Para que o aplicativo Leka seja 100% funcional, o banco de dados precisa ser bem estruturado, de modo a armazenar informações dos usuários, progresso, questionários, metas e interações entre os usuários. Aqui está um exemplo de como o banco de dados pode ser modelado:

**1. Tabela de Usuários (usuarios)**

Armazena as informações dos usuários do aplicativo.

| **Campo** | **Tipo** | **Descrição** |
| --- | --- | --- |
| id | INT (PK) | ID único do usuário |
| nome | VARCHAR(100) | Nome do usuário |
| email | VARCHAR(100) | E-mail (único) |
| senha | VARCHAR(100) | Senha criptografada |
| data\_criacao | DATETIME | Data de criação da conta |
| nivel | ENUM('iniciante', 'medio', 'avancado') | Nível de experiência com programação |

**2. Tabela de Progresso (progresso)**

Armazena o progresso do usuário em diferentes cursos e atividades.

| **Campo** | **Tipo** | **Descrição** |
| --- | --- | --- |
| id | INT (PK) | ID único do progresso |
| usuario\_id | INT (FK) | ID do usuário |
| etapa\_atual | INT | Número da etapa atual |
| metas\_diarias | VARCHAR(100) | Metas diárias definidas pelo usuário |
| progresso\_percentual | INT | Progresso percentual (0-100) |
| data\_atualizacao | DATETIME | Última data de atualização do progresso |

**3. Tabela de Questionários (questionarios)**

Armazena as respostas dos questionários de cada usuário.

| **Campo** | **Tipo** | **Descrição** |
| --- | --- | --- |
| id | INT (PK) | ID único da resposta |
| usuario\_id | INT (FK) | ID do usuário |
| motivo\_aprendizado | ENUM('profissional', 'diversao', 'trabalho', 'outro') | Motivo pelo qual está aprendendo |
| categoria\_educacional | ENUM('ensino\_medio', 'universitario', 'empregado', 'nenhuma') | Nível de educação |
| aspecto\_favorito | ENUM('aparencia', 'logico', 'ambos') | O que mais cativa o usuário na programação |
| data\_resposta | DATETIME | Data da resposta ao questionário |

**4. Tabela de Atividades (atividades)**

Registra as atividades interativas e seus detalhes.

| **Campo** | **Tipo** | **Descrição** |
| --- | --- | --- |
| id | INT (PK) | ID único da atividade |
| descricao | TEXT | Descrição da atividade |
| data\_cadastro | DATETIME | Data de cadastro da atividade |
| data\_execucao | DATETIME | Data de execução da atividade |
| etapa\_numero | INT | Etapa a que essa atividade pertence |

**5. Tabela de Resumos (resumos)**

Armazena os resumos das aulas criados pelos usuários.

| **Campo** | **Tipo** | **Descrição** |
| --- | --- | --- |
| id | INT (PK) | ID único do resumo |
| usuario\_id | INT (FK) | ID do usuário |
| conteudo\_resumo | TEXT | Conteúdo do resumo |
| data\_criacao | DATETIME | Data de criação do resumo |

**6. Tabela de Amigos (amigos)**

Gerencia o relacionamento entre os usuários e seus amigos.

| **Campo** | **Tipo** | **Descrição** |
| --- | --- | --- |
| id | INT (PK) | ID único da amizade |
| usuario\_id | INT (FK) | ID do usuário |
| amigo\_id | INT (FK) | ID do amigo |

**7. Tabela de Conquistas (conquistas)**

Armazena as conquistas alcançadas pelos usuários ao longo do aprendizado.

| **Campo** | **Tipo** | **Descrição** |
| --- | --- | --- |
| id | INT (PK) | ID único da conquista |
| usuario\_id | INT (FK) | ID do usuário |
| descricao | TEXT | Descrição da conquista |
| data\_conquista | DATETIME | Data de obtenção da conquista |

**Relacionamentos:**

* **usuarios -> progresso** (1

): Um usuário pode ter múltiplos registros de progresso.

* **usuarios -> questionarios** (1

): Um usuário pode responder múltiplos questionários.

* **usuarios -> resumos** (1

): Um usuário pode criar múltiplos resumos.

* **usuarios -> atividades** (1

): Um usuário pode realizar várias atividades.

* **usuarios -> amigos** (1

): Um usuário pode ter múltiplos amigos.

* **usuarios -> conquistas** (1

): Um usuário pode ganhar várias conquistas.

Esse modelo oferece uma visão completa para gerenciar usuários, progresso, atividades e interações sociais no aplicativo Leka.

1/2

4o

**Você disse:**

ok a partir do bano me de como o aplicativo ira funcionar passo a passo desde o login até o fim por gentileza

**O ChatGPT disse:**

ChatGPT

Aqui está o fluxo de funcionamento do aplicativo Leka, desde o login até o acompanhamento do progresso e interações, baseado no banco de dados estruturado:

**1. Login e Autenticação**

* **Passo 1**: O usuário acessa a tela de login.
  + O aplicativo envia uma requisição para verificar o **e-mail** e **senha** na tabela usuarios.
  + Se os dados forem válidos, o sistema autentica o usuário.
  + Após a autenticação, o usuário é redirecionado para a página principal.

**2. Questionário Inicial (Onboarding)**

* **Passo 2**: Caso o usuário seja novo, o aplicativo exibe um questionário inicial para personalizar a experiência.
  + As respostas são armazenadas na tabela questionarios, capturando informações como:
    - Motivo do aprendizado (motivo\_aprendizado).
    - Nível educacional (categoria\_educacional).
    - Aspecto favorito da programação (aspecto\_favorito).
* **Passo 3**: Com base nas respostas, o aplicativo sugere uma trilha de aprendizado personalizada, que é exibida ao usuário.

**3. Definição de Metas Diárias e Estabelecimento de Rotina**

* **Passo 4**: O aplicativo sugere que o usuário defina metas diárias de estudo.
  + Essas metas são registradas na tabela progresso, no campo metas\_diarias.
  + Um lembrete diário pode ser ativado para o usuário se comprometer com o aprendizado.
* **Passo 5**: O usuário define o tempo que deseja dedicar ao aprendizado, e isso é registrado no campo metas\_diarias da tabela progresso.

**4. Progresso e Etapas Interativas**

* **Passo 6**: O usuário começa a acompanhar seu progresso através de etapas interativas.
  + As etapas são representadas na tabela progresso com o campo etapa\_atual.
  + Conforme o usuário avança, o progresso percentual (progresso\_percentual) é atualizado.
  + O aplicativo mostra uma visualização de "etapas interativas" para motivar o usuário a continuar aprendendo.

**5. Realização de Atividades**

* **Passo 7**: O aplicativo apresenta atividades interativas relacionadas à trilha de aprendizado personalizada.
  + Essas atividades são armazenadas na tabela atividades.
  + O usuário realiza as atividades e a conclusão é registrada, associando-as à etapa correspondente (etapa\_numero).
  + O campo data\_execucao é atualizado conforme o usuário completa as atividades.

**6. Interação com Amigos (Competição Saudável)**

* **Passo 8**: O usuário pode convidar amigos para participar da competição.
  + A tabela amigos armazena essas conexões.
  + O aplicativo mostra uma interface onde o usuário pode ver o progresso de seus amigos, promovendo uma competição saudável.
  + A interação pode ocorrer através de desafios e comparações de conquistas.

**7. Conquistas e Recompensas**

* **Passo 9**: Ao atingir metas específicas ou completar determinadas atividades, o usuário ganha conquistas.
  + As conquistas são registradas na tabela conquistas.
  + O aplicativo exibe essas conquistas na interface, como uma forma de reconhecimento e motivação adicional para o usuário.

**8. Criação de Resumos Personalizados**

* **Passo 10**: Durante o aprendizado, o usuário pode criar resumos personalizados das aulas.
  + Esses resumos são armazenados na tabela resumos.
  + O usuário pode acessar os resumos a qualquer momento para revisar o conteúdo aprendido.

**9. Acompanhamento do Progresso Geral**

* **Passo 11**: O usuário pode visualizar um painel de progresso, onde vê o progresso acumulado de suas atividades e metas diárias.
  + Isso inclui progresso percentual, etapas já concluídas e metas futuras.
  + A visualização é atualizada dinamicamente conforme o usuário avança.

**10. Gerenciamento de Perfil**

* **Passo 12**: O usuário pode gerenciar seu perfil e dados de aprendizado.
  + Isso inclui alteração de informações pessoais na tabela usuarios (nome, e-mail, senha, etc.).
  + O aplicativo também exibe os certificados alcançados e o progresso total do usuário.

**Exemplo de fluxo de um novo usuário:**

1. **Login/Cadastro**: O usuário cria uma conta e faz login.
2. **Onboarding (Questionário Inicial)**: O usuário responde ao questionário e define metas diárias.
3. **Primeira Etapa**: O usuário inicia a primeira etapa de aprendizado sugerida pelo aplicativo.
4. **Progresso e Conquistas**: O usuário completa as atividades e começa a ganhar conquistas.
5. **Competição**: O usuário convida amigos e acompanha o progresso deles.
6. **Resumos**: O usuário cria resumos personalizados para revisar o conteúdo.
7. **Perfil e Acompanhamento**: O usuário acompanha seu progresso total e pode ajustar suas metas e dados.

**1. Estrutura Adicional do Banco de Dados**

Vamos criar três novas tabelas para gerenciar as salas, as perguntas e o desempenho dos alunos:

* **Tabela salas**: Gerencia as salas criadas pelos professores.
* **Tabela perguntas**: Contém as perguntas que os professores criam para os jogos.
* **Tabela respostas\_alunos**: Armazena as respostas dos alunos e acompanha seus acertos e erros.

**Tabela salas**

sql

Copiar código

CREATE TABLE salas (

id INT PRIMARY KEY AUTO\_INCREMENT,

professor\_id INT,

codigo\_sala VARCHAR(20) UNIQUE,

data\_criacao DATETIME DEFAULT CURRENT\_TIMESTAMP,

nome\_sala VARCHAR(100),

FOREIGN KEY (professor\_id) REFERENCES usuarios(id)

);

* **professor\_id**: Referencia o professor que criou a sala.
* **codigo\_sala**: Um código único que será usado pelos alunos para ingressar na sala.
* **nome\_sala**: Nome dado à sala pelo professor (por exemplo, "Jogo de Lógica").

**Tabela perguntas**

sql

Copiar código

CREATE TABLE perguntas (

id INT PRIMARY KEY AUTO\_INCREMENT,

sala\_id INT,

pergunta\_texto TEXT,

resposta\_correta VARCHAR(100),

FOREIGN KEY (sala\_id) REFERENCES salas(id)

);

* **sala\_id**: Relaciona a pergunta à sala onde será usada.
* **pergunta\_texto**: O enunciado da pergunta.
* **resposta\_correta**: A resposta correta definida pelo professor.

**Tabela respostas\_alunos**

sql

Copiar código

CREATE TABLE respostas\_alunos (

id INT PRIMARY KEY AUTO\_INCREMENT,

aluno\_id INT,

pergunta\_id INT,

resposta\_aluno VARCHAR(100),

acerto BOOLEAN,

FOREIGN KEY (aluno\_id) REFERENCES usuarios(id),

FOREIGN KEY (pergunta\_id) REFERENCES perguntas(id)

);

* **aluno\_id**: Identifica o aluno que respondeu.
* **pergunta\_id**: Identifica a pergunta que foi respondida.
* **resposta\_aluno**: Armazena a resposta dada pelo aluno.
* **acerto**: Um campo booleano que indica se o aluno acertou (TRUE) ou errou (FALSE).

**2. Funcionamento do Aplicativo: Passo a Passo**

**1. Criação da Sala pelo Professor**

* **Passo 1**: O professor cria uma nova sala para o jogo.
  + Ele fornece um nome para a sala.
  + Um **código da sala** é gerado automaticamente, e o professor pode compartilhar esse código com seus alunos.
  + O sistema cria uma nova entrada na tabela salas, associando o professor à sala criada.

**2. Criação de Perguntas**

* **Passo 2**: O professor adiciona perguntas à sala.
  + Ele insere o texto da pergunta e define a resposta correta.
  + As perguntas são armazenadas na tabela perguntas, vinculadas à sala.

**3. Convite aos Alunos**

* **Passo 3**: O professor compartilha o **código da sala** com os alunos.
  + Os alunos entram no aplicativo e inserem o código para participar da sala.
  + Eles são adicionados à sala e podem começar a responder as perguntas.

**4. Resposta dos Alunos**

* **Passo 4**: Os alunos começam a responder as perguntas da sala.
  + Ao responder cada pergunta, o aplicativo verifica se a resposta está correta ou errada.
  + A resposta do aluno é registrada na tabela respostas\_alunos junto com um campo que indica se ele acertou ou errou.

**5. Acompanhamento dos Acertos e Erros**

* **Passo 5**: O professor pode acompanhar em tempo real o desempenho dos alunos.
  + O aplicativo pode gerar relatórios mostrando o **percentual de acertos e erros** de cada aluno para cada pergunta.
  + O professor pode visualizar a tabela respostas\_alunos, que terá o histórico de cada resposta dos alunos e o campo acerto indicando o desempenho.

**Exemplo de Fluxo de Uso**

1. **Criação de uma Sala**: O professor cria uma sala chamada "Jogo de Lógica" e recebe um código de sala (ex: LOG123).
2. **Adição de Perguntas**: O professor adiciona perguntas como:
   * "Qual é o valor de 2 + 2?" (Resposta correta: "4").
   * "O que é uma variável em programação?" (Resposta correta: "Um espaço na memória para armazenar valores").
3. **Convite aos Alunos**: O professor compartilha o código LOG123 com seus alunos.
4. **Resposta dos Alunos**: Os alunos entram na sala e começam a responder as perguntas.
   * O aluno A responde "4" para a primeira pergunta e "Função" para a segunda pergunta.
   * A primeira resposta é correta, mas a segunda está errada.
5. **Acompanhamento do Desempenho**: O professor pode ver que o aluno A acertou 50% das perguntas e pode visualizar o percentual geral da turma.

**Relatórios para o Professor**

O professor pode ter acesso a relatórios detalhados, como:

* **Percentual de Acertos por Aluno**: Exibe quantas perguntas cada aluno acertou em uma determinada sala.
* **Resumo por Pergunta**: Mostra a porcentagem de alunos que acertaram ou erraram cada pergunta.
* **Progresso Geral da Turma**: Um painel que mostra o desempenho da turma como um todo, destacando pontos fortes e fracos.