



Instituto Superior de Engenharia de Lisboa  
Licenciatura em Engenharia Informática e de  
Computadores

## **Proposta de Projeto Final i-on ClassCode**

André David dos Santos  
933 507 757 - A48309@alunos.isel.pt

Ricardo de Freitas Henriques  
926 945 808 - A48322@alunos.isel.pt

João Diogo da Anunciação Magalhães  
914 750 039 - A48348@alunos.isel.pt

Docente:  
Pedro Miguel Henriques Santos Félix  
pedro.felix@isel.pt

20 de março de 2023

# 1 Introdução

O **i-on ClassCode** tem como objetivo a disponibilização de um sistema para a criação e gestão de repositórios **GitHub**[1], para uso em contexto académico, de forma a melhorar a experiência de utilização e o aumento de produtividade para docentes e alunos.

O projeto **i-on ClassCode** aplica a experiência dos dois projetos anteriores, **i-on CodeGarten**[2] e **Teams**[3], através da melhoria de lacunas da aplicação web, bem como o desenvolvimento de uma aplicação móvel, como ferramenta pessoal do docente.

Na área da engenharia informática é comum as instituições académicas usarem a plataforma **GitHub** para o armazenamento e partilha de projetos através de repositórios. A ferramenta **GitHub Classroom**[4] é usada também em contexto académico para a criação de repositórios associados a grupos de alunos ou a trabalhos individuais.

## 2 Contexto

Em ambiente académico, os docentes usam os repositórios **GitHub** como espaço para os alunos poderem entregar e guardar as suas cópias locais dos trabalhos tendo ao seu dispor um mecanismo de controlo de versões. Os repositórios fornecem ainda a possibilidade de visualizar o progresso, contribuições de cada aluno e entregas através do sistema de criação de *tags*.

O **GitHub Classroom** é a ferramenta que é usada atualmente para estabelecer os grupos de alunos através da criação de *teams* e associar esses grupos a um repositório de trabalho. É também usada como ferramenta para a identificação dos alunos e fornece capacidade de *feedback* entre o docente e os grupos de alunos.

## 3 Problema

O **GitHub Classroom** apesar de ser útil apresenta limitações. É necessário a criação e configuração de permissões de novos repositórios para os vários grupos de alunos em todas as novas execuções de uma unidade curricular e a respetiva associação à turma correspondente, ao novo ano letivo e a uma *team*. Outros problemas do **GitHub Classroom** são a identificação institucional do aluno através da conta **GitHub** e o processo lento de *cloning* dos repositórios para a máquina local do docente durante o processo de avaliação. Relacionado ainda com os alunos, a existência de *feedback* entre o docente e os grupos de alunos é bastante reduzida.

## 4 Solução

O **i-on ClassCode** disponibilizará um serviço que expõe uma **API HTTP** que será usada por duas aplicações *frontend*, *web* e *mobile*, de modo a fornecer suporte para os problemas existentes. O sistema será responsável por aceder e manipular uma base de dados com finalidade de gerir e guardar os dados do serviço. De forma a aceder a dados e operações externas relacionadas com o **GitHub**, o serviço irá aceder à **API** pública do **GitHub**. Como poderão ser realizadas ações utilizando a interface **GitHub** e a **API HTTP** existirão duas fontes de informação sendo assim necessário sincronizar os dados do serviço com a **API** do **GitHub** e vice-versa.

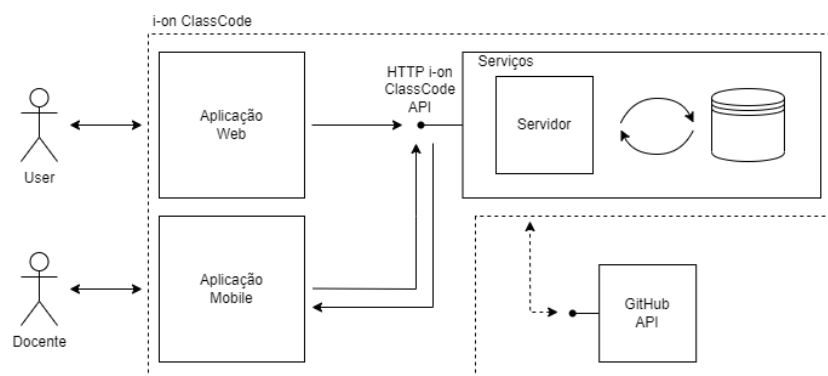


Figura 1: Arquitetura de alto nível

Em relação às tecnologias adotadas a **API HTTP** será desenvolvida utilizando a framework **Spring**[5]. A base dados relacional será criada usando **PostgreSQL**[6] e a ligação à mesma utilizará a biblioteca **JDBI**[7]. As respostas da **API** utilizarão **Hypermedia** e o seu formato irá adotar a especificação **Siren**[8]. A aplicação web será desenhada com a biblioteca **React**[9] e a linguagem **TypeScript**[10]. A aplicação Mobile será criada para o ambiente **Android**[11] e fará uso do *toolkit* **Jetpack Compose**[12].

## 5 Desafios e Riscos

Os desafios e riscos associados ao desenvolvimento do sistema será a dependência do serviço com um sistema externo, neste caso a **API** do **GitHub**, a existência de dados de utilizador sensíveis, devido ao facto da **API** do **GitHub** utilizar tokens de acesso capazes de operações destrutivas e o uso da tecnologia **Jetpack Compose**, uma vez que esta tecnologia é uma tecnologia com atualizações recentes tal como as versões **Android**.

## 6 Plano Semestral do Projeto

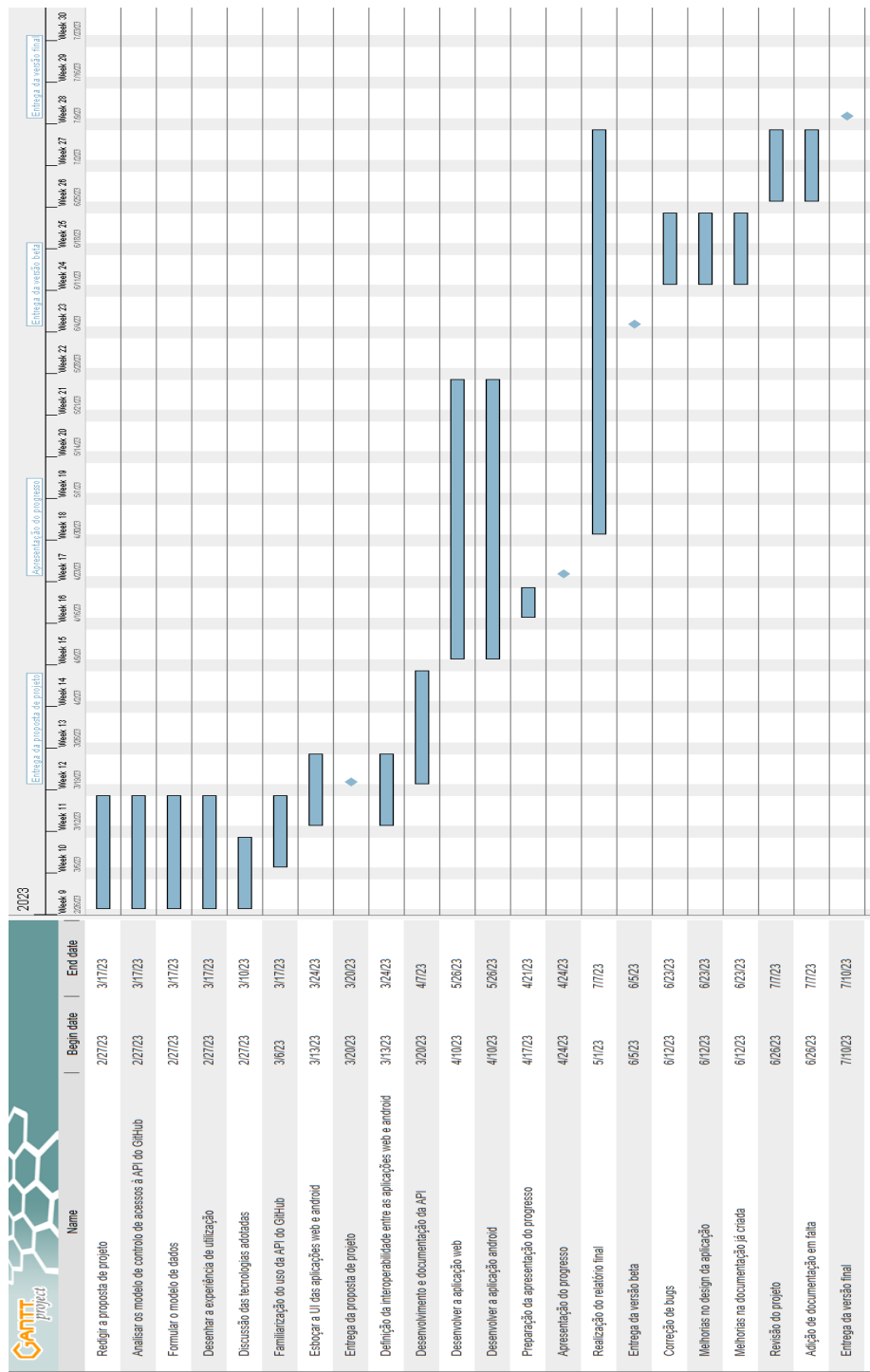


Figura 2: Plano semanal de desenvolvimento

## Referências

- [1] GitHub. (2008), URL: <https://github.com> (acedido em 02/03/2023).
- [2] Diogo Sousa, Tiago David e João Moura. «i-on CodeGarten.» (2021), URL: <https://github.com/i-on-project/codegarten> (acedido em 02/03/2023).
- [3] Afonso Machado e Martim Francisco. «i-on Teams.» (2022), URL: <https://github.com/i-on-project/teams> (acedido em 02/03/2023).
- [4] GitHub Classroom. (2015), URL: <https://classroom.github.com/> (acedido em 02/03/2023).
- [5] Spring. (2003), URL: <https://spring.io/> (acedido em 04/03/2023).
- [6] PostgreSQL. (1996), URL: <https://www.postgresql.org/> (acedido em 16/03/2023).
- [7] JDBC. (2011), URL: <https://jdbc.org/jdbi2/> (acedido em 04/03/2023).
- [8] Siren. (2017), URL: <https://github.com/kevinswiber/siren> (acedido em 04/03/2023).
- [9] React. (2013), URL: <https://reactjs.org/> (acedido em 04/03/2023).
- [10] Microsoft Corporation. (2012), URL: <https://www.typescriptlang.org/> (acedido em 16/03/2023).
- [11] Various (mostly Google). (2008), URL: <https://www.android.com/> (acedido em 16/03/2023).
- [12] Jetpack Compose. (2022), URL: <https://developer.android.com/jetpack/compose> (acedido em 07/03/2023).