## WS Project

This activity intends to stimulate the students to develop a prototype called Mytube using Web Services technologies. This system should allow users to remotely manage digital content information (not the content itself) within a web services system. The contents are already uploaded to a P2P network node.

**Basic Functions [5 points]**

- The web service generates a unique key for the new uploaded digital content if it does not exist in the global directory (WS).
- Store in the global directory (WS) content id, title, description and node.
- Modify content's title/description.
- Delete content.
- Search digital content using the key or search them using textual description title/description/keyword (full search).
- Download digital content information. By Id or listing all the contents in the system.

**Advanced Functions [5 points]**

- Search digital content using textual description (partial search).
- Manage user's access to the contents: sign up, sign in (name and pass). Every content should have an owner.
- List user's contents.
- User can modify/delete just its own content.
- Store and retrieve all of your content's information in a database in your web services server.

**Integration Functions [2.5 points]**

1- Create a new content on the WS after uploading a file to a network node.
2- User should validate its ID before modify/delete contents.
3- Update/delete contents to the WS after doing it in a node.

You are free to implement new features that you consider convenient as long as you maintain the main features and architecture.

## Instructions

Working in pairs or individually and develop the project. There will be a penalization in case of late delivery (after Christmas) up to -2 points.

Deliveries:

- Prepare a **presentation** in English language in order to show the project and its execution. The document should include the following contents:
  1. Describe the classes implemented and the corresponding UML diagram.
  2. Include the endpoints with their operations, resulting codes and representations as in the movie database.
  3. Include a screenshot of each case execution. Describe and explain the outcomes.
  4. Include the GitHub reference to the code. It is important to produce clean and encapsulated code.
  5. You should justify your solution. Remember to include the hours dedicated to this project.

- Execute the main use cases at the end of the presentation in order to show the result of your project.

## Execution

WS DEMO

1- Create four different digital contents, the first and the second with similar description (e.g., First episode of the wire, first episode of the wire HBO tv show)
2- Modify 3rd's description
3- Search a content by its description (full)
4- Delete the fourth one
5- List all contents
6- Create two different users
7- Upload a different content to each previous user
8- Delete the second content uploaded
9- Search content by its description (partial)
10-Show its user
11-Check some ID from an existent user
12-Check some other nonexistent ID

INTEGRATION DEMO with, at least, two different clients/servers

1- Create a new content on the WS after uploading a file to a network node.
2- User should validate its ID before modify/delete contents.
3- Update/delete contents to the WS after doing it in a node.

## References

**The movie database**. https://developers.themoviedb.org/3/getting-started/introduction