



Facultad de Informática
Departamento de asuntos humorísticos

PROYECTO FIN DE CARRERA
Ingeniería en Informática

Cómo el pollo cruzó la carretera

Alumno: Dolores Fuertes de Barriga
Director: Armando Bronca Segura
Tutor: Alex Plosivo Seguro
Fecha: 14 de septiembre de 2014

D. ARMANDO BRONCA SEGURA
Profesor, Facultad de Informática
Departamento de asuntos humorísticos
Universidad de A Coruña

CERTIFICA:

Que la memoria titulada *“Cómo el pollo cruzó la carretera”* ha sido realizada por DOLORES FUERTES DE BARRIGA de acuerdo a la descripción inicialmente propuesta, bajo mi dirección y constituye su Proyecto Fin de Carrera de Ingeniería en Informática. Por la presente, autorizo su presentación para que el Proyecto sea defendido en esta convocatoria.

En A Coruña, a 14 de septiembre de 2014

D. ARMANDO BRONCA SEGURA
Director del proyecto

Título

Cómo el pollo cruzó la carretera

Título longo do proxecto

Long title of the proyect

Clase: Proyecto clásico de ingeniería

Autor: Dolores Fuertes de Barriga

Tutor: Alex Plosivo Seguro

Director: Armando Bronca Segura

Fecha: 14 de septiembre de 2014

Tribunal

**Fecha de
defensa:**

Calificación:

Resumen

A lo largo de los últimos años ...

Lista de Palabras Clave

lista, palabras, clave

A todo el mundo.

Agradecimientos

A los profesores Armando Bronca Segura y Alex Plosivo Seguro por sus consejos durante el desarrollo del proyecto.

A María de la O por ...

Índice general

1. Introducción	1
1.1. Dominio de Aplicación	2
2. Conceptos Previos	3
2.1. Coordenadas Geográficas	4
3. Plan de Proyecto	5
3.1. Análisis de Viabilidad	6
4. Tecnología	7
4.1. Base Tecnológica	8
5. Diseño	9
5.1. Ciclo de vida	10
6. Implementación	11
6.1. Organización del código fuente	12
7. Conclusiones	13
7.1. Análisis Económico	14
A. Apéndice	15
A.1. PhotoPlace	15
Glosario	17
Acrónimos	19

Índice de figuras

Capítulo 1

Introducción

1.1. Dominio de Aplicación	2
--------------------------------------	---

El proceso de desarrollo del software comienza con un análisis del entorno y del dominio de la aplicación. En un contexto amplio, se entiende por dominio como “una esfera de actividad o interés”, también denominado campo. Es decir, el área de aplicación donde el software fue creado.

En el presente capítulo se describe el entorno y el dominio de la aplicación, y la evolución de los servicios de geolocalización en los últimos años. A su vez, también se exponen los objetivos de este proyecto, dando una visión de su estructura y las partes que la forman.

1.1. Dominio de Aplicación

La evolución de los SIG¹ ha sido imparable en la última década. Motivado -sin duda- por el interés de gigantes tecnológicos como Google, Apple y Microsoft, ha provocado que se hayan popularizado y extendido tecnologías que antes sólo estaban al alcance de profesionales y científicos.

¹Sistemas de Información Geográfica

Capítulo 2

Conceptos Previos

2.1. Coordenadas Geográficas	4
--	---

En los últimos años, la información geográfica -también llamada información geoespacial- a cobrado una gran importancia. La facilidad -y necesidad- de identificar el lugar donde se han obtenido los datos ha provocado la aparición de nuevos términos. A continuación se presenta una serie de definiciones que ayudarán a comprender el dominio y alcance de este proyecto.

2.1. Coordenadas Geográficas

Para determinar la posición de un objeto en el globo terrestre se emplean coordenadas geográficas, un sistema de referencia que utiliza dos coordenadas angulares: latitud (eje Norte-Sur) y longitud (eje Este-Oeste) y una tercera coordenada elevación para indicar la altitud.

Latitud es el ángulo que existe entre un punto cualquiera y el Ecuador, medida sobre el meridiano que pasa por dicho punto. En cartografía, la latitud se suele expresar en grados sexagesimales. Todos los puntos ubicados sobre el mismo paralelo tienen la misma latitud. Aquellos que se encuentran al norte del Ecuador reciben la denominación Norte (N), los que se encuentran al sur del Ecuador reciben la denominación Sur (S). El rango de valores es de 0° para el Ecuador a 90° para los polos Norte y Sur con latitud 90° N y 90° S respectivamente.

Longitud es el ángulo a lo largo del ecuador desde cualquier punto de la Tierra. Por convenio, está aceptado internacionalmente que Greenwich en Londres es la longitud 0. Las líneas de longitud son círculos máximos que pasan por los polos y se llaman meridianos. La longitud también se expresa en grados sexagesimales entre 0° y 180° indicando a qué hemisferio (occidental W —del inglés *West*— y oriental E —*East*—) o también entre 0° y 180° positivos indicando Este, o negativos indicando hacia el Oeste.

Elevación o altitud es la distancia vertical de un punto de la Tierra respecto al nivel del mar, llamada elevación sobre el nivel medio del mar, en contraste con la altura, que indica la distancia vertical existente entre dos puntos de la superficie terrestre.

Combinando estos dos ángulos (latitud y longitud) junto con la elevación, es posible expresar la posición de cualquier punto de la superficie de la Tierra.

Capítulo 3

Plan de Proyecto

3.1. Análisis de Viabilidad	6
---------------------------------------	---

Como paso previo a la realización de un proyecto es necesario realizar una serie de análisis que -dependiendo del tipo de sistema a desarrollar- determinen su viabilidad y permitan intentar predecir los problemas y los riesgos que se encontrarán durante su elaboración. Por ejemplo, en un proyecto de ingeniería, si los problemas de desarrollo superan a las expectativas vinculadas a las predicciones y previsiones, no compensaría el esfuerzo de implementación. En este capítulo, se estudia el plan de proyecto para la propuesta realizada, sin tener en cuenta las futuras ampliaciones.

3.1. Análisis de Viabilidad

Al emprender el desarrollo de un proyecto los recursos y el tiempo deben ser realistas para su materialización a fin de evitar pérdidas económicas o frustración profesional. La viabilidad y el análisis de riesgos están relacionados de muchas maneras, si el riesgo del proyecto es alto, la viabilidad de producir software de calidad se reduce. Así pues, en el presente capítulo aborda los siguientes puntos:

- **Descripción** formal del proyecto.
- **Objeto**, motivos y necesidades del usuario.
- **Objetivos**, declaración del objetivo final del proyecto.
- **Metodología**, introducción al proceso del ciclo de vida elegido.
- **Requerimientos y recursos necesarios** para alcanzar el objetivo.
- **Sistema de control** para el seguimiento de los hitos principales.
- **Elementos de riesgo** que pueden hacer inviable el proyecto.
- **Beneficios** esperados tras la realización del proyecto.
- **Conclusiones** y resoluciones finales.

Capítulo 4

Tecnología

4.1. Base Tecnológica	8
---------------------------------	---

La base tecnológica es el grupo de tecnologías sobre las que se apoya el software y su desarrollo. Constituyen una parte importante en el análisis del proyecto, puesto que, de su elección dependen otros factores tales como: curva de aprendizaje y facilidad de uso, soporte y evolución futura, documentación y comunidad, etc.

Este capítulo se centra en la última versión, a fecha de entrega de esta memoria, del programa. Es probable que algunas módulos aquí expuestos sean sustituidos por otros en futuras iteraciones siguiendo el ciclo de vida del software.

4.1. Base Tecnológica

A continuación se describen las tecnologías más importantes sobre las que se ha desarrollado el programa **PhotoPlace**. El nexo común entre todas las tecnologías expuestas es su capacidad multiplataforma y formar parte del ecosistema del software libre.

Capítulo 5

Diseño

5.1. Ciclo de vida	10
------------------------------	----

Como se ha visto, la realización del proyecto ha seguido directrices marcadas por la metodología ágil Scrum-Ban y por el Diseño Guiado por el Dominio. Además, la arquitectura de la aplicación se ha dividido en distintas capas apoyadas por patrones de diseño e integración.

En este capítulo se muestran, en primer lugar, los conceptos básicos necesarios para comprender el proceso de diseño del software y, posteriormente, los modelos del software logrados tras la última iteración de la aplicación.

5.1. Ciclo de vida

El ciclo de vida está basado en los principios de las metodologías ágiles y se organiza en iteraciones que incluyen: análisis de requisitos, diseño, codificación, revisión y documentación.

Capítulo 6

Implementación

6.1. Organización del código fuente	12
---	----

En el presente capítulo se prestará atención a la arquitectura de la aplicación, es decir, a los bloques de construcción del programa. Los componentes son: paquetes, librerías, frameworks, *APIs*, etc descritos en términos de módulos, clases, funciones y algoritmos. En esencia, se trata del software y de la organización del código fuente. Se mostrarán fragmentos de código del software implementado con los aspectos más significativos. Esta parte de la documentación técnica es de gran importancia para comprender el proyecto.

6.1. Organización del código fuente

El código fuente de todo el proyecto, incluida la presente memoria realizada en L^AT_EX, se ha gestionando en Git, usando el flujo de trabajo Git-Flow. Se ha estructurado el código fuente de la siguiente forma:

Capítulo 7

Conclusiones

7.1. Análisis Económico	14
-----------------------------------	----

En este último capítulo del proyecto se presenta una estimación de los costes de desarrollo totales para, después de las conclusiones, finalizar el capítulo con unos apuntes acerca de las direcciones de las posibles futuras ampliaciones.

7.1. Análisis Económico

En esta sección se muestra un estudio de los costes de la aplicación basándose principalmente en el tiempo empleado. Se ha realizado un análisis económico desde dos puntos de vista diferentes: costes de desarrollo y retorno de inversión.

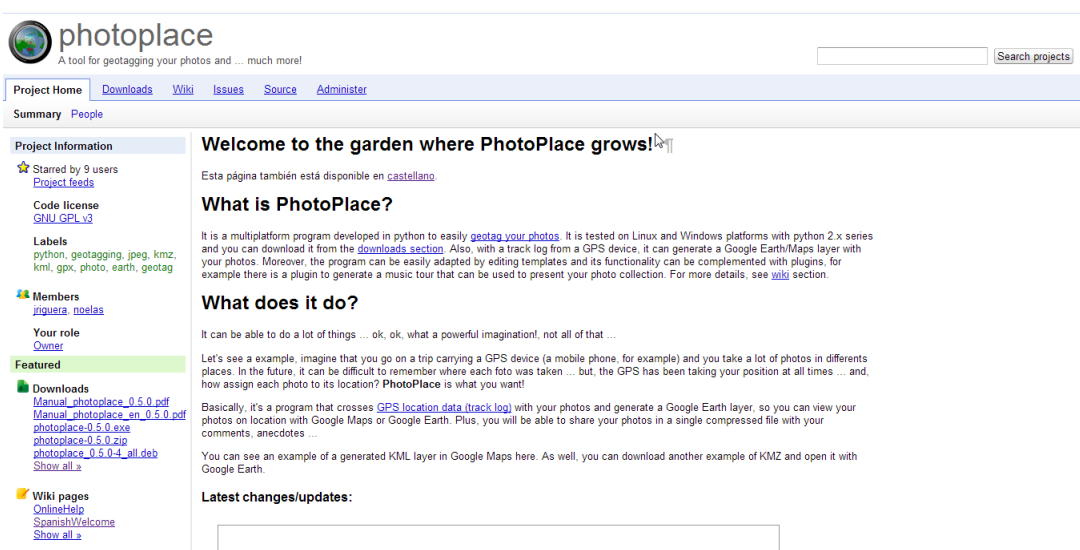
Apéndice A

Demostración

En este apéndice se mostrará el funcionamiento de la aplicación, primero con la versión para computador y a continuación la aplicación web. Para ello, se mostrará el caso de uso para generar una presentación KML. No se pretende detallar todos los parámetros del proceso, puesto que están cubiertos por la documentación de usuario disponible en la web, sólo mostrar la apariencia de la aplicación y su filosofía de funcionamiento.

A.1. PhotoPlace

Como se ha comentado, el proyecto ha sido desarrollado en abierto usando la plataforma Google Code. El nombre elegido para el software ha sido PhotoPlace¹ y se ha desarrollado un logotipo que refuerza la identidad del programa:



Web de desarrollo de PhotoPlace en Google Code

¹Web oficial del proyecto <https://code.google.com/p/photoplace>

Glosario

Diseño Guiado por el Dominio o *Domain Driven Design* (DDD) es un enfoque para el desarrollo de software con necesidades complejas mediante una profunda conexión entre la implementación y los conceptos del modelo y núcleo del negocio. 10

iteración en las Metodogías Ágiles es una unidad de tiempo en la que se desarrollan las funcionalidades acordadas. También conoce como *Sprint* en la terminología Scrum. Típicamente se corresponde con dos semanas de desarrollo, pero esto depende del proyecto. Cada iteración del ciclo de vida incluye normalmente diseño, codificación, pruebas y documentación. 8, 10

metodología ágil método de ingeniería del software basado en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan mediante la colaboración de grupos auto organizados y multidisciplinarios. 10

Scrum-Ban método de desarrollo que emplea Kanban para la planificación de tareas en cada iteración y Scrum para la planificación de las iteraciones, combinando ambas metodologías en un mismo ciclo de desarrollo. 10

Acrónimos

SIG Sistema de Información Geográfica 2

Bibliografía

Alistair Cockburn (2001). *Agile Software Development*, Addison–Wesley.

Django Software Foundation (2014). The django framework.

<https://www.djangoproject.com/>

Eric Evans (2004). *Domain-Driven Design, Tackling Complexity in the Heart of Software*, Addison-Wesley.

Erich Gamma, Richard Helm, R. J. and Vlissides, J. (2000). *Design Patterns, Elements of Reusable Object-Oriented Software*, Addison-Wesley.

Fog Creek Software (2014). Trello, organize anything, together.

<https://trello.com/>

Henrik Kniberg (2008). *Scrum y XP desde las trincheras*, InfoQ.

Jeff Sutherland, K. S. (2013). *The Scrum Guide*, Scrum.org.

Kent Beck, James Grenning, R. C. M. et al. (2014). Manifesto for Agile Software Development.

<http://agilemanifesto.org/>

Pressman, R. S. (2005). *Ingeniería del software. Un enfoque práctico (sexta edición)*, McGraw Hill Higher Education.

Python Software Foundation (2014). Python documentation 2.7.x.

<https://docs.python.org/2/>

Wikimedia Foundation Inc. (2014). Wikipedia, the free encyclopedia.

<http://www.wikipedia.org>

Everything will be okay in the end. If it's not okay, it's not the end.

Copyright (c) José Riguera.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with the Invariant Sections being: DEDICATORIA, AGRADECIMIENTOS and BIBLIOGRAFÍA, with the Front-Cover Texts being PORTADA UDC, and with no Back-Cover Texts.

