

Memorial Descritivo

O projeto consiste na modelagem de um sistema de controle de acesso e monitoramento de capacidade em academias. A proposta busca otimizar o registro de check-ins, gestão de clientes e administração de unidades, permitindo maior controle sobre a lotação em tempo real e o gerenciamento de usuários administrativos.

Justificativa

Com a crescente necessidade de controlar a ocupação de ambientes, especialmente no setor fitness, este sistema visa substituir métodos manuais e sistemas legados de check-in. O modelo adotado foi desenhado de forma a garantir rastreabilidade dos eventos, simplicidade estrutural e escalabilidade para múltiplas unidades dentro de uma academia.

Objetivos

1. Estruturar um banco de dados com cinco entidades centrais: **Cliente**, **Academia**, **Usuário**, **Unidades** e **Checkin**.
2. Garantir que o processo de **check-in** seja vinculado de forma clara ao cliente e à unidade frequentada.
3. Permitir que a **Academia** **gerencie múltiplas unidades**, mantendo controle sobre a ocupação de cada uma.
4. Segregar as responsabilidades administrativas através da entidade **Usuário**.

Descrição das Entidades e Seus Campos

Entidade: Cliente

Representa a pessoa física que frequenta a academia.

- **id (ObjectId, obrigatório):** Identificador único do cliente.
- **nome (String, obrigatório):** Nome completo.
- **email (String, obrigatório):** Principal meio de contato.
- **telefone (String, opcional):** Telefone secundário.
- **endereço (String, opcional):** Endereço do cliente.

Entidade: Academia

Representa a organização principal, podendo ter diversas unidades.

- **id (ObjectId, obrigatório):** Identificador único da academia.
- **nome (String, obrigatório):** Nome da academia.
- **telefone (String, opcional):** Telefone de contato.
- **usuario_id (ObjectId, obrigatório):** Referência ao usuário responsável (dono/administrador).

- **endereco (String, obrigatório):** Localização principal da academia.

Entidade: Usuário

Perfis administrativos com acesso ao sistema de gestão.

- **id (ObjectId, obrigatório):** Identificador único do usuário.
- **username (String, obrigatório):** Nome de login único.
- **senha (String, obrigatório):** Senha do usuário (armazenada com hash).
- **role (String, obrigatório):** Nível de permissão (ex.: Admin, Gerente, Recepcionista).

Entidade: Unidades

Subdivisões físicas da academia principal, onde ocorrem os check-ins.

- **academia_id (ObjectId, obrigatório):** Identificador da academia a que pertence.
- **nome (String, obrigatório):** Nome da unidade.
- **endereco (String, obrigatório):** Endereço físico da unidade.
- **current_occupancy (Integer, obrigatório):** Número atual de clientes presentes.
- **telefone (String, opcional):** Telefone da unidade.

Entidade: Checkin

Registra o evento histórico de entrada de um cliente em determinada unidade.

- **id (ObjectId, obrigatório):** Identificador único do check-in.
- **cliente_id (ObjectId, obrigatório):** Referência ao cliente que realizou o check-in.
- **unidade_id (ObjectId, obrigatório):** Referência à unidade onde ocorreu o check-in.
- **checkin_time (Date, obrigatório):** Data e hora exata do check-in.
- **is_active (Boolean, opcional):** Indica se o cliente ainda está presente (true) ou já saiu (false).

Justificativa das Relações e Cardinalidades

Relação 1: Academia ↔ Usuário

- **Tipo:** Composição.
- **Justificativa:** O Usuário (administrador/dono) pode gerenciar várias academias, mas seu ciclo de vida é independente delas.
- **Cardinalidade:** 1 Usuário gerencia 0..* Academias.

Relação 2: Academia ↔ Unidades

- **Tipo:** Composição.
- **Justificativa:** As unidades dependem da academia para existir. Se a academia for excluída, todas as suas unidades também devem ser removidas.
- **Cardinalidade:** 1 Academia possui 1..* Unidades.

Relação 3: Cliente ↔ Academia

- **Tipo:** Associação.
- **Justificativa:** Um Cliente pode estar vinculado a diversas academias (por exemplo, ser aluno de várias redes). A academia mantém a lista de clientes associados.
- **Cardinalidade:** 0..* Clientes ↔ 0..* Academias.

Relação 4: Cliente ↔ Checkin

- **Tipo:** Composição.
- **Justificativa:** O Checkin só faz sentido no contexto do Cliente. Se o Cliente for removido, todos os seus check-ins devem ser apagados.
- **Cardinalidade:** 1 Cliente pode ter 0..* Check-ins.

Relação 5: Unidade ↔ Checkin

- **Tipo:** Composição.
- **Justificativa:** O Checkin só existe se estiver vinculado a uma unidade específica. Sem a unidade, o evento perde o sentido.
- **Cardinalidade:** 1 Unidade pode registrar 0..* Check-ins.