



I-on ClassCode

Licenciatura em Engenharia Informática e de Computadores
Descrição da Organização, Utilização e Instalação do Projeto

André David dos Santos
A48309@alunos.isel.pt

Ricardo de Freitas Henriques
A48322@alunos.isel.pt

João Diogo da Anunciação Magalhães
A48348@alunos.isel.pt

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

10 de julho de 2023

Índice

1	Repositório GitHub e Organização	1
2	Manual de utilização	1
3	Disponibilização do projeto	1
4	Instalação local	2
4.1	Pré-requisitos	2
4.2	Variáveis de Ambiente	2
4.3	Sistema de contentores Docker	3
5	Execução dos Testes Automáticos	4
5.1	Serviço Backend	4
5.2	Aplicação Web	4

1 Repositório GitHub e Organização

O código desenvolvido bem como a documentação do projeto **i-on ClassCode** encontram-se disponíveis publicamente no repositório [Github](#). A entrega da fase final do projeto foi marcada no repositório com a *tag* 1.0.0.

2 Manual de utilização

O documento referente às várias funcionalidades e à utilização do sistema do projeto **i-on ClassCode** encontra-se disponível tanto em inglês como em português na pasta [docs](#) do repositório **Github**.

3 Disponibilização do projeto

A disponibilização do sistema **i-on ClassCode** foi feita utilizando a Google *Cloud Platform* (GCP) e o sistema de contentores Docker, e está disponível [aqui](#). A disponibilização da **Aplicação Móvel** foi feita usando o arquivo *apk* gerado pelo *Android Studio* e a instalação da aplicação é feita usando o *blob storage* da GCP, *Cloud Storage*. O *download* da **Aplicação Móvel** está disponível na **Aplicação Web** para os professores na página principal.

A disponibilização do sistema **i-on ClassCode** pode não estar sempre disponível devido ao facto de os créditos da GCP serem limitados.

4 Instalação local

A instalação dos vários componentes do projeto e como proceder à sua utilização, encontram-se documentadas em:

- [Aplicação JVM e Web](#)
- [Aplicação Mobile](#)

4.1 Pré-requisitos

Antes de qualquer instalação é necessário fazer *clone* do repositório **GitHub** para uma diretoria da máquina local. Em seguida é necessária a instalação de uma base de dados **PostgreSQL** na máquina local. O URL associado à base de dados deverá ser configurado como variável de ambiente (JDBC_DATABASE_URL). A base de dados deverá ser passível de ser acedida publicamente pelas instâncias do **Serviço Backend** em execução nos contentores Docker. Os *scripts* SQL que deverão ser executados para a criação da base de dados são [createTables.sql](#) e [triggers.sql](#).

Para além da instalação da base de dados deve-se proceder à instalação do ambiente de execução [Node.js](#).

Por fim, é necessário a instalação da aplicação **Ngrok** e a criação de uma [conta Ngrok](#). Após ser realizado o registo será atribuído um *AuthToken* à conta. Para conectar a aplicação Ngrok à conta registada, em primeiro lugar, deverá ser aberta a aplicação e em seguida deverá ser inserido o comando 'ngrok config add-authtoken [*AuthToken*]'. Por fim, para utilizar a ferramenta Ngrok juntamente com o projeto desenvolvido deverá ser utilizado sempre o comando 'ngrok http 80' ou 'ngrok http 443'. Após a execução do comando surgirá o URL de *forwarding* que deverá ser configurado como variável de ambiente (NGROK_URI) na máquina local. Em qualquer nova execução da aplicação deverá ser reconfigurada a variável de ambiente. Se necessário poderá ser consultado o seguinte [guia](#). Para carregar a **Aplicação Web** deverá ser inserido no *browser* o URL de *forwarding*.

4.2 Variáveis de Ambiente

As variáveis de ambiente necessárias ao funcionamento dos vários componentes do projeto são:

- [Aplicação JVM](#)
- [Aplicação Web](#)
- [Aplicação Mobile](#)

Durante o desenvolvimento do projeto os valores das chaves usadas foram os seguintes:

- CLASSCODE_ENCRYPTION_KEY:
n2r5u8x/A%D*G-KaPdSgVkYp3dd90pdB
- SENDGRID_CLASSCODE_API_KEY:
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

Os valores das variáveis de ambiente `CLIENT_ID` e `CLIENT_SECRET` devem corresponder às credenciais de uma aplicação OAuth **GitHub** que tenha configurada como URL de *callback* o URL de *forwarding* Ngrok. Um guia para a criação e configuração de uma aplicação OAuth **GitHub** pode ser encontrado [aqui](#).

No contexto da **Aplicação Mobile** existe uma única variável de ambiente com o nome `NGROK_URI`, que deve ser definida no ficheiros de propriedades local *local.properties* localizado na raiz do projeto *android*. O arquivo deve incluir a propriedade *sdk.dir*, que é o caminho para o Android SDK. O valor da variável `NGROK_URI` deve também corresponder ao URL de *forwarding* Ngrok.

4.3 Sistema de contentores Docker

A disponibilização do **Serviço Backend** bem como da **Aplicação Web** foi feita através do uso do sistema de contentores Docker e da ferramenta Docker Compose. A ferramenta Docker Compose permite definir múltiplas instâncias de serviços docker responsáveis pela execução dos componentes do sistema **i-on ClassCode** em múltiplos contentores. Para disponibilizar tanto o **Serviço Backend** e a **Aplicação Web** criou-se um ficheiro yaml, `docker-compose`, que define dois serviços docker. O ficheiro `docker-compose` é responsável por criar:

- uma ou múltiplas instâncias do **Serviço Backend**.
- uma instância do NGINX para servir a **Aplicação Web** e aproveitar as múltiplas instâncias do **Serviço Backend** para balanceamento de carga.

Para iniciar os vários contentores docker deve ser executado o comando `'docker-compose up'` ou `'gradlew composeUp'`. Caso se pretenda criar múltiplas instâncias do **Serviço Backend** é possível substituir o comando `'gradlew composeUp'` pelo comando `'docker-compose up -build -scale spring-service=[número de instâncias]'`.

5 Execução dos Testes Automáticos

O processo de execução dos testes encontra-se documentado em:

- [JVM](#)
- [JS](#)

Para a execução dos testes do sistema é necessário alterar os valores das variáveis de ambiente `CLIENT_ID` e `CLIENT_SECRET` para que o **Serviço Backend** utilize uma aplicação OAuth GitHub de teste que utilize o endereço *localhost* como domínio. Além disso deve ser eliminada a variável de ambiente `NGROK_URI`.

5.1 Serviço Backend

Para a execução dos testes do **Serviço Backend** deve-se executar o comando `'gradlew test'` na diretoria `code/jvm`. É necessário também que a base de dados se encontre vazia e que sejam executados os *scripts* SQL [createTables.sql](#), [triggers.sql](#) e [insert.sql](#).

5.2 Aplicação Web

Para a execução dos testes automáticos correspondentes à aplicação *frontend* é necessário que pelo menos uma instância do **Serviço Backend** se encontre em execução e a criação de quatro utilizadores numa base de dados vazia através da execução do *script* SQL [script.sql](#) que se encontra na pasta `code/js/tests`. Estes utilizadores correspondem a um professor que se encontra registado, um professor em processo de validação, um estudante registado e um estudante ainda pendente que se encontra na fase de inserção do seu identificador escolar. Os *usernames* e *passwords* de cada uma das contas email registadas no **GitHub** são:

- Teacher: `i-on-classcode@outlook.pt` / `ClassCodeTeacher`
- Student: `i-on-classcode-teacher@outlook.pt` / `ClassCodeStudent`
- PendingTeacher: `i-on-classcode-pending-teacher@outlook.pt` / `ClassCodePendingTeacher`
- PendingStudent: `i-on-classcode-pending-student@outlook.pt` / `ClassCodePendingStudent`