



04 DE MARZO DE 2020

**PRACTICA 1**  
DESARROLLO E INTEGRACIÓN DEL SOFTWARE

CAMILO VALDÉS, WILLEM VAN VROENHOVEN, ROGER URIBE  
MONTERO, MIGUEL ANGEL VARA  
INGENIERÍA INFORMÁTICA  
Universidad Francisco de Vitoria

## Índice de contenido

Introducción	2
Desarrollo	3
Control de Versiones	3
Estructura del árbol de versiones	3
Responsables y Roles	4
Definición del Modelo de Datos	4
Generación del Documento XML	4
Generación del Documento DTD	6
Desarrollo de la aplicación en Java	6
Objetivo	7
Dependencias	7
Conclusiones	8
Bibliografía	9

## Introducción

Para la elaboración de este documento se ha tomado como apoyo la practica 1 de la asignatura de Desarrollo e Integración del Software.

En la práctica que se plantea se pretende implementar un modelo de datos estructurado tomando como contexto la gestión de un almacén, para un posterior desarrollo de una aplicación que sea capaz de leer y escribir en el formato del almacén y modificar el contenido del modelo de datos anterior. Este desarrollo se pretende implementar mediante XML, DTD, JSON y Java.

Además, todo el desarrollo ha de estar bajo el control de versiones de git, el cual ayudará al equipo del proyecto con el desarrollo de este.

Debido al tamaño del repositorio se añade a continuación el enlace de este utilizando GitHub para el almacenamiento del repositorio remoto (<https://github.com/CarRaxX/dis-practica1>).

## Desarrollo

Para abordar el desarrollo de la práctica se ha decidido dividir en dos partes ya definidas en el enunciado, tomándose en primer lugar el desarrollo del Modelo de Datos a obtener, basado en el almacén y, por otro lado, el desarrollo de la aplicación en Java que sea capaz de escribir y leer los ficheros XML con la estructura del modelo anteriormente definido.

Además, todo el proceso de desarrollo se ha mantenido bajo un control de versiones, utilizando para ello, la herramienta de control de versiones de git.

## Control de Versiones

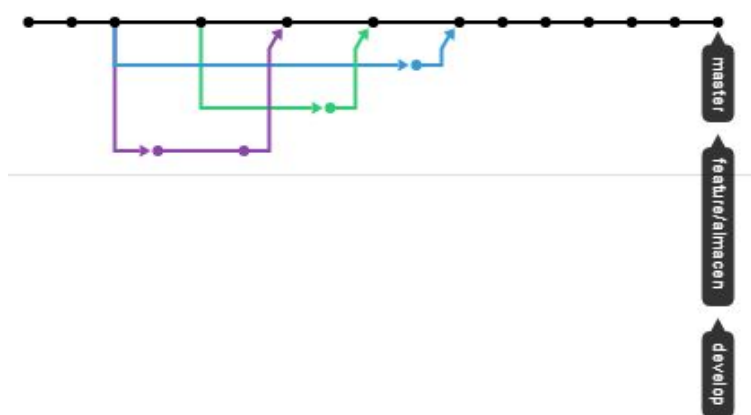
Para el desarrollo del proyecto se ha utilizado el control de versiones de git junto a un repositorio remoto privado en la plataforma de GitHub.

Así pues, se mostrará el árbol de versiones planteado para el control de este proyecto, los diferentes roles y las personas a las que se les asignan dichos roles y las funciones que han desempeñado en el proyecto respecto al control de versiones de Git.

## Estructura del árbol de versiones

Para elaborar el árbol de versiones se ha tomado como referencia la metodología empleada en Gitflow:

1. Rama **master**: Rama principal del proyecto que, en este caso, indica el inicio y el fin de este, teniendo por medio las diferentes versiones con grandes cambios que han acontecido en las diferentes versiones.
2. Rama **develop**: Rama secundaria donde se han ido incorporando las versiones finales de la rama de desarrollo **feature/almacen**. Además, las grandes implementaciones que se terminen en esta rama desembocarán en la rama principal o **master**.
3. Rama **feature/almacen**: Rama de desarrollo del documento XML, DTD donde se han ido implantando versiones parciales o funciones sencillas, cuya suma se implementará en la rama **develop**.



*Árbol de versiones*

Cada modificación realizada por un miembro del equipo se actualizaba a nivel local, realizando entregas incrementales al repositorio remoto medida que se iban realizando grandes cambios al proyecto o a la rama con la que se trabajase.

También, en ciertos momentos de necesidad se utilizaron comandos como “git reset” para revertir los cambios cuando se cometían ciertos errores en el desarrollo y estos se habían registrado en un “commit”.

### Responsables y Roles

Para el desarrollo de la práctica se ha tenido en cuenta la metodología GitFlow, pero por temas de rapidez y de que era una practica sencilla hemos decidido crearnos una rama aparte de la Master cada uno.

El proyecto se ha realizado entre los cuatro integrantes del grupo propuesto para la práctica y, ya divididas las partes de la página en el plan de trabajo, se asignaron los diferentes roles y trabajos a cada uno para controlar el desarrollo del proyecto.

Así pues:

- Roger Uribe: Se encargó del desarrollo en la rama **feature/almacen**.
- Willem: Se encargó del desarrollo en la rama **feature/almacen**.
- Camilo Valdés: Se encargó del desarrollo y mantenimiento de las ramas **develop** y **master**.
- Miguel Ángel: Se encargó del desarrollo y mantenimiento de las ramas **feature/almacen** y **develop**.

Cada miembro del grupo trabaja y se encarga de sus o su rama asignada. En un inicio, cada uno debe de realizar un “git pull” para obtener el contenido remoto más actualizado, mientras que cuando termina la parte de su desarrollo debe de realizar un nuevo “commit” y un “git push” para mandar esas nuevas modificaciones al repositorio remoto.

### Definición del Modelo de Datos

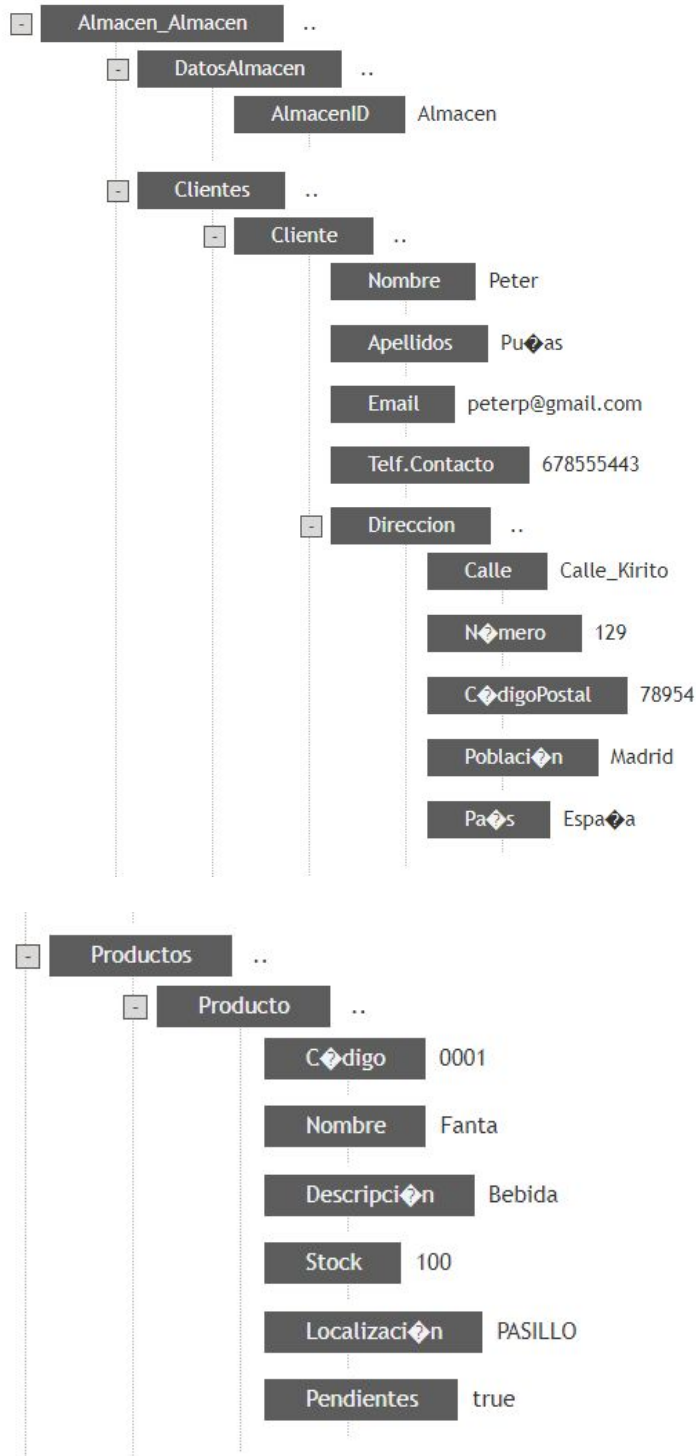
A continuación, se mostrarán los documentos necesarios para la elaboración del modelo de datos planteado, así como las funciones y la forma en la que se ha realizado a lo largo del desarrollo.

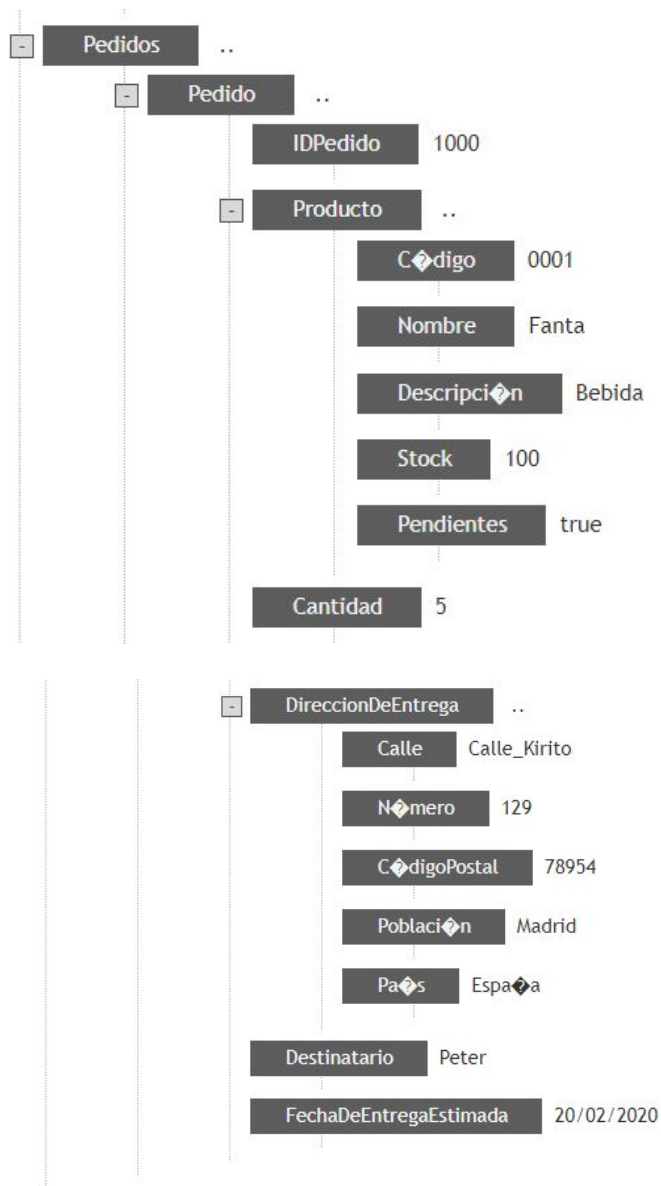
Para este desarrollo se han tomado como herramientas, aplicaciones como Eclipse, las cuales facilitan el desarrollo en este tipo de lenguaje debido al reconocimiento de etiquetas que utiliza este.

### Generación del Documento XML

La primera parte del desarrollo consiste en la generación del documento XML que dotará la estructura de los diferentes elementos presentes en un Almacén.

Así pues, la estructura del Almacén se ha seguido tal y como se dicta en el enunciado, pasándose a una estructura en XML que se define por la siguiente composición:





### Estructura del fichero XML

#### Generación del Documento DTD

Una vez definida la versión final del documento XML, se procede a la elaboración del documento DTD el cual otorgará al modelo una validez y verificación de su estructura ya definida.

Para ello, se realizará un fichero DTD que contenga los elementos y sus propiedades bien definidas de tal forma que sea aceptada dicha estructura.

#### Desarrollo de la aplicación en Java

A continuación, se mostrarán los apartados en el proceso de desarrollo de la aplicación en Java, la cual consta de varias funcionalidades a implementar que se explicarán posteriormente.

Para este desarrollo se han tomado como herramienta Eclipse con soporte en el lenguaje de Java, la cual facilita el desarrollo debido a su ya incorporación del lenguaje Java y la

herramienta de gestión de dependencias Maven, necesaria para implementar las dependencias necesarias para este proyecto.

### Objetivo

El objetivo de implementar esta aplicación es el de obtener mediante la lectura o escritura de ficheros XML la estructura del modelo explicado anteriormente para el Almacén.

Así pues, la aplicación:

- Deberá leer el fichero XML y mostrar en consola el contenido que se desee o todo su conjunto.
- Permitirá añadir o eliminar la información contenida en el fichero XML.
- Permitirá exportar el fichero XML a JSON.

### Dependencias

Será necesario el uso de ciertas funciones que permitan el objetivo deseado, así pues, se ha de utilizar el gestor de dependencias Maven, ya implementado en la herramienta de Eclipse, facilitando el desarrollo del proyecto.

Seguidamente se darán a conocer las dependencias utilizadas y el motivo de su uso en esta aplicación.

- **JUnit**: Framework utilizada para la realización de TDD's.
- **Gson**: Librería de Google utilizada para la traducción a lenguaje JSON.
- **JDOM**: Librería usada para el manejo de los archivos XML.



## Bibliografía

*Gson User Guide*. (s.f.). Obtenido de <https://sites.google.com/site/gson/gson-user-guide>

*JDOM: Documentation*. (s.f.). Obtenido de <http://www.jdom.org/downloads/docs.html>

*JSON*. (s.f.). Obtenido de <https://www.json.org/>

*JUnit 5 User Guide*. (s.f.). Obtenido de <https://junit.org/junit5/docs/current/user-guide/>

*Online XML Editor*. (s.f.). Obtenido de [https://www.tutorialspoint.com/online\\_xml\\_editor.htm](https://www.tutorialspoint.com/online_xml_editor.htm)

*Tutorial de XML*. (s.f.). Obtenido de <https://www.abrirllave.com/xml/>

*Tutorial DTD*. (s.f.). Obtenido de <https://www.abrirllave.com/dtd/guion-del-tutorial.php>