

# APLIÇÕES DISTRIBUIDAS PELA INTERNET

MEEC

---

School resources management system

---

**Authors:**

Miguel Rocha (93138)  
José Graça (96256)

[miguel.a.rocha@tecnico.ulisboa.pt](mailto:miguel.a.rocha@tecnico.ulisboa.pt)  
[josemendesgraca@tecnico.ulisboa.pt](mailto:josemendesgraca@tecnico.ulisboa.pt)

Group 8
---------

2025/2026 – 1<sup>st</sup> Semester, P1

# Contents

<b>1</b>	<b>Portos e Endpoints</b>	<b>2</b>
<b>2</b>	<b>Funções Implementadas</b>	<b>3</b>
2.1	QRCodegenerator . . . . .	3
2.2	FoodService / FoodAdminApp . . . . .	3
2.2.1	FoodService . . . . .	3
2.2.2	FoodAdminApp . . . . .	3
2.3	RoomService / RoomAdminApp . . . . .	4
2.3.1	RoomService . . . . .	4
2.3.2	RoomAdminApp . . . . .	4
2.4	Check-inApp . . . . .	4
2.5	MessageApp . . . . .	5
<b>3</b>	<b>Uso de AI</b>	<b>5</b>

# 1 Portos e Endpoints

## 8001 - QRCodeGenerator

- / (POST) – gera e devolve a imagem do QRCode (PNG)

## 5000 - FoodApp

- /restaurant/<int:restId> – página do restaurante de id *restId*
- /rate (POST) – submeter form de avaliação
- /api – API endpoint para admin app

## 5003 - Food admin

- /add – página de adicionar restaurantes
- /addRestaurant (POST) – submeter o form de adicionar restaurante
- /update – página de atualizar menu
- /update.menu – submeter o form de atualizar o menu

## 5001 - RoomApp

- /submit.room (POST) – submeter o form de adicionar sala
- /update.schedule (POST) – submeter o form de atualizar horário
- /api – API endpoint para admin app

## 5007 - Room admin

- /createRoom – página de criação de sala
- /create – submeter o form de adicionar sala
- /list – página de lista de salas
- /updateSchedule – página de atualizar horário

## 8000 - Checkin App

- /checkin (POST) – submeter registo de check-in
- /checkout (POST) – submeter registo de check-out
- /listcheck (POST) – apresenta lista de check-ins/outs do utilizador

## 5010 - MessageApp

- /send (POST) – submeter envio de nova mensagem
- /inbox (GET/POST) – listar mensagens (todas ou filtradas)

## 2 Funções Implementadas

### 2.1 QRCodegenerator

Nesta aplicação, foram implementadas as seguintes funcionalidades:

- Gera códigos QR a partir de texto introduzido pelo utilizador.
- Apresenta o QRCode diretamente no navegador.

### 2.2 FoodService / FoodAdminApp

#### 2.2.1 FoodService

Nesta aplicação, foram implementadas as seguintes funcionalidades:

- No **index** mostra uma lista de restaurantes com hipertexto para a página do restaurante.
- Nas páginas dos restaurantes mostra o menu e o rating e permite avaliar o restaurante calculando a média das avaliações.
- Para navegar para o **index** basta clicar no título.

A base de dados está estruturada da seguinte maneira:

- **ID**: chave primária
- **Nome**: string
- **Menu**: string
- **Rating**: float
- **Number of ratings**: float

O menu pode ser alterado para acumodar mais prato ou espaços no futuro.

#### 2.2.2 FoodAdminApp

Nesta aplicação, foram implementadas as seguintes funcionalidades:

- Adicionar restaurante, com filtro de repetição, atualizando na base de dados da *Food App* através do proxy.
- Atualizar o menu, com deteção de restaurante não existente, também por proxy.
- Ver lista de restaurantes.
- Ver os ratings.

## 2.3 RoomService / RoomAdminApp

### 2.3.1 RoomService

Esta aplicação apenas mostra a lista de salas e horários.

A base de dados tem a seguinte estrutura:

- **ID:** primary key
- **Name:** String
- **Capacity:** Integer
- **Schedule:** String

Dependendo das necessidades da segunda parte, o horário poderá ser mudado para um código semelhante a "xxxoooxxxxxx" para acomodar mais precisão no horário.

### 2.3.2 RoomAdminApp

Nesta aplicação, foram implementadas as seguintes funcionalidades:

- Criar uma sala, com deteção de repetição, por proxy.
- Mostrar a lista de salas.
- Alterar horários, com deteção de não existência, por proxy.

## 2.4 Check-inApp

Nesta aplicação, foram implementadas as seguintes funcionalidades:

- Permite registar um *check-in* numa sala/local.
- Permite registar um *check-out*.
- Lista o histórico de check-ins e check-outs de um utilizador.

A base de dados tem a seguinte estrutura:

- **ID:** Primary Key (Integer)
- **Username:** String – identifica o utilizador
- **Location:** String – local/sala associado ao check-in/out
- **Spec:** String – indica o tipo de registo (*Checked In* ou *Checked Out*)
- **Date:** Date – data do registo

## 2.5 MessageApp

Nesta aplicação, foram implementadas as seguintes funcionalidades:

- Permite enviar mensagens entre utilizadores.
- Lista mensagens por remetente
- Disponibiliza uma API *XML-RPC* para operações remotas de mensagens. (Para uso futuro)

A base de dados tem a seguinte estrutura:

- **ID:** Primary Key (Integer)
- **Sender:** String – remetente da mensagem
- **Receiver:** String – destinatário da mensagem
- **Content:** String – conteúdo da mensagem
- **Created\_at:** DateTime – data e hora de criação da mensagem, valor por omissão é o tempo atual (UTC).

## 3 Uso de AI

Miguel: A minha parte do trabalho foi as rooms e a food e parte das mensagens, portanto usei o copilot para adaptar as funções da food para a rooms e um pouco para a message, para tudo o que seja estético e para apanhar formatos de mensagens entre servidores e clientes.

José: Fiz o QRCodeGenerator e também a CheckInApp e implementei uma parte final da MessageAPP, tendo usado ChatGPT apenas para corrigir o ocasional erro de HTML e para tornar as páginas web mais visualmente apelativas.