

Hybrid Flow

Sistema de Reserva de Salas

Equipe: João Pedro Coelho Maran, Victor Mathias Pereira Porto e Iago Henrique Bortolo

01. Introdução

- A Hybrid tem como objetivo otimizar a gestão e utilização de ambientes compartilhados. Para isso, foi projetada uma plataforma que permite **reservar salas, monitorar a temperatura e acompanhar a porcentagem de ocupação em tempo real**.
A solução busca melhorar a eficiência no uso dos espaços, oferecendo recursos que facilitam o controle de disponibilidade, o conforto dos usuários e a segurança na lotação dos ambientes.
Dessa forma, o sistema apoia tomadas de decisão e promove uma experiência mais organizada, inteligente e integrada na administração das salas.

02. Banco de dados

id_usuario	nome	email	senha_hash	tipo_usuario
1	Ana Ribeiro	ana@empresa.com	HASH123456	admin
2	Carlos Mendes	carlos@empresa.com	HASH654321	colaborador
3	Juliana Alves	juliana@empresa.com	HASH987654	colaborador
4	Roberto Lima	roberto@empresa.com	HASH192837	colaborador

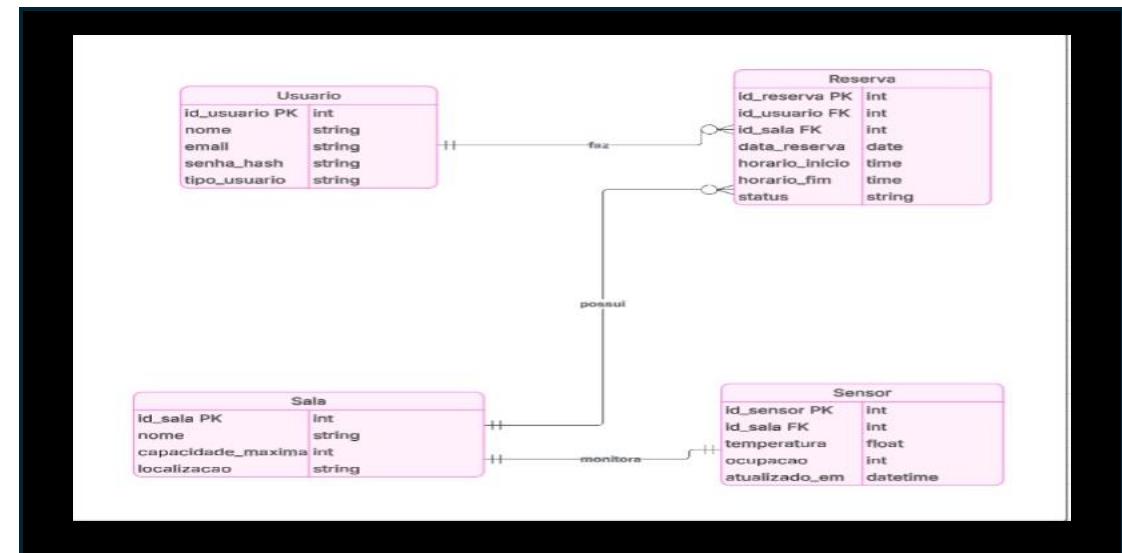
id_sala	nome	capacidade_maxima	localizacao
1	Sala Azul	12	Bloco A - 2º andar
2	Sala Verde	8	Bloco B - 1º andar
3	Auditório Principal	50	Bloco C - Térreo
4	Sala Reunião 304	6	Bloco A - 3º andar

id_sensor	id_sala	temperatura	ocupacao	atualizado_em
1	1	23.5	40	25/11/2025 14:03
2	2	24.1	20	25/11/2025 14:05
3	3	21.8	65	25/11/2025 14:02
4	4	25.3	10	25/11/2025 14:04

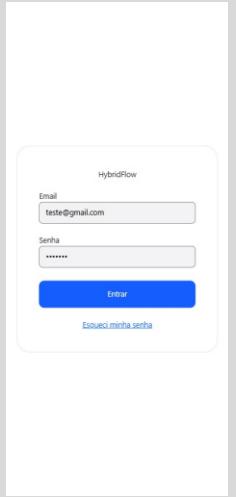
id_reserva	id_usuario	id_sala	data_reserva	horario_inicio	horario_fim	status
1	1	1	1	26/11/2025	09:00	10:00 ativa
2	2	2	2	26/11/2025	14:00	15:00 ativa
3	3	3	3	27/11/2025	10:00	12:00 confirmada
4	4	4	4	28/11/2025	08:00	09:00 cancelada
5	1	3	3	29/11/2025	13:00	15:00 ativa

02.1 Modelo ER

- Usuário - id_usuario PK
Reserva - id_reserva PK; id_usuario FK; id_sala FK
Sala - id_sala PK
Sensores (IoT) - id_sensor PK; id_sala FK



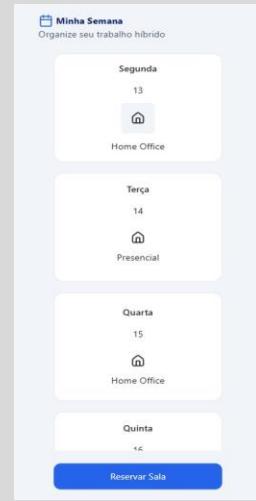
03. Mobile / Prototipo



Login

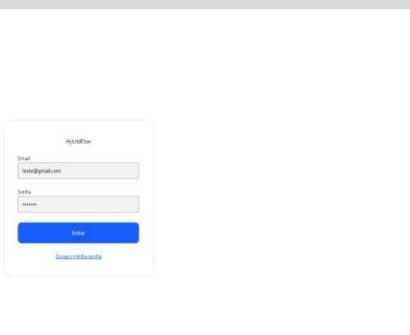


Reserva

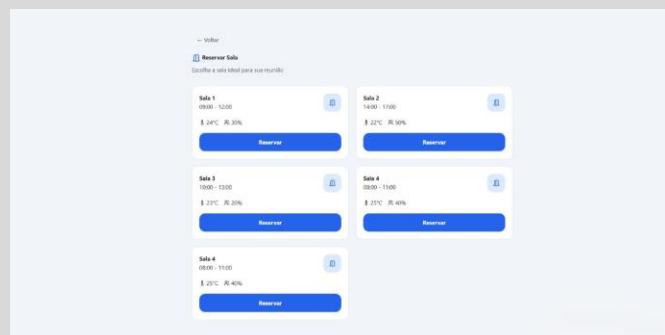


Agenda

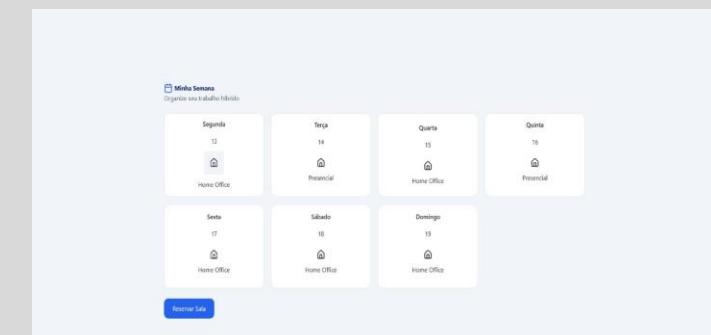
03.1 Web



Login



Reserva



Agenda

04. Segurança da Aplicação

- **04.1 Autenticação**
- 1. Senhas Criptografadas (BCrypt)
 - Todas as senhas são armazenadas utilizando hash com BCrypt, garantindo que nenhuma senha seja salva em texto puro no banco.
 - Senha do usuário é criptografada no momento do cadastro.
 - Durante o login, a senha digitada é comparada com o hash usando BCrypt.Verify.
- **04.2. Controle de Acesso e Autorização**
 - Com base no tipo de usuário, a API restringe rotas e operações sensíveis, como:
 - Criação e exclusão de salas
 - Gestão de sensores

05. IoT (Sensores de Temperatura)

- O sistema foi projetado para ser totalmente compatível com dispositivos IoT, permitindo a integração de sensores de temperatura instalados nas salas. Esses sensores realizam medições contínuas do ambiente e enviam os dados para a plataforma por meio de um serviço de comunicação (API ou gateway IoT), garantindo atualizações rápidas e precisas.
- Cada sensor transmite informações como **temperatura atual**, **horário da leitura** e **identificação da sala**. Ao receber os dados, o sistema valida as informações e as registra no banco de dados, tornando possível acompanhar a condição térmica de cada espaço em tempo real.

Conclusão

O sistema reúne banco de dados, API, IoT, segurança e mobile de forma integrada. O banco armazena usuários, salas, sensores e reservas. A API conecta os módulos, enquanto os sensores IoT enviam temperatura e ocupação. A segurança garante autenticação e hashing de senhas e nos permite visualizar salas e fazer reservas. Tudo funciona de forma simples, conectada e segura.

05. Testes

Cadastro Users:

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:5000/usuarios' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "id_original": 0,
    "nome": "Iago",
    "email": "iago@fiap.com",
    "senha_hash": "1234"
}'
```



Request URL

<http://localhost:5000/usuarios>

Server response

Code Details

200

Response body

Usuário criado com senha segura



Download

Response headers

content-type: text/plain; charset=utf-8
date: Sat,29 Nov 2025 04:16:58 GMT
server: Kestrel
transfer-encoding: chunked

Responses

Code Description

Links

200

OK

No links

05. Testes

Login Users com senha incorreta:

Responses

Curl

```
curl -X 'POST' \
'http://localhost:5000/Usuario/login' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
"id_original": 0,
"nome": "Iago",
"email": "iago@fiap.com",
"senha_hash": "$2a$11$5pd2TMB7a8eLfmm0i4NF1.yvTShoaic2fzJdz218.pqB05gWDzWrm"
}'
```



Request URL

<http://localhost:5000/Usuario/login>

Server response

Code Details

401 Undocumented Error: Unauthorized

Response body

Senha incorreta



Download

Response headers

content-type: text/plain; charset=utf-8
date: Sat,29 Nov 2025 04:21:30 GMT
server: Kestrel
transfer-encoding: chunked

Responses

Code Description

Links

200 OK

No links

05. Testes

Email em branco:

Responses

Curl

```
curl -X 'POST' \
'http://localhost:5000/Usuario/login' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
  "id_original": 0,
  "nome": "",
  "email": "",
  "senha_hash": ""
}'
```



Request URL

<http://localhost:5000/Usuario/login>

Server response

Code Details

401 Error: Unauthorized

Undocumented

Response body

E-mail não encontrado



Download

Response headers

content-type: text/plain; charset=utf-8
date: Sat, 29 Nov 2025 04:40:25 GMT
server: Kestrel
transfer-encoding: chunked

Responses

Code Description

Links

200 OK

No links

05. Testes

Login Users com senha correta :

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:5000/Usuario/login' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "id_original": 0,
    "name": "Iago",
    "email": "iago@fiap.com",
    "senha_hash": "1234"
}'
```



Request URL

<http://localhost:5000/Usuario/login>

Server response

[Code](#) [Details](#)

200

Response body

Login autorizado



[Download](#)

Response headers

content-type: text/plain; charset=utf-8
date: Sat, 29 Nov 2025 04:22:56 GMT
server: Kestrel
transfer-encoding: chunked

Responses

[Code](#) [Description](#)

[Links](#)

200 OK

No links

05. Testes

Reserva Criada:

Responses

Curl

```
curl -X 'POST' \
'http://localhost:5000/reservas' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
  "id_usuario_original": 0,
  "id_sala_original": 4,
  "data_reserva": "2025-02-03",
  "horario_inicio": "09:00",
  "horario_fim": "10:30",
  "status": "ativa"
}'
```



Request URL

<http://localhost:5000/reservas>

Server response

[Code](#) [Details](#)

200

Response body

Reserva criada



[Download](#)

Response headers

content-type: text/plain; charset=utf-8
date: Sat, 29 Nov 2025 04:30:37 GMT
server: Kestrel
transfer-encoding: chunked

Responses

[Code](#) [Description](#)

[Links](#)

200 OK

No links

05. Testes

Atualizar Sala:

Responses

Curl

```
curl -X 'PUT' \
'http://localhost:5000/Sala/0' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
  "id": "string",
  "id_original": 0,
  "nome": "Nova_Sala A4",
  "capacidade_maxima": 20,
  "localizacao": "Sala A4"
}'
```



Request URL

<http://localhost:5000/Sala/0>

Server response

Code Details

200

Response body

Sala atualizada com sucesso



Download

Response headers

content-type: text/plain; charset=utf-8
date: Sat, 29 Nov 2025 04:33:20 GMT
server: Kestrel
transfer-encoding: chunked

Responses

Code Description

200

OK

Links

No links

05. Testes

Deletar Reserva:

Responses

Curl

```
curl -X 'DELETE' \
'http://localhost:5000/reservas/0' \
-H 'accept: */*'
```



Request URL

```
http://localhost:5000/reservas/0
```

Server response

Code Details

200

Response body

Reserva removida



Download

Response headers

```
content-type: text/plain; charset=utf-8
date: Sat, 29 Nov 2025 04:34:07 GMT
server: Kestrel
transfer-encoding: chunked
```

Responses

Code Description

Links

200

OK

No links

05. Testes

Sensores:

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:5000/Sensor' \
  -H 'accept: */*'
```



Request URL

```
http://localhost:5000/Sensor
```

Server response

Code	Details
------	---------

200

Response body

```
{
  "-snsA1": {
    "id": "-snsA1",
    "temperatura": 22.1,
    "ocupacao": 3,
    "atualizado_em": "NOW()"
  },
  "-snsA2": {
    "id": "-snsA2",
    "temperatura": 23.4,
    "ocupacao": 5,
    "atualizado_em": "NOW()"
  },
  "-snsA3": {
    "id": "-snsA3",
    "temperatura": 21.9,
    "ocupacao": 1,
    "atualizado_em": "NOW()"
  },
  "-snsA4": {
    "id": "-snsA4",
    "temperatura": 24,
    "ocupacao": 2,
    "atualizado_em": "NOW()"
  },
  "-snsA5": {
    "id": "-snsA5",
    "temperatura": 23.5,
    "ocupacao": 4,
    "atualizado_em": "NOW()"
  }
}
```



Download

Response headers

```
content-type: application/json; charset=utf-8
date: Sat, 29 Nov 2025 04:35:48 GMT
server: Kestrel
transfer-encoding: chunked
```

Responses

Code	Description	Links
------	-------------	-------

200

OK

No links

06. Como usar

- Para a utilização dos mesmos pode acessar esses links depois de rodado o Teste.exe:
 - <https://localhost:5000/usuarios>
 - <https://localhost:5000/sensor>
 - <https://localhost:5000/reserva>
 - <https://localhost:5000/sala>
 - <https://localhost:5000/swagger>