
A Eng. de Software em meio a uma pandemia

Como desenvolver soluções de sucesso!



INSTITUTO FEDERAL

Baiano

Campus Guanambi

SUMÁRIO

- 1. Como capacitar a equipe para produzir?;**
- 2. Como gerenciar a equipe (ambiente de reuniões, organização de tarefas)?;**
- 3. Técnicas (DevOps e desenvolvimento ágil) e tecnologias (git/gitlab, dockerização, teste e produção)**
- 4. Entrega continua;**
- 5. Melhorias para próximos ciclos de produção.**

Como capacitar a equipe para produzir

FATORES IMPORTANTES:

0. Equipe comprometida e disposta a enfrentar desafios;

Como capacitar a equipe para produzir

FATORES IMPORTANTES:

0. Equipe comprometida e disposta a enfrentar desafios;

NEM SEMPRE VOCÊ TEM UM TIME ESPECIALIZADO
APRENDER IMPLICA COMPROMETIMENTO

Como capacitar a equipe para produzir

FATORES IMPORTANTES:

- 0. Equipe comprometida e disposta a enfrentar desafios;
- 1. Conhecer o cenário a qual a solução será aplicada;

Como capacitar a equipe para produzir

FATORES IMPORTANTES:

- 0. Equipe comprometida e disposta a enfrentar desafios;
- 1. Conhecer o cenário a qual a solução será aplicada;

NO NOSSO CASO, **WEB/MOBILE**

Como capacitar a equipe para produzir

FATORES IMPORTANTES:

- 0. Equipe comprometida e disposta a enfrentar desafios;
- 1. Conhecer o cenário a qual a solução será aplicada;
 - WEB/MOBILE
- 2. **DOMÍNIO** de algumaS tecnologiaS que resolva a demanda;

Como capacitar a equipe para produzir

FATORES IMPORTANTES:

0. Equipe comprometida e disposta a enfrentar desafios;
1. Conhecer o cenário a qual a solução será aplicada;
 - WEB/MOBILE
2. **DOMÍNIO** de algumaS tecnologiaS que resolva a demanda;
 - LinguagenS
 - BibliotecaS e FrameworkS
 - DataBase (Físico e Lógico)
 - Infraestrutura

Como capacitar a equipe para produzir

1. Tecnologias envolvidas;

Linguagens aplicadas



Como capacitar a equipe para produzir

1. Tecnologias envolvidas;

BibliotecaS e FrameworkS



Como capacitar a equipe para produzir

1. Tecnologias envolvidas;

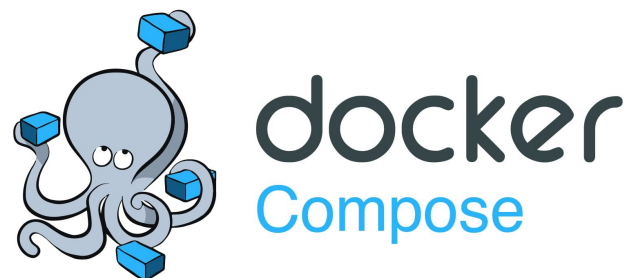
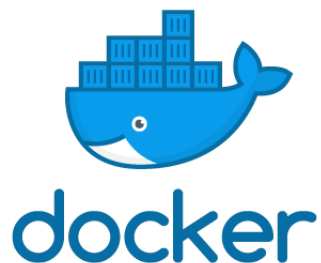
DataBase (Físico e Lógico)



Como capacitar a equipe para produzir

1. Tecnologias envolvidas;

Infraestrutura



Como gerenciar a equipe

ENG. DE SOFTWARE

- GERÊNCIA DE PROJETOS (Fábio)
 - Relacionamento com Stakeholders
 - Tempo:
 - 0. Manter cronograma
 - 1. Fazer cumprir as etapas do projeto
 - Garantir o necessário para equipe se manter ativa trabalhando;
 - Definir prioridade dos artefatos;
- Teste Funcional (Fábio)
 - Testar as saídas geradas pelo sistema
 - Garantir que as demandas do cliente estava sendo executada.



Como gerenciar a equipe

ENG. DE SOFTWARE

- METODO ÁGIL
 - ReMIX (Scrum + Kamban)
 - Product Owner
 - Tasks, Issues, Quadro (via GitLab)
 - Reuniões semanais
 - Entregas semanais

Como gerenciar a equipe

ENG. DE SOFTWARE

- METODO ÁGIL
 - ReMIX (Scrum + Kamban)
 - Product Owner
 - Tasks, Issues, Quadro (via GitLab)
 - Reuniões semanais
 - Entregas semanais
- Modelo de Trabalho;



Como gerenciar a equipe

ENG. DE SOFTWARE

- METODO ÁGIL
 - ReMIX (Scrum + Kamban)
 - Product Owner
 - Tasks, Issues, Quadro (via GitLab)
 - Reuniões semanais
 - Entregas semanais
- Modelo de Trabalho;



Como gerenciar a equipe

ENG. DE SOFTWARE

- METODO ÁGIL
 - ReMIX (Scrum + Kamban)
 - Product Owner
 - Tasks, Issues, Quadro (via GitLab)
 - Reuniões semanais
 - Entregas semanais
- Modelo de Trabalho REMOTO;

O GRANDE DESAFIO



Como gerenciar a equipe

ENG. DE SOFTWARE

COMO ERAM FEITAS AS REUNIÕES???



Como gerenciar a equipe

ENG. DE SOFTWARE

COMO ERAM FEITAS AS REUNIÕES???

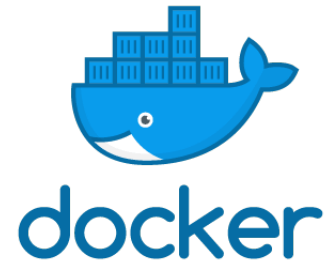
Para mais o que era utilizado?

1. Entender demandas do cliente
2. Treinamentos
3. Debugar
4. Programação em Par
5. Aprender uma determinada tecnologia nova
6. Teste conjunto
7. Compartilhar conhecimento



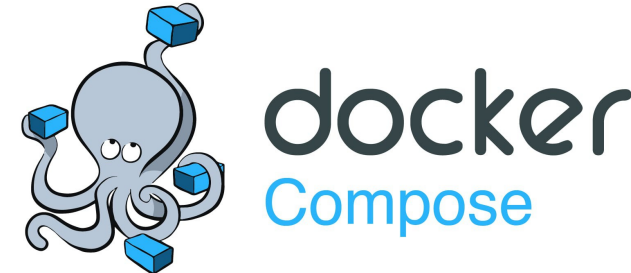
Técnicas (DevOps e desenvolvimento ágil) e tecnologias (git/gitlab, dockerização)

Controle de Versão:
(Local e Remoto)
Teste de Integração
(GitLab-CI)



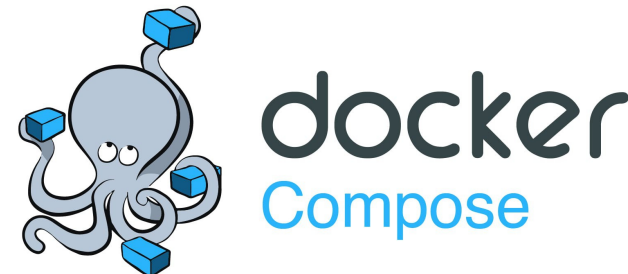
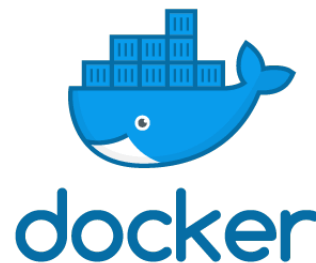
Infraestrutura Limpa
Servidor de Container
(Apenas docker e docker-compose
instalados)

Orquestração de serviços
Servidor de Aplicação
Servidor de arquivos estaticos
Servidor de banco de dados
(usando replicação)



Técnicas (Teste e Produção)

Controle de Versão:
Controlando estagios de
desenvolvimento via
Branchs



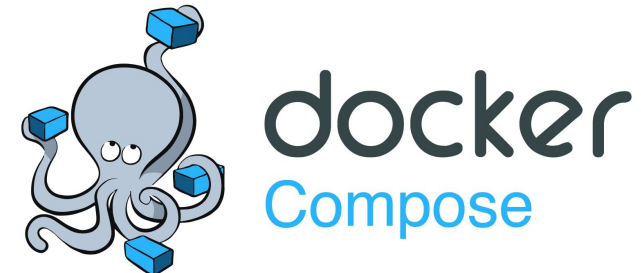
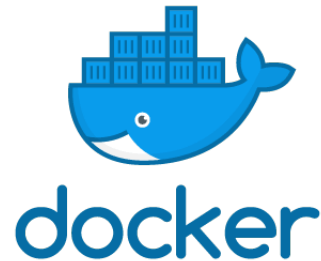
Duas instâncias separadas da aplicação:
Teste (Onde eram aplicadas as novas funcionalidade e Correções).
Quando o teste passa:
Um update era feito na instância de Produção

Entrega contínua

Ciclo semanal de entregas

Controle de branches:

- Master (Princial - Última versão estável)
 - v0.1 (Primeira entrega)
- v1.0 (Primeira versão funcional)
- v2.0 (Segunda versão funcional)



Sempre que uma versão nova era lançada – aplicação de patches, bugfix, features – sua atualização no servidor de teste era feita (manualmente).
Ao passar um update era feito no servidor de produção

Melhorias para próximos ciclos de produção.

Ao final do ciclo de vida de uma versão, era feita reuniões para discutir as novas features/funcionalidades desejadas pelo cliente –
Uma TODO LIST e um CHANGELOG eram necessários



TODO LIST: LISTA DE AFAZERES

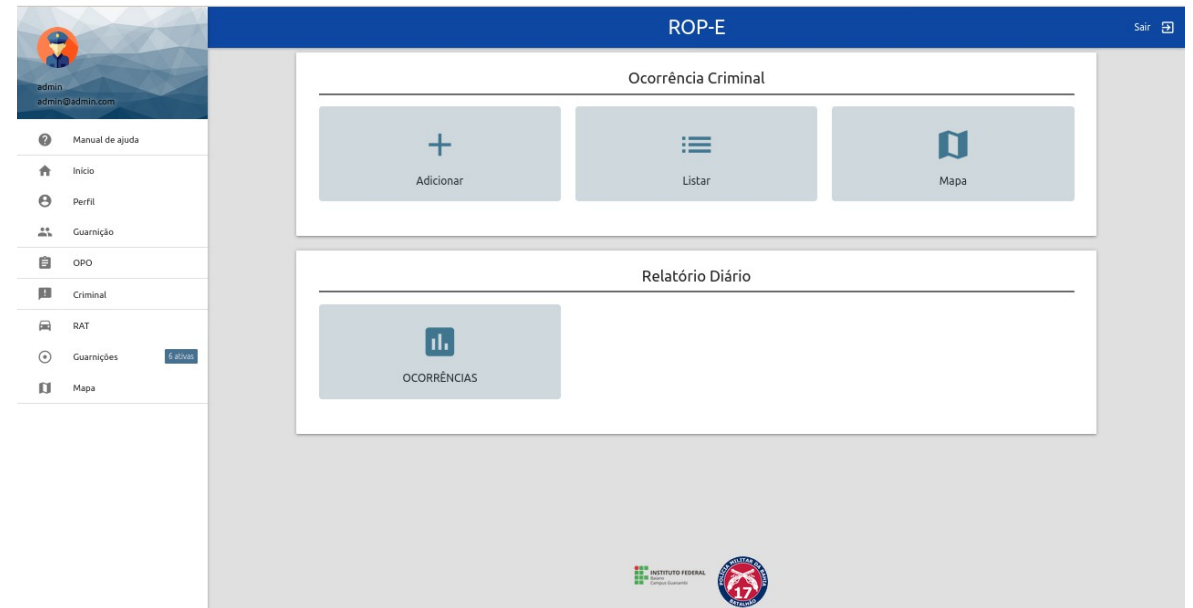
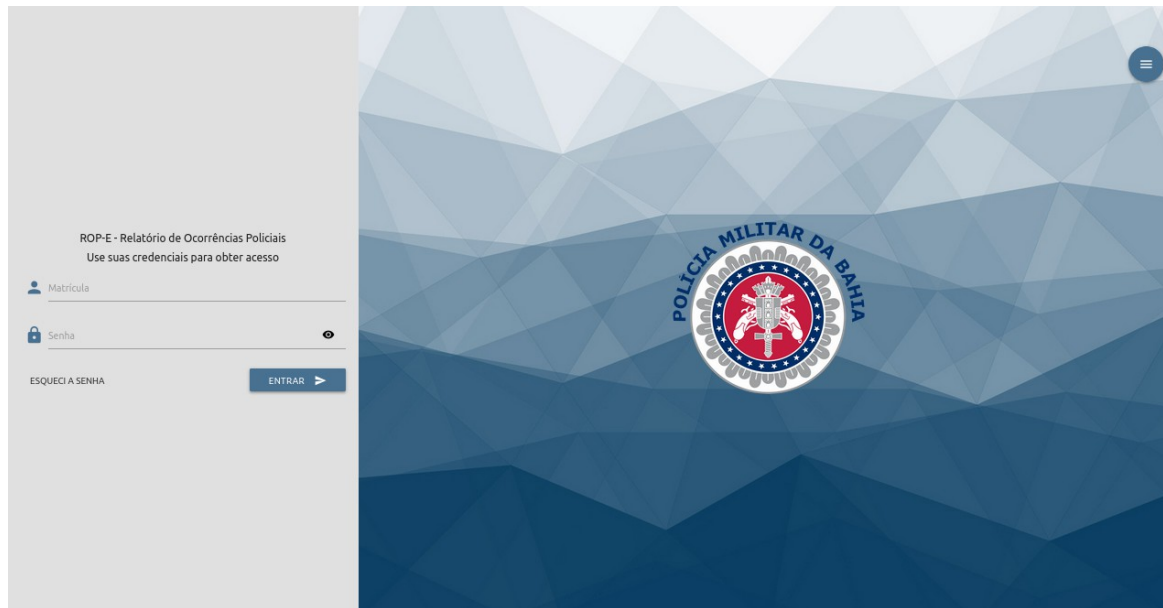
CHANGE LOGS: CONTROLE DE MUDANÇAS

HOUVERAM FALHAS?

CLARO, NADA É TÃO PERFEITO!

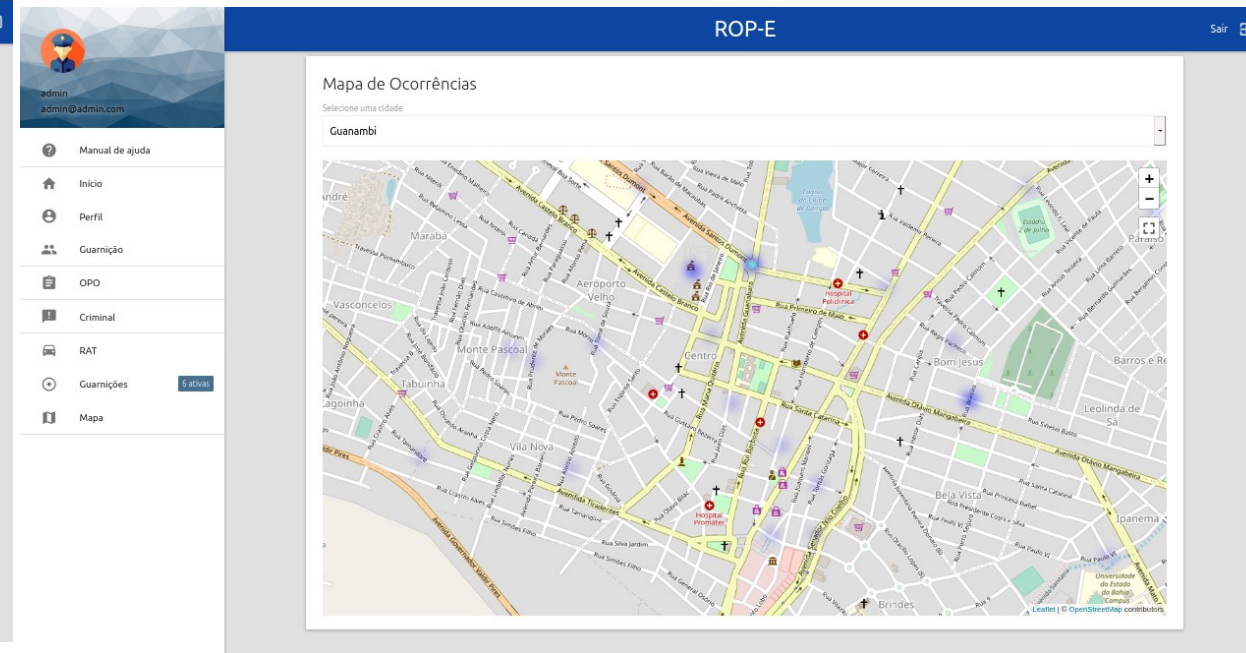
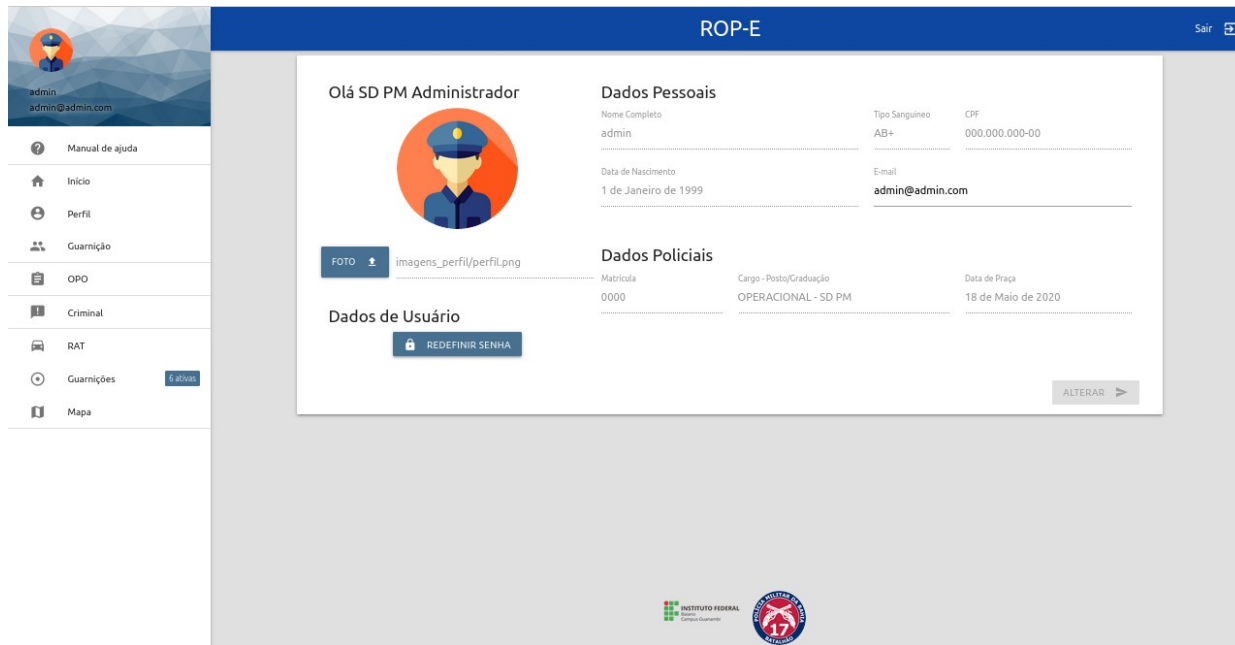
PRODUTO FINAL

INTERFACE WEB



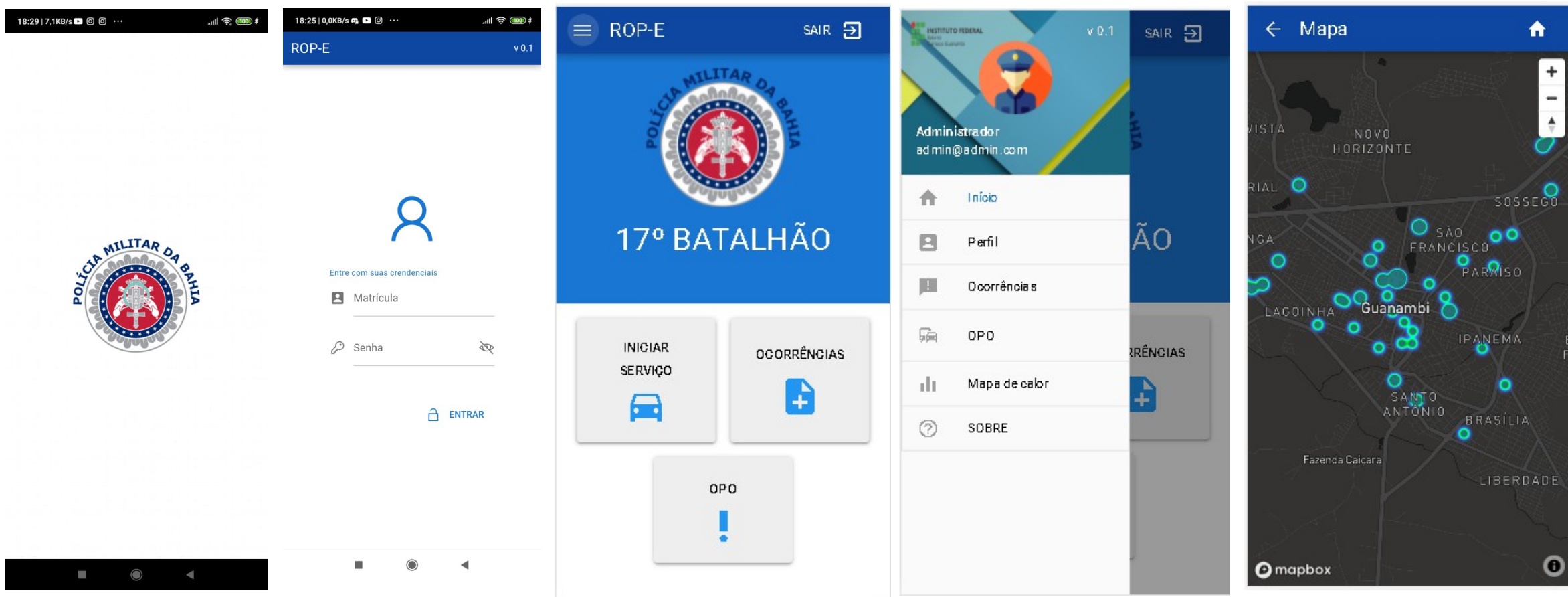
PRODUTO FINAL

INTERFACE WEB



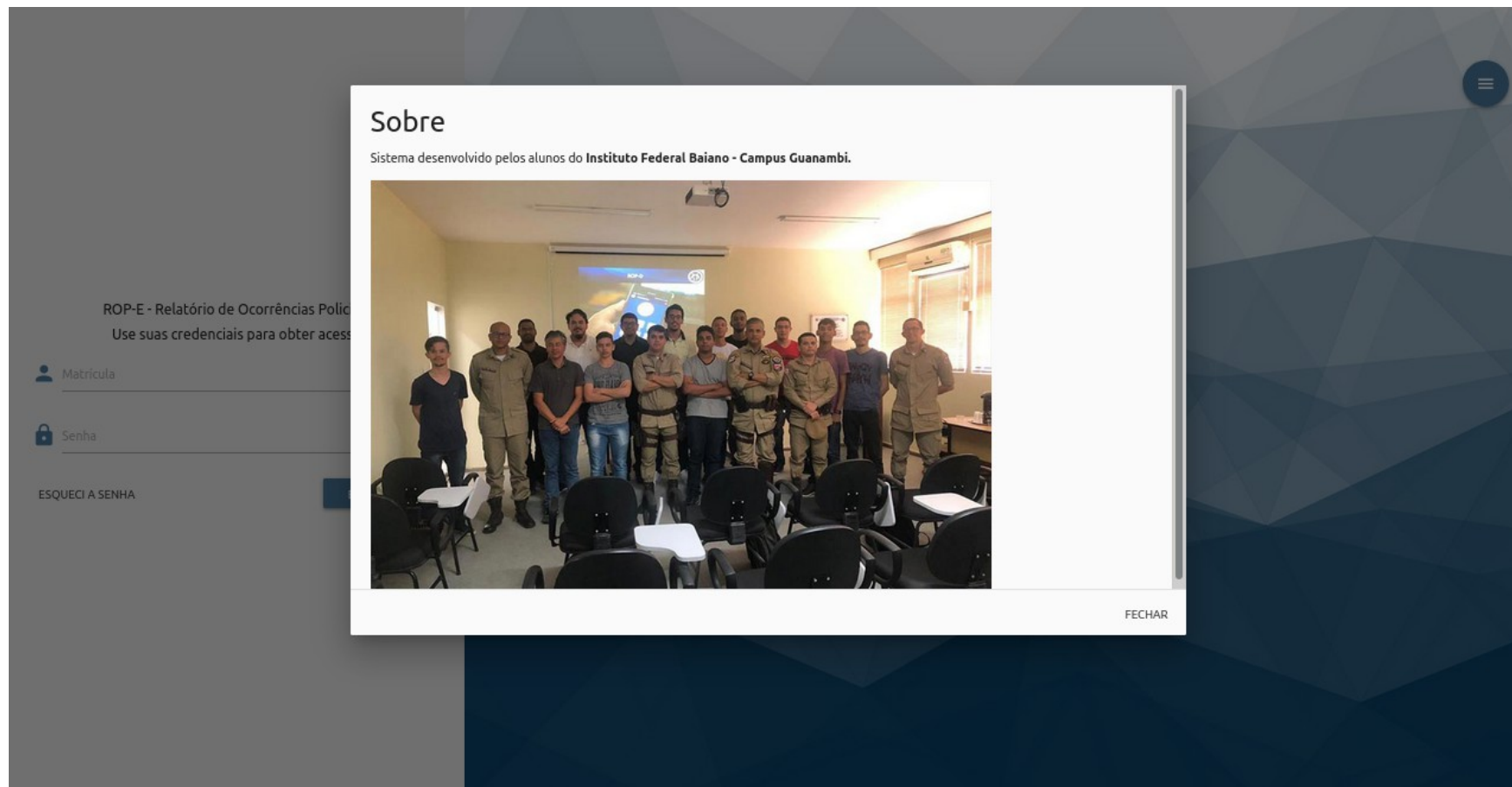
PRODUTO FINAL

INTERFACE MOBILE



PRODUTO FINAL

EQUIPE



WWW.ROP-E.COM:8888

PERGUNTAS?

