

Faculdade de Informática e Administração Paulista

**API Meraki - Documentação**

DANIELLE RODRIGUES LOBO – rm88913

GIOVANNA FRANÇA MAGALHÃES SANTOS – rm87019

JOÃO PEDRO MARQUES NARDI – rm85846

MATHEUS SANCHES BARREIROS – rm87426

OTÁVIO VIÑÉ MASELLA – rm89110

2022

São Paulo - SP

A ansiedade é um descaso que cada vez mais aflige a sociedade, causando mal às pessoas de diferentes formas e intensidades. E em contrapartida, algo que também avança rapidamente em meio a atualidade é a tecnologia.

Ajudar milhares de pessoas que sofrem com os sintomas da ansiedade é o nosso objetivo principal. O contexto atual de agitações de exigências tornam esse problema a realidade de muitos, por isso, nossa equipe desenvolveu o projeto do app Meraki com diversas funcionalidades para auxiliar em situações que os ansiosos são submetidos no dia a dia.

Nossa equipe irá desfrutar da tecnologia, criando uma plataforma que auxilia pessoas ansiosas por meio da inteligência artificial.

Segue a arquitetura da solução:

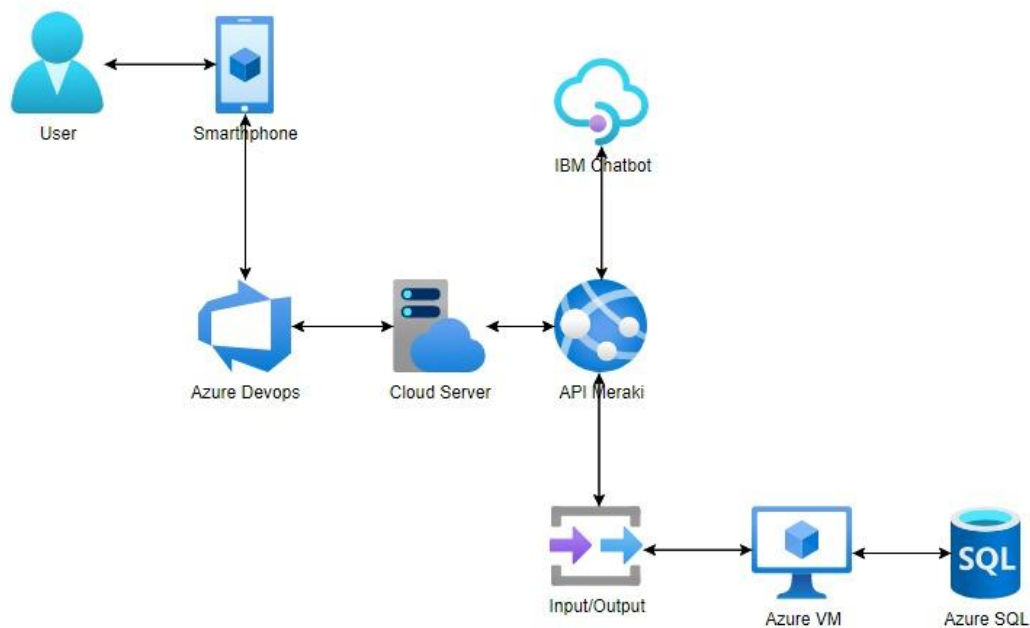


Tabela de **Endpoints**:

Nome	Path	Verbo	Ação
index	/api/usuario	GET	Listar todos os usuários
create	/api/usuario	POST	Criar um novo usuário (cadastro)
getById	/api/usuario/{id}	GET	Devolver um usuário específico
deletar	/api/usuario/{id}	DELETE	Deletar um usuário
update	/api/usuario/{id}	PUT	Atualizar um usuário

Requisição do tipo GET de endpoint: /api/usuario

Não recebe parâmetros.

Devolve uma lista de usuários do Meraki em formato Application JSON.

Só pode ser realizada por quem estiver autorizado.

Requisição do tipo POST de endpoint: /api/usuario

Recebe por parâmetro, através do corpo da requisição, um objeto do tipo usuário com os seguintes dados: nome(obrigatório), email(obrigatório), senha(obrigatório) e foto(não obrigatório). A senha é criptografada e o objeto é salvo no banco de dados.

Se tudo correr bem, a requisição devolve o status 201 do protocolo Http (CREATED) e, no corpo, o usuário criado.

Não precisa de autorização.

Requisição do tipo GET de endpoint: /api/usuario/{id}

Recebe por parâmetros o id do usuário que deseja encontrar.

Devolve o usuário em formato Application JSON.

Só pode ser realizada por quem estiver autorizado.

Requisição do tipo PUT de endpoint: /api/usuario/{id}

Recebe por parâmetro, através do corpo da requisição, o id e um objeto do tipo usuário com dados novos/alterados. O método confere se existe usuário com esse id. Caso não, devolve 404 (NOT FOUND). Caso sim, as propriedades alteradas são copiadas para dentro do objeto e é salvo no banco de dados. Status 200(ok).

Só pode ser realizada por quem estiver autorizado.

Requisição do tipo DELETE de endpoint: /api/usuario/{id}

Recebe por parâmetros o id do usuário que deseja deletar.

Confere se o usuário com aquele id existe, se existir, ele é deletado e o método devolve o status 204 (NO CONTENT) do http.

Se o usuário com id indicado não existir, ele devolve o status 404 (NOT FOUND).

Só pode ser realizada por quem estiver autorizado.

Tabela de **Endpoints**:

Nome	Path	Verbo	Ação
index	/api/diario	GET	Listar todos os diários
create	/api/diario	POST	Criar um novo diário
getById	/api/diario/{id}	GET	Devolve um diário específico
deletar	/api/diario/{id}	DELETE	Deletar diário
update	/api/diario/{id}	PUT	Editar um diário

Requisição do tipo GET de endpoint: /api/diario

Não recebe parâmetros.

Devolve páginas com 6 diários por vez.

Só pode ser realizada por quem estiver autorizado.

Requisição do tipo POST de endpoint: /api/diario

Recebe por parâmetro, através do corpo da requisição, um diário com os seguintes dados: título(obrigatório), conteúdo(obrigatório) e data(obrigatório).

Se tudo correr bem, a requisição devolve o status 201 do protocolo Http (CREATED) e, no corpo, o diário criado.

Só pode ser realizada por quem estiver autorizado.

Requisição do tipo GET de endpoint: /api/diario/{id}

Recebe por parâmetros o id do diário que deseja encontrar.

Devolve o diário em formato Application JSON.

Só pode ser realizada por quem estiver autorizado.

Requisição do tipo PUT de endpoint: /api/diario/{id}

Recebe por parâmetro, através do corpo da requisição, o id e um objeto do tipo diário com seus dados alterados. O método confere se ele existe. Caso não, devolve 404(NOT FOUND). Caso sim, as propriedades alteradas são copiadas para dentro do objeto e é salvo no banco de dados e o método devolve o http status de 200.

Só pode ser realizada por quem estiver autorizado.

Requisição do tipo DELETE de endpoint: /api/diario/{id}

Recebe por parâmetros o id do diário que deseja deletar.

Confere se o diário com aquele id existe, se existir, ele é deletado e o método devolve o status 204 (NO CONTENT) do http.

Se o diário com id indicado não existir, ele devolve o status 404 (NOT FOUND).

Só pode ser realizada por quem estiver autorizado.