

# ZOOPLANET

Fases en el desarrollo del software

Javier Vañó, Rodolfo Biasini, Maria Adrover

ENTORNS DE DESENVOLUPAMENT

Curso 2017/18

# Contenido

Introducción .....	2
DESARROLLO.....	3
a) Contextualización .....	3
#Nuestra empresa.....	3
#Descripción del proyecto .....	3
#Características del cliente .....	3
b) Modelo de ciclo de vida .....	4
c) Documento ERS (Análisis de Requisitos) .....	5
d) Diseño .....	5
#Introducción.....	5
#Programas .....	5
#Interfaz del software .....	6
e) Codificación .....	8
# Lenguaje de programación.....	8
# Codificación .....	9
# Herramientas.....	9
f) Pruebas.....	10
g) Documentación.....	11
h) Explotación .....	13
i) Mantenimiento .....	13
Conclusiones.....	14
Herramientas para el desarrollo.....	15
Bibliografía y Documentación .....	16

## Introducción

En este documento se incluye toda la información relativa al proyecto de desarrollo de la aplicación ZOOPLANET. En él, se realizará una descripción detallada de las diferentes etapas del ciclo de vida del software, y de los aspectos que deberían tenerse en cuenta para garantizar el éxito del proyecto.

El desarrollo de la aplicación ha sido asignado a nuestra empresa tras haber ganado un concurso público convocado por el Ministerio de Educación y Medioambiente. El proyecto contempla el desarrollo de un software educativo y de divulgación dirigido a fomentar la concienciación y respeto medioambientales, mejorar el conocimiento del reino animal, y promocionar las visitas a los zoos y acuarios.

En función de la acogida que tenga la aplicación, se pactará un nuevo contrato para la elaboración de una segunda versión del software con nuevas funcionalidades. Al estar la aprobación de esta segunda fase sujeta al éxito de la aplicación, en este documento no se tratarán los detalles de su desarrollo sino tan sólo los de la primera versión.

## DESARROLLO

### a) Contextualización

#### #Nuestra empresa

Somos una pequeña empresa con 12 años de experiencia en el sector informático, y desde hace 7 nuestra actividad profesional se ha centrado principalmente en el desarrollo de aplicaciones multiplataforma. Nuestro equipo está formado por 8 trabajadores especializados en diferentes campos en el mundo de la informática y el diseño.

#### #Descripción del proyecto

El proyecto a desarrollar consiste en la implementación de la aplicación ZOOPLANET, de descarga gratuita. El software permitirá la observación en directo de diferentes animales mediante la conexión del dispositivo con las cámaras instaladas en zoológicos y acuarios. Así mismo, se incluirá en la aplicación información detallada sobre todos los animales visualizados (enciclopedia), y sobre los zoos y acuarios participantes en el proyecto.

La aplicación se desarrollará para Windows, Apple y Linux en su versión de escritorio, y para Android, iOS y Windows Phone en su versión móvil.

Será necesario que el usuario cuente con una conexión a internet para poder aprovechar todas las funcionalidades de la aplicación, si bien será posible consultar la parte de la enciclopedia si se está trabajando offline.

Los usuarios que descarguen la aplicación en el móvil obtendrán un descuento para visitar los zoos y acuarios que participen en el proyecto, simplemente enseñando el móvil con la aplicación descargada al acceder al recinto.

La aplicación está dirigida al público en general. Se desarrollará una interfaz amigable y atractiva, que ofrezca una navegación sencilla, asequible para un rango de edad lo más amplio posible.

#### #Características del cliente

El desarrollo del proyecto está promocionado, gestionado y financiado por el Ministerio de Educación y Medioambiente. Se ha conseguido pactar los plazos que se consideran necesarios para un desarrollo exitoso. El cliente, para esta 1ª

fase del proyecto, nos ha marcado de forma clara y detallada los objetivos y funcionalidades del software, si bien ciertos detalles referentes a servidores son susceptibles de evolución.

En el proyecto están implicados también como colaboradores un gran número de parques zoológicos y acuarios, tanto públicos como privados, que a cambio de su implicación gratuita en el proyecto, aportarán su infraestructura en cuanto a instalación de cámaras, y colaborarán en la redacción de los contenidos sobre los animales que se incluirán en la aplicación. Por consiguiente, será necesario el contacto con estos colaboradores, si bien la gestión de designación de los mismos correrá por cuenta del cliente.

## b) Modelo de ciclo de vida

Tras haber analizado las características del proyecto hemos optado por el Modelo en Cascada con Realimentación para su desarrollo. Para tomar esta decisión hemos tenido en cuenta las siguientes cuestiones:

- El proyecto tiene cierta envergadura y damos por hecho que tendremos que volver a etapas anteriores en ciertos momentos del desarrollo para poder corregir errores, y modificar o depurar algún aspecto, por lo que hemos descartado la utilización del modelo en cascada clásico.
- A pesar de esta necesidad de contar con una realimentación entre las diferentes etapas, los requisitos del software están claros y bien especificados desde el principio; es un proyecto rígido con unas funcionalidades y objetivos bien definidos; no se prevé una evolución en el software durante su desarrollo.
- Los presupuestos y plazos pactados, en principio se prevén suficientes para el desarrollo del proyecto.

### #Introducción

El software a desarrollar tendrá dos versiones: un programa para ordenadores, tanto para Windows, Apple y Linux, y una aplicación para móviles, tanto para Android, IOS y Windows Phone. El contenido de ambas versiones será la misma, con excepción de la información acerca de los descuentos, solo disponible en la aplicación para móviles.

### #Programas

El software a desarrollar estará constituido por varios programas, cada uno con diferente función. Es posible que, durante el desarrollo, se tome la decisión de dividir alguno de los presentes programas en varios de menor tamaño y complejidad. El diseño inicial contempla la programación de cinco programas principales:

#### : *Gestión de cámaras*

Uno de los programas a desarrollar, será el software que gestionará la conexión con las cámaras a través de internet. Este programa será especialmente complejo, puesto que puede que cada zoo gestione sus cámaras de vigilancia de distinta manera y el trabajo de los programadores consistirá en integrar los sistemas de todas las cámaras, con la interfaz gráfica de nuestra aplicación. De esta forma, este programa permitirá sincronizar la señal de vídeo de las cámaras de los recintos de animales, en los diferentes Zoos, y será posible hacer streaming de vídeo directamente a la aplicación. Este programa, como es lógico, solo funcionará si el usuario está conectado a internet y si los servidores de los diferentes Zoos son accesibles.

#### : *Interfaz*

La interfaz del software, explicada más en profundidad en apartados posteriores, consistirá en la programación del sistema de navegación de la aplicación. Desde la selección funcional de elementos, la redirección a otras páginas dentro de la misma aplicación, hasta la función de pantalla completa de la reproducción de cámaras en tiempo real. En definitiva, este programa cubrirá los aspectos formales e interactivos de la aplicación. El desarrollo de esta parte del software implicará diseño gráfico y ergonomía para que el usuario tenga la mejor experiencia posible.

### *: Buscador*

En diversas partes de la interfaz de la aplicación, existirá un cuadro de búsqueda en el que el usuario podrá introducir texto para buscar cámaras, Zoos o animales. Este buscador identificará elementos presentes en las alternativas posibles y seleccionará los resultados coincidentes (tanto total- como parcialmente) y los mostrará. La programación de este motor de búsqueda, estará integrado con la interfaz de usuario, pero se programará por separado, haciendo uso de la propiedad modular de los lenguajes de programación orientados a objetos. Esto permitirá modificar la interfaz al margen del sistema de búsqueda, que podrá amoldarse al diseño de ésta.

### *: Sistema de Filtrado*

Además del motor de búsqueda, la aplicación a desarrollar también contará con un sistema de filtrado. Éste ofrecerá unos filtros u otros, en función de la sección en la que se encuentre el usuario. Por ejemplo, si el usuario se encuentra en el apartado de Zoos y acuarios, se podrá filtrar por tipo de parque, comunidad autónoma, extensión...etc. Si, por el contrario, el usuario se encuentra en la enciclopedia, se podrá filtrar por continente, clima, familia del animal, dieta...etc.

El sistema de filtrado mostrará unos resultados u otros en función de los filtros seleccionados, por lo que será un software de ordenación y selección, desarrollado por separado por la misma razón que el sistema de búsqueda: para poder adaptarse mejor a la interfaz de usuario, aunque ésta sufra cambios.

### *: Base de Datos*

Finalmente, la información que se mostrará en la aplicación, tanto sobre las cámaras, los diferentes Zoos y acuarios y las fichas de los animales (enciclopedia), se guardará en una base de datos que se actualizará cuando se añadan nuevas cámaras o cualquiera de los elementos antes mencionados. Por esa razón, será necesaria la creación/implementación de una base de datos en la que almacenar la información que luego se mostrará en el software. Esto es un nuevo ejemplo de cómo se separará el contenido de la aplicación, de su presentación (interfaz). De esta manera, será posible añadir nuevos datos sin alterar el sistema de navegación, pues este podrá acomodar los nuevos añadidos.

## *#Interfaz del software*

Al iniciar la aplicación, aparecerán tres opciones de navegación entre las que el usuario puede elegir, en función de sus intereses: cámaras, zoos y enciclopedia. Adicionalmente, en la esquina superior izquierda existirá un botón que llevará al usuario a una pantalla de ayuda, con información sobre cómo funciona la aplicación. Además, exclusivamente en la aplicación para móvil, habrá un botón en la esquina superior derecha que redirigirá a una página con información sobre descuentos y ofertas en los diferentes zoos y acuarios.

Las tres opciones de navegación (cámaras, zoos y enciclopedia) estarán representadas por 3 botones grandes que ocuparán gran parte de la pantalla. El usuario podrá seleccionar uno de ellos y será redirigido a una pantalla de búsqueda y filtrado. Dicha pantalla contará con un cuadro de búsqueda en la parte superior y diversos filtros en la parte izquierda. En el resto de la pantalla aparecerán todos los resultados compatibles con la búsqueda y filtro utilizados (al iniciar la aplicación se mostrarán todos los ítems). Los elementos mostrados, así como los filtros disponibles, variarán en función de la opción de navegación elegida por el usuario: si se ha hecho click en la opción de cámaras, el usuario verá todas las cámaras disponibles y podrá filtrar por nombre de animal, tipo, zoo al que pertenece, tipo de hábitat, continente, clima...etc; si el usuario ha seleccionado la navegación por zoos, éste verá los zoos disponibles y podrá filtrar por tipo de parque (zoo o acuario), comunidad autónoma, animales que posean...etc; finalmente, si el usuario selecciona la visualización por animales (enciclopedia), la pantalla de búsqueda mostrará los animales que se encuentren en, al menos, un zoo y podremos filtrar por nombre de animal, tipo, zoo, tipo de hábitat, continente, clima...etc.

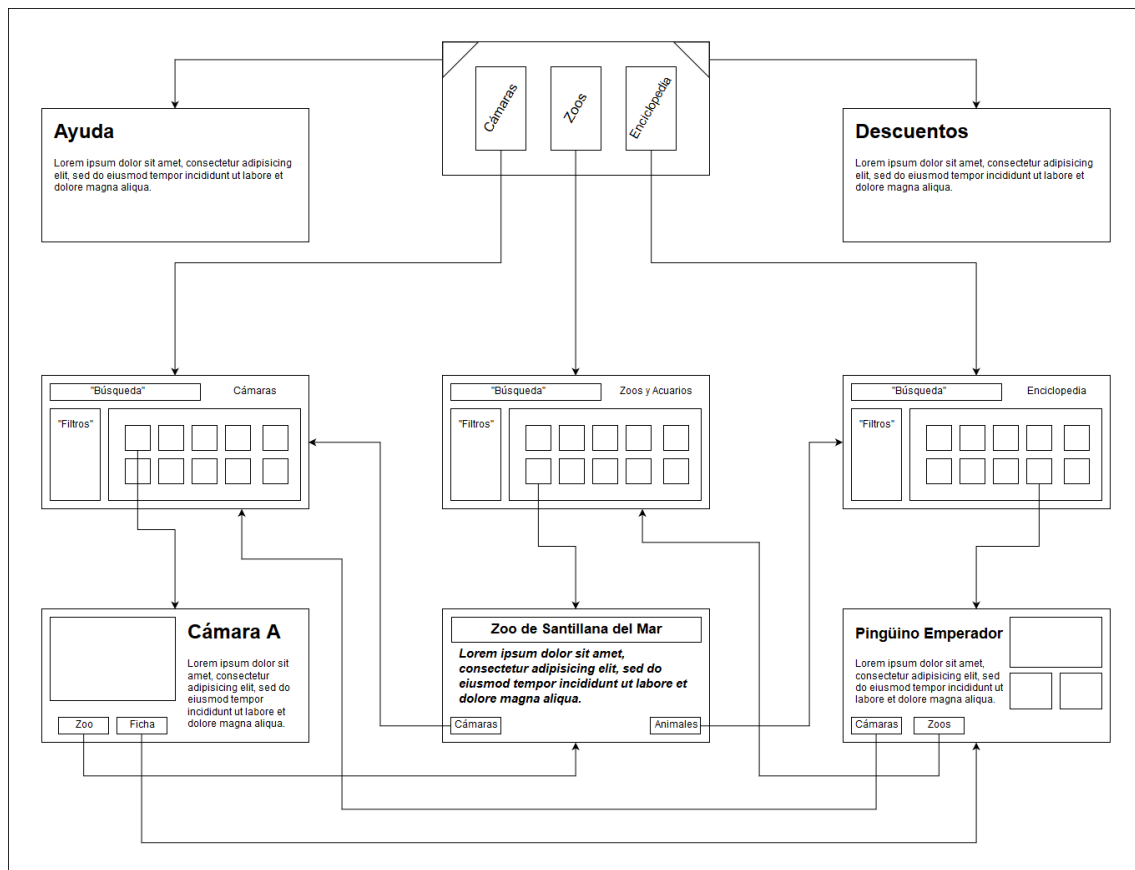
En el apartado de cámaras, al seleccionar una cámara concreta, el usuario accederá a una pantalla en la que podrá ver en tiempo real la cámara seleccionada, además de información sobre el zoo al que pertenece y una breve descripción del animal que visualizará. En la misma pantalla habrá un enlace a la página del zoo al que pertenece y otro enlace a la ficha del animal en la enciclopedia. De esta forma, las tres formas de navegar por la aplicación estarán enlazadas de forma dinámica.

En el apartado zoos, al seleccionar un zoo o acuario concreto, el usuario será redireccionado a la página de dicho parque y ahí encontrará toda la información relacionada con él, como su horario, precios y localización. Además, en la pantalla del zoo seleccionado, podemos encontrar dos enlaces: uno redireccionará al usuario a la pantalla de búsqueda del apartado de cámaras, mostrando solo las cámaras del zoo seleccionado; y el segundo enlazará la pantalla de búsqueda de la enciclopedia, mostrando solo las fichas de los animales que se encuentran en el zoo seleccionado.

En el apartado de la enciclopedia, el usuario puede seleccionar el animal que prefiera y, al seleccionarlo, verá la ficha técnica de dicho animal, con diversas imágenes y abundante información sobre dicho animal. Además, en dicha página habrá dos enlaces: uno llevará a la pantalla de búsqueda del apartado de las cámaras, mostrando solo las cámaras correspondientes a dicho animal (en diferentes zoos); y el otro enlace llevará a la pantalla de búsqueda del apartado de zoos, mostrando solo los Zoos en los que se encuentre el animal seleccionado. De esta forma, tal y cómo se ha dicho anteriormente, los tres modos de visualización quedan interconectados y permitirían una navegación fluida y continua.

A continuación puede verse un gráfico que describe el funcionamiento de la interfaz del software explicada en los párrafos anteriores.





## e) Codificación

### # Lenguaje de programación

Hemos elegido Java como lenguaje de programación para desarrollar todas las versiones de la aplicación. Los motivos que nos han llevado a tomar esta decisión son los siguientes:

- Java es un lenguaje de programación multiplataforma que nos permitirá desarrollar la aplicación para todos los sistemas operativos y plataformas que nos ha solicitado el cliente.
- Dadas las características de la aplicación nos permitirá implementar y optimizar el software de forma adecuada para todas ellas.
- Al tener que desarrollar la aplicación para tantos sistemas operativos, utilizando este lenguaje podremos ahorrar mucho tiempo y gasto en el proceso de desarrollo del proyecto.

- Damos por supuesto que la necesidad de contar con el intérprete de Java en el dispositivo no supone un problema ni limitación dada la popularidad de este lenguaje. Es muy probable que el usuario final ya tenga instalada la máquina virtual de Java en su dispositivo ya que hay muchísimas aplicaciones que necesitan este software para su ejecución.
- Las características modulares de Java como lenguaje de programación orientado a objetos se adaptan a las necesidades asociadas a las características de la aplicación. Los distintos programas de nuestra aplicación estarán formados por objetos independientes entre si que nos facilitarán la localización y depuración de errores y la reutilización eficaz de código.

## # Codificación

Como paso previo a la codificación propiamente dicha, trabajaremos en el desarrollo en pseudocódigo de algoritmos que nos permitan resolver los diferentes problemas que nos plantee el desarrollo nuestra aplicación. Esto nos permitirá ganar tiempo y efectividad a la hora de escribir el código fuente de nuestros programas.

Partiremos de las etapas anteriores de análisis y diseño para la elaboración de nuestro algoritmo y una vez elaborado se procederá a su codificación en Java.

## # Herramientas

Las herramientas con las que trabaja generalmente nuestro equipo y que nos serán útiles en esta fase de desarrollo del proyecto, teniendo en cuenta sus características concretas son:

- JDK (Java Development Kit) que incluye el JRD (Java Runtime Environment) con su JVM (Java Virtual Machine). Android requiere de otro software específico, SDK con AVD (maquina virtuales que nos permitirá ver la aplicación en funcionamiento).
- IDE (Integrated Development Environment) NetBeans: entorno de desarrollo que nos permitirá escribir, compilar, depurar y ejecutar los programas de nuestra aplicación. Permite el desarrollo rápido de interfaz de usuario, la gestión organizada de proyectos e integra Git, Subversion y Mercurial como control de versiones (nosotros trabajaremos con Git).
- Git y GitHub: control de versiones, backup de todo el proyecto completo con el código final y también de todo el historial de modificaciones que el código ha sufrido desde el primer día, auditoría del código, control de seguridad, etc.

## f) Pruebas

En el presente proyecto de desarrollo se llevaría a cabo dos tipos de pruebas: unitarias y de integración. Las primeras suponen poner a prueba las distintas partes del programa por separado, mientras que la segunda propone una prueba completa del software, con el objetivo de comprobar el funcionamiento general de la aplicación, así como de observar si todas las partes del software se coordinan correctamente.

Las pruebas unitarias cubrirían, en este caso, las comprobaciones del correcto funcionamiento de cada una de las partes de nuestro software: gestión de cámaras, interfaz, buscador, sistema de filtrado y base de datos.

La gestión de las cámaras, el buscador y el sistema de filtrado serían objeto de pruebas similares en las que, mediante una interfaz sencilla (incluso el propio entorno de desarrollo), se comprobaría que cada uno de estos programas funciona correctamente. Se comprobaría que se visualizan correctamente a tiempo real todas las cámaras que formarán parte de la aplicación final, también se probaría que el cuadro de búsqueda registra y procesa correctamente el texto introducido y, por último, se constataría que el sistema de filtrado es capaz de seleccionar correctamente los resultados a mostrar y que es capaz de lidiar con múltiples filtros en el orden correcto y lógico.

La interfaz de la aplicación, por otra parte, sería objeto de pruebas más extensas, comprobando que se visualiza correctamente en todo tipo de dispositivos, desde pantallas de ordenador, tablets o móviles. Especialmente en éstos últimos, también se comprobará que la visualización sea correcta, tanto en vertical como en horizontal.

Finalmente, se comprobaría que la base de datos responde correctamente a las consultas que se le hagan y que registra correctamente la información introducida. También se revisará que las tablas de la base de datos cuenten con las medidas de seguridad necesarias (especialmente en lo referente a copias de seguridad). Además, se comprobará que las tablas están correctamente enlazadas y conectadas entre sí.

Cabe destacar que, como punto intermedio entre las pruebas unitarias y las de integración, podrían realizarse algunas pruebas intermedias, probando, por ejemplo, la integración entre el buscador o el sistema de filtrado y la interfaz de usuario. Esto permitiría reducir la cantidad de pruebas integradas, permitiendo comprobar que dos programas ya funcionan correctamente de forma simultánea.

Las pruebas de integración comprobarían, como se ha dicho anteriormente, el correcto funcionamiento del software en su totalidad. Estas pruebas se

realizarían una vez hechas todas las pruebas unitarias y una vez resueltos todos los problemas que éstas hayan podido revelar. En el caso del presente proyecto, debido a que la aplicación está pensada para dispositivos móviles, además de para ordenadores, será necesario comprobar el correcto funcionamiento de la aplicación en dichos sistemas operativos y en el mayor número de configuraciones de hardware posible.

Un primer paso en las pruebas de integración sería probar la aplicación de escritorio en los tres sistemas operativos: Windows, Mac y Linux. Una vez comprobado el correcto funcionamiento, pasaríamos a utilizar máquinas virtuales de los sistemas operativos de móviles: Android, IOS y Windows Phone. Después de corregir los posibles errores, pasaríamos a probar la aplicación en terminales móviles, incluidas tablets y repetiríamos las pruebas en el mayor número de configuraciones posible (sobre todo en dispositivos Android, dada su mayor variabilidad). Esto sería lo más recomendable puesto que una aplicación puede funcionar correctamente en un determinado Smartphone y dar errores en un terminal totalmente distinto, a nivel de componentes, a pesar de que utilicen la misma versión de Android.

En definitiva, las pruebas de integración nos permitirían observar la interacción entre las distintas partes del software y detectar errores o fallos en ella. Debido a que, la aplicación del presente proyecto, no cuenta con la posibilidad de crear cuentas de usuario o de editar el contenido de la misma, las pruebas no serían excesivamente extensas, ya que el contenido sería el mismo para todos los usuarios. Esto hace que las pruebas determinen más fácilmente todos los posibles escenarios y casos, haciendo que el tiempo de prueba sea menor, optimizando el proceso de desarrollo.

## g) Documentación

La documentación es una fase muy importante, y muchas veces descuidada, en el desarrollo de software. Sirve para registrar todo el proceso de la creación del programa o programa, así como para ayudar al usuario final en la utilización del software. Esto es así porque la documentación se divide en tres partes, cada una de ellas escrita con un nivel de tecnicismo distinto y dirigidas a un público distinto. Más que hablar de tres fases distintas de la documentación, deberíamos hablar de tres formas de documentar el software, puesto que cada una de ellas cubre la totalidad del proceso, de principio a fin.

La guía técnica es una de las tres partes de la documentación y forma la explicación más técnica del desarrollo de software. Es una guía orientada al personal informático y debe explicar de la forma más detallada posible, toda la información referente a la elaboración del programa. Deben constar obligatoriamente el diseño de la aplicación, la codificación de los diversos programas que la forman, así como las pruebas realizadas. De esta forma queda constancia de todo lo que se ha hecho, lo que facilita ampliaciones futuras en el software, y proporciona una corrección de errores mucho más eficiente, ya que el personal informático cuenta con información detallada de la estructura interna del software. En este proyecto concreto, la guía técnica explicaría y justificaría el diseño de la aplicación (apartados C y D de este trabajo), documentaría la codificación de los distintos programas (apartado E) y detallaría las pruebas realizadas (apartado F).

La guía de uso es otra de las tres partes de la documentación y está dirigida a los usuarios del software. Ésta parte de la documentación debe incluir información acerca del funcionamiento de la aplicación, debe explicar las diferentes opciones y funciones que ofrece el programa, además de contar con posibles soluciones a los principales problemas que puede sufrir la aplicación. En el caso de este proyecto, la guía de uso estará incluida en la propia aplicación, en la ventana de "Ayuda", que aparecerá en la esquina superior izquierda de la pantalla principal de la aplicación. De esta forma, el usuario podrá seleccionar el botón triangular de dicha esquina, lo que desplegará toda la información acerca del funcionamiento del programa, así como los errores más habituales que pueden ocurrir y cómo solucionarlos. Por ejemplo, en el caso de que las cámaras no se vean correctamente, la pantalla de ayuda le recordará al usuario que debe estar conectado a internet...etc.

La guía de instalación es la última de las tres partes de la documentación y recoge toda la información necesaria para la instalación y la puesta en marcha del software. Esta documentación es importante para que se realice una buena instalación de la aplicación, lo que permitirá una correcta, segura y confiable implantación del software. En el presente caso, esta guía solo estará disponible en la versión de escritorio, introducida en el ejecutable, que guiará al usuario durante la instalación del software, requiriendo la información necesaria de éste, como la ruta de instalación o las opciones en el menú de inicio. Esta guía no figurará en la versión para móviles y tablets, puesto que la instalación correrá a cargo de la tienda de aplicaciones correspondiente (Play Store, App Store o Windows Store), que realiza todo el proceso sin requerir ninguna acción por parte del usuario, una vez que esté ha elegido instalar la aplicación desde la tienda.

## h) Explotación

## i) Mantenimiento

En el contrato firmado con el cliente, además del desarrollo de la aplicación, se ha pactado el mantenimiento de la misma por un período de dos años, prorrogable a partir de la fecha de finalización de dicho período.

El ciclo de vida del desarrollo del software no termina cuando la aplicación está terminada, instalada y en funcionamiento. Falta la etapa final, que puede llegar a ser la más larga, que es la de mantenimiento. Una vez finalizadas las fases anteriores y con el software funcionando, hay que controlar, mejorar y optimizar el software, y realizar las modificaciones que se consideren necesarias para hacer frente a las diferentes situaciones que vayan surgiendo

En este proyecto, los tipos de cambios que se prevén son los siguientes:

- Correctivos: se revisará el comportamiento del software y se corregirán los fallos que surjan.
- Perfectivos: es posible que haya que implementar la conexión de la aplicación con nuevas cámaras, ya sea porque los zoos y acuarios participantes en el proyecto añadan nuevos dispositivos, o bien porque surjan zoos y acuarios que se incorporen al proyecto. Por otro lado, se irán ampliando las bases de datos de la aplicación añadiendo información e imágenes sobre nuevos animales, o modificando los datos ya existentes; esto supondrá la gestión de las modificaciones que estos cambios puedan provocar en los filtros y criterios de búsqueda.
- Evolutivos: Como ya hemos comentado, está previsto que se desarrolle una segunda versión de la aplicación con nuevas funcionalidades que cubran las necesidades que observe el cliente después de estudiar la acogida de la primera versión. pero al estar sujeto el desarrollo de esta segunda versión a la acogida que tenga la primera versión de la aplicación, y las nuevas necesidades que el cliente observe, en caso de que se aprobara el desarrollo de la nueva versión, constituiría un nuevo proyecto y estaría sujeta a un nuevo, por lo que no se ha incluido en el servicio de mantenimiento del proyecto que nos ocupa.
- Adaptativos: si se introdujeran cambios en las tecnologías de las cámaras y equipos de los zoos y acuarios, se realizarán los cambios necesarios para garantizar el correcto funcionamiento del sistema.

## Conclusiones

Tras haber reflexionado y desarrollado todas las etapas y demás contenidos descritos en el presente documento sobre el ficticio proyecto ZOOLANDER, nos damos cuenta de la complejidad y dificultad que conlleva el proceso de desarrollo de un software cualquiera.

La cantidad de conceptos y conocimientos técnicos implicados en el desarrollo del software es enorme y además evoluciona continuamente: el hardware, las técnicas de gestión y análisis, herramientas de desarrollo, de diseño, de depuración, lenguajes de programación, técnicas de prueba... Es evidente que se hace necesaria una buena formación y actualización de los conocimientos para ser un profesional competente.

Sin embargo, la competencia técnica no es la única cuestión clave en el resultado de un proyecto; más allá de la formación y capacidad particulares de los miembros de un equipo de desarrollo, hay un elemento determinante en el éxito o fracaso de cualquier proceso de desarrollo, y es la planificación y gestión adecuada, rigurosa y correcta de todas y cada una de las etapas del ciclo de vida del software. En el proceso de desarrollo, todos los elementos y fases que lo componen están, directa o indirectamente, relacionados entre sí; todos son importantes, y todos, hasta el más mínimo detalle, tienen de una forma u otra su repercusión en fases posteriores. Obviamente todos los errores, carencias, faltas de previsión... que se puedan dar se pagarán exponencialmente en el resto de etapas y por consiguiente, en el resultado final. De la misma forma, los aciertos y buenos hábitos tendrán también su recompensa, y facilitarán y agilizarán el resto de etapas. Salvando las distancias, podríamos establecer cierto paralelismo entre las dificultades que hemos tenido que afrontar elaborando el presente trabajo y las que resultarían del desarrollo de una aplicación: problemas de organización, problemas de documentación, problemas técnicos, falta de información y de formación, problemas de tiempo... y todos estos problemas han tenido su repercusión en un momento u otro del proceso. Por otra parte, hay que contar con que, por muy escrupulosos que seamos en la ejecución de nuestro trabajo, nunca podremos preverlo todo; hemos de contar con que siempre sucederán imprevistos y sucesos que escapen a nuestro control. Éste es por tanto un motivo más para resaltar la importancia que tiene aprovechar las cosas que sí dependen de nosotros, como el desarrollo correcto y meticuloso cada etapa, dejando así el mínimo de cosas al azar.

Como conclusión final, diremos que, aunque no debe ser nada fácil poner en práctica la teoría en un entorno laboral real (plazos excesivamente cortos, carga de trabajo excesivamente larga, mala planificación y gestión, falta de medios, etc.), siempre hay que intentar aplicar en la medida de lo posible las buenas prácticas descritas en este documento.

## Herramientas para el desarrollo

Hay muchas aplicaciones que pueden ayudarnos en el desarrollo del software, son las herramientas CASE. Éstas nos permiten ahorrar tiempo y ganar fiabilidad en las etapas de desarrollo de un proyecto, con el consecuente ahorro económico que esto supone.

Las principales herramientas que, teniendo en cuenta sus características concretas, nos ayudarán en este proyecto son:

- Control de versiones Git junto con GitHub como plataforma para gestionar repositorios y organizar el trabajo del equipo.
- Framework Apache Maven: herramienta para la gestión y el diseño de proyectos en Java.
- Microsoft Project, para gestionar el desarrollo de planes, asignación de recursos a tareas, seguimiento del progreso, administrar presupuesto y analizar las cargas de trabajo.
- Visual Paradigm, herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software.
- JDK (Java Development Kit) que incluye el JRE (Java Runtime Environment) con su JVM (Java Virtual Machine). Android requiere de otro software específico, SDK con AVD (maquina virtuales que nos permitirá ver la aplicación en funcionamiento).
- IDE (Integrated Development Environment) NetBeans: entorno de desarrollo - una herramienta para puedan escribir, compilar, depurar y ejecutar programas. Permite el desarrollo rápido de interfaz de usuario, la gestión organizada de proyectos e integra - Git, Subversion y Mercurial como control de versiones.



## Bibliografía y Documentación

Información sobre lenguajes de programación

<https://www.muypymes.com/2016/03/28/lenguajes-programacion-desarrollo-apps>

<https://es.quora.com/Que-lenguaje-de-programación-debería-saber-para-el-2017>

Información sobre el desarrollo de aplicaciones en Java

<https://www.taringa.net/posts/linux/19187248/Herramientas-utiles-o-necesarias-para-un-desarrollador-Java.html>

<https://desarrolloweb.com/articulos/iniciacion-java-caracteristicas-programacion-ideeclipse.html>

<https://www.genbetadev.com/java-j2ee/herramientas-imprescindibles-para-un-desarrollador-de-java>

Información sobre desarrollo de aplicaciones para móvil

<http://perso.wanadoo.es/emiliotoboso/android/preparacion.htm>

Información sobre herramientas CASE

<http://wpmallorca.com/2013/02/12/pero-que-es-github/>

<http://www.monografias.com/trabajos73/herramientas-case-proceso-desarrollo-software/herramientas-case-proceso-desarrollo-software2.shtml>