



Facultade de Informática
Departamento de Computación

PROXECTO FIN DE CARREIRA
Enxeñaría en Informática

Servizo en liña para a publicación de gravacións de radio e podcasts.

Alumno: Fernando Liñares Varela
Director: José María Casanova Crespo
Data: 28 de mayo de 2018

D. JOSÉ MARÍA CASANOVA CRESPO
Profesor, Facultade de Informática
Departamento de Computación
Universidade de A Coruña

CERTIFICA:

Que a memoria titulada “*Servizo en liña para a publicación de gravacións de radio e podcasts.*” foi realizada por FERNANDO LIÑARES VARELA conforme á descrición inicialmente proposta baixo a miña dirección e constitúe o seu Proxecto Fin de Carreira de Enxeñaría en Informática. Pola presente, autorizo a súa presentación para que o Proxecto sexa defendido nesta convocatoria.

En A Coruña, a 28 de mayo de 2018

D. JOSÉ MARÍA CASANOVA CRESPO
Director do proxecto

Título

Servizo en liña para a publicación de gravacións de radio e podcasts.

Servicio en línea para la publicación de grabaciones de radio y podcasts.

Online service for publishing radio broadcasting recordings and podcasts

Clase: Proxecto clásico de enxeñaría

Autor: Fernando Liñares Varela

Director: José María Casanova Crespo

Data: 28 de mayo de 2018

Tribunal

**Data de
defensa:**

Calificación:

Resumo

Nos últimos anos, Internet converteuse nunha peza clave para os medios de radiodifusión xa que permite un alcance global e o acceso baixo demanda aos contidos emitidos. Isto é especialmente interesante para as pequenas emisoras locais, a miúdo comunitarias, culturais e con orzamento limitado.

A estas últimas está orientado este proxecto. Consiste nun punto de encontro online para promocionar contidos radiofónicos e favorecer a súa reemisión por parte de distintos medios de comunicación (Emisoras, canles de podcasting...) así coma o seu consumo directo por parte dos visitantes da web. Para o seu desenvolvemento, utilizouse Django 1.11, un framework web de Python, ferramentas HTML5, Javascript e CSS-grid. Tamén se utilizaron ferramentas de sindicación RSS para o acceso aos contidos de terceiros.

Nesta memoria trátase o proceso completo de desenvolvemento do proxecto desde as fases de análise e deseño até os detalles de implementación. Mencionáranse tamén as liñas de traballo que se pretenden seguir no futuro.

Lista de Palabras Clave

Radio, Podcast, Web, Django, Postgres, Python, Javascript, jQuery, CSS Grid.

Dedicatoria por hacer

Agradecimientos

A los profesores José María Casanova Crespo por sus consejos durante el desarrollo del proyecto. POR HACER

Índice general

1. Introducción	1
1.1. Marco do proxecto	2
1.2. Motivación	2
1.3. Obxectivos	3
2. Conceptos Previos	5
3. Plan de Proyecto	7
3.1. Análisis de Viabilidade	8
4. Tecnoloxías Utilizadas	9
4.1. Linguaxes	10
4.1.1. Python	10
4.1.2. HTML	11
4.1.3. CSS	12
4.1.4. JavaScript	12
4.1.5. SQL	13
4.1.6. XML	14
4.2. Django Framework	15
4.3. Celery	16
4.4. RabbitMQ	17
4.5. Bootstrap	17
4.6. Apache HTTP server	18
4.7. PostgreSQL	18
4.8. Ferramentas de desenvolvemento	18
4.8.1. Eclipse	18
4.8.2. Git	19
4.8.3. Dia	19

ÍNDICE GENERAL

5. Diseño	21
5.1. Ciclo de vida	22
6. Implementación	23
6.1. Organización del código fuente	24
7. Conclusiones	25
7.1. Análisis Económico	26
A. Apéndice	27
A.1. PhotoPlace	27

Índice de figuras

1.1. Ouvintes de radio promedio diario do ano 2017 en España.	2
4.1. Exemplo de árbore HTML DOM[1]	11

Capítulo 1

Introducción

1.1. Marco do proxecto	2
1.2. Motivación	2
1.3. Obxectivos	3

1.1. Marco do proxecto

A radiodifusión tradicional, entendendo esta coma a retransmisión de contidos de audio a través de ondas analóxicas, presenta a día de hoxe unha serie de limitacións. A máis importante, se cadra, é o feito de que a demanda de frecuencias é superior ao que o espectro radioeléctrico pode ofrecer. Cómpre, por iso, a existencia de unha autoridade que outorgue licenzas de emisión sendo ditas institucións, na nosa sociedade, a Secretaría de Estado de Telecomunicaciones y para la Sociedad de la Información e máis a Secretaría Xeral de Medios[2]. Isto implica a imposibilidade de emitir contidos por parte de aqueles que ou ben non poidan facer fronte á inversión que unha licenza supón ou ben non lles fose outorgada.

A medida que o acceso a Internet se fai máis cotián, a emisión por streaming eríxese coma solución aos problemas da radio en FM. A través deste medio, unha emisora pode emitir contidos sen necesidade de licenzas, acadando, ademáis, unha cobertura global de xeito centralizado, sen necesidade de emitir mediante cadeas de radio enlace, como é costume nas grandes emisoras en FM do Estado Español, coa inversión en infraestruturas que tal cousa require.

Internet permite non só a emisión en directo mediante streaming, senón tamén o acceso a contidos baixo demanda co nacemento do podcasting a mediados da década dos 2000[3]. A aparición de ferramentas de uso diario que permiten un acceso fácil a estes contidos está a afectar ao comportamento dos usuarios: Actualmente, un 7.5 % dos ouvintes escoitan a radio por Internet; aínda lonxe dos ouvintes de FM, porén máis do dobre dos de Onda Media[4].

OYENTES DE RADIO PROMEDIO DIARIO								
EGM acumulado 2017	Total Población	Total Oyentes	FM	OM	Total Internet	Directo / Streaming	Diferido / Podcast	TDT
TOTAL OYENTES (000)	39.783	23.605	21.628	725	1.775	1.440	384	476
Penetración %	100,0	59,3	54,4	1,8	4,5	3,6	1,0	1,2
Cuota por onda		100,0	91,6	3,1	7,5	6,1	1,6	2,0

Figura 1.1: Ouvintes de radio promedio diario do ano 2017 en España.

1.2. Motivación

Ao entender que os suxeitos que atopan dificultades para emitir por FM son a miúdo medios do chamado terceiro sector. É dicir: entidades pequenas, comunitarias, sen ánimo de lucro, independentes e con finalidade maioritariamente cultural e social[5]. O impacto

positivo na pluralidade informativa e no desenvolvemento da sociedade civil destes medios é explicitamente recoñecido pola UNESCO[6].

Se ben a emisión por streaming e o podcast se perfilan coma a alternativa nun futuro inmediato, tamén veñen acompañados de certa aura de incertidume. Hai que ter en conta que os intereses deste tipo de emisoras adoitan ser de ámbito local. Os eventos que unha radio comunitaria ten capacidade de cubrir rara vez son internacionais e os patrocinios aos que adoitan ter acceso son PEMEs da propia localidade na que se atope a emisora.

Deste xeito, a vantaxe de de globalidade que ofrece Internet acaba por non ser tal mentres que, pola contra, a súa visibilidade si queda diluída pola numerosa oferta existente, esta si, a nivel global.

1.3. Obxectivos

Este proxecto pretende servir de axuda aos colectivos do terceiro sector da comunicación, favorecendo a colaboración entre eles e achegándolles as ferramentas necesarias para acadar unha maior presenza na rede.

O produto resultante consistirá nun portal web onde os usuarios poderán acceder a un catálogo de programas. O mencionado catálogo estará constituído por arquivos de son enlazados por outros usuarios que desexen compartilos desde o seu propio almacenamento.

Desde ese portal web poderase:

- Acceder a ficheiros de audio mediante streaming por Internet e descarga directa desde servidores alleos ao servizo.
- Como usuario, publicar de xeito manual ou automatizado o seu contido no propio sitio web, de xeito organizado por programas e categorías.
- Procurar contidos mediante ferramentas web e a opción de subscribirse aos diferentes programas ou emisoras.
- Colaborar entre usuarios na xestión dos contidos publicados.

Capítulo 2

Conceptos Previos

A continuación preséntanse unha serie de definicións coa finalidade de clarificar algúns conceptos que se manexan neste proxecto

Emisora

Capítulo 3

Plan de Proyecto

3.1. Análisis de Viabilidad	8
---------------------------------------	---

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

Tecnoloxía

4.1. Linguaxes	10
4.1.1. Python	10
4.1.1.1. Anaconda	10
4.1.2. HTML	11
4.1.3. CSS	12
4.1.3.1. CSS Grid	12
4.1.4. JavaScript	12
4.1.4.1. jQuery	13
4.1.5. SQL	13
4.1.6. XML	14
4.1.6.1. RSS	14
4.2. Django Framework	15
4.3. Celery	16
4.4. RabbitMQ	17
4.5. Bootstrap	17
4.6. Apache HTTP server	18
4.7. PostgreSQL	18
4.8. Ferramentas de desenvolvemento	18
4.8.1. Eclipse	18
4.8.2. Git	19
4.8.2.1. GitHub	19
4.8.3. Dia	19

Este capítulo contempla a base tecnolóxica utilizada na última versión do proxecto até a data de entrega desta memoria. Os conceptos listados nesta sección están suxeitos a posibles cambios por mor dos futuros traballos de actualización e despregue nun entorno de produción que se pretenden efectuar.

Á hora de elixir o uso destas ferramentas valorouse a súa natureza open source xa que se pretende que o resultado dispoña dunha licencia de software libre compatible coa definición da Free Software Foundation.

Tamén se valoraron, dado que nos capítulos anteriores insistiuse na importancia dos dispositivos móbiles no consumo da radio a través de Internet, as posibilidades de ditas ferramentas de ofrecer un bo resultado en distintos dispositivos e distintas plataformas software.

4.1. Linguaxes

4.1.1. Python

Python é unha linguaxe de propósito xeral de alto nivel. Trátase dunha linguaxe interpretada polo que é necesario ter un intérprete para executar o código. Ao ser o intérprete unha capa intermedia de software entre o programa e o sistema, Python é unha linguaxe fácilmente portable entre dispositivos de distinta natureza.

A súa sintaxe baseada na indentación está pensada para favorecer a comprensión entre distintos desenvolvedores e facilitar o mantemento do código.

O seu sistema de tipado implícito e a súa riqueza de bibliotecas para executar comandos de consola de xeito programático fan que sexa moitas veces descrito coma “unha linguaxe de scripting orientada a obxectos” [7], o cal serve coma mostra da súa flexibilidade, un dos motivos polo que foi elixida para este proxecto.

4.1.1.1. Anaconda

Anaconda é unha distribución de Python, libre e de código aberto (licenza BSD), utilizada normalmente para análise estatística e machine learning. Inclúe, ademais dunha completa instalación do intérprete de Python, un rico conxunto de librarías de uso común e o xestor de paquetes conda, sendo esas dúas últimas melloras o motivo primordial polo que foi a elección para este proxecto. Conda utiliza os paquetes da comunidade de Conda Forge [8]

Anaconda é responsabilidade de Anaconda Incorporated, antigo Continuum Analytics.

A versión utilizada foi a Anaconda 4.4.0 de 64 bits para Python 3, a máis nova no momento de comezar o traballo. Inclúe o intérprete para Python versión 3.6.1.

4.1.2. HTML

HTML (acrónimo de HyperText Markup Language) é a linguaxe estándar para a creación da estrutura das páxinas web. Os elementos presentes decláranse mediante bloques representados por etiquetas(tags), de ahí que reciba o nome de “markup language” (linguaxe de marcado). Estas etiquetas son interpretadas polos navegadores para renderizar os contidos[9].

A estrutura declarada no código HTML pode ser representada pola interface de DOM (Document Object Model). O DOM é unha árbore creada polo navegador ao cargar a páxina e é moi útil para acceder aos elementos da páxina de xeito programático, como veremos máis adiante.

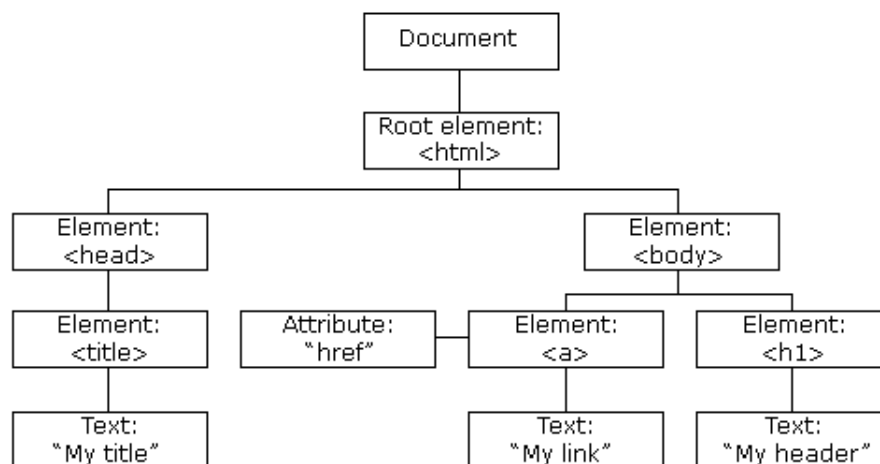


Figura 4.1: Exemplo de árbore HTML DOM[1]

Dado á natureza multimedia da web realizada, utilizouse HTML5. Esta quinta revisión do estándar inclúe novas etiquetas para o tratamento das imaxes e dos contidos de audio e vídeo sen necesidade de utilizar outras tecnoloxías complementarias, simplificando así o desenvolvemento e o mantemento posterior do portal. Desde decembro do pasado ano 2017, a versión 5.2 é a última estable recomendada polo World Wide Web Consortium(W3C)[10].

Ao ser o uso de HTML un estándar tan lonxevo, consolidado e popular, non se consideraron alternativas para este proxecto.

4.1.3. CSS

CSS (Acrónimo de Cascading Style Sheets) ou “Follas de estilo” é unha linguaxe empregada para definir a estética dos elementos definidos anteriormente no código HTML: A súa cor, fonte, posición... Ao separar o estilo da estrutura, favorécese a reutilización de código xa que un mesmo ficheiro de CSS pode ser utilizado en diversas páxinas ao mesmo tempo. CSS permite tamén adaptar o contido das páxinas a dispositivos de distinto tamaño[11].

4.1.3.1. CSS Grid

O módulo de “Grid” úsase para definir un deseño de interface gráfica consistente nunha cuadrícula de dúas dimensións en CSS. Nun modelo deste tipo, declárase un conxunto de elementos no HTML coma pertencentes a un “grid container”(contedor da grella ou da cuadrícula) e, á súa vez, zonas fillas que se posicionarán dentro dese contedor dependendo das características coas que este último fose definido na folla de estilos.

Unha cela nun contedor pode, á súa vez, consistir noutro contedor, permitindo así deseños asimétricos. O tamaño das celas pode ser fixo, relativo á páxina ou automático dependendo do contido da cela. Por todo o dito, este módulo proporciona un nivel de flexibilidade superior ao que poderíamos obter utilizando outras alternativas populares coma CSS Flexbox.

O nivel primeiro de CSS Grid non acadou aínda, na data de entrega desta memoria, o status de “recomendación” da World Wide Web Consortium(W3C)[12], porén, xa é soportado polas últimas versións dos navegadores máis populares[13] polo que non se considerou un risco utilizalo neste proxecto.

4.1.4. JavaScript

JavaScript é unha linguaxe de scripting interpretada e de alto nivel. Utilízase principalmente (e tamén neste proxecto) coma unha ferramenta para mellorar a interactividade da interface web ao permitir executar código orientado a eventos no lado do cliente, aforrando recargas da páxina e incluso accesos innecesarios á base de datos xa que permite o uso do disco local mediante cookies.

A pesares do seu nome, non garda relación algunha coa linguaxe Java e serven a propósitos claramente distintos. O núcleo desta linguaxe está regulado polo estándar ECMAScript®, na súa 8ª versión[14] no momento de entregar esta memoria.

Como se comentou no apartado 4.1.1, que sexa interpretada implica a necesidade dun intérprete para executar o código. Ese intérprete, comunmente chamado JavaScript En-

gine preséntase embebido nos navegadores web. Non obstante, non existe unha única implementación senón que distintos navegadores presentan distintas versións. Para o desenvolvemento deste proxecto utilizáronse os navegadores Mozilla Firefox 57.0.1 e Google Chrome 66.0, cuxos motores de JavaScript son SpiderMonkey e V8 respectivamente[15], ambos os dous libres e de código aberto.

Pese a que o seu uso nos navegadores continúa a ser a principal razón de ser desta linguaxe, tamén se utiliza a día de hoxe noutro tipo de produtos coma Node.js(para correr JavaScript no lado do servidor) ou Apache Couch DB (Para o manexo de bases de datos na nube)[16]

Ademáis do código en JavaScript puro, este proxecto tamén fai uso nalgúns partes do código de funcións de jQuery.

4.1.4.1. jQuery

jQuery é unha biblioteca de JavaScript creada co fin de simplificar o “scripting” no lado do cliente. Trátase dunha ferramenta libre, publicada baixo licenza MIT e é mantida pola comunidade jQuery Team. No 2015, xa era empregada polo 63 % do Top Million Websites[17], incluíndo sitios populares coma Netflix, Amazon ou Microsoft. O seu uso segue a ser moi xeralizado a día de hoxe.

Cunha API funcional e válida en gran parte dos navegadores web, evita moitos dos problemas de incompatibilidade que aparacen ao utilizar JavaScript puro e o código resultante adoita ser moito máis conciso, mellorando a súa comprensibilidade e o seu mantemento. Isto débese en grande medida pola súa expresión “selector” que facilita o acceso aos distintos elementos da árbore do HTML DOM (ver figura 4.1).

A versión de jQuery utilizada foi a 2.1.4 por ser a máis actual incluída na biblioteca django-static-jquery-2.1.4[18] do framework utilizado (ver sección 4.2) ao comezo do desenvolvemento.

4.1.5. SQL

SQL é a linguaxe utilizada para a manipulación dos datos dentro do eido das bases de datos relacionais. Nace coma un refinamento ou “secuela” (SQL é unha simplificación da verba inglesa “sequel”) da linguaxe SQUARE, que á súa vez consistía nunha simplificación da linguaxe DSI/Alpha proposta polo mesmo Edgar F. Codd no momento de presentar o modelo relacional de Bases de Datos. O primeiro estándar foi publicado no ano 1986

pola American National Standards Institute (ANSI)[19]. Permite definir a estruturación dos datos, insertalos, eliminalos, editalos e, por suposto, consúltalos.

Neste proxecto, o seu uso explícito correspóndese só coas primeiras fases do traballo debido á utilización do framework Django coma se detalla no apartado 4.2. É necesario especificar tamén que, pese á existencia dun estándar, existen distintas versións desta linguaxe que a fan “non completamente portable” entre distintos sistemas de xestión de bases de datos[20]. Neste proxecto utilizouse a variante correspondente a PostgreSQL.

4.1.6. XML

XML é o acrónimo de “eXtensible Markup Language” (Linguaxe de marcado extensible). Consiste nunha serie de normas que dividen un documento en distintas partes, asignándolle unha identidade a cada unha delas. A diferenza doutras linguaxes de marcado coma o HTML, non existe un conxunto de etiquetas válido. No XML queda á responsabilidade do autor do documento establecer as etiquetas necesarias dependendo do entorno no que ese documento vaia ser utilizado. Por suposto, existen unha serie de normas estruturais para que o etiquetado se poida considerar coma válido, pero desde o punto semántico, a natureza destas etiquetas é moi flexible. Isto fai que moitas veces se denomine o XML coma unha Meta-Linguaxe, é dicir, unha linguaxe para definir linguaxes[21].

A función de XML é definir a semántica e a estrutura dun documento, pero nunca o estilo. Ao igual que HTML, pódese asociar unha folla de estilos CSS.

Neste proxecto, XML é utilizado na súa modalidade RSS do xeito detallado no apartado 4.1.6.1.

4.1.6.1. RSS

RSS é un tipo de formato XML utilizado para a “sindicación” web, isto é, a emisión de contidos actualizados dunha páxina web a un número indefinido de subscriptores. Utilízase para evitar a procura manual de novos contidos naquelas páxinas nas que a actualización é frecuente: Páxinas de novas, blogs, podcasts...

O seu funcionamento consiste nun ficheiro RSS (comunmente chamado “feed”) accesible desde a web polos programas clientes de rss que posúan os subscriptores. Ese feed mostra información resumida do contido publicado no sitio web ao que fai referencia. Cada vez que o contido se actualice, o feed actualizarase tamén e ese cambio será detectado polo cliente rss de xeito automático.

O nome procede do acrónimo de “Really Simple Syndication” (Sindicación Realmente Sinxela) e o estándar é mantido polo “RSS Advisory Board”. Na data de entrega desta memoria e desde o ano 2009, a última revisión é a 2.0.11 [\[22\]](#)

Neste proxecto, o interese céntrase na actualización dos podcasts. Existen alternativas a este formato de sindicación mais, dado o seu uso xeralizado entre os usuarios potenciais da aplicación web a desenvolver, non foron consideradas.

4.2. Django Framework

O proxecto Django nace no ano 2005 coma un conxunto de ferramentas Python para o desenvolvemento web publicadas baixo licenza BSD. A motivación dos seus creadores, Simon Wilson e Adrian Holovaty, naquel tempo programadores nun medio periodístico, era a homoxeneización e a reutilización de código. Isto daba resposta á necesidade de reducir o alto custo que supón manter código ad-hoc para cada novo artigo ou función, especialmente nunha páxina que requira unha constante actualización de contidos coma a dun diario online. É por iso que se adoita utilizar o termo “newsroom schedule” cando se fala da rapidez de desenvolvemento que permite o framework[\[23\]](#).

Django ofrece un conxunto de bibliotecas que dan soporte ás mecánicas máis comúns do desenvolvemento web:

- **Persistencia:** Django permite crear a estrutura da base de datos sen necesidade de escribir SQL. Isto conséguese mediante a superclase Model da librería “db” de Django. Ao declarar unha clase de Python coma extensión de “Model”, a biblioteca ocúpase automaticamente de manter a coherencia entre as súas instancias en memoria e as táboas da base de datos.
- **Peticións web:** A lectura e a interpretación destas peticións son levadas a cabo polo conxunto de bibliotecas de HTTP do framework, encapsulando as peticións e as respostas en obxectos para formar un estándar sinxelo e garantir a correcta formación das mesmas.
- **Enrutamento e Validación:** O framework ofrece un estándar para asignar as distintas vistas definidas no código ás URL’s desexadas. Os posibles formularios presentes nas vistas poden ser abstraídos en obxectos de Python simplificando a validación dos datos introducidos polos usuarios. A utilización destes sistemas é non obstante, opcional, puidendo o programador optar por utilizar sinxelos formularios HTML no caso de que isto axilizase o desenvolvemento.

- Sistema para embeber datos dinámicos no HTML: O chamado “template system” de Django marca uns procedementos sinxelos para acceder aos datos xenerados no código. Tamén ofrece ferramentas para implementar certa lóxica presentacional.
- Interface de administración nativa: Por defecto, este framework dispón dun panel de administración de acceso reservado aos superusuarios que evita os accesos directos á base de datos no caso de que un administrador necesite facer cambios manuais nos datos.
- Soporte por defecto para a autenticación e autorización: Django permite ao desenvolvedor invocar a clase nativa Usuario e efectuar con ela accións de rexistro, autenticación e concesión/revocación de permisos sen máis programación. Se ben, como se explica máis adiante, neste proxecto estendeuse esa clase Usuario e creouse unha xerarquía de permisos propia.

Para este proxecto utilizouse concretamente Django 1.11, a versión estable máis avanzada no tempo en que se comezou o traballo. Django sufriu unha actualización importante desde entón que rematou coa publicación de Django 2.0 en Decembro do pasado ano 2017. Estas dúas versións non son totalmente compatibles de modo que unha actualización implicaría cambios no código. Tendo en conta que a Django Software Foundation, responsable do mantemento do framework, non prevee retirar o soporte á versión utilizada nun futuro próximo e que a versión de Python utilizada é soportada tamén por Django 2[24], considerouse aceptable o grao de actualidade da tecnoloxía utilizada.

Ademáis do mencionado anteriormente, confiouse en Django para este proxecto pola súa madurez e pola súa popularidade, sendo utilizado en sitios coma Instagram, Disqus, Pinterest ou Open Knowledge Foundation.

4.3. Celery

Celery é unha cola de traballos asíncrona baseada na mensaxería distribuída (distributed message passing). Utilízase para distribuír traballos entre máquinas ou threads. Celery levanta un conxunto de procesos chamados “workers” nos que se executarán de xeito concurrente os diferentes traballos ou “tasks”. Eses workers consultan constantemente a cola de traballos para executar os traballos que podan estar pendentes[25].

Neste proxecto, utilizouse Celery para controlar a execución dos demos que corren no lado do servidor. Eses dous demos son: O encargado de actualizar a información do programa e os episodios e o encargado de calcular a popularidade dos programas. Utilizouse a

versión de Celery 4.1, a máis actual no repositorio de Anaconda no momento de comezar o traballo.

Django ofrece bibliotecas para traballar con Celery de xeito máis sinxelo, por exemplo, permitíndolle gardar información na base de datos do proxectos e engadindo ferramentas de administración dos procesos de Celery ao menu de administración de Django. Utilizáronse as bibliotecas `django-celery-results` 1.0.1 e `django-celery-beat` 1.1.1

Como xa se mencionou, Celery baséase na mensaxería, mais non inclúe un sistema de seu senón que lle hai que proporcionar un. Neste caso, optouse pola utilización de RabbitMQ (ver sección 4.4), un dos recomendados na documentación de Celery.

4.4. RabbitMQ

RabbitMQ é un software de “message passing” tipo “broker” mensaxes. Isto é, un sistema onde diferentes aplicacións se conectan ao broker coa fin de enviar a ou lelos deste, funcionando o dito broker coma unha cola. As mensaxes enviadas á cola por un aplicativo permanecerán aí até que sexan lidos por outro aplicativo. Unha mensaxe pode ser calquera cousa, desde meta información dos procesos até texto plano[26].

Neste proxecto utilizouse a versión 3.6.10-1 por seres esta a máis actual dispoñible nos repositorios do Sistema Operativo no momento de comezar o traballo.

4.5. Bootstrap

Bootstrap é un framework libre e de código aberto (publicado baixo licenza MIT) para á construción de interfaces web. O Framework ofrece unha serie de modelos ou “templates”, cada unha cunha estrutura HTML, declaracións en CSS e, nalgúns casos, extensións javascript. O uso neste proxecto, dado que o aspecto xeral da páxina ven determinado polas CSS Grid (ver apartado 4.1.3.1), vese reducido a un modelo que controla a tipografía e as cores das fontes.

Neste proxecto, utilizouse Bootstrap a través da biblioteca de `django-bootstrap4` para facilitar o seu uso desde os templates do framework de Django.

4.6. Apache HTTP server

4.7. PostgreSQL

PostgreSQL é un sistema de xestión de bases de datos (SXBD) relacionais libre e de código aberto (Licenza PostgreSQL, permisiva) desenvolvido polo PostgreSQL Global Development Group. A linguaxe SQL soportada pretende ser o máis fiel posible ao estándar e as súas operacións cumpren coas regras ACID (Acrónimo inglés para Atomicidade, Consistencia, Aillamento e Durabilidade)[27]. A súa orixe remóntase a 1996, polo que existe unha grande dispoñibilidade de recursos de documentación. Ese feito unido á certa experiencia persoal no seu uso xa antes de comezar o proxecto foron valorados á hora de elixir este sistema.

A versión utilizada é a 9.6-3 por ser a máis actual no momento de comezar o traballo. Para facer posible o acceso mediante Python, utilizouse o driver psycopg2 na súa versión 2.7.1

4.8. Ferramentas de desenvolvemento

4.8.1. Eclipse

Eclipse é un IDE(Integrated Development Environment ou Entorno de Desenvolvemento Integrado), isto é, un aplicativo consistente nunha serie de ferramentas de desenvolvemento de software dispoñibles arredor dun editor de texto tales coma, por exemplo, unha consola de saída estándar e ferramentas para compilar e depurar código desde a interface proporcionada.

É principalmente empregado nos proxectos de Java, C e C++, porén, dada a súa popularidade, existen extensións dispoñibles no seu propio “marketpace” que nos permiten utilizalo para outras linguaxes. No caso deste proxecto, ese plugin é PyDev, que proporciona as ferramentas necesarias para o desenvolvemento non só de proxectos de Python en xeral senón especificamente de Django.

Outra extensión utilizada foi EGit na súa versión 4.8. Esta serve para manexar o control de versións desde a interface gráfica de Eclipse. Egit utiliza JGit, unha implementación puramente Java do sistema Git (ver apartado 4.8.2).

Optouse pola versión de Eclipse máis nova no momento de comezar o traballo pese a que aínda non estaba dispoñible daquela nos repositorios do sistema operativo do equipo de

desenvolvemento: Eclipse Oxygen 4.7.0. A versión de PyDev utilizada foi a 5.9.2. Valorous PyCharm, un IDE específico para Python, coma alternativa; mais a versión de comunidade non tiña soporte para desenvolvemento específico de Django[28].

4.8.2. Git

Git é un sistema de control de versións libre e open source (licenza GPL v2). Un sistema de control de versións utilízase para gardar os diferentes estados (versións) do código nas distintas fases de desenvolvemento. Permite a creación de ramas para a división do traballo e de etiquetas para marcar certos hitos no proceso. Estas versións pódense gardar no equipo local ou nun repositorio remoto. No caso de Git, mantéñense copias tanto no repositorio coma nos equipos dos programadores, polo cal adoita calificarse coma un sistema de control de versións distribuído[29].

Como se mencionou no apartado 4.8.1, utilizouse sobre todo na súa implementación de EGit, mais as veces recurriuse ao paquete de sistema para realizar operacións sen necesidade de acceder a Eclipse. Nesas casos, utilizouse a versión 2.9.5.

Existen outros sistemas coma, por exemplo, Apache Subversion, pero elixiuse Git pola posibilidade de publicar o código en GitHub

4.8.2.1. GitHub

GitHub é un servizo de hosting web para repositorios de control de versións. É moi popular para o desenvolvemento colaborativo de proxectos de código aberto. Neste proxecto, utilizouse ata o momento coma alternativa gratuita para obter un repositorio online pero, idealmente, tamén será utilizado para colaborar con outros desenvolvedores no futuro.

4.8.3. Dia

Dia é un editor de Diagramas libre e de código aberto(Licenza GPL). Utilizaronse neste proxecto as súas extensións de UML para o diagrama de clases do sistema e a de Entidade Relación para o esquema de deseño da base de datos. A versión utilizada foi a 0.97.3.

Consideráronse outros editores coma, por exemplo, Umbrello, pero a elección foi DIA por criterios de estética e comodidade de uso debido á súa interface sinxela.

Capítulo 5

Diseño

5.1. Ciclo de vida	22
------------------------------	----

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	24
---	----

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	26
-----------------------------------	----

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.

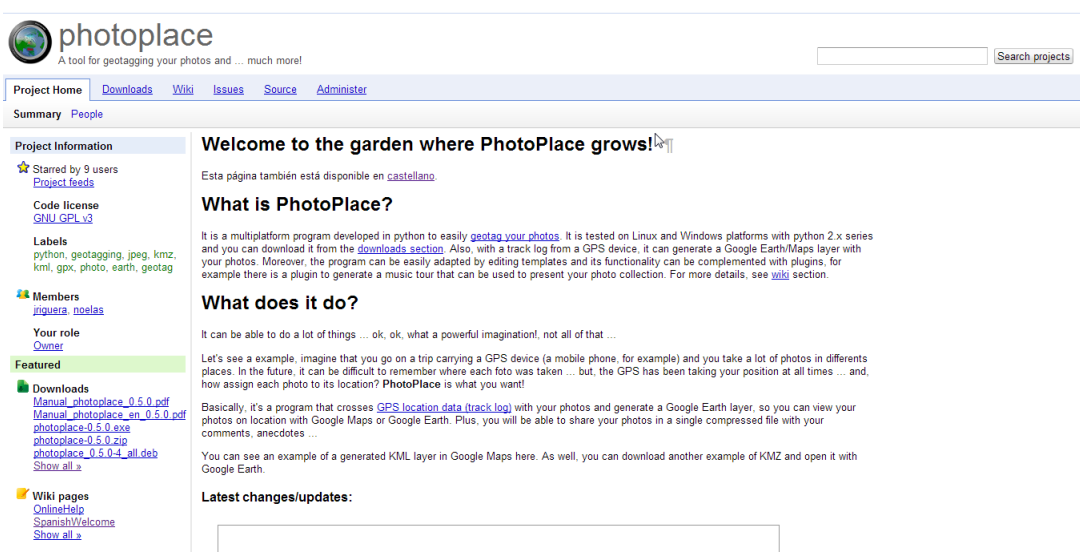
Apéndice A

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>

Bibliografía

- [1] W3Schools. Javascript html dom, . URL https://www.w3schools.com/js/js_htmlDOM.asp. III, 11
- [2] España, lei 7/2010, do 31 de marzo, Xeral da Comunicación Audiovisual. *Boletín Oficial Del Estado, xoves 1 de abril de 2010, suplemento en lingua galega ao núm. 79, sec I, páx. 30157*. URL https://www.boe.es/boe_gallego/dias/2010/04/01/pdfs/BOE-A-2010-5292-G.pdf. 2
- [3] Ben Hammersley. Audible revolution: Online radio is booming thanks to ipods, cheap audio software and weblogs. *The Guardian*, 12/04/2004. URL <https://www.theguardian.com/media/2004/feb/12/broadcasting.digitalmedia>. 2
- [4] Asociación para la investigación de medios de comunicación (AIMC). El streaming supera a la onda media en la escucha de radio. *3ª Ola EGM*, 30/11/2017. URL https://www.aimc.es/aimc-c0nt3nt/uploads/2017/11/171130_NP_EGM_2017ola3.pdf. 2
- [5] Investigan a España por no dar licencias a medios comunitarios. *Federación de Sindicatos de Periodistas (FeSP)*, 14/03/2018. URL <http://www.fesp.org/index.php/noticias/item/8136-investigan-a-espana-por-no-dar-licencias-a-medios-comunitarios>. 2
- [6] Frances J. Berrigan. *La Comunicación Comunitaria: Cometido de los medios de comunicación comunitaria en el desarrollo*. Editorial de la Unesco, 1981. URL <http://unesdoc.unesco.org/images/0013/001343/134355so.pdf>. 3
- [7] Mark Lutz. *Learning Python*. O'Reilly, 5 edition, 2013. ISBN 978-1-449-35573-9. 10
- [8] Anaconda Inc (Continuum Analytics). Anaconda distribution, 2018. URL <https://www.anaconda.com/distribution>. 10
- [9] W3Schools. HTML Introduction, . URL https://www.w3schools.com/html/html_intro.asp. 11
- [10] Steve Faulkner, Aaron Eicholz, Travis Leithead, Alex Danilo, Sangwhan Moon. HTML 5.2. Technical report, W3C, 14/12/2017. URL <https://www.w3.org/TR/2017/REC-html52-20171214>. 11
- [11] W3Schools. What is CSS?, . URL https://www.w3schools.com/css/css_intro.asp. 12

BIBLIOGRAFÍA

- [12] Tab Atkins Jr., Erika J. Etemad, Rossen Atanassov. CSS Grid layout module level 1. Technical report, W3C, 14/12/2017. URL <https://www.w3.org/TR/2017/CR-css-grid-1-20171214>. 12
- [13] W3Schools. CSS Grid layout module, . URL https://www.w3schools.com/css/css_grid.asp. 12
- [14] VV. AA. ECMAScript® 2017 Language Specification (ECMA-262, 8th edition). Technical report, ECMA International, Xuño 2017. URL <https://www.ecma-international.org/ecma-262/8.0>. 12
- [15] javascript.info. An Introduction to JavaScript. URL <https://javascript.info/intro>. 13
- [16] MDN Web docs. JavaScript. URL <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. 13
- [17] Aurelio De Rosa Bear Bibeault, Yehuda Katz. *jQuery in action*. Manning, 3 edition, 2015. ISBN 978-1-617-29207-1. 13
- [18] Python Software Foundation. django-static-jquery 2.1.4, 06/05/2015. URL <https://pypi.org/project/django-static-jquery>. 13
- [19] Alan Beaulieu. *Learning SQL*. O'Reilly, 2005. ISBN 978-0-596-00727-0. 14
- [20] W3Schools. Introduction to SQL, . URL https://www.w3schools.com/sql/sql_intro.asp. 14
- [21] Elliotte Rusty Harold. *XML Bible*. IDG Books Worldwide, 1999. ISBN 0-7645-3236-7. 14
- [22] RSS Advisory Board. RSS 2.0 Specification, 30/03/2009. URL <http://www.rssboard.org/rss-specification>. 15
- [23] James Bennett. *Practical Django projects*. Apress, 2009. ISBN 978-1-4302-1938-5. 15
- [24] Django Software Foundation. Django 2.0 release notes, 02/12/2017. URL <https://docs.djangoproject.com/en/2.0/releases/2.0>. 16
- [25] The Celery Project. Introduction to Celery. URL <http://docs.celeryproject.org/en/latest/getting-started/introduction.html>. 16
- [26] Lovisa Johansson. *Getting started with RabbitMQ and CloudAMQP*. Codes AB, 2 edition, 2017. 17

- [27] The PostgreSQL Global Development Group. About PostgreSQL. URL <https://www.postgresql.org/about>. 18
- [28] Jet Brains S.R.O. PyCharm Editions Comparison. URL https://www.jetbrains.com/pycharm/features/editions_comparison_matrix.html. 19
- [29] Ben Straub Scott Chacon. *Pro Git*. Apress, 2 edition, 2018. 19

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.

