



# Deusto

Facultad de Ingeniería  
Ingeniaritza Fakultatea

## DEUZUM

Terres Escudero, Erik B.  
Zarate Jayo, Gorka  
Zugazaga Alonso, Aritz  
Acha Aristegui, Amaia  
Gago López, Leire  
San Millan Langa, Lander

Curso 2019/2020

Ing. Informática  
Programación III

# 1 INTRODUCCIÓN

DeuZum es una aplicación de gestión de dinero online para la gestión de pagos, proyectos o costes grupales. DeuZum ofrece un sistema seguro de transferencias de forma que el cliente no tenga que preocuparse de que este pasando con su dinero en todo momento.

DeuZum estará implementado en un servidor y daría servicio a los usuarios a través de un programa en Windows y una aplicación en Android. Cada una de ellas implementada para conectarse a un servidor central, en el cual se almacenarán los datos, procesarán las peticiones y se garantizará la seguridad de la información.

El servicio que ofrece DeuZum está dividido en diferentes sectores, algunos puramente económicos y otros de análisis de la conducta o de las transacciones. El servidor ejecutara todas las peticiones de los usuarios y devolverá la información que se haya solicitado. Toda la información mientras se envía se encontrara encriptada para la seguridad del usuario.

## 2 IDEAS

## 3 ESTRUCTURA DEL PROYECTO

### 3.1 Apartados

### 3.2 Directorios y Estructura

### 3.3 Implementación del Server

### 3.4 IA

### 3.5 Data Analysis

### 3.6 Bases de Datos I

## 4 FORMA DE TRABAJO

### 4.1 Division del trabajo

Para la realización del proyecto, hemos optado por la división del trabajo en grupos con temáticas separadas. Los dos grupos tendrán que trabajar en una categoría central entre IA y Seguridad, y, trabajar en una parte de la implementación y algoritmia. Las categorías y subcategorías serian las siguientes:

Implementacion	Uso de datos	Seguridad
Servidor	Data Analysis	Database
Windows	IA	Criptografia
Android	Big Data	Proteccion

Cada uno de los grupos tendra un conjunto de tareas las cuales se diviran en *tareas principales* y *tareas secundarias*. Las tareas principales son aquellas a las que se les daran prioridad y buscamos que esten terminadas a la hora de terminar el proyecto. Las tareas secundarias son aquellas que no tienen tanta importancia y que se empezaran una vez se vayan acabando las tareas principales.

A su vez, las tareas estaran numeradas segun importancia, dificultad y tiempo de trabajo con el fin de poder establecer un orden de prioridad para que su progreso haga que el proceso de

construccion del proyecto sea relativamente optimo y de forma que su estructura no se vea afectada por software anterior.

Por ultimo, hay un sector no especializado que trata los temas de algoritmia sobre como manejar la forma de minimizar el total de pagos que hay realizar en forma de peticiones. Este algoritmo estara pensado en forma de algoritmo greedy al tener complejidad  $O(n)$ , pero en caso de que consigamos un algoritmo que no dependa de combinatorial algorithms o complejidades exponenciales para encontrar el minimo numero de pagos, tal vez aniadamos la opcion. Por ahora no sabemos como de rentable resultaria este algoritmo, ni tampoco sabemos ninguna upperbound del numero de movmientos que sea inferior a  $n$  aproximadamente. Esto se mostrara en la bibliografia final o en la documentacion sobre el software.

El apartado de algoritmia se dividira segun convenga, aunque se mostrara dentro de la documentacion quien ha hecho que tareas y como ha resuelto los problemas.

## 4.2 Error Handling

Para evitar errores que afecten al codigo a nivel global, hemos optado por utilizar github y hacer que cada miembro del grupo tenga su propia cuenta. Los errores, comentarios y todo se enviarian a traves de github excepto la informacion mas urgente que se enviaria a traves de un grupo de whatsapp o telegram/discord en si defecto.

## 4.3 Preparacion del temario

Para la preparacion del temario que queremos trabajar en el trabajo hemos optado por utilizar un conjunto de libros que consideramos basicos para las personas que no tienen mucha experiencia y utiles/importantes para las personas que tengan un cierto control en ciertos aspectos. Esta lista de libros se encontrara en el apendice de este documento categorizada.

# 5 PLANIFICACIÓN

## 5.1 8 de octubre E1. Propuesta de proyecto y planificación

Server	<ul style="list-style-type: none"><li>• Conseguir conectar un servidor SQL a el servidor.</li><li>• Hacer que el servidor lea archivos properties.</li></ul>
Desktop	<ul style="list-style-type: none"><li>• Crear la ventana junto con sus componentes.</li><li>• Crear el metodo para enviar sockets.</li></ul>
Android	<ul style="list-style-type: none"><li>• Leer un poco sobre Android</li></ul>
Database	<ul style="list-style-type: none"><li>• Crear la estructura de la base de datos.</li></ul>
Seguridad	<ul style="list-style-type: none"><li>• Haber leído un mínimo sobre RSA, Blowfish, MD5, SHA512, Cesar...</li><li>• Plantear los métodos de encriptación que se usarán para cada tarea dentro del servidor.</li></ul>
IA	<ul style="list-style-type: none"><li>• Plantear que metodos usaremos en el proceso</li><li>• Plantear que tipo de bots queremos usar</li></ul>
Miscelanea	<ul style="list-style-type: none"><li>• Entender como funciona la documentación del proyecto.</li><li>• Entender como funciona el sistema de comandos del servidor.</li><li>• Organizar el repositorio.</li><li>• Estudiar la posibilidad de cambiar la documentación a una wiki.</li></ul>

## 5.2 29 de octubre E2. Fase I del proyecto

Server	<ul style="list-style-type: none"><li>• Estructurar bien el servidor.</li><li>• Todos los mensajes de los sockets deberán ir encriptados.</li></ul>
Desktop	<ul style="list-style-type: none"><li>• Cliente funcional para enviar sockers</li><li>• Cliente visualmente no horrible</li></ul>
Android	<ul style="list-style-type: none"><li>• Cliente funcional para enviar sockers</li></ul> Cliente visualmente no horrible
Database	<ul style="list-style-type: none"><li>• La base de datos debe estar creada y con algunos datos de ejemplo</li></ul>
Seguridad	<ul style="list-style-type: none"><li>• Las contraseñas de los usuarios deben estar encriptadas en algun tipo de hash.</li><li>• Los usuarios deben tener su clave privada para contactar con el servidor</li></ul>
IA	<ul style="list-style-type: none"><li>• Tener un bot construido</li><li>•</li></ul>
Miscelanea	<ul style="list-style-type: none"><li>• .</li><li>• .</li><li>• .</li><li>• .</li></ul>

## 5.3 10 de diciembre E4. Fase II del proyecto

## 5.4 13 de diciembre E5. Informe de desarrollo

# 6 MATERIAL

# 7 LIBRERÍAS