



Licenciatura em Informática – 2º Ano

Unidade Curricular – Desenvolvimento Web 1

Tema do trabalho – Base de Dados para uso numa cadeia de hotéis

Grupo - Inf24dw1g32

Alunos:

António Manuel Ferreira Lopes dos Santos Filipe – nº 044351

João Pedro Freitas Gomes – nº045235

Para o desenvolvimento do trabalho relativo ao M2, iremos utilizar a Base de Dados utilizada no M2. Iremos também adotar uma abordagem Code-first para a construção de serviços web baseados numa arquitetura REST. Para tal, utilizaremos a framework Loopback4 para o desenvolvimento expedito da API e a biblioteca React-Admin para a criação de um cliente web de backoffice.

A Base de Dados tem 9 tabelas (hotel, regime, quarto, hospede, reserva, tipo_quarto, reserva_hospede, restaurante e serviço), para o trabalho, iremos desenvolver os seguintes recursos, utilizando os métodos GET, POST, PUT e DELETE.

Hotels	/api/hotels	GET	List all hotels
	/api/hotels/{id}	GET	Get details of a specific hotel
	/api/hotels	POST	Create a new hotel
	/api/hotels/{id}	PUT	Update data of a specific hotel
	/api/hotels/{id}	DELETE	Delete a hotel
Guests	/api/guests	GET	List all guests
	/api/guests/{id}	GET	Get details of a specific guest
	/api/guests	POST	Create a new guest
	/api/guests/{id}	PUT	Update data of a specific guest
	/api/guests/{id}	DELETE	Delete a guest
Reservations	/api/reservations	GET	List all reservations
	/api/reservations/{id}	GET	Get details of a specific reservation
	/api/reservations	POST	Create a new reservation
	/api/reservations/{id}	PUT	Update data of a specific reservation
	/api/reservations/{id}	DELETE	Delete a reservation
	/reservations/report	GET	Get reservations grouped by guest, hotel and room type
Rooms	/api/rooms	GET	List all rooms
	/api/rooms/{id}	GET	Get details of a specific room
	/api/rooms	POST	Create a new room
	/api/rooms/{id}	PUT	Update data of a specific room
	/api/rooms/{id}	DELETE	Delete a room
Room Status	/room-status	GET	Get available rooms in a hotel for a specific date

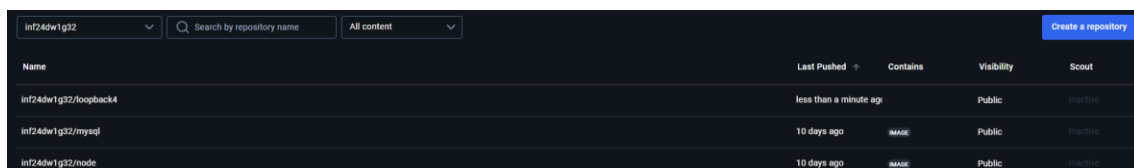
A API será documentada no formato OpenAPI 3.0, utilizando os recursos da framework Loopback4. Além disso, será criada uma collection Postman, que permitirá a consulta e teste dos endpoints.

O cliente de backoffice será desenvolvido com a biblioteca React-Admin. A aplicação será configurada para um ambiente multi-container, utilizando Docker, com pelo menos duas imagens: uma para o backend Node.js e outra para a base de dados MySQL.

Configurações

Link GitHub da organização - <https://github.com/inf24dw1g32/M3.git>

DockerHub configurado com o Username – inf24dw1g32 e criados três repositórios (mysql, node, loopback4)

A screenshot of the Docker Hub interface for the user 'inf24dw1g32'. The page shows a list of three repositories: 'inf24dw1g32/loopback4', 'inf24dw1g32/mysql', and 'inf24dw1g32/node'. Each repository entry includes a 'Name' column, a 'Last Pushed' column with a timestamp, a 'Contains' column with a 'IMAGE' icon, a 'Visibility' column set to 'Public', and a 'Scout' column with a 'Inactive' status.

Name	Last Pushed	Contains	Visibility	Scout
inf24dw1g32/loopback4	less than a minute ago		Public	Inactive
inf24dw1g32/mysql	10 days ago	IMAGE	Public	Inactive
inf24dw1g32/node	10 days ago	IMAGE	Public	Inactive