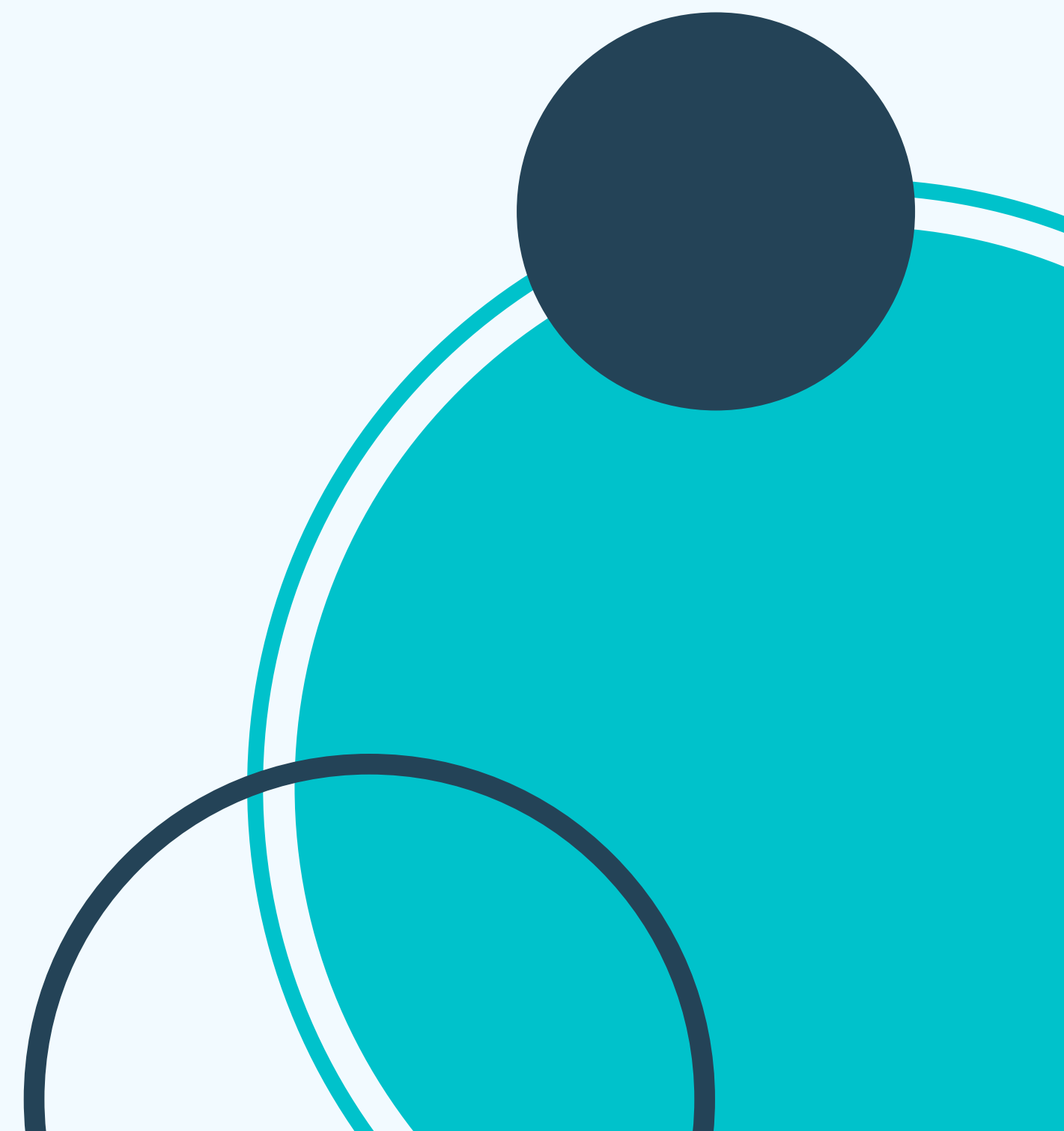




REC.   
TECHNOLOGY



# SOBRE NÓS

- Investimos no desenvolvimento de soluções focadas no mercado audiovisual;
- Buscamos aumentar os resultados e diminuir os prejuízos;
- Inovar para melhorar;



# CONTEXTO

- Crescimento audiovisual de 5% ao ano;
- 300 mil empregos gerados no Brasil;



# PRINCIPAIS PROBLEMAS

- Falta de insights de negócios e técnicos sobre recursos;
- BPMN de Atendimento arcaico e manual;
- Falta de visibilidade sobre a operação de infraestrutura nas filiais.



# TAKE ONE

---

- Solução para os problemas
- Aplicação em JAR
- Captura de componentes e insight

# DIAGRAMA DE VISÃO DE NEGÓCIO



Captação de clientes  
apresentando a empresa



Instalação do sistema Take  
One nos ATMs



Ativação do sistema  
de gerenciamento



Monitoramento  
dos dados capturados



Suporte 24H e  
contato com o cliente



Análise dos insights para gerar  
poder decisório

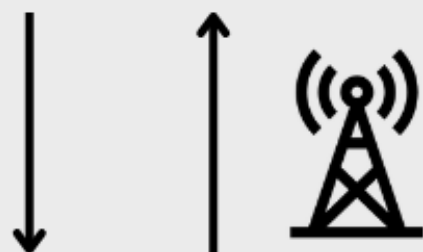
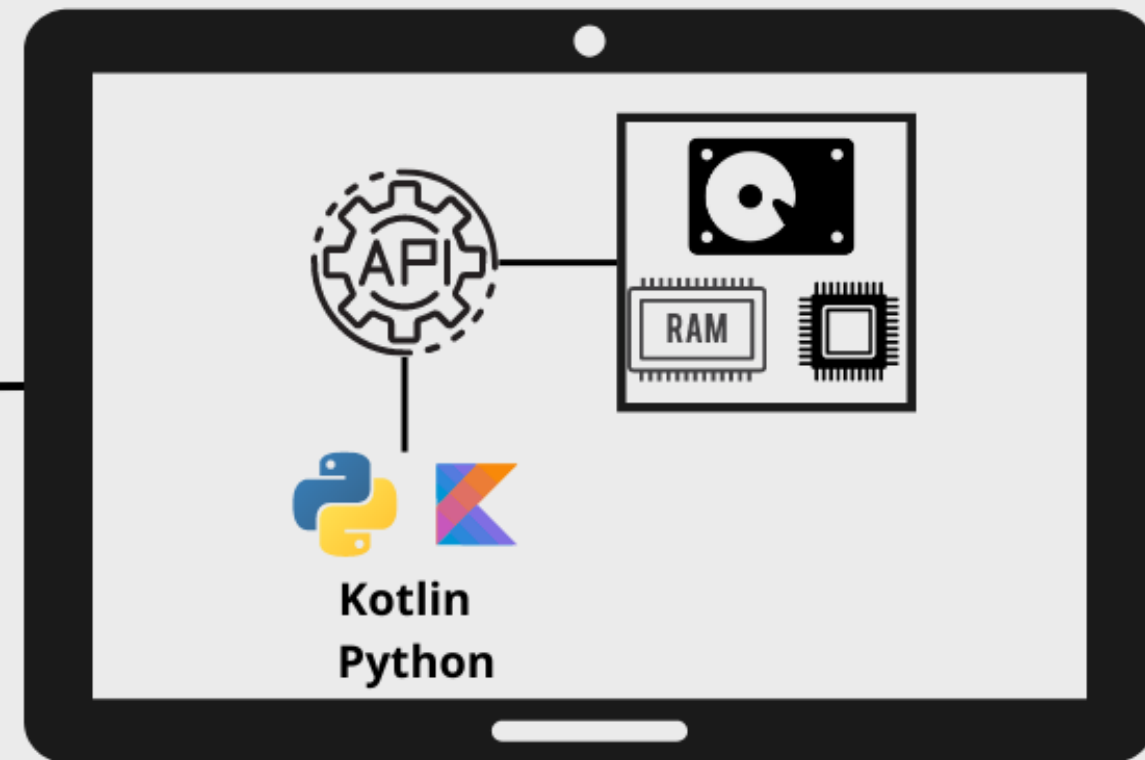


Conversão de dados  
em gráficos e KPIs

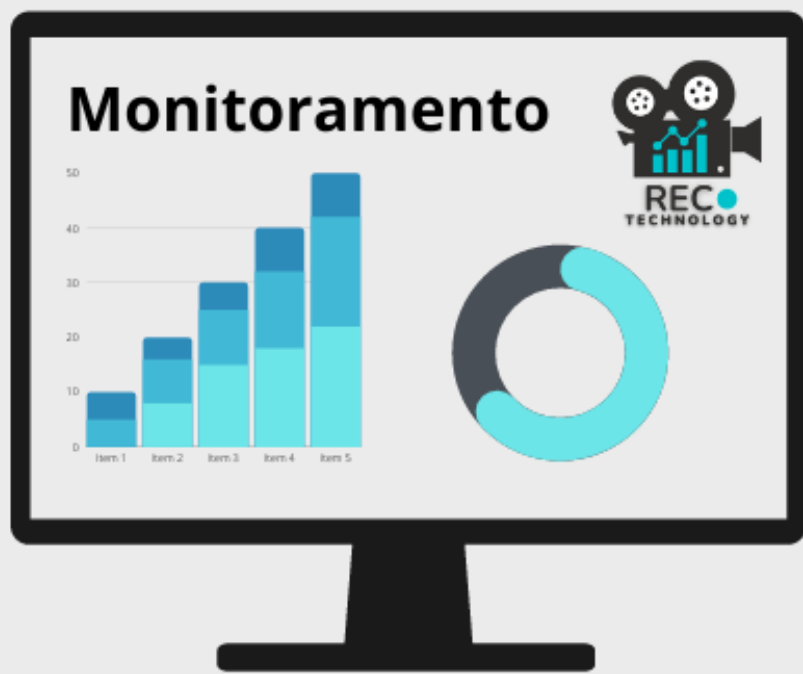
# DIAGRAMA DE SOLUÇÃO

---

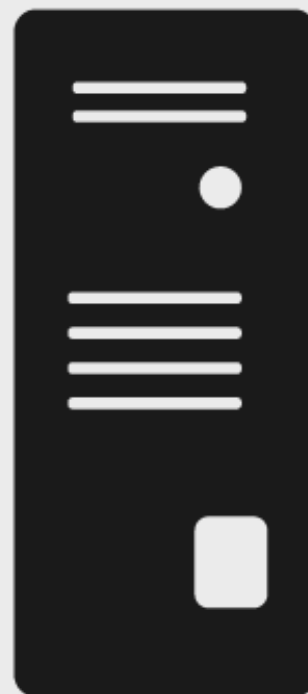
Sistema



aws



Aplicação web



Slack  
Pipefy



Alertas

Container

Kotlin  
Python



EC2











# PRODUCT BACKLOG

---



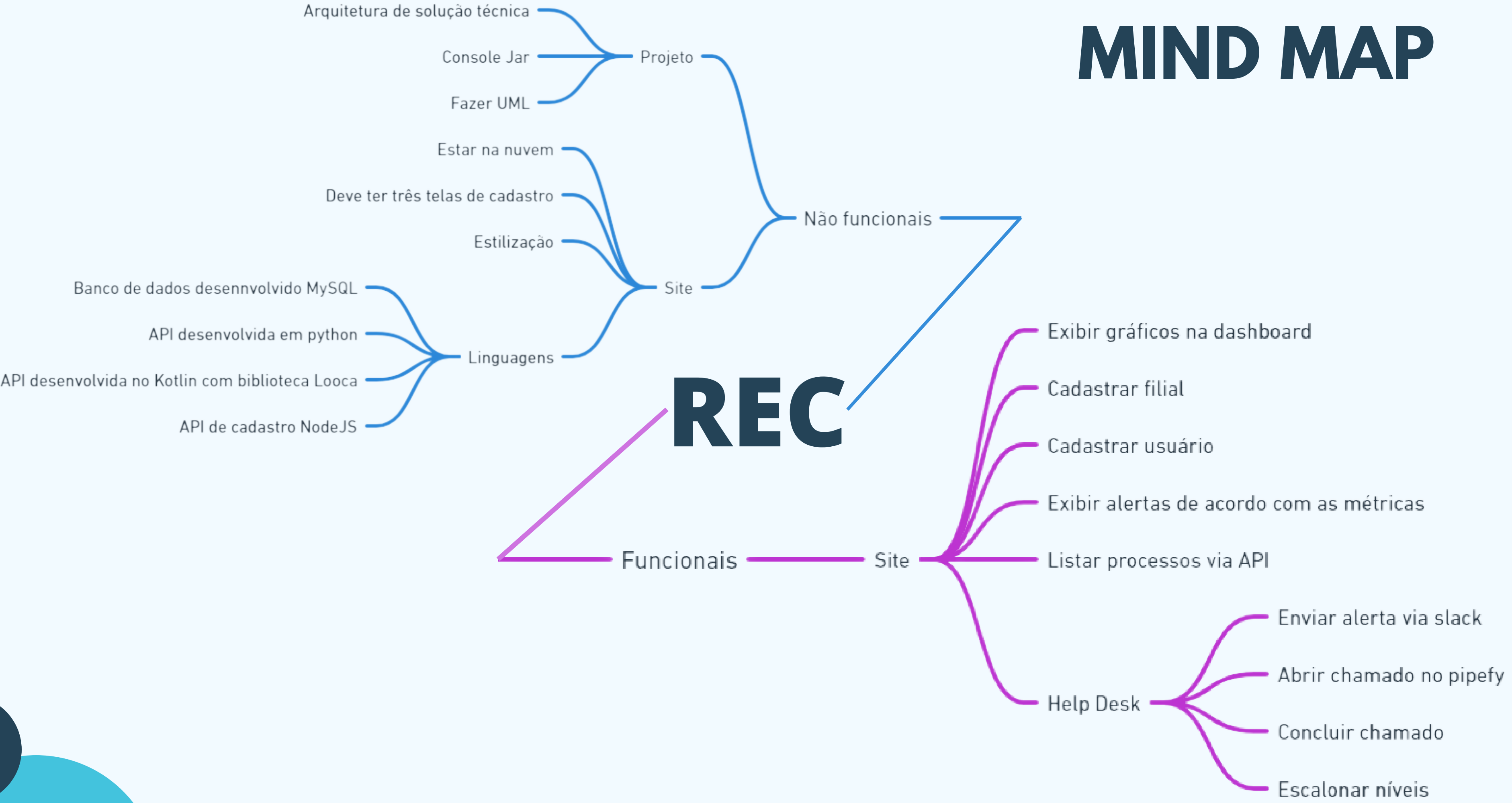
Product BackLog Qualified (PBQ)

Cód. do Requisit 	Cód. do Artefat 	Estado 	Descrição do Requisito 	Prioridad 	Observação 
1	US001	Em Andamento	O sistema deve ter um mecanismo de recebimento de chamados de forma detalhada, contendo todas as informações necessárias.	Importante	
2	US002	Em Andamento	O sistema deve realizar de forma agil envio do incidente (chamado), ocorrendo assim a diminuição do tempo médio para reparo (MTTR)	Importante	
3	US003	Feito	O sistema deve monitorar o processo de funcionamento de maquinas PDVs e ATMs	Importante	
4	US004	Planejada	O sistema deve conter um grafico exibindo informações financeiras a respeito de vendas provenientes dos totens para auxiliar na tomada de decisão sobre a infraestrutura de TI	Importante	
5	US005	Planejada	O sistema deve criar e enviar chamados de incidentes computacionais de forma automatizada	Importante	
6	US007	Feito	O sistema deve conter um grafico exibindo informações gerais da filial para medição de investimentos	Importante	
7	US008	Feito	O sistema deve gerar insights de negócios e técnicos sobre recursos atrelados as maquinas para haver uma visão sintetica sobre os componentes e seu funcionamento	Importante	
8	US009	Feito	O sistema deve gerar graficos com informações tecnicas integradas para haver um monitoramento agil	Importante	
9	US010	Feito	O sistema deve realizar monitoramento 24h de equipamentos distribuidos	Importante	
10	SB001	Feito	Contexto + Justificativa do Projeto	Importante	
11	SB002	Feito	Proto-Persona	Essencial	
12	DER001	Feito	Modelagem de Dados Lógica - versão 1	Essencial	
13	ARQ001	Feito	Projeto Funcional - Kotlin	Essencial	
14	ARQ002	Feito	Client Linux local com camada de segurança	Essencial	
15	EC001	Feito	Entrevista de campo sobre o negocio	Essencial	
16	LUX001	Feito	Mapeamento de problemas, soluções, usuarios e beneficios	Essencial	
17	ARQ003	Feito	Capturar Dados de maquina via Python, info do sistema operacional e discretizar.	Essencial	
18	ARQ004	Feito	Armazenar a captura de dados em BD	Essencial	
19	BE001	Feito	Cadastro e estrutura do Backend NodeJS funcional	Essencial	
20	UC001	Feito	Site institucional	Essencial	
20.1	UC002	Feito	Abertura de chamados via Pipefy e Slack	Essencial	
20.2	UC003	Feito	MVP do dashboard (funcional)	Essencial	
21	DS001	Feito	Diagrama de Solução	Importante	
22	KOT001	Feito	Jar inicial do projeto conectado com banco de dados usando a biblioteca Looca	Essencial	
23	VM001	Feito	O sistema deve listar processos via API através da VM.	Essencial	
23.1	VM002	Feito	VM na nuvem AWS	Essencial	
24	MIN001	Feito	Mineração de dados	Essencial	
25	MET001	Feito	Definição de métricas (Dimensões)	Essencial	
26	ALT001	Em Andamento	Definição de Alertas	Essencial	

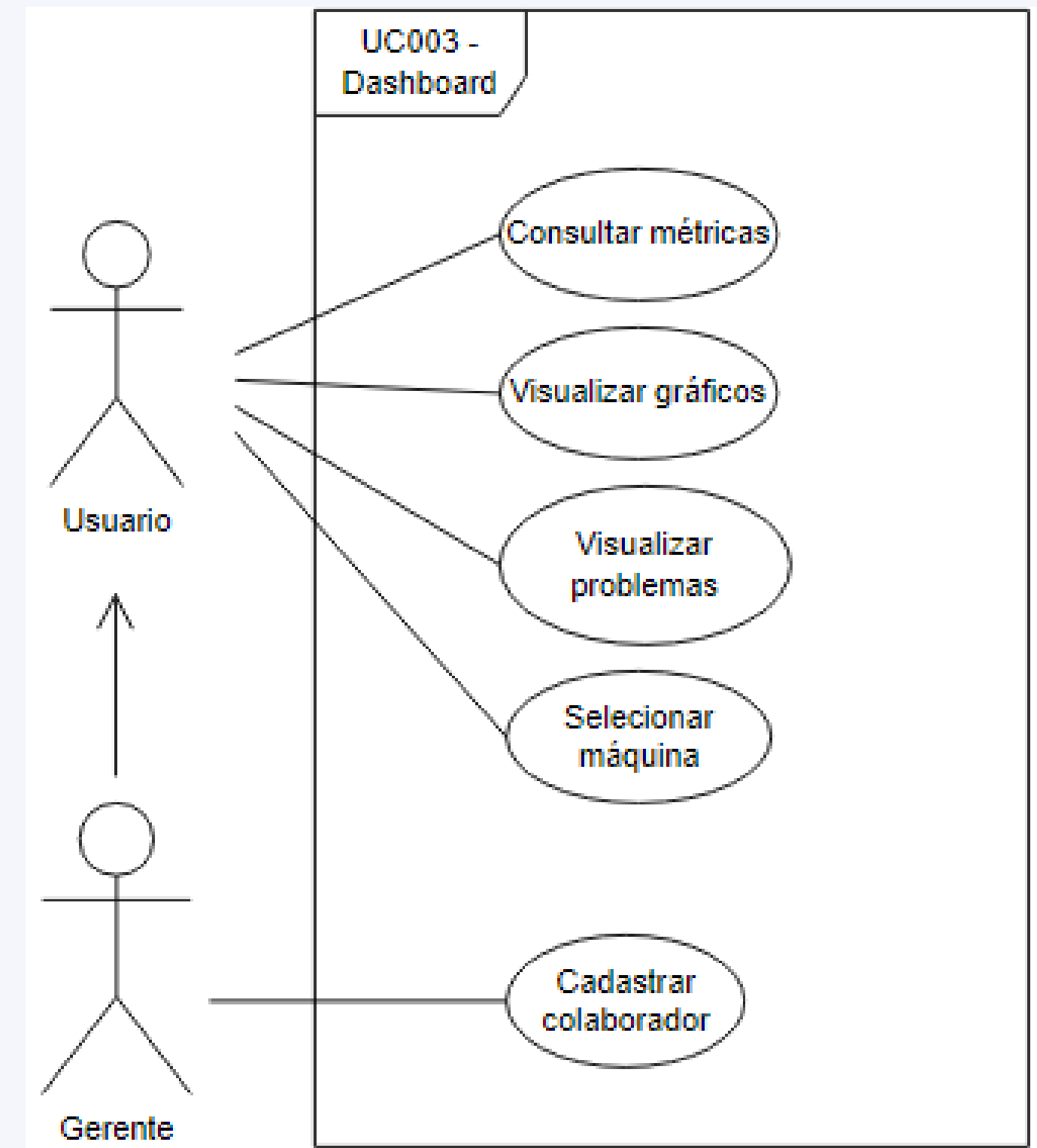
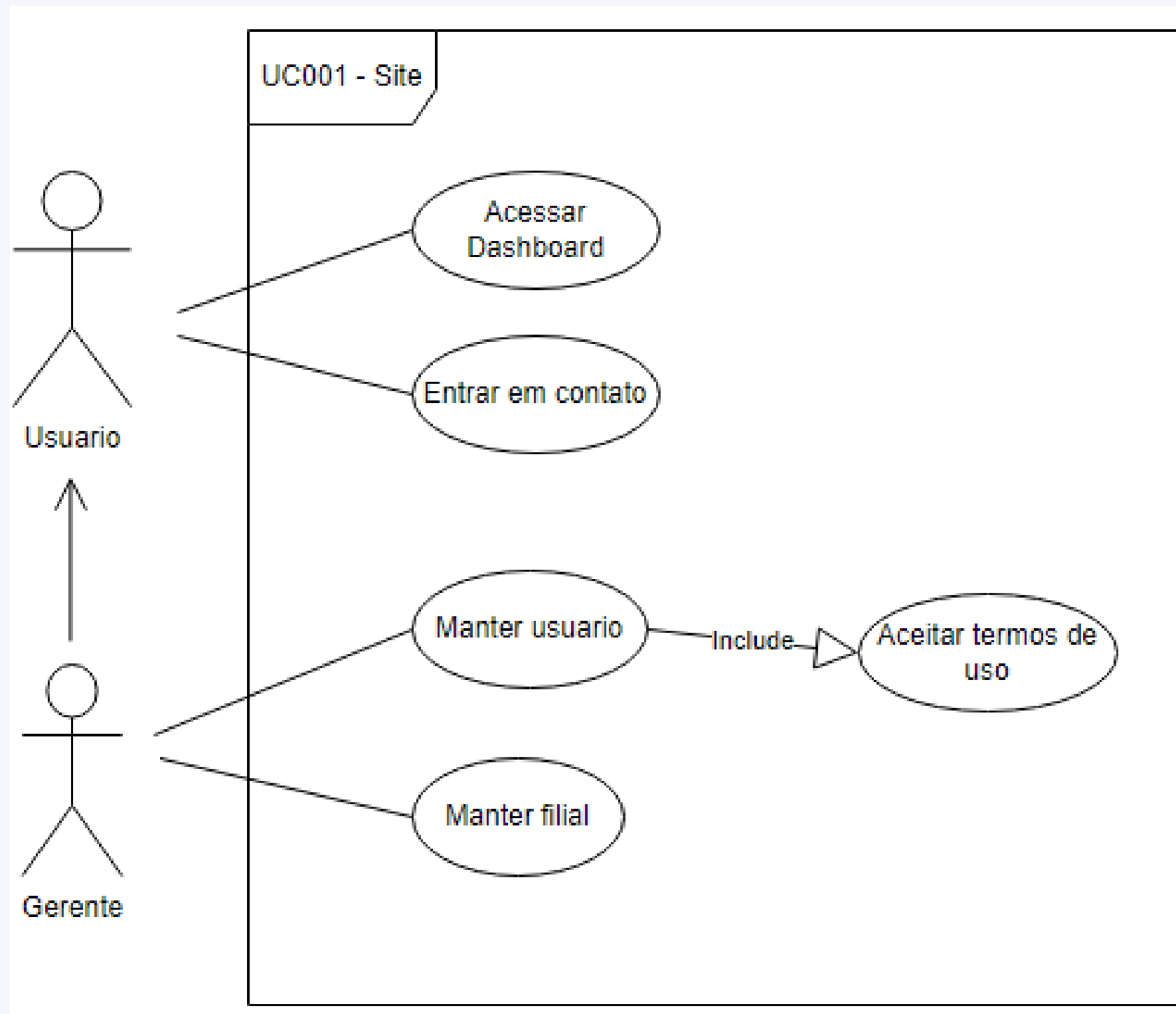
# COMO NOS ORGANIZAMOS?



# MIND MAP



# UML - UNIFIED MODELING LANGUAGE



# UML - UNIFIED MODELING LANGUAGE

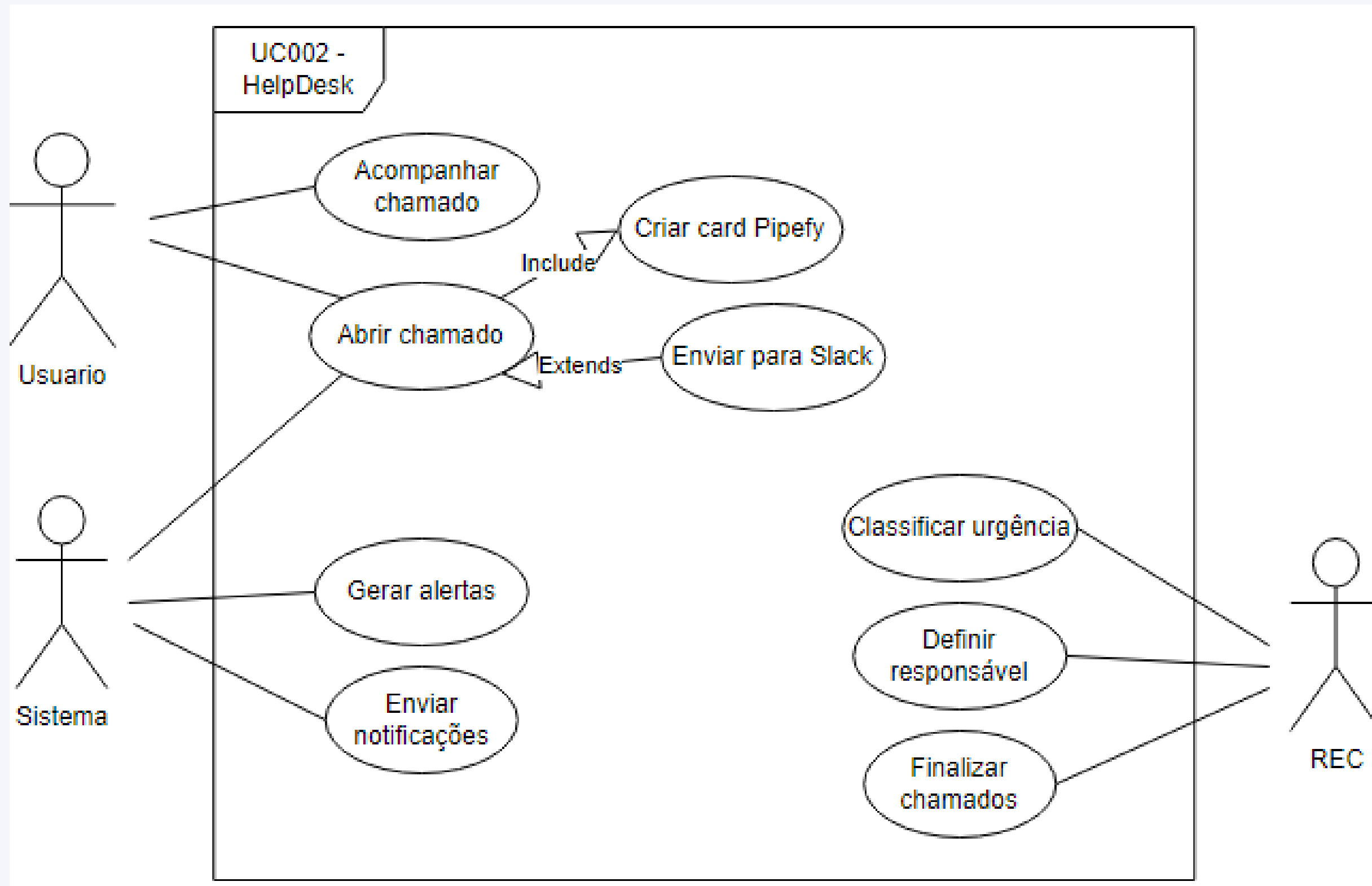


Tabela com dados capturados

Leitura	
idLeitura	INT
fkAtm	INT
CPU	FLOAT
Memoria	FLOAT
Disco	FLOAT
dataHora	DATETIME
Indexes	

Tabela com dados capturados

Atm	
idAtm	INT
fkFilial	INT
nome	VARCHAR(45)
maquina	VARCHAR(45)
sistemaOp	VARCHAR(45)
Indexes	

3ª CRUD - Tela de Cadastro de Usuario

Usuario	
idUsuario	INT
fkFilial	INT
fkLicenca	INT
email	VARCHAR(45)
nome	VARCHAR(45)
senha	VARCHAR(45)
Indexes	



Tabela com dados alinhados

Empresa	
idEmpresa	INT
razaoSocial	VARCHAR(45)
cnpj	CHAR(14)
logradouro	VARCHAR(45)
numero	INT
bairro	VARCHAR(45)
cidade	VARCHAR(45)
estado	CHAR(2)
cep	CHAR(8)
Indexes	

Filial	
idFilial	INT
fkEmpresa	INT
nomeFantasia	VARCHAR(45)
numero	INT
logradouro	VARCHAR(45)
bairro	VARCHAR(45)
cidade	VARCHAR(45)
estado	CHAR(2)
cep	CHAR(8)
complemento	VARCHAR(45)
receita	FLOAT
Indexes	

1ª CRUD - Tela de Cadastro da Filial

Tabela com dados alinhados

Licenca	
idLicenca	INT
fkEmpresa	INT
tipoLicenca	VARCHAR(45)
Indexes	

# MÉTRICAS E DEFINIÇÃO DE ALERTAS

Uso de CPU(%)			
Ideal	Alerta	Emergência	Crítico
De 1 a 40	De 41 a 80	De 81 a 94	De 95 a 100

Temperatura CPU(°C)			
Ideal	Alerta	Emergência	Crítico
De 65 a 85	De 86 a 92	De 93 a 99	100

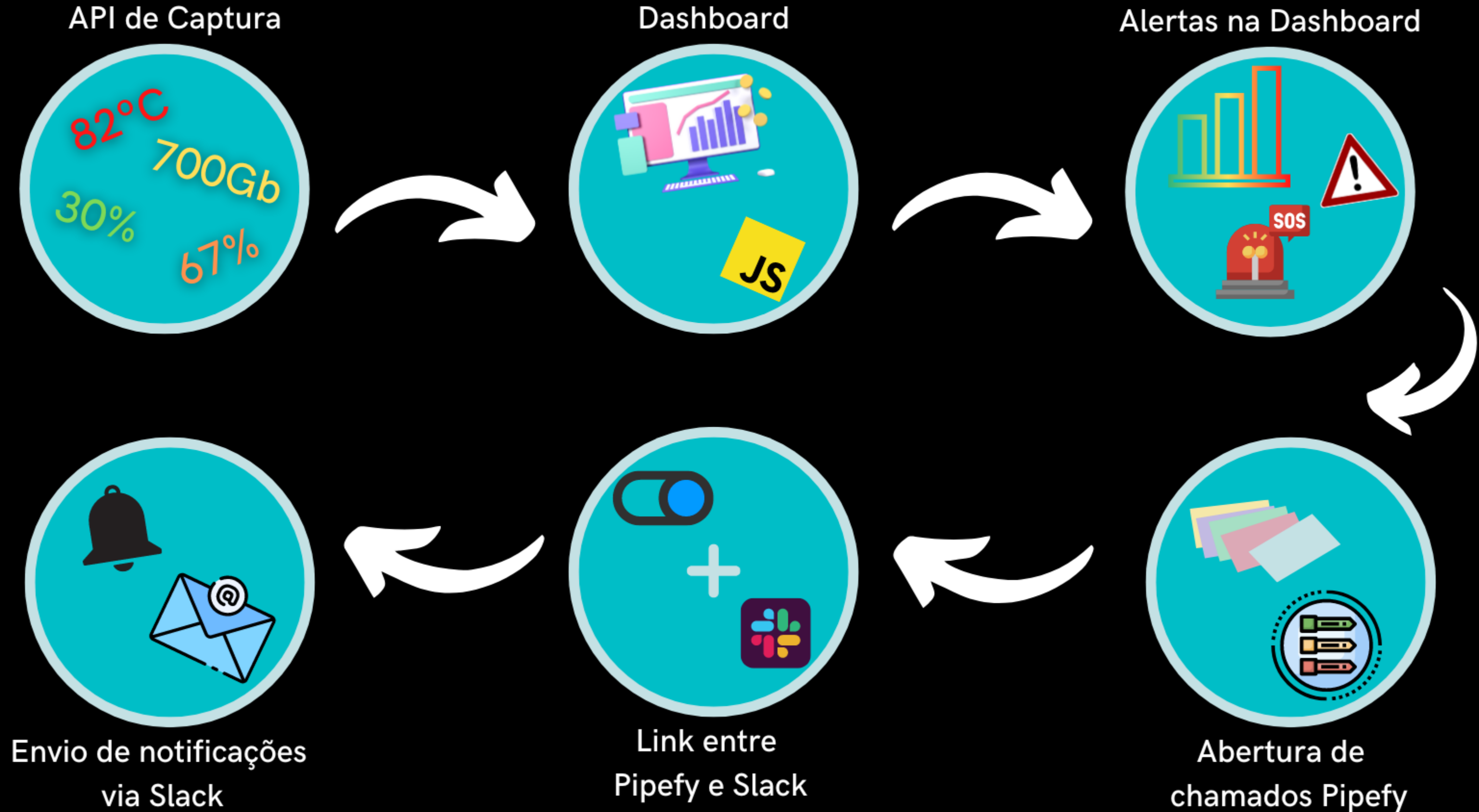
Uso de RAM (%)			
Ideal	Alerta	Emergência	Crítico
Menos de 60	De 60 a 79	De 80 a 85	86 a 100



# PARTINDO PARA O SITE ...



# Fluxo de Alertas



# MINERAÇÃO DE DADOS

**É O PROCESSO DE  
EXPLORAR DADOS À  
PROCURA DE PADRÕES  
CONSISTENTES**



# PROJETO KOTLIN

**API LOOCA  
CAPTURA DE DADOS**

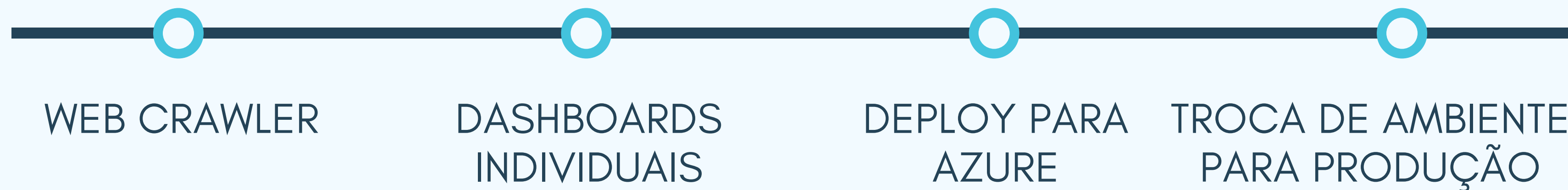


# MÁQUINA VIRTUAL

**EC2**  
**API PYTHON**  
**LISTAR PROCESSOS**



# PLANOS FUTUROS



**AGRADECEMOS A SUA  
ATENÇÃO**



