagen actual, manteniendo la esencia de la naturaleza, pero uniéndola a las nuevas tecnologías.

Figura 62. Carteles promocionales

Del mismo modo se creó el sitio web para el proyecto, así como un par de propuestas para la página oficial de ZOO XXI, donde se pretenden alojar todos los proyectos que formen parte éste. Presentamos un prototipo más moderno e innovador, y otro, un poco más tradicional.

Figura 63. Sitio web del proyecto

Figura 64. Propuesta 1 proyecto ZOO XXI

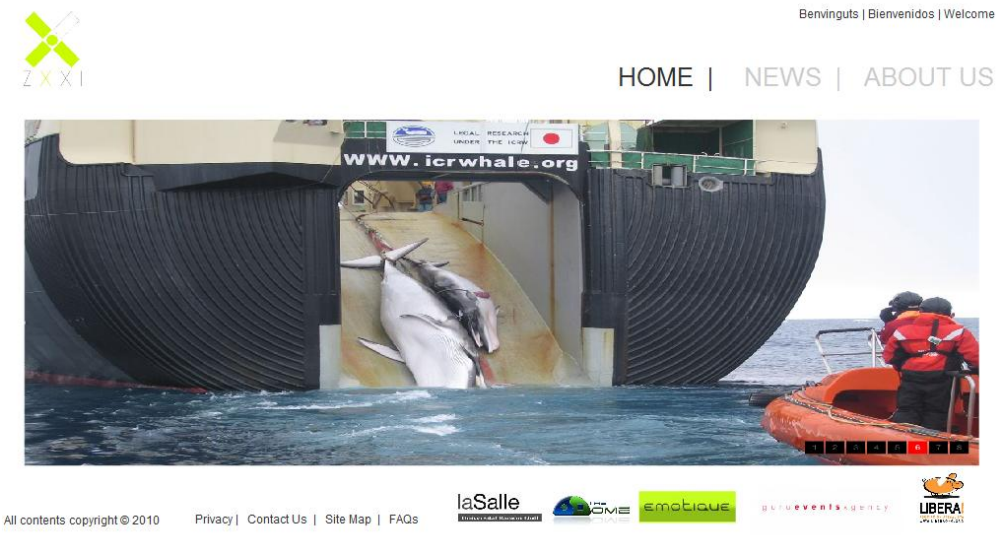


Figura 65. Propuesta 2 proyecto ZOO XXI

3.2.4 Análisis de la situación

A raíz de las reuniones con el cliente, después de un par de meses nos dimos cuenta de que estábamos perdiendo de vista la visión global del proyecto. Por lo que decidimos hacer un replanteamiento.

¿Cuál era exactamente nuestro objetivo? ¿Realmente estábamos cumpliendo con las expectativas del cliente? Nos habíamos centrado en cerrar un proyecto cuando en realidad, de lo que se trataba era de hacer un proyecto abierto al máximo posible para que cualquier persona pudiera contribuir en él.

Qué es Zoo XXI? Un proyecto que pretende ser una alternativa y/o un complemento a los zoológicos existentes. ¿Zoo XXI lleva el sello de una población en concreto? Precisamente la respuesta era que no, Zoo XXI pretende ser un proyecto genérico, extrapolable a cualquier población a nivel mundial. Y entonces surgió una nueva perspectiva. ¿Porqué no intentar llevamos el zoo a la ciudad? Que cualquiera tuviera acceso sin necesidad de desplazarse.

Hicimos balance de lo que teníamos, y hacia dónde nos llevaba este nuevo enfoque. Valoramos los riesgos que implicaba un cambio de dirección (tiempo, tecnología, etc.), pero teniendo en cuenta que el concepto es la base de cualquier tipo de conocimiento, tanto científico como filosófico y, que precisamente esto era lo que queríamos enfatizar, concluimos que valía la pena dar un paso atrás y retomar la visión global del proyecto.

La base propiamente dicha es el concepto. A partir de éste se construyen todos los proyectos en los que existe un equilibrio entre aspecto (diseño) y funcionalidad (desarrollo). La usabilidad debe ir implícita. Y la finalidad concreta vendrá definida por cada proyecto en particular. El triángulo se encuentra inmerso en una enorme esfera formada por millones de usuarios. Así pues se debe pensar en ellos desde la base hasta la cúspide a la hora de afrontar cualquier proyecto. Solamente los usuarios decidirán si se alcanza la finalidad marcada en el punto inicial.

Figura 66. Pirámide del proyecto

3.3 Fase de merchandising

La publicidad tiene dos objetivos. En primera instancia, idealmente, informar al consumidor sobre los beneficios de un determinado producto, resaltando la diferenciación sobre otras marcas. En segundo lugar, la publicidad busca inclinar la balanza motivacional del sujeto hacia el producto anunciado por medios psicológicos.

AIDA es una de las teorías más antiguas que describe los efectos que produce un mensaje publicitario. Basándonos en este modelo clásico, nos centramos en sus cuatro escalones. ¿Cómo se capta la Atención? ¿Cómo se consigue despertar el Interés? ¿Cómo se despierta el Deseo? ¿Cómo se conduce a la Acción?

3.3.1 Atención

Para despertar la curiosidad de los usuarios hemos ideado unos carteles publicitarios para promocionar el proyecto, haciendo especial énfasis en el hecho que ZOO XXI es mucho más que un zoo.

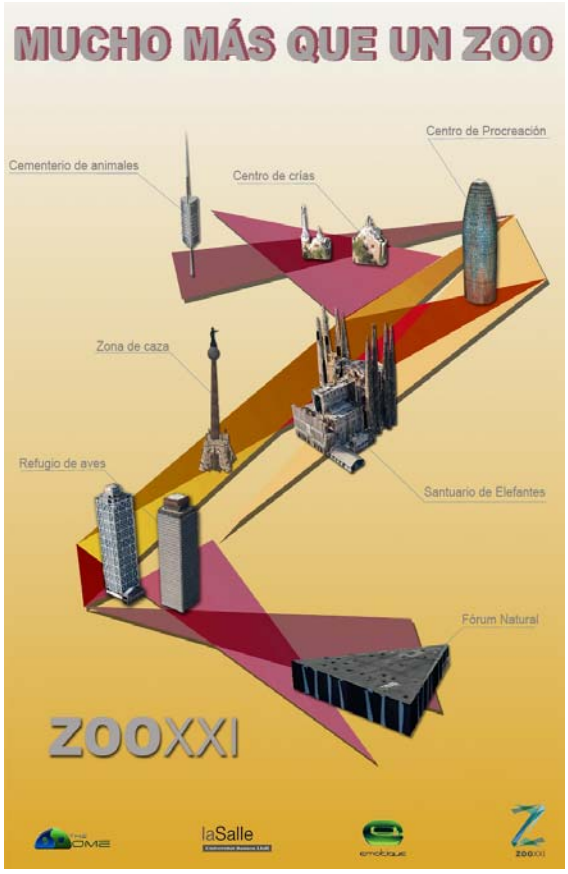


Figura 67. Cartel promocional

Del mismo modo, se ha ideado una cuartilla para poder hacer una mayor difusión, ya sea en bares y cafeterías, restaurantes, tiendas, centros comerciales, estaciones de transporte público, centros de educación y universidades, etc.

Figura 68. Cuartilla promocional

Y, en esta misma línea, se ha pensado en la opción de fabricar una chapa promocional con el logo del proyecto, extendiendo más aún la campaña promocional, y promover un marketing boca a oreja, basado en el marketing de 3ª generación apoyado también por los canales que facilita internet (blogs, foros y e-mails).

3.3.2 Interés

Despertar el interés es consecuencia inmediata de captar la atención, lo que se pretende conseguir con la acción continuada de promoción del proyecto, orientado a la participación en el mismo.

3.3.3 Deseo

El proyecto pretende despertar el deseo del usuario, convenciéndole de los beneficios del producto demostrándole hasta dónde se puede llegar. Un proyecto de estas características es muy jugoso a la hora de disfrazarlo según los intereses del cliente. El deseo de poseer existe de forma natural, la clave está en encontrar la manera de vender el producto personalizado al destinatario.

3.3.4 Acción

La acción se concreta con la adquisición del producto, totalmente gratuito para nuestro público objetivo.

3.4 Conclusiones

El concepto es la clave de nuestro proyecto. Nuestro objetivo es demostrar la fuerza de nuestro concepto, el resto se alimenta de éste. No se trata de un producto definido, sino de todo lo que gira a su alrededor. Ello permite la implementación de diversas aplicaciones que puedan adaptarse a cualquier público, manteniendo el objetivo de llevar el zoológico y sus animales a la ciudad.

Hemos partido de una fase de generación de ideas dónde un equipo multidisciplinar ha sido la clave para poder enfocar el proyecto desde un punto de vista suficientemente global como equipo, teniendo en cuenta los intereses del cliente desde las diferentes perspectivas y complementando unas con otras.

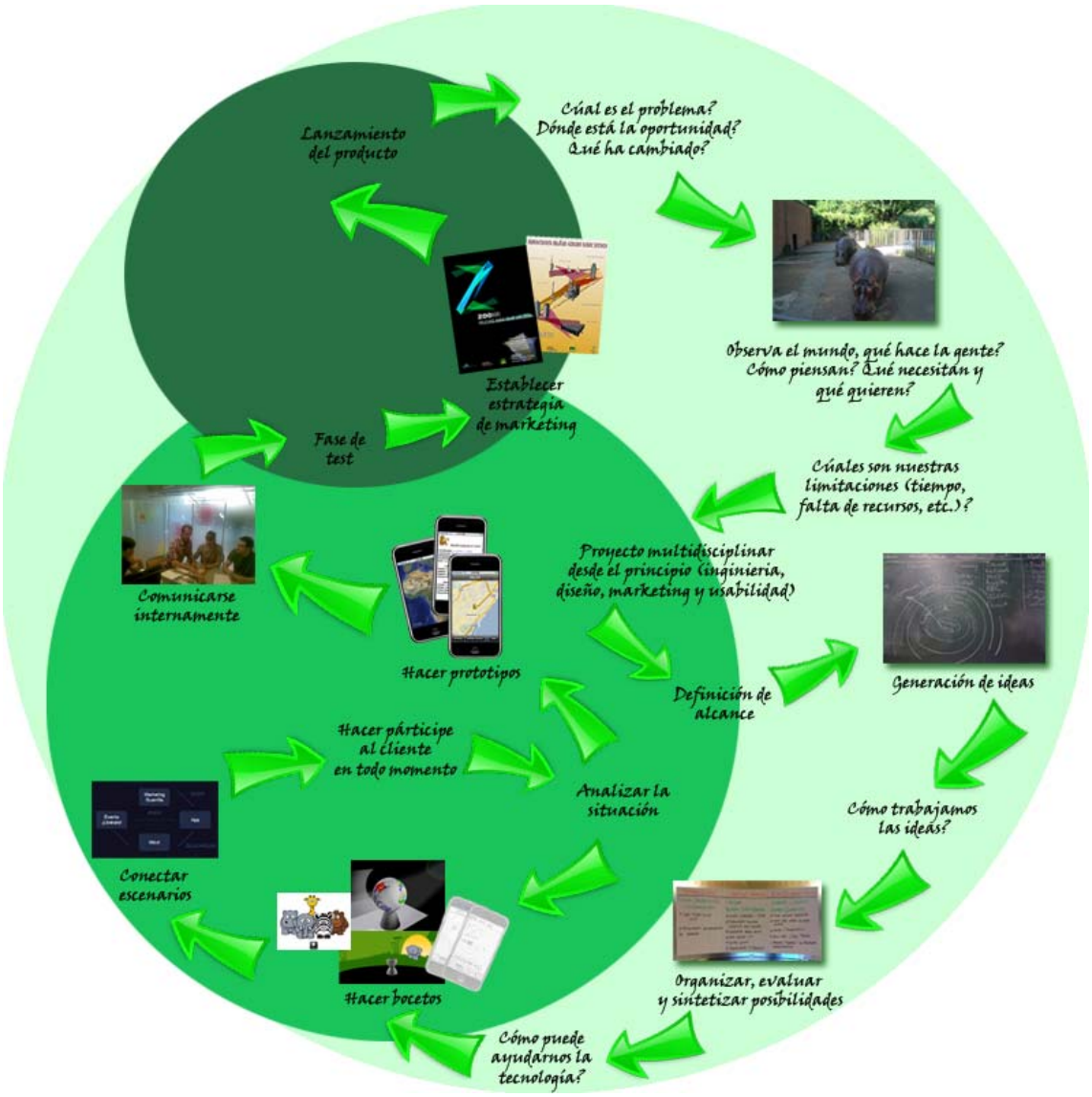


Figura 69. ZOO XXI: mucho más que un zoo

4. Resultados

4.1 Prototipo 1: Wikipedia

4.1.1 Investigaciones previas

Representación de datos

Mapa de la biodiversidad, Museo de las Ciencias Naturales de Barcelona. La base de datos del museo contiene más de 50,000 registros recolectados por todo el mundo a lo largo del siglo, con mayor densidad en la península ibérica, así como al oeste del mar mediterráneo.

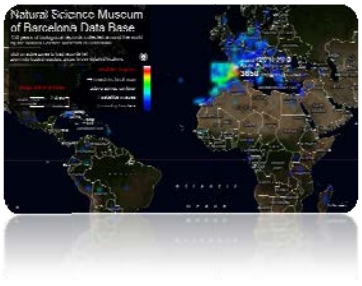


Figura 70. Mapa de la biodiversidad [39]

Figura 71. Atlas de Innovación [40]

Atlas de Innovación, World Alliance for Innovation. Este Atlas presenta la primera visión panorámica de parques científicos y tecnológicos e incubadoras de empresas. Los datos fueron recogidos entre más de 600 organizaciones de base tecnológica extendidas a través de 76 países.

Proyectos de Conservación, WWF. Google Earth permite el uso de capas a diferentes organizaciones para que hagan públicos sus proyectos, entre ellos, la WWF que muestra todos los proyectos de conservación que están en marcha alrededor del mundo.

Figura 72. Proyectos de Conservación [31]

Figura 73. NOAH [41]

NOAH, Networked Organisms and Habitats. Noah es una herramienta que los amantes de la naturaleza puede utilizar para explorar y documentar la fauna y flora. Del mismo modo, los grupos de investigación pueden aprovechar los conocimientos de los ciudadanos de todas partes del mundo.

4.1.2 Diseño conceptual

Esta aplicación tiene como finalidad constituir una herramienta de consulta disponible y de interés para distintos ámbitos. Desde universidades, pasando por centros de investigación, protectoras de animales, organizaciones e instituciones, centros de educación, hasta particulares.

4.1.3 Diseño funcional

El diseño funcional nos sirve para tener una idea de qué se puede hacer en la aplicación desde el punto de vista del usuario. A continuación se muestran las principales funcionalidades definidas.

- Visualización de datos
- Filtrado de datos
- CRUD (crear, obtener, actualizar y borrar; funcionalidades básicas en una aplicación)

En la imagen podemos ver el diseño funcional realizado con la herramienta Balsamiq Mockups[42].

Figura 74. Diseño de funcionalidades

4.1.4 Diseño gráfico

El diseño gráfico ha sido desarrollado teniendo en cuenta tanto la usabilidad de la aplicación, basándonos en la guía de interfaz para iPhone publicada por Apple Inc., cómo la innovación en el formato de representación de los datos.

La pantalla de inicio nos muestra el logo del proyecto, que dista mucho del que se ideó en un principio, éste, jugando también con figuras geométricas básicas, busca un poco más la integración con la instalación central de ZOO XXI que tiene estructura de rizoma.

Figura 75. Logo ZOO XXI

Figura 76. Edificio central ZOO XXI

Toda la información aparece georreferenciada en el mapa del mundo, pudiéndose utilizar de manera alternativa un formato de lista. Cada entrada de información tiene asociada una ficha dónde puede consultarse tanto el nombre, como la ubicación, como una descripción; existiendo también la opción de tener una imagen asociada.

Figura 77. Diseño gráfico aplicación

4.1.5 Diseño tecnológico

La arquitectura del servidor, la conexión de éste con la aplicación, así como el desarrollo propiamente dicho de ésta última quedan detallados en el Apéndice.

4.2 Prototipo 2: Juego

4.2.1 Investigaciones previas

Social Gaming

Los juegos sociales, permiten interactuar a los distintos jugadores, permitiendo la intercomunicación y la invitación de nuevos usuarios para participar del mismo generando grupos sociales virtuales en el contexto del juego. Esta acción genera, a la vez un grado de fidelidad al juego, ya sea para mantener ese contacto o por la interacción misma con los otros usuarios, incentivando a la vez el espíritu competitivo, creativo y el entretenimiento interactivo. El concepto de viralidad es uno de los factores que hace a este negocio tan atractivo. Social gaming es un concepto bastante amplio que engloba diferentes categorías.

Board games, caracterizados por la utilización de un tablero en dónde se desarrolla el juego.

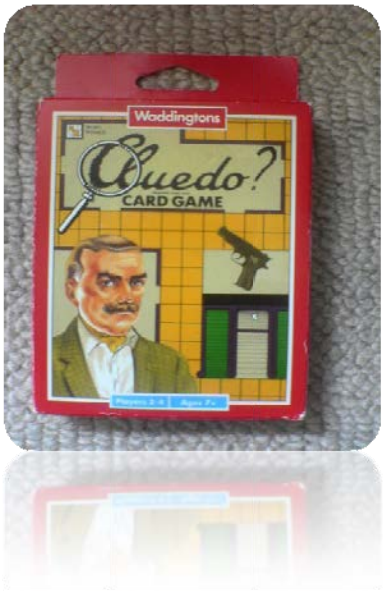


Figura 78. Cluedo, Waddingtons [43]

Multiplayer games, caracterizados por poder jugar más de un jugador en un mismo escenario al mismo tiempo.

Figura 79. Invizimals, Sony [44]

Role-playing games son aquellos en que el jugador debe asumir un rol.

Figura 80. Spore, EA [45]

Social casual games, se caracterizan por tener una red social integrada.

Figura 81. PetVille, Zynga [46]

4.2.2 Diseño conceptual

Esta aplicación coge cómo punto de partida el concepto de capa de Google Maps. Es decir, crear una única capa con un identificador, destinada a contener nuestro Google Map. Una capa contenedora de ZOO XXI que permitirá, gracias a la conexión entre el dispositivo y un servidor, conocer a tiempo real la ubicación de todos los usuarios, pudiendo interactuar entre ellos en un ecosistema urbano.

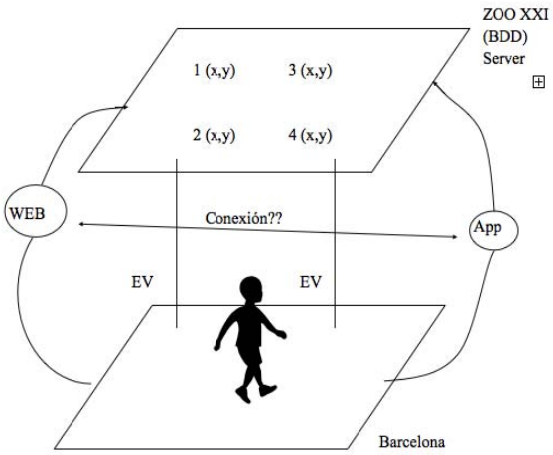


Figura 82. Diseño conceptual

4.2.3 Diseño funcional

La idea es poder escoger cualquier animal del mundo, para que éste se convierta en nuestro avatar, y por consiguiente nos represente a partir de ese momento.

Para que la aplicación sea atractiva, se han determinado tres acciones que el usuario tiene la posibilidad de realizar, hacer manada, procrearse y alimentarse. Acciones básicas de supervivencia que permiten al usuario conocer y aprender cómo se comporta el animal que ha escogido.

Figura 83. Diseño funcional

4.2.4 Diseño gráfico

Este es el diseño gráfico de la aplicación, basado en el del prototipo anterior.

Figura 84. Diseño gráfico aplicación

4.2.5 Diseño tecnológico

La arquitectura del servidor, la conexión de éste con la aplicación, así como el desarrollo propiamente dicho de ésta última quedan detallados en el Apéndice.

4.3 Prototipo 3: Acción de guerrilla

Se trata de una acción de guerrilla en la ciudad, e impulsada como una acción de marketing, tanto para dar a conocer ZOO XXI, como para ver las reacciones de los ciudadanos frente a acciones de estas características.

El concepto reside en el hecho de llenar las calles de animales mediante sonidos, esculturas realizadas con cartón-piedra, pintadas en el suelo aprovechando las bocas de acceso a las redes de alcantarillado de la ciudad, en carteleras publicitarias mediante material reflejante, etc., pero siempre grabando con una cámara oculta la reacción de la gente al verlos.

Figura 85. Antwerp Zoo, Bélgica [47]

Figura 86. Zoo Santa Fe, Colombia [48]

Las acciones serán puntuales y en espacios públicos transitados. La gente que sea objeto de la acción será informada posteriormente, y sólo si se autoriza se incluirá el material recogido como parte del proyecto, respetando en todo momento la Ley Orgánica 15/1999 de 13 de diciembre de Protección de Datos de Carácter Personal (LOPD) y la Ley 1/82, de 5 de mayo, de Protección Civil del Derecho al Honor, a la Intimidad Personal y Familiar, y a la Propia Imagen.

5. Conclusiones

El éxito global de un proyecto requiere obtener un considerable valor integrado para el cliente. Esto a su vez, requiere una combinación aceptable de éxito tecnológico, fabricando un producto con la tecnología adecuada que funcione según lo previsto, y de éxito en el mercado, obteniendo la participación deseada.

La mayor parte de los intentos de crear un producto afrontaron cierto grado de dificultad para responder a las expectativas en uno de estos aspectos, o en ambos. Así pues, para aumentar las probabilidades del éxito apostamos por una innovación conceptual.

ZOO XXI: mucho más que un zoo es **extrapolable** a cualquier otra ciudad del mundo, adaptándose fácilmente a su ubicación. Es un proyecto **adaptable** a las posibilidades de crecimiento constante del concepto, pudiéndose ampliar a las necesidades de la sociedad. Y, del mismo modo, facilitando la **integración** con proyectos existentes. Un proyecto **sostenible**, que mantiene un equilibrio con los recursos de su entorno. Facilita la interacción con los animales, evitando la explotación de éstos y satisfaciendo las necesidades de la actual generación.

Sin ir más allá, la fácil incursión en el mundo en el que vivimos, dónde la falta de tiempo es una característica en el día a día de los ciudadanos, lo convierte en un proyecto **omnipresente**.

El proyecto complace pues, el objetivo del cliente de llegar a todos los públicos. Por un lado, para el público joven podemos ofrecer una aplicación iPhone que permita una finalidad educativa y de concienciación mientras juegan e interactúan con otros jugadores (concepto de Serious Game). Por otro lado, para un público adulto interesado por el tema del conocimiento y protección de la biodiversidad podemos ofrecer otra aplicación, básicamente informativa y divulgativa (tipo Wikipedia), en la que los usuarios puedan crear, compartir y consultar información. Y, destinado a captar la atención de todos los públicos, se ha desarrollado una acción de guerrilla. Se trata de una acción de marketing a desarrollar en los espacios abiertos de la ciudad, grabando las diversas situaciones que puedan derivarse de las acciones desarrolladas. Del mismo modo, se ofrece la posibilidad de integrar cualquier aplicación en una instalación, cumpliendo con el objetivo de introducir una nueva realidad al espectador.

6. Líneas de futuro

Si por alguna cosa se caracteriza este proyecto, es por ser un proyecto abierto. Los usuarios forman parte del ZOO XXI, por lo que hacer que crezca está en las manos de todos. Estas propuestas son tan solo una muestra de las múltiples posibilidades de desarrollo.

Este proyecto es fácilmente integrable, y cómo ZOO XXI es un proyecto escalable, la primera propuesta es la integración con otra instalación existente [49], realizada por nuestros compañeros del ETC (Carnegie Mellon, Pittsburgh, PA, EUA). Una forma más de enfatizar la importancia del concepto, el zoológico en la ciudad. Cómo miran los animales? Era lo que ellos se preguntaron antes de desarrollar su proyecto. Éste consiste en una instalación interactiva que permite al usuario, mediante unos sensores lumínicos, sentirse observados por animales. Si en nuestro proyecto nosotros adoptamos el rol de ser un animal, ¿Cómo nos miran aquellos que están a nuestro alrededor? La integración de ambos conceptos podría unirse en un único espacio. Un zeppelín itinerante dónde el público se ha de sentir transportado al mundo de los animales, en un recorrido a lo largo de toda la diversidad de la fauna y el mundo vegetal.

Figura 87. Zeppelín Itinerante

Otra línea de futuro es la posibilidad de introducir la realidad aumentada en cualquier soporte tecnológico con la finalidad de poder visualizar animales en distintos puntos de la ciudad.

También podría pensarse en una aplicación iPhone interactiva que podría utilizarse en la visita a las instalaciones físicas del Parque Zoológico, integrando la información que habitualmente se facilita mediante grabaciones de voz, con la interacción de realidad virtual con juegos diseñados específicamente en relación al itinerario de la visita, con un objetivo a la vez lúdico y formativo.

Del mismo modo, este proyecto abre las puertas a muchísimos sectores como el turismo, la educación y la investigación, adaptándose a los objetivos específicos que puedan ser necesarios para cada sector.

Bibliografía

[1] ZOOLOGICAL SOCIETY OF LONDON (ZSL). [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.zsl.org

[2] PARQUE ZOOLÓGICO DE BARCELONA. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.zoobarcelona.cat

[3] PARQUE DE LA NATURALEZA DE CABÁRCENO. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.parquedecabarceno.com/

[4] FAUNIA. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.faunia.es/

[5] COSMOCAIXA. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://obrasocial.lacaixa.es/

[6] BIOPARC VALENCIA. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.bioparcvalencia.es/

[7] EUROPEAN ASSOCIATION OF ZOOS AND AQUARIA (EAZA). [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.eaza.net

[8] ASSOCIATION OF ZOOS AND AQUARIUMS (AZA).). [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.aza.org/

[9] DBK ANÁLISIS DE SECTORES. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.dbk.es

[10] APPLE DEVELOPER. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://developer.apple.com

[11] COCOA FUNDAMENTALS GUIDE. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://developer.apple.com/iphone/library/documentation/Cocoa/Conceptual/CocoaFundamentals/

[12] OBJECTIVE-C. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://developer.apple.com/iphone/library/documentation/General/Conceptual/DevPedia-CocoaCore/ObjectiveC.html

[13] MEMORY MANAGEMENT. [Consultado: Febrero – Mayo 2010]. Disponible en internet:

http://developer.apple.com/iphone/library/documentation/General/Conceptual/DevPedia-CocoaCore/MemoryManagement.html

[14] DELEGATION. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://developer.apple.com/iphone/library/documentation/General/Conceptual/DevPedia-CocoaCore/Delegation.html

[15] NOTIFICATION. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://developer.apple.com/iphone/library/documentation/General/Conceptual/DevPedia-CocoaCore/Notification.html

[16] MODEL VIEW CONTROLLER. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://developer.apple.com/iphone/library/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html

[17] KEY-VALUE CODING PROGRAMMING GUIDE. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://developer.apple.com/iphone/library/documentation/Cocoa/Conceptual/KeyValueCoding/

[18] KEY-VALUE OBSERVING PROGRAMMING GUIDE. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://developer.apple.com/iphone/library/documentation/Cocoa/Conceptual/KeyValueObserving/

[19] CORE DATA PROGRAMMING GUIDE. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://developer.apple.com/iphone/library/documentation/Cocoa/Conceptual/CoreData/

[20] CREATING A MANAGED OBJECT MODEL WITH XCODE. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://developer.apple.com/iphone/library/documentation/Cocoa/Conceptual/CreatingMOMWithXcode/

[21] THE CORE PROJECT. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.codeproject.com

[22] NETBEANS. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://netbeans.org

[23] HTC. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.htc.com

[24] APPLE, INC. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.apple.com/iphone/

[25] TAMAGOTCHI. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.tamagotchi.com/

[26] ZOO TYCOON. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://zootycoon.com

[27] NINTENDOGS. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://nintendogs.com/

[28] EYEPET. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.eyepet.com

[29] FARMVILLE. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.farmville.com/

[30] FACEBOOK DEVELOPER. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://developers.facebook.com/

[31] GOOGLE EARTH. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://earth.google.es/

[32] MICROSOFT SPHERE. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://research.microsoft.com/en-us/um/people/benko/projects/sphere/

[33] GLOBAL IMAGINATION. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.globalimagination.com/

[34] IGOR POLYAKOV, ART DIRECTOR. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://polyakov.org/

[35] INTERACTIVE MULTIMEDIA TECHNOLOGY. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://interactivemultimediatechnology.blogspot.com/

[36] DISPLAX INTERACTIVE SYSTEMS. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.displax.com

[37] WHITEVOID. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.whitevoid.com/

[38] OPEN CLIP ART LIBRARY. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.openclipart.org/

[39] NATURAL SCIENCE MUSEUM OF BARCELONA DATA BASE. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://mapa.bioexplora.cat/

[40] WORLD ALLIANCE FOR INNOVATION. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.wainova.org/

[41] NOAH, NETWORKED ORGANISMS AND HABITATS. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.networkedorganisms.com/

[42] BALSAMIQ. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.balsamiq.com

[43] BOARD GAME GEEK. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.boardgamegeek.com/

[44] INVIZIMALS. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.invizimals.com/

[45] SPORE. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.spore.com/

[46] ZINGA. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.zynga.com/

[47] ANTWERPEN ZOO. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.zooantwerpen.be/

[48] PARQUE ZOOLÓGICO DE SANTA FE. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.zoologicosantafe.com/

[49] ZOO XXI, ETC PROJECT. [Consultado: Febrero – Mayo 2010]. Disponible en internet: http://www.etc.cmu.edu/projects/emotique/

Índice de figuras