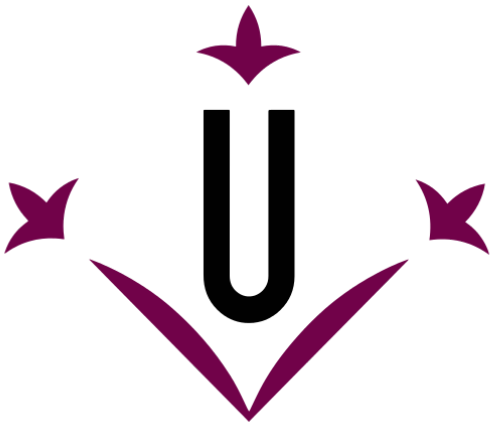

Universitat de Lleida

GRAU EN ENGINYERIA INFORMATICA

Deliverable 1

Basic Web Application 1.0

Bustos Gracia, Marc	49753478Q
Badaoui Mahdad, Faysal	53923695B
Bernárdez Matalobos, Andrés	53197271L
Batlle Gispert, Miquel	48250329D

Delivery Date

31/03/2023

INTRODUCTION

Our goal in this report is to state the servers we have in our web project, the connections and dependencies between them, and also mentioning which of those are required and which are not.

1. Number and function of servers

Our web project is composed by six servers, which interact with eachother to satisfy a function.

- Project: has the capability of creating new tasks and deleting them.
- Department: here we can create a new projcet or delete an already existent one.
- User: an user can sign in or sign up, create or delete projects, and also he can create or delete groups of users
- Company: it can add or delete users for a company, and the same with groups and departments.

Important to know, that all the servers will have the functions to get the main information of each database that corresponds to them.

2. Connections and dependences among them

In case of *Project* and *Task*, the multiplicity is 1 – 0…*, that means that from only one project we can have zero or many tasks, but tasks can not be repeated.

Unlike the rest of connections, we have ternary between *Project*, *Departament* and *Companies*.
It all has 0…*, because zero or more can be associated with 0 or more departaments or 0 or more companies and the same reasoning for the leftovers.

The multiplicity of *User* and *Project* is 0…* in both directions. This is due to, that a project might not be started by a single user, but insted, by a company. Moreover, a project can be handled by more that one user, and at the same time, a user, can be the author of as many projects as he desire.

Between *User* and *Company*, the multiplicity is 1…* – 0 - 1, respectively, meaning that a user can be part of none or one company, and a company can have one or many users. Originally we had thought that, because of a user could be working by himself, the multiplicity should be 0…*. But this does not make sense, considering that this would imply that a company could be composed of no users.

Last but not least, when a *User* deals with the *Database*, the multiplicity is 0…* – 1. We dispose of a single database which can contain a large amount of users.

3. State which are required and which are optional

In the present state of the project, we consider that all the current dependencies are required. Later on during the development of the following deliveries this might change, as we add more dependencies.