

Techmentor

Plataforma para gerenciar monitorias, conectar alunos e monitores.

- Mateus Fasciani,
- Lucas Marinho,
- Felipe Portes
- Leticia Duarte

Requisitos Funcionais

RF1: Cadastro e login de contas

O usuário deve poder criar conta e entrar na plataforma.

RF2: Conta de Aluno/Coordenador/Monitor

O usuário deve poder escolher se cria uma conta de Aluno ou Coordenador.

RF3: Verificar Horário do monitor

Acesse o perfil dos monitores para conferir se eles estão disponíveis na sala de monitoria no momento.

RF4: Calendários

Consultar calendário para verificar os dias de aulas

Requisitos Funcionais

RF5: Marcação de aulas

Usar o calendário para marcar aulas, informando a data-hora, sala e matéria do aula.

RF6: Bater ponto

Usar seu perfil para informar se está disponível para atendimento, o local onde se encontra, registrando na sua frequência.

RF7: Verificar Horário do monitor

Usar seu perfil para informar se está disponível para atendimento, e o local onde se encontra

RF8:Fazer atendimentos

Preencher um formulário de atendimento.

Requisitos Funcionais

RF9: Edição de Horários

Editar os horários de monitoria de cada monitor

RF10: Dar cargo de monitoria para Alunos

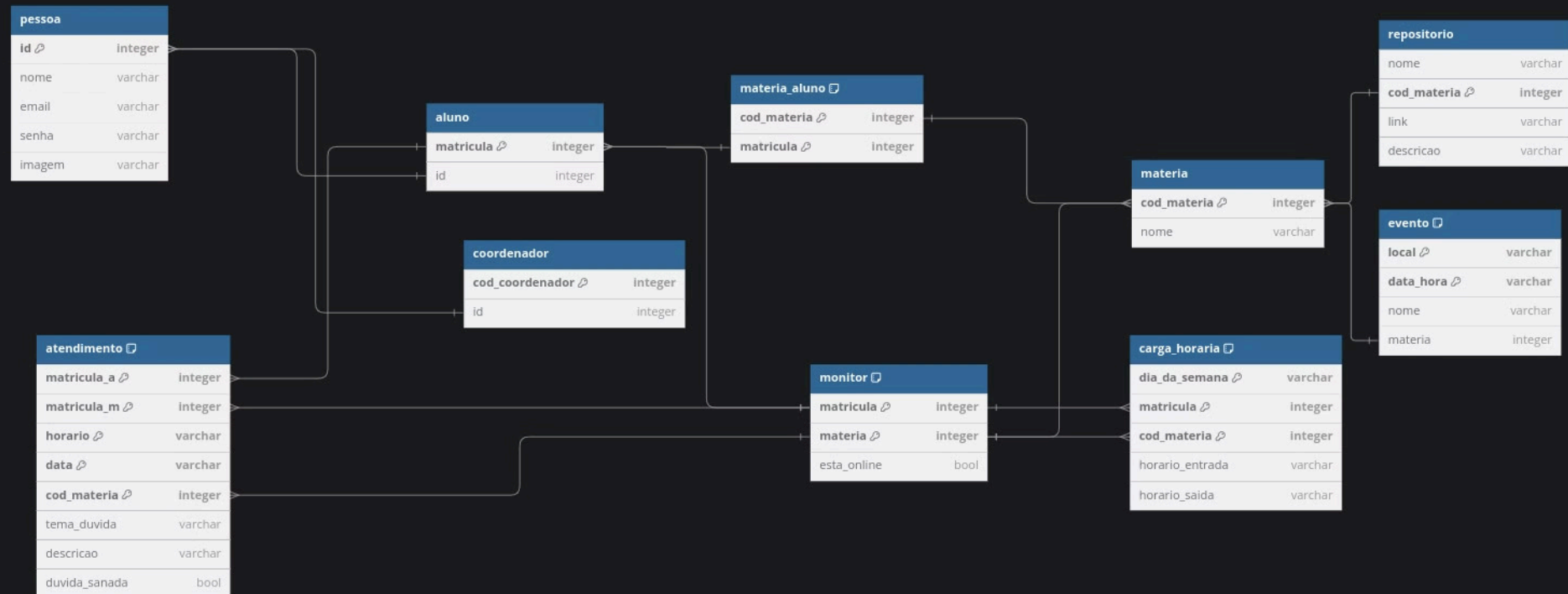
Dar e retirar o cargo de monitor para os Alunos.

RF11: Verificar frequência dos monitores

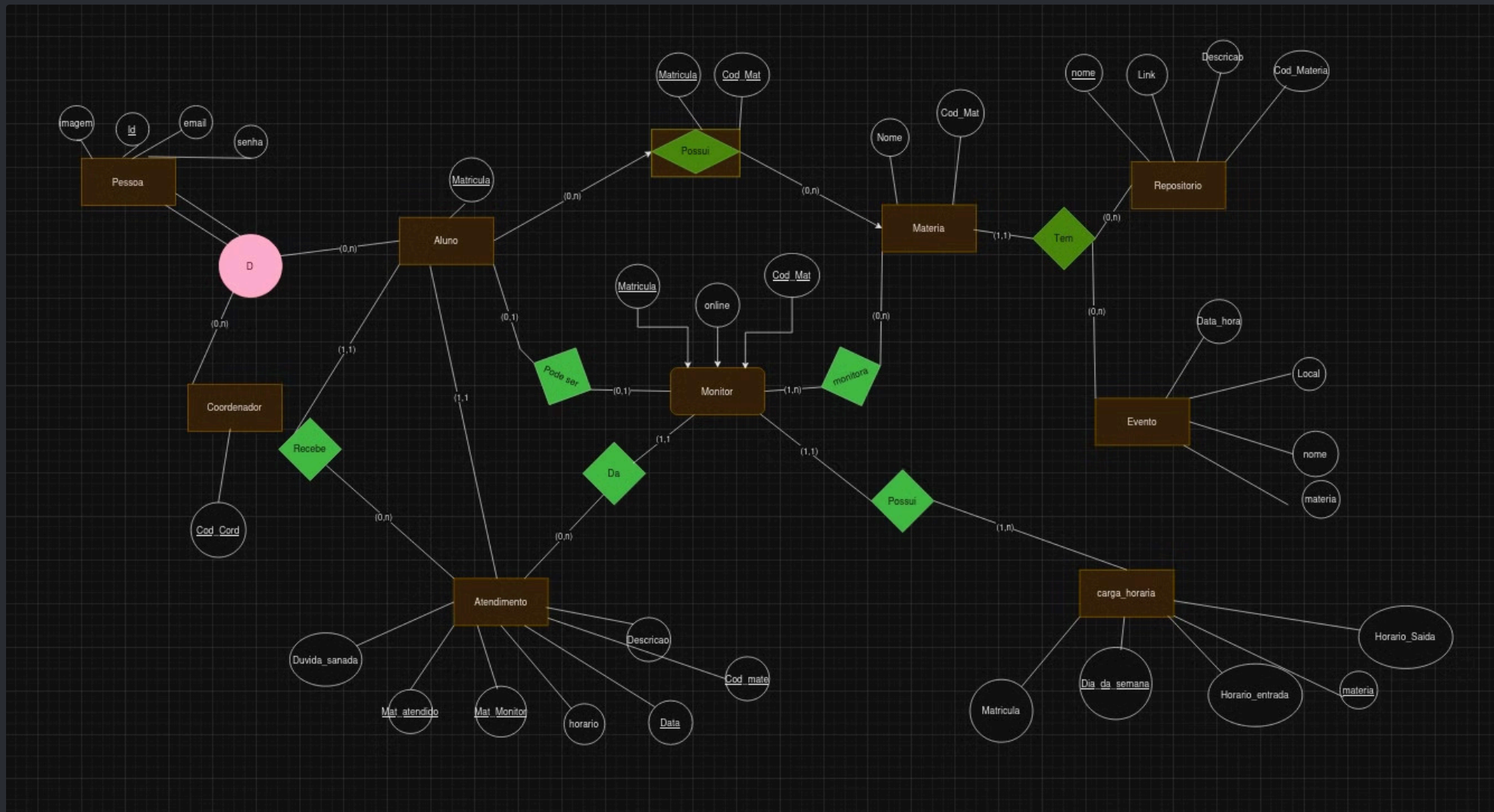
RF12: Verificar dados

Verificar a quantidade de atendimentos no mês e quem fez os atendimentos.

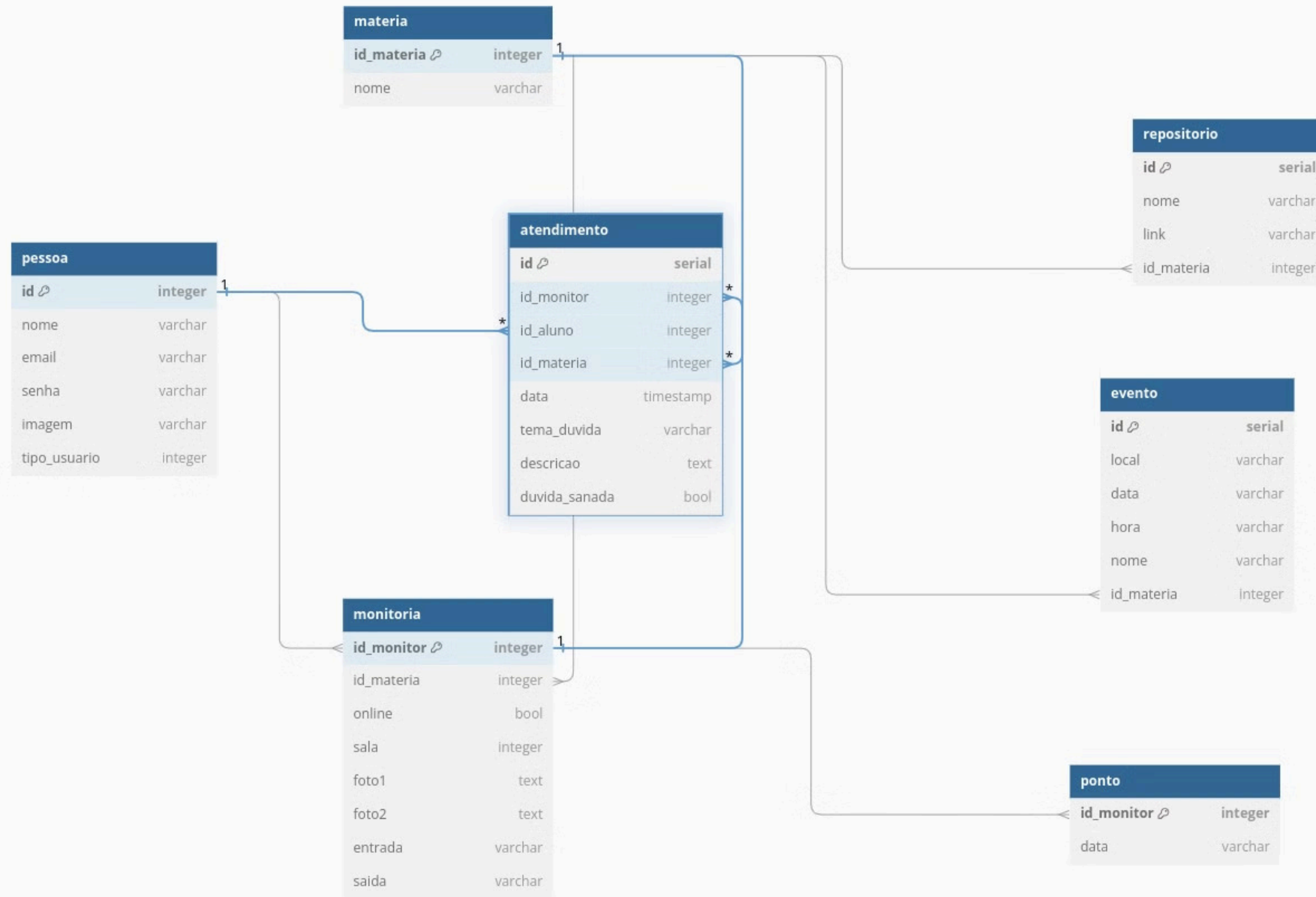
Modelo de banco de dados no formato de diagrama de entidade-relacionamento no início do projeto.



Modelo de banco de dados no formato de diagrama de entidade-relacionamento no início do projeto.



Modelo de banco de dados no formato de diagrama de entidade-relacionamento no fim do projeto.

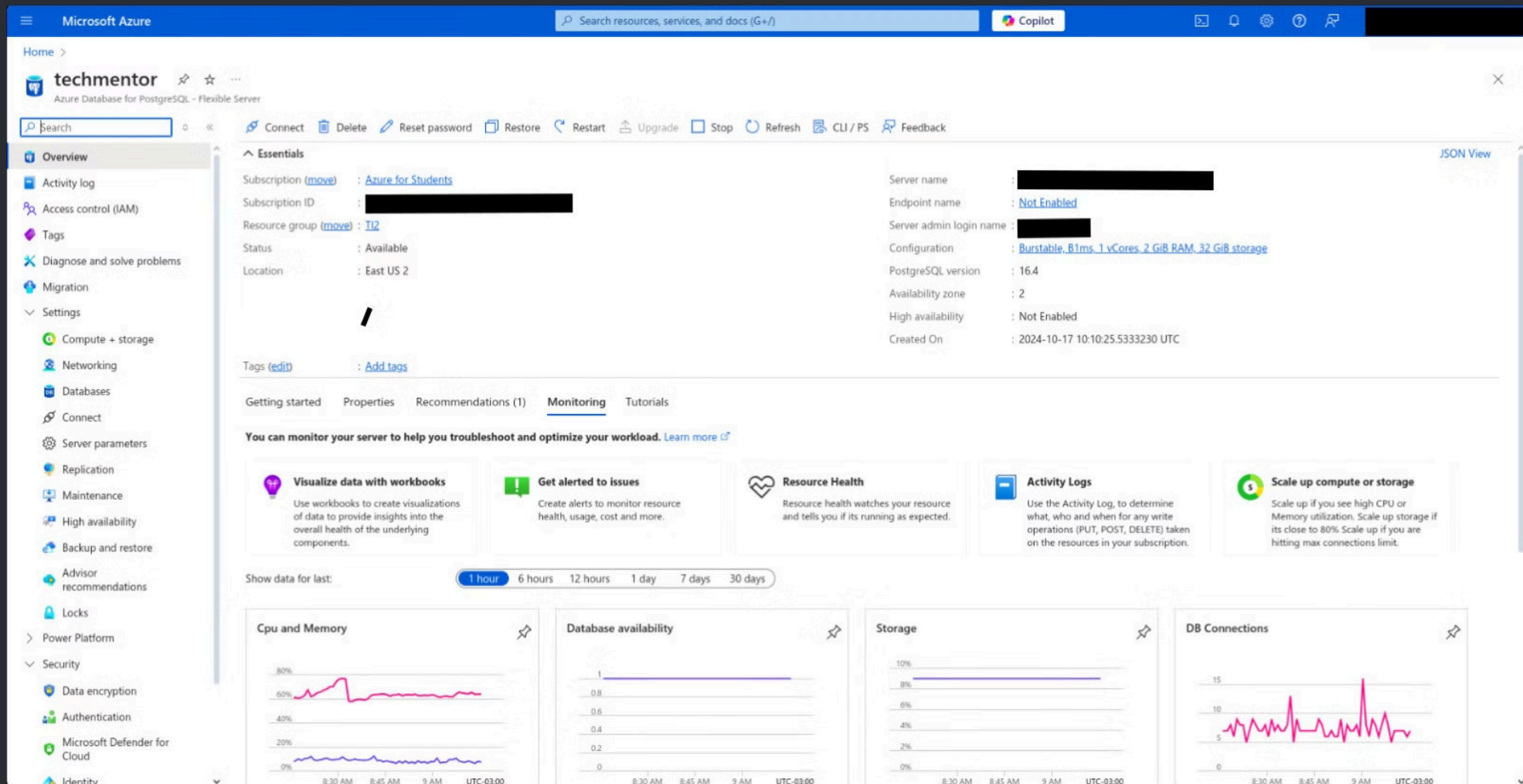


Serviços de IA utilizado: face-api.js

API de JavaScript client-side utilizada para detecção e reconhecimento facial, destinada a confirmar a presença do monitor em seu horário estabelecido. A API retorna um valor booleano indicando se a face detectada corresponde ou não ao monitor, permitindo assim o registro de ponto.

[Link do Github da API](#)

Banco de Dados hospedado na nuvem: Azure



Sistema Hospedado na Nuvem:Render

The screenshot displays the Render dashboard interface. At the top, the navigation bar includes the 'Render' logo, 'Dashboard', 'Blueprints', and 'Env Groups' tabs. On the right, there is a '+ New' button and a user profile icon. The main content area shows details for a 'WEB SERVICE' named 'Techmentor'. It includes a 'Connect' button, a 'Manual Deploy' button, and a link to 'Upgrade your instance'. Below this, the repository path 'ICEI-PUC-Minas-CC-TI / plmg-cc-ti2-2024-2-g20-techmentor' and a 'master' branch are listed, along with the service URL 'https://plmg-cc-ti2-2024-2-g20-techmentor-1.onrender.com'. A left sidebar contains a menu with options: Events, Logs, Disks, Environment, Shell, Previews, Jobs, Metrics, Scaling, and Settings (which is highlighted). The 'Settings' panel is open, showing the 'General' tab. It contains fields for 'Name' (Techmentor), 'Region' (Oregon (US West)), and 'Instance Type' (Free, 0.1 CPU, 512 MB). A blue banner at the bottom of the settings panel states: 'Please ask the owner (lucasfernandesmarinho7@gmail.com) to enter their payment information.' and provides a link to 'See remaining free usage, or learn about free service limits.'

Criptografia de senhas: Bcrypt

A criptografia de senha foi implementada utilizando o método **Bcrypt**, que é amplamente reconhecido por sua segurança e eficiência. O Bcrypt aplica um algoritmo de hashing adaptativo, o que significa que sua complexidade pode ser ajustada ao longo do tempo para resistir a avanços em poder computacional. Isso garante que as senhas armazenadas estejam protegidas contra ataques de força bruta ou tabelas arco-íris.

	123 id	A-Z nome	A-Z email	A-Z senha	A-Z foto	123 tipo_usuario
3	151.515	Lucas Teixeira	usuario@example.com	\$2a\$10\$npV.hGFjfbZXwtqsKOyNeO96HRIBy9ov	https://res.cloudinary.com/deycrrjpb/image/	2
4	1.575.201	Gabriel Carvalhaes	gabriel@leolins	\$2a\$10\$tbq1xf8Tpjp6JNNLIE9MpuDHIN1a130V		3
5	1.525.533	Lucas Marinho	lucas@gmail.com	\$2a\$10\$N9MRaXiBDXXceRUPzemBi.Aec5Ze2Z	https://res.cloudinary.com/deycrrjpb/image/	2
6	9.999	teste	teste	\$2a\$10\$hCKvLzpFcNs7metyu/EkYubfpo.HX678	https://res.cloudinary.com/deycrrjpb/image/	3
7	12.341.234	Rommel	rommel@pucminas	\$2a\$10\$QLzm3OEDkYKfwQc6ERERKuXcZcLzFB	https://res.cloudinary.com/deycrrjpb/image/	1
8	1.527.678	Felipe Fenske	asd2@ESF	\$2a\$10\$yIND1.c7HzysUMnvTWk8P.Bdw8GYbF	https://res.cloudinary.com/deycrrjpb/image/	2
9	15.141.312	Lucas Grossi	grossi@gmail.com	\$2a\$10\$OQaUTeHDKHPBZvKnUBgjPOHPG/Ibxi	https://res.cloudinary.com/deycrrjpb/image/	3
10	1.528.527	Bernardo Alves	biernierdo@techmentor	\$2a\$10\$rKVgWUft5cpBW6Vuoy6EWOjsT/4N8fC	https://res.cloudinary.com/deycrrjpb/image/	2
11	1.455.565	Matheus Viana	viana@gmail.com	\$2a\$10\$LWXRLHQ8jtnTm47WpPfSgedqzLINT9n	https://res.cloudinary.com/deycrrjpb/image/	3
12	777	Ronaldo Lins	ronaldo@email	\$2a\$10\$yDnfdtLu3DjRMcuaj6o76eC3GDpPFRa		3
13	11	Joana Mundim	teste	\$2a\$10\$JekCFTYFU7Qhrb..f1TMHuUvAzyMiomi	https://res.cloudinary.com/deycrrjpb/image/	3
14	333	Matheus Brasileiro Aguiar	usuario@example.com	\$2a\$10\$ETT3dOk4fKjsaICTClqwY.EgXFGgJZL81A		3
15	1	Natsumi Yuki	usuario@example.com	\$2a\$10\$najg3CiYHImZ8Tc4VZyLC.XfkKg4WXox	https://res.cloudinary.com/deycrrjpb/image/	3
16	152.526	Chris	chris@gmail.com	\$2a\$10\$TI7ukAXJAKMzjkV6OtnCceImHbxd2Hb	https://res.cloudinary.com/deycrrjpb/image/	3

```
if(result.next()){  
    String senhaCripto = result.getString("senha");  
    autenticado = BCrypt.checkpw(pessoa.getSenha(), senhaCripto);  
}
```

s Fernandes Marinho, last month • Adicionado método de login de pessoa

```
String senhaCripto = BCrypt.hashpw(pessoa.getSenha(), BCrypt.gensalt());
```

Defesa contra SQL injection: prepared statements

A defesa contra SQL Injection foi implementada utilizando **prepared statements**, que parametrizam as consultas ao banco de dados, evitando a injeção de comandos maliciosos ao tratar os dados de entrada como parâmetros seguros.

```
try(Connection conn = DataBaseConnection.getConnection()){  
  
    String sql = "SELECT senha FROM pessoa WHERE id = ?";  
    PreparedStatement pstmt = conn.prepareStatement(sql);  
  
    pstmt.setInt(1, pessoa.getId());  
  
    ResultSet result = pstmt.executeQuery();  
  
    if(result.next()){  
        String senhaCripto = result.getString("senha");  
        autenticado = BCrypt.checkpw(pessoa.getSenha(), senhaCripto);  
    }  
}
```

Vídeo do projeto :

[Clique aqui](#)

Muito obrigado!

