**FireWatch Sistema Inteligente de Monitoramento e Combate às Queimadas**

**Equipe**

Leonardo de Farias Silva – RM: 555211

Gustavo Laur – RM: 556603

Giancarlo Cestarolli – RM:555248

**Desafio Escolhido**

**Subtema:** Monitoramento e Resposta Rápida às Queimadas Urbanas e Rurais

O FireWatch foi desenvolvido para resolver a problemática crítica das queimadas no Brasil, que causam:

* Danos ambientais: Destruição de ecossistemas e biodiversidade
* Riscos à saúde pública: Problemas respiratórios e cardiovasculares
* Perdas econômicas: Prejuízos na agricultura, turismo e infraestrutura
* Mudanças climáticas: Emissão de gases de efeito estufa

**Solução Proposta**

O **FireWatch** é uma plataforma tecnológica integrada que combina:

* Detecção Rápida: Sistema de reporte via WhatsApp permite que cidadãos reportem focos de incêndio instantaneamente
* Coordenação Inteligente: Algoritmos de priorização alocam automaticamente equipes de combate baseado em severidade e localização
* Comunicação Efetiva: Notificações automáticas alertam população em áreas de risco
* Monitoramento Geográfico: Mapeamento em tempo real com coordenadas GPS precisas
* Análise Preditiva: Histórico de dados para identificação de padrões e prevenção

**Diferenciais Tecnológicos:**

* Integração WhatsApp via Twilio API
* Interface web responsiva com React + Leaflet
* Backend robusto em Spring Boot
* Simulador Python com algoritmos de otimização
* Arquitetura containerizada com Docker

**Públicoalvo Principal**

**Usuários Primários**

* Corpo de Bombeiros: Profissionais de combate às queimadas
* Defesa Civil: Órgãos de proteção e defesa civil estaduais e municipais
* IBAMA/ICMBio: Órgãos de proteção ambiental federal
* Prefeituras: Gestores públicos municipais

**Usuários Secundários**

* Cidadãos: População em geral para reporte de ocorrências
* Produtores Rurais: Fazendeiros e empresas do agronegócio
* ONGs Ambientais: Organizações de proteção ambiental
* Empresas de Seguros: Avaliação de riscos e sinistros

**Estimativas de Impacto e Dados de Mercado**

**Dimensão do Problema no Brasil**

* 816.000 km²de área queimada em 2023 (INPE)
* R$ 164 bilhões em prejuízos econômicos anuais por queimadas
* 11.000 focos de calor detectados diariamente em período crítico
* 2.300 municípios em situação de emergência por queimadas (2023)

**Mercado Potencial**

* 5.570 municípios brasileiros comoUSuários potenciais
* 27 Corpos de Bombeiros estaduais + 1 distrital
* 220 milhões deUSuários WhatsApp no Brasil
* R$ 2,5 bilhões mercado de tecnologia para emergências no Brasil

**Impacto Estimado da Solução**

* Redução de Tempo de Resposta:
* Tempo atual médio: 4560 minutos
* Tempo com FireWatch: 1525 minutos
* Redução de 60% no tempo de resposta

**Economia de Recursos:**

* 40% redução em custos de combate através de alocação otimizada
* 30% menos área queimada devido à resposta mais rápida
* R$ 65 bilhões economia anual estimada

**Vidas Salvas:**

* 85% redução em acidentes por queimadas em áreas monitoradas
* 70% menos problemas respiratórios em cidades atendidas

**Dados de Validação de Mercado**

* 76% dos bombeiros relatam dificuldades na coordenação de emergências (Pesquisa CBMERJ 2023)
* 89% dos brasileirosUSam WhatsApp diariamente (DataReportal 2024)
* R$ 4,2 milhões investimento médio anual em tecnologia por estado
* 92% das prefeituras buscam soluções digitais para emergências (IBGE Digital 2023)

**PERSONA E MAPA DE EMPATIA**

**Persona Principal:** Comandante Carlos Silva

**Perfil Demográfico**

* Nome: Carlos Eduardo Silva
* Idade: 42 anos
* Cargo: Comandante do Corpo de Bombeiros Região Metropolitana
* Localização: São Paulo, SP
* Formação: Engenharia Civil + Especialização em Gestão de Emergências
* Experiência: 18 anos no Corpo de Bombeiros

**Características Pessoais**

* Casado, 2 filhos
* Renda: R$ 15.000/mês
* Tecnologia:USuário intermediário
* Personalidade: Metódico, responsável, orientado para resultados

**MAPA DE EMPATIA**

Diagrama

O conteúdo gerado por IA pode estar incorreto.

**Requisitos Funcionais**

| **Código** | **Nome** | **Descrição** | **Prioridade** | **Complexidade** |
| --- | --- | --- | --- | --- |
| RF01 | Cadastro deUSuários | Permitir cadastro deUSuários (cidadãos, bombeiros, administradores) com informações pessoais e de contato | Alta | Baixa |
| RF02 | Registro de Ocorrências via WhatsApp | Processar mensagens do WhatsApp com localização GPS e criar ocorrências automaticamente | Crítica | Alta |
| RF03 | Geolocalização de Ocorrências | Extrair e validar coordenadas GPS de mensagens ou entrada manual | Crítica | Média |
| RF04 | Classificação Automática de Severidade | Analisar descrição e atribuir severidade (1 a 10) | Alta | Alta |
| RF05 | Alocação Inteligente de Equipes | Atribuir equipe mais próxima e disponível automaticamente | Crítica | Alta |
| RF06 | Envio de Notificações WhatsApp | Enviar alertas automáticos paraUSuários em raio de risco | Alta | Média |
| RF07 | Dashboard de Monitoramento | Painel com mapa interativo, ocorrências ativas, equipes e estatísticas em tempo real | Alta | Média |
| RF08 | Gestão de Status de Equipes | Permitir atualização de status das equipes e localização | Alta | Baixa |
| RF09 | Histórico de Ocorrências | Registro completo de ocorrências com timestamps e ações | Média | Baixa |
| RF10 | Relatórios de Performance | Gerar relatórios automáticos com métricas de performance | Média | Média |
| RF11 | Gestão de Recursos das Equipes | Controlar inventário de equipamentos, veículos e recursos das equipes | Média | Média |
| RF12 | Sistema de Priorização | Ordenar automaticamente ocorrências por prioridade | Alta | Alta |

**Requisitos NãoFuncionais**

| **Código** | **Nome** | **Descrição** | **Métrica** | **Categoria** |
| --- | --- | --- | --- | --- |
| RNF01 | Performance de Resposta | Processar mensagens e criar ocorrências em até 5s | Tempo de resposta < 5s | Performance |
| RNF02 | Disponibilidade do Sistema | Disponibilidade de 99,9% do tempo | Uptime ≥ 99,9% | Disponibilidade |
| RNF03 | Escalabilidade deUSuários | Suportar até 10.000USuários ativos e 1.000 ocorrências/h | 10.000USuários concorrentes | Escalabilidade |
| RNF04 | Segurança de Dados | Criptografia de comunicações e dados pessoais (AES256) | 100% criptografia | Segurança |
| RNF05 | Compatibilidade Mobile | Web responsivo para dispositivos móveis | Compatível com 95% dos dispositivos | Usabilidade |
| RNF06 | Integração WhatsApp | Integração estável com Twilio API e taxa de entrega ≥ 99% | Taxa de entrega ≥ 99% | Integração |
| RNF07 | Backup e Recuperação | Backup automático diário e recuperação em até 2h | RTO=2h, RPO=24h | Recuperação |
| RNF08 | Localização Geográfica | Precisão GPS de no máximo 10 metros | Precisão ≤ 10 metros | Precisão |
| RNF09 | Conformidade Legal | Conformidade com LGPD | 100% conformidade | Conformidade |
| RNF10 | Monitoramento de Sistema | Logs detalhados e métricas de monitoramento para operações críticas | 100% das operações logadas | Observabilidade |
| RNF11 | Usabilidade da Interface | Usuários realizam tarefas principais em até 3 cliques/toques | Máximo 3 interações | Usabilidade |
| RNF12 | Tolerância a Falhas | Operar mesmo com falha de componentes individuais | Zero singlepointoffailure | Confiabilidade |

**Requisitos Técnicos**

| **Código** | **Nome** | **Descrição** | **Tecnologia** | **Categoria** |
| --- | --- | --- | --- | --- |
| RT01 | Arquitetura de Microserviços | Uso de microserviços para escalabilidade e manutenibilidade | Spring Boot + Docker | Arquitetura |
| RT02 | Base de Dados Relacional | Utilizar MySQL 8.0+ como SGBD principal | MySQL 8.0+ | Persistência |
| RT03 | API RESTful | Expor APIs REST seguindo padrões OpenAPI 3.0 | Spring Boot REST + OpenAPI 3.0 | Integração |
| RT04 | Interface Web Responsiva | Interface web em React.js com mapas interativos | React.js + Leaflet | Frontend |
| RT05 | Integração API Twilio | Integração com Twilio WhatsApp Business API | Twilio WhatsApp API | Integração |
| RT06 | Containerização Docker | Containerização completa para deployment | Docker + Docker Compose | DevOps |
| RT07 | Cache Redis | Uso de Redis para cache e sessões deUSuário | Redis 7.0+ | Performance |
| RT08 | Monitoramento Prometheus | Métricas personalizadas com Prometheus e Grafana | Prometheus + Grafana | Observabilidade |
| RT09 | Processamento de Coordenadas | Algoritmos para extração e validação de coordenadas GPS | Regex + GeoDB | Processamento |
| RT10 | Algoritmos de Otimização | Algoritmos (heap, queue) para priorização e alocação de recursos | Python + Algoritmos de Grafos | Algoritmos |
| RT11 | Versionamento de API | Suporte a versionamento de API | Spring Boot API Versioning | Integração |
| RT12 | Logs Estruturados | Logging estruturado em formato JSON | Logback + JSON Encoder | Observabilidade |

**Regras de Negócio**

| **Código** | **Nome** | **Regra / Lógica** | **Impacto** |
| --- | --- | --- | --- |
| RN01 | Prioridade por Severidade | IF severidade >= 8 THEN prioridade = CRÍTICA | Alocação automática |
| RN02 | Alocação de Equipe Mais Próxima | MIN(distancia) WHERE status\_equipe = 'DISPONIVEL' | Otimização de resposta |
| RN03 | Raio de Alerta para População | IF severidade >= 6 THEN notify\_users\_in\_radius(5km) | Segurança da população |
| RN04 | Limite de Ocorrências por Equipe | IF count\_ocorrencias\_equipe < 3 THEN assign\_team | Gestão de recursos |
| RN05 | Timeout de Resposta de Equipe | IF timeout > 10min THEN reallocate\_team | Garantia de atendimento |
| RN06 | Validação de Coordenadas | IF lat BETWEEN 34,5 AND long BETWEEN 74,32 THEN valid | Qualidade dos dados |
| RN07 | Cadastro Automático deUSuário | IF telefone NOT EXISTS THEN create\_user(tipo='CIDADAO') | Facilitação do processo |
| RN08 | Status de Equipe Automático | IF assign\_ocorrencia THEN status\_equipe = 'EM\_ACAO' | Controle operacional |
| RN09 | Finalização Automática de Ocorrência | IF last\_update > 24h THEN status = 'FINALIZADA' | Gestão de dados |
| RN10 | Restrição de Acesso por Tipo | IFUSer\_type IN ['BOMBEIRO','ADMIN'] THEN allow\_update | Segurança operacional |
| RN11 | Custo Calculado por Severidade | custo = (severidade \* 1000) + (membros \* 500) | Controle orçamentário |
| RN12 | Backup de Dados Críticos | IF status IN ['ABERTA','EM\_ATENDIMENTO'] THEN replicate\_realtime | Continuidade operacional |

**PRODUCT BACKLOG**

**Épicos do Sistema**

**ÉPICO 1: Gestão deUSuários e Autenticação**

Desenvolvimento do sistema de cadastro, autenticação e gestão de perfis deUSuários

**ÉPICO 2: Sistema de Detecção e Registro de Ocorrências**

Implementação do core do sistema para recepção, processamento e registro de ocorrências

**ÉPICO 3: Alocação Inteligente de Recursos**

Desenvolvimento dos algoritmos de otimização para alocação de equipes e recursos

**ÉPICO 4: Sistema de Notificações e Alertas**

Implementação do sistema de comunicação via WhatsApp e outras notificações

**ÉPICO 5: Interface de Monitoramento e Dashboards**

Desenvolvimento da interface web com mapas interativos e painéis de controle

**ÉPICO 6: Integração WhatsApp e APIs Externas**

Implementação das integrações com Twilio e outras APIs necessárias

**Histórias deUSuário**

**US001 Cadastro deUSuário Cidadão**

* Como cidadão
* Eu quero me cadastrar no sistema fornecendo meus dados pessoais
* Para que eu possa reportar ocorrências de incêndio e receber alertas

**Critérios de Aceite:**

* Formulário deve solicitar: nome, telefone, email, endereço, cidade
* Sistema deve validar formato do telefone e email
* Usuário deve receber confirmação via WhatsApp após cadastro
* Senha deve ser gerada automaticamente e enviada por WhatsApp

**Definition of Done:**

* Formulário de cadastro implementado
* Validações de campo funcionando
* Integração com WhatsApp para confirmação
* Testes unitários e de integração
* Documentação da API

**US002 Registro de Ocorrência via WhatsApp**

* Como cidadão
* Eu quero reportar um incêndio enviando uma mensagem via WhatsApp
* Para que as autoridades sejam notificadas rapidamente

**Critérios de Aceite:**

* Sistema deve aceitar mensagens de texto com coordenadas
* Sistema deve aceitar localização compartilhada do WhatsApp
* Confirmação automática deve ser enviada aoUSuário
* Ocorrência deve aparecer no sistema em tempo real

**Definition of Done:**

* Webhook do Twilio configurado
* Parser de coordenadas implementado
* Criação automática de ocorrência
* Resposta automática implementada
* Logs de auditoria funcionando

**US003 Visualização de Mapa de Ocorrências**

* Como bombeiro
* Eu quero visualizar todas as ocorrências ativas em um mapa interativo
* Para que eu possa ter visão geral da situação operacional

**Critérios de Aceite:**

* Mapa deve mostrar ocorrências com diferentes cores por severidade
* Clicar na ocorrência deve mostrar detalhes
* Mapa deve atualizar automaticamente a cada 30 segundos
* Filtros por status e severidade devem estar disponíveis

**Definition of Done:**

* Integração Leaflet implementada
* API de listagem de ocorrências
* Update automático via WebSocket
* Filtros funcionais
* Interface responsiva

**US004 Atribuição Manual de Equipe**

* Como comandante
* Eu quero atribuir manualmente uma equipe específica a uma ocorrência
* Para que eu possa override da alocação automática quando necessário

**Critérios de Aceite:**

* Lista de equipes disponíveis deve ser apresentada
* Status da equipe deve mudar para 'EM\_ACAO' automaticamente
* Equipe deve receber notificação da atribuição
* Histórico da atribuição deve ser registrado

**Definition of Done:**

* Interface de seleção de equipe
* API de atribuição implementada
* Notificação para equipe
* Atualização de status automática
* Log de auditoria

**US005 Recebimento de Alerta de Risco**

* Como cidadão
* Eu quero receber alertas via WhatsApp quando houver risco de incêndio na minha região
* Para que eu possa tomar medidas preventivas

**Critérios de Aceite:**

* Alertas devem ser enviados paraUSuários em raio de 5km
* Mensagem deve conter localização e nível de risco
* Usuário deve poder confirmar recebimento
* Frequência máxima de 1 alerta por hora porUSuário

**Definition of Done:**

* Algoritmo de geolocalização
* Template de mensagem de alerta
* Rate limiting implementado
* Confirmação de leitura
* Métricas de entrega

**US006 Dashboard de Estatísticas**

* Como administrador
* Eu quero visualizar estatísticas de desempenho do sistema
* Para que eu possa monitorar a eficiência das operações

**Critérios de Aceite:**

* Gráficos de ocorrências por período, região e severidade
* Métricas de tempo de resposta das equipes
* Taxa de resolução de ocorrências
* Comparativo mensal de performance

**Definition of Done:**

* Componentes de gráficos implementados
* APIs de métricas desenvolvidas
* Cálculos de KPIs funcionando
* Interface responsiva
* Cache de dados otimizado

**US007 Gestão de Status de Equipe**

* Como bombeiro
* Eu quero atualizar o status da minha equipe (disponível/indisponível)
* Para que o sistema possa alocar recursos corretamente

**Critérios de Aceite:**

* Interface simples com toggle de status
* Localização atual deve ser atualizada automaticamente
* Mudanças devem ser refletidas imediatamente no sistema
* Histórico de mudanças deve ser mantido

**Definition of Done:**

* Interface de toggle implementada
* Geolocalização automática
* API de atualização de status
* Sincronização em tempo real
* Auditoria de mudanças

**US008 Busca e Filtro de Ocorrências**

* Como comandante
* Eu quero buscar e filtrar ocorrências por data, região, status e severidade
* Para que eu possa analisar padrões e tomar decisões estratégicas

**Critérios de Aceite:**

* Filtros múltiplos devem funcionar simultaneamente
* Busca por texto livre na descrição
* Exportação de resultados em CSV
* Paginação para grandes volumes de dados

**Definition of Done:**

* Interface de filtros implementada
* API de busca com parâmetros múltiplos
* Exportação CSV funcionando
* Paginação implementada
* Performance otimizada

**US009 Histórico Detalhado de Ocorrência**

* Como investigador
* Eu quero visualizar o histórico completo de uma ocorrência
* Para que eu possa analisar o desenvolvimento do caso

**Critérios de Aceite:**

* Timeline com todas as ações realizadas
* Responsáveis por cada ação identificados
* Anexos e comentários organizados cronologicamente
* Possibilidade de adicionar observações

**Definition of Done:**

* Interface de timeline implementada
* Rastreamento de todas as mudanças
* Sistema de comentários
* Upload de anexos
* Auditoria completa

**US010 Configuração de Perfil deUSuário**

* ComoUSuário do sistema
* Eu quero editar meu perfil e preferências de notificação
* Para que eu possa manter meus dados atualizados

**Critérios de Aceite:**

* Edição de dados pessoais (exceto telefone)
* Configuração de tipos de notificação preferidos
* Definição de raio de alertas desejado
* Alteração de senha

**Definition of Done:**

* Formulário de edição implementado
* Validações de dados
* Criptografia de senha
* Confirmação de mudanças
* Interface de preferências

**US011 Relatório de Performance de Equipe**

* Como comandante
* Eu quero gerar relatórios de performance individual das equipes
* Para que eu possa avaliar e melhorar a eficiência operacional

**Critérios de Aceite:**

* Métricas por equipe: tempo médio de resposta, ocorrências atendidas
* Comparativo entre equipes da mesma região
* Gráficos de tendência temporal
* Exportação em PDF

**Definition of Done:**

* Cálculo de métricas por equipe
* Interface de relatórios
* Gráficos comparativos
* Geração de PDF
* Agendamento de relatórios

**US012 Sistema de Backup Manual**

* Como administrador do sistema
* Eu quero executar backup manual dos dados críticos
* Para que eu possa garantir a segurança das informações

**Critérios de Aceite:**

* Backup deve incluir todas as tabelas principais
* Processo deve ser executado em background
* Notificação de conclusão deve ser enviada
* Backup deve ser validado automaticamente

**Definition of Done:**

* Script de backup implementado
* Interface para execução manual
* Processamento assíncrono
* Validação de integridade
* Notificações de status

**US013 Alocação Automática Inteligente**

* Como sistema
* Eu quero alocar automaticamente equipes baseado em proximidade e disponibilidade
* Para que o tempo de resposta seja otimizado

**Critérios de Aceite:**

* Algoritmo deve considerar distância e disponibilidade
* Prioridade deve ser dada à severidade da ocorrência
* Equipe deve ser notificada automaticamente
* Fallback para próxima equipe se não houver resposta

**Definition of Done:**

* Algoritmo de otimização implementado
* Cálculo de distâncias funcionando
* Sistema de fallback
* Notificação automática
* Logs de decisão

**US014 Monitoramento de Saúde do Sistema**

* Como administrador técnico
* Eu quero monitorar a saúde de todos os componentes do sistema
* Para que eu possa prevenir falhas e garantir disponibilidade

**Critérios de Aceite:**

* Health checks de todos os serviços
* Métricas de performance em tempo real
* Alertas automáticos para falhas
* Dashboard de status do sistema

**Definition of Done:**

* Endpoints de health check
* Integração Prometheus
* Alertas configurados
* Dashboard Grafana
* Documentação de monitoramento

**US015 Integração com Sistemas Externos**

* Como administrador
* Eu quero integrar o FireWatch com sistemas do Corpo de Bombeiros
* Para que os dados sejam sincronizados automaticamente

**Critérios de Aceite:**

* API deve permitir sincronização bidirecional
* Autenticação segura deve ser implementada
* Logs de integração devem ser mantidos
* Tratamento de erro e retry automático

**Definition of Done:**

* APIs de integração desenvolvidas
* Autenticação OAuth implementada
* Sistema de retry
* Logs de auditoria
* Documentação técnica

**US016 Simulação de Cenários de Emergência**

* Como comandante
* Eu quero simular cenários de múltiplas ocorrências simultâneas
* Para que eu possa treinar equipes e testar procedimentos

**Critérios de Aceite:**

* Criação de cenários com múltiplas ocorrências
* Simulação de disponibilidade de equipes
* Cronômetro para medir tempo de resposta
* Relatório de performance da simulação

**Definition of Done:**

* Interface de criação de cenários
* Engine de simulação
* Cronometragem automática
* Relatórios de simulação
* Modo de treinamento isolado

**US017 Configuração de Alertas Personalizados**

* Como comandante
* Eu quero configurar regras personalizadas para envio de alertas
* Para que eu possa adaptar o sistema às necessidades específicas da região

**Critérios de Aceite:**

* Criação de regras baseadas em severidade, localização e horário
* Definição de destinatários específicos para cada regra
* Teste de regras antes da ativação
* Histórico de alertas enviados por regra

**Definition of Done:**

* Interface de configuração de regras
* Engine de processamento de regras
* Sistema de teste
* Rastreamento de execução
* Validação de regras