

**Serviço de Autenticação Distribuído**

**M**estrado em **E**ngenharia **I**nformática

**Laboratórios de Engenharia Informática**

(4º ano, 2º semestre)

**A78322** André Filipe Ferreira de Mira Vieira

**A77048** Eduardo Gil Ribeiro Rocha

**A78764** Ricardo André Araújo Neves

**Data**:

6 de maio de 2019

Índice

[Introdução 3](#_Toc8050680)

[Solução e Objetivos 4](#_Toc8050681)

[Diagrama de Gantt 6](#_Toc8050682)

[OAuth 2.0 7](#_Toc8050683)

[Implementação 9](#_Toc8050684)

[Conclusões e Trabalho Futuro 9](#_Toc8050685)

# Introdução

Este trabalho prevê a produção de um serviço de autenticação distribuído, no âmbito da Unidade Curricular de Laboratórios de Engenharia Informática. Este foi um das dezenas de projetos apresentados pelos coordenadores da UC, sendo que o grupo, depois de uma reunião e de troca de ideias, decidi-mos escolher este devido ao desafio de criar o sistema proposto. Foi um dos temas disponíveis que nos suscitou uma maior atenção, pelo que falamos com o coordenador deste projeto, aceitando este desafio.

Como sabemos, os utilizadores precisam de memorizar um par *username* / *password* para cada plataforma que utilizam. A nossa conta GitHub tem um par de credenciais, a conta Twitter, possivelmente, tem outro par diferente, e assim em diante para todas as contas em nossa posse. Isto, como é obvio, não é a abordagem mais correta, por uma questão de segurança e comodidade. Quantas vezes já nos esquecemos do nosso par de credenciais de um certo website, sendo que tivemos de gastar mais uns minutos a repor estes?

Pois bem, isto deve-se ao facto de cada serviço ou aplicação desenvolve o seu próprio sistema de autenticação e autorização, o que é perfeitamente natural, uma vez que são implementados por diferentes entidades. No entanto, o nosso serviço de autenticação vem solucionar este problema, pelo que falaremos um pouco mais sobre os nossos objetivos em seguida.

# Solução e Objetivos

Assim, podemos afirmar que o nosso objetivo para este trabalho prático será criar um sistema central que seja capaz de trabalhar e gerir a autenticação dos futuros utilizadores.

Deste modo, os utilizadores do serviço apenas precisarão de memorizar um único par de credenciais, qualquer que seja o número de aplicações vinculadas ao nosso sistema. Quer o nosso serviço agrupe as autenticações de 2, 20 ou 200 aplicações distintas, o utilizador apenas irá ter de se relembrar das credenciais que usou para criar o seu registo na plataforma.

No entanto, pretendemos facilitar ainda mais a tarefa de login numa determinada aplicação. Assim, a ideia é possibilitar o utilizador a conectar uma ou mais contas das suas redes sociais. Este aspeto é bastante interessante devido ao facto de, com isto, o utilizador pode efetuar o seu login na plataforma através dos atalhos disponíveis.

Com isto, o login no serviço passa a ser possível com apenas um clique, não sendo necessário ao utilizador recordar e escrever o seu *username* e *password* registados.

O seguinte esquema traduz, de uma forma bastante simples e direta, o que foi retratado até aqui:

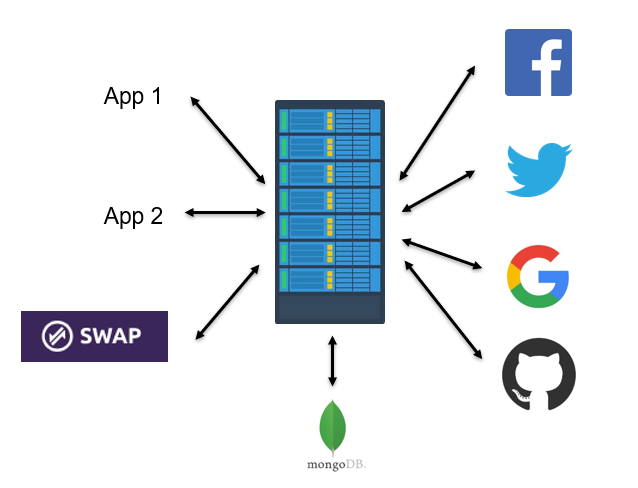


Figura - Esquema do projeto

Como podemos observar, do lado esquerdo temos as aplicações que se podem vincular ao nosso serviço. A plataforma SWAP, por exemplo, da Universidade do Minho é uma potencial aplicação que pode usufruir das vantagens da nossa implementação.

Do outro lado, temos as 4 das maiores redes sociais, utilizadas quase diariamente por estudantes, e não só: o Facebook, o Twitter, o Google e o GitHub. Escolhemos estas 4 entidades porque, como já referimos, são redes sociais gigantes e qualquer possível utilizador da nossa aplicação tem, pelo menos, uma conta numa destas redes. Para além deste facto, todas estas redes sociais usam o protocolo OAuth para a autenticação dos utilizadores. Iremos debruçar-nos sobre este protocolo OAuth mais à frente neste documento.

No fim, no rodapé da imagem, está presente o “mongoDB”, uma base de dados não relacional, onde iremos armazenar e gerir os dados necessários.

# Diagrama de Gantt

Diagrama de Gantt é um gráfico utilizado para ilustrar os intervalos de tempo relativos ao início e à finalização das diferentes etapas de um projeto.

Deste modo, o grupo debateu e desenvolveu o seguinte Diagrama de Gantt, de forma a funcionar como uma ferramenta de controlo relativamente ao trabalho realizado, como o dia de incio, de fim e a duração de uma determinada fase do mesmo.

Por aqui, podemos observar que o grupo começou a debater e discutir o problema a abordar, sendo que prevemos que o projeto irá se prolongar até meados do mês de junho. A etapa mais trabalhosa e que dependeu de mais tempo para ser concluída foi a implementação da associação de contas, prevendo que iria demorar cerca de 46 dias.

Todas estas informações podem ser encontradas, então, no seguinte Diagrama de Gantt:

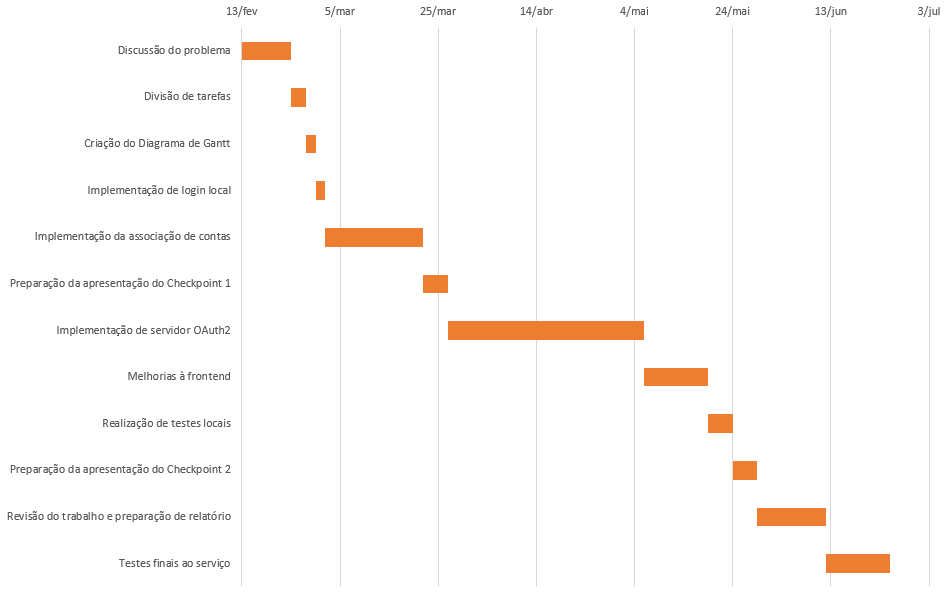


Figura - Diagrama de Gantt

# OAuth 2.0

O *OAuth 2.0* é uma estrutura de autorização, lançada em outubro de 2012, que permite que as aplicações obtenham acesso limitado às contas dos usuários em serviço HTTP, tendo como os exemplos mais conhecidos o Google e o Facebook, sem que estes precisem de expor as suas credenciais.

Podemos até usar uma analogia para especificar mais facilmente este protocolo: imaginemos que temos um carro de luxo, sendo que, com este, uma “valet key”. Esta chave pode ser utilizada para quando queremos que um arrumador de carros estacione o nosso automóvel. No entanto, por se tratar de uma chave especial, este manobrista só pode dirigir no carro até uma distância de 2 quilómetros, digamos. Assim, tiramos daqui a ideia que, se queremos fornecer a alguém um acesso limitado, usamos a chave especial; se queremos um acesso total e sem limitações, utilizamos a nossa chave normal.

O *OAuth* define 4 papéis principais: proprietário do recurso (utilizador), cliente (aplicação que requisita recursos protegidos, através da autorização do utilizador), servidor de autenticação (emite *tokens* de acesso ao cliente, depois da sua autenticação e obtenção de autorização) e servidor de recurso (hospeda os recursos protegidos).

OAuth é um protocolo de autorização que permite que as aplicações obtenham acesso limitado às contas de usuários em um serviço HTTP.

O nosso sistema irá usar o protocolo OAuth 2.0 para enviar tokens para clientes (aplicações terceiras).

O OAuth 2.0 tem várias melhorias sobre o OAuth 1, em segurança e número de funcionalidades.

O Oauth é um protocolo de autorização que possibilita às aplicações obter acesso limitado à conta do usuário que pretende aceder a um determinado serviço.

Por acesso limitado, queremos dizer que a aplicação apenas tem acesso a informação pessoal, como o nome, o email ou a fotografia, se for necessário, e nunca a acesso privado, ou seja, a password do utilizador para aquele serviço.

Assim, temos o objetivo de usar o protocolo Oauth 2.0 no nosso sistema, de modo a enviar tokens para os clientes, as aplicações terceiras.

Depois de uma breve pesquisa, chegamos à conclusão que esta seria a melhor abordagem, uma vez que o Oauth 2.0 apresenta várias melhorias relativas ao seu antecessor, o Oauth 1, em termos de segurança e em número de funcionalidades.

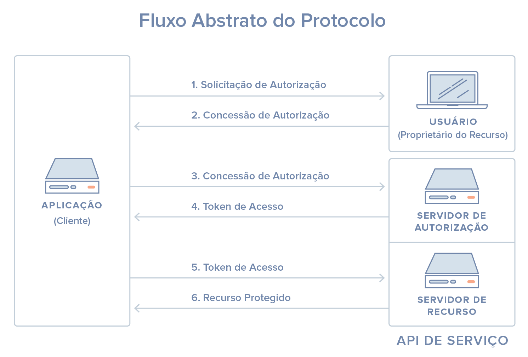


Figura - Esquema de Fluxo do OAuth 2.0

-- EXPLICAR O DIAGRAMA ACIMA –

# Implementação

# Conclusões e Trabalho Futuro