



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**Integración del CENIEH en
ARIADNEplus**



Presentado por Gonzalo Cuesta Marín
en Universidad de Burgos — 22 de marzo
de 2020

Tutores: Dr. Carlos López Nozal
y Dr. Mario Juez Gil



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. Carlos Lopez Nozal y D. Mario Juez Gil, profesores del Departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos.

Exponen:

Que el alumno D. Gonzalo Cuesta Marín, con DNI 71310247N, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado título de TFG.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 22 de marzo de 2020

Vº. Bº. del Tutor:

Vº. Bº. del tutor:

D. nombre tutor

D. nombre tutor

Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android ...

Abstract

A **brief** presentation of the topic addressed in the project.

Keywords

keywords separated by commas.

Índice general

Índice general	III
Índice de figuras	V
Índice de tablas	VI
Introducción	1
Objetivos del proyecto	3
2.1. Objetivos generales	3
2.2. Objetivos específicos	3
Conceptos teóricos	5
3.1. Secciones	5
3.2. Referencias	5
3.3. Imágenes	6
3.4. Listas de items	6
3.5. Tablas	7
Técnicas y herramientas	9
4.1. Metodologías	9
4.2. Cliente de Control de versiones	9
4.3. <i>Hosting</i> del repositorio	10
4.4. Entorno de desarrollo integrado (IDE)	10
4.5. Documentación	10
Aspectos relevantes del desarrollo del proyecto	11

Trabajos relacionados	13
Conclusiones y Líneas de trabajo futuras	15
Bibliografía	17

Índice de figuras

3.1. Autómata para una expresión vacía	6
--	---

Índice de tablas

3.1. Herramientas y tecnologías utilizadas en cada parte del proyecto	8
---	---

Introducción

El CENIEH, Centro Nacional de Investigación sobre la Evolución Humana, es una instalación científica y tecnológica donde se desarrollan multitud de investigaciones relacionadas con la evolución humana. Además, son responsables de la conservación, restauración, gestión y registro de una gran cantidad de colecciones paleontológicas y arqueológicas procedentes de las excavaciones de Atapuerca y otros yacimientos tanto nacionales como internacionales de similares características.

Todas estas actividades generan un gran volumen de datos que, sin el uso de nuevas tecnologías, su almacenamiento sería una tarea verdaderamente compleja. Actualmente disponen de un sistema de base de datos interno que sirve como catálogo de colecciones. Este les permite gestionar la información conceptual (metadatos) de todos los elementos pertenecientes a cada colección como, por ejemplo, su identificación mediante etiquetas RFID y códigos de barras. Además, cuentan con un repositorio online de colecciones llamado CIR (CENIEH Institutional Repository), construido sobre la infraestructura DSpace, donde gestionan una pequeña cantidad de documentos. Al igual que con el sistema anterior, cada documento almacenado es representado como un ítem el cual tiene asignado un conjunto de metadatos y está asociado a una comunidad y colección determinada.

Exceptuando la información recogida en el CIR, la gran mayoría de datos están almacenados de forma local, es decir, no son accesibles desde el exterior. Esto puede significar un problema ya que cualquier investigador ajeno al CENIEH que pretenda consultar qué documentos tienen en posesión debe personarse en sus instalaciones para llevar a cabo dicha consulta. Como veremos a continuación, AriadnePlus es un proyecto que se presenta como solución a este problema.

AriadnePlus es un proyecto europeo que tiene como finalidad construir una infraestructura de investigación enfocada a la arqueología que fomenta la enseñanza, aprendizaje e investigación a través del acceso a recursos digitales y servicios. El pilar principal de esta infraestructura es su catálogo de colecciones digitales. En él todos los socios del proyecto vuelcan su contenido fruto de investigaciones, excavaciones, trabajos de laboratorio y otros procesos. Además, en el mismo portal donde se encuentra el catálogo, ofrecen multitud de servicios que contribuyen a mejorar la calidad del contenido.

La integración del CENIEH en este proyecto permitirá que todo el contenido almacenado de forma local salga a la red para que investigadores y estudiantes de toda Europa puedan visualizar y acceder a toda esta información de forma remota a través del portal oficial AriadnePlus.

La finalidad de este proyecto será llevar a cabo esta integración. Para ello trataré de diseñar e implementar una infraestructura software que permita gestionar cada uno de los conjuntos de datos que almacena el CENIEH de forma que estos puedan ser importados a AriadnePlus.

Objetivos del proyecto

Para alcanzar las metas fijadas durante el desarrollo del proyecto, he planteado los siguientes objetivos:

2.1. Objetivos generales

- Diseñar e implementar una infraestructura software que contribuya a realizar, de manera eficiente, los procesos de importación, gestión, modelado y exportación de los datos ya existentes en el CENIEH.
- Supervisar que todos los procesos cumplan con los requisitos establecidos por AriadnePlus.
- Crear un esquema de metadatos común para todas las colecciones que sea compatible entre ambas plataformas y facilite las tareas de gestión e importación de los datos.
- Adaptar todos los recursos y servicios del sistema al usuario objetivo de forma que pueda hacer uso de ellos sin ningún tipo de dificultad.

2.2. Objetivos específicos

- Desarrollar un gestor de contenidos que cumpla con un mínimo de requisitos:
 - ◊ Haga uso de un esquema de metadatos compatible con *CIDOC-CRM* o alguna de sus variantes utilizadas en el proyecto como *ACDM* o *AO-CAT*.
 - ◊ Implemente el protocolo de interoperabilidad *OAI-PMH* (Open Archive Initiative-Protocol for Metadata Harvesting).

- ◊ Cuento con un sistema de importación y exportación de metadatos con formato *.csv* y *.xml*.
- ◊ Interfaz amigable e intuitiva adaptada al usuario objetivo.
- Diseñar e implementar un esquema de metadatos en formato *.xml* que satisfaga las necesidades de ambas partes, es decir, ser compatible con *CIDOC-CRM* | *ACDM* | *AO-CAT* y además, tenga la capacidad de representar los conjuntos de datos que posee el CENIEH.
- Utilizar como sistema de documentación continua '*Read the Docs*'.
- Trabajar con '*Docker*' para facilitar el despliegue de la infraestructura sobre el servidor.
- Hacer uso de las herramientas existentes en la plataforma *AriadnePlus Gateway*.

Conceptos teóricos

En aquellos proyectos que necesiten para su comprensión y desarrollo de unos conceptos teóricos de una determinada materia o de un determinado dominio de conocimiento, debe existir un apartado que sintetice dichos conceptos.

Algunos conceptos teóricos de L^AT_EX¹.

3.1. Secciones

Las secciones se incluyen con el comando `section`.

Subsecciones

Además de secciones tenemos subsecciones.

Subsubsecciones

Y subsecciones.

3.2. Referencias

Las referencias se incluyen en el texto usando `cite` [4]. Para citar webs, artículos o libros [3].

¹Créditos a los proyectos de Álvaro López Cantero: Configurador de Presupuestos y Roberto Izquierdo Amo: PLQuiz

3.3. Imágenes

Se pueden incluir imágenes con los comandos standard de \LaTeX , pero esta plantilla dispone de comandos propios como por ejemplo el siguiente:



Figura 3.1: Autómata para una expresión vacía

3.4. Listas de items

Existen tres posibilidades:

- primer item.
- segundo item.

1. primer item.
2. segundo item.

Primer item más información sobre el primer item.

Segundo item más información sobre el segundo item.

-

3.5. Tablas

Igualmente se pueden usar los comandos específicos de \LaTeX o bien usar alguno de los comandos de la plantilla.

Herramientas	App	AngularJS	API REST	BD	Memoria
HTML5		X			
CSS3		X			
BOOTSTRAP		X			
JavaScript		X			
AngularJS		X			
Bower		X			
PHP			X		
Karma + Jasmine		X			
Slim framework			X		
Idiorm			X		
Composer			X		
JSON		X	X		
PhpStorm		X	X		
MySQL				X	
PhpMyAdmin				X	
Git + BitBucket		X	X	X	X
MikTeX					X
TeXMaker					X
Astah					X
Balsamiq Mockups		X			
VersionOne		X	X	X	X

Tabla 3.1: Herramientas y tecnologías utilizadas en cada parte del proyecto

Técnicas y herramientas

4.1. Metodologías

Scrum

Scrum es un marco de trabajo de procesos ágiles que contribuye al desarrollo y mantenimiento de productos *software*. Por ello, está catalogado como una metodología ágil la cual se caracteriza por trabajar con el ciclo de vida iterativo e incremental, donde se va liberando el producto software de forma periódica a través de *sprints* (iteraciones) [2].

Gitflow

Gitflow es un modelo de *branching* o ramificación que sirve de gran ayuda en la gestión de repositorios de tipo Git. El trabajo se organiza en dos ramas principales, la rama *master*, donde se efectúan los *commits* listos para producción, y la rama *develop*, donde está el código que conformará la siguiente versión planificada del proyecto. Además se proponen otras tres ramas auxiliares, *feature* para desarrollar nuevas características de la aplicación, *release* para preparar el siguiente código en producción y, por último, *hotfix* para corregir errores y bugs en el código en producción [1].

4.2. Cliente de Control de versiones

- Herramientas consideradas: [Gitg](#), [SmartGit](#), [GitKraken](#) y [GitHub Desktop](#)
- Herramienta elegida: [GitKraken](#).

GitKraken es un cliente para el control de versiones *Git* que nos permite realizar todas y cada una de sus tareas a través de una intuitiva y elegante interfaz gráfica. Además, incorpora funciones adicionales como *GitFlow*, que nos permite gestionar las diferentes ramificaciones del proyecto. De entre todas las opciones posibles, ha sido sin duda alguna la más competente de todas.

4.3. *Hosting* del repositorio

- Herramientas consideradas: [GitHub](#) y [GitLab](#).
- Herramienta elegida: [GitHub](#).

4.4. Entorno de desarrollo integrado (IDE)

PHP | CSS | JavaScript | XML

- Herramientas consideradas: [Atom](#), [Eclipse](#), [Zend Studio](#) y [Komodo](#)
- Herramienta elegida: [Zend Studio](#).

Zend Studio es un IDE para PHP que ha sido construido tomando como base Eclipse. Considero que es la opción ideal ya que da soporte a todos y cada uno de los lenguajes de programación utilizados por mi infraestructura software. Además, permite la instalación de *plugins* y acepta herramientas tales como *Docker* y *Gitflow*.

LaTeX

- Herramientas consideradas: [TeXstudio](#) y [Texmaker](#).
- Herramienta elegida: [Texmaker](#).

4.5. Documentación

- Herramientas consideradas: [LaTeX](#), [Markdown](#), [reStructuredText](#) y [Microsoft Word](#)
- Herramienta elegida: [Markdown](#) + [reStructuredText](#) + [L^AT_EX](#).

Aspectos relevantes del desarrollo del proyecto

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros³, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.

Trabajos relacionados

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Bibliografía

- [1] Alfonso. ¿qué es git-flow?, 2020. [Internet; accedido el 22-marzo-2020].
- [2] Agile Alliance. What is scrum?, 2020. [Internet; accedido el 22-marzo-2020].
- [3] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.