



EQUIPE FOCUS QUEST

API 2º Semestre - DSM

Sistema de Chamados

SÃO JOSÉ DOS CAMPOS - SP

2023

Sumário

INTRODUÇÃO	3
OBJETIVO	4
TECNOLOGIAS ESCOLHIDAS.....	5
METODOLOGIA	6
Levantamento de Requisitos	6
Product Backlog	6
Sprint backlog	6
Análise e Design	7
Desenvolvimento	7
Modelo de dados	7
Testes e Validação	8
Documentação Completa	8
Usabilidade e UX/UI	8
Implantação e Homologação	8
Avaliação e Ajustes	8
CONCLUSÃO	9
ANEXOS	10
Product Backlog	10
Sprint Backlog	11
Modelo de dados	12

1 INTRODUÇÃO

O projeto visa à criação de uma solução de software avançada, para acompanhamento e controle no atendimento a chamadas de serviços à universidades e que seja capaz de fornecer informações relacionadas à disponibilização de funcionalidades de planejamento.

Adicionalmente, busca-se a implementação de relatórios analíticos, que representam pontos essenciais de agregação de valor ao projeto.

Para atingir esses objetivos, o grupo constituído pelo Development Team Gustavo, Katiane, Lilian, Marcus, Sarah, Product Owner Maria Gabriela e Scrum Master Silmara, desenvolverão, no prazo de 4 sprints, cada qual com 21 dias, uma documentação completa (a qual refere-se este documento), um Sistema para Gestão de Chamados de Serviços, utilizando-se de conhecimentos e práticas essenciais, conforme detalhado na metodologia a seguir.

2 OBJETIVO

O objetivo principal deste projeto é desenvolver uma solução robusta e escalável que aprimore significativamente a Gestão de Chamadas de Serviços. Os objetivos específicos incluem:

- Implementar funcionalidades de planejamento, acompanhamento e controle de chamadas de serviços: Desenvolver um sistema que permita aos usuários planejar, rastrear e controlar o progresso das chamadas de serviço de forma eficiente e intuitiva.
- Integrar relatórios analíticos: Implementar recursos de geração de relatórios avançados que forneçam insights valiosos para a tomada de decisões informadas.
- Documentação abrangente: Criar documentação detalhada do sistema, incluindo user stories, wireframes, documentação de API e rotas, bem como requisitos funcionais e não funcionais.
- Garantir usabilidade e experiência do usuário (UX/UI) de alta qualidade: Priorizar uma interface amigável e eficaz que melhore a experiência dos usuários.

3 TECNOLOGIAS ESCOLHIDAS

O produto será desenvolvido utilizando-se as seguintes tecnologias:

- React.js: uma biblioteca do JavaScript que permite a criação de interfaces gráficas com a sintaxe JSX;
- HTML e CSS: linguagens de marcação e estilização, respectivamente, de páginas web;
- Node.js: uma tecnologia que permite o uso do JavaScript no “lado do servidor” (pensando na arquitetura cliente-servidor);
- TypeScript: um superset de JavaScript que permite o uso de tipagem e aplicação de conceitos do paradigma da Programação Orientada a Objetos;
- Git: um sistema de versionamento para as entregas a cada sprint, onde os códigos e versões estão disponíveis também através do Git Hub da equipe (acesso por: <https://github.com/FocusQuest/FocusQuestAPI.git>).

4 METODOLOGIA

Para atingir os objetivos delineados, será adotada uma metodologia que compreende as seguintes etapas:

4.1 Levantamento de Requisitos

Realizar entrevistas e reuniões com as partes interessadas para entender completamente os requisitos funcionais e não funcionais do sistema.

Assim, foi iniciada a documentação de todo e qualquer requisito levantado, construindo em cima deles User Stories e classificando-os em requisitos funcionais e não funcionais.

4.1.1 Product Backlog

Após concentrar certa quantidade de requisitos, a realização da organização deles em um backlog de produto ordenado e priorizado se fez necessária, acordando sempre junto do cliente o que é imprescindível e o que é desejável, resultando no Product Backlog descrito pela figura 6.1 na seção de Anexos.

4.1.2 Sprint backlog

Com o Product Backlog em mãos, foi realizada a cerimônia de Sprint Planning, onde todo o time se encontra para estimar as tarefas e quebrá-las em tarefas menores para melhor execução de cada uma, separando-as entre as 4 sprints que irão decorrer até o fim do projeto. A separação de cada requisito pelas sprints se encontra ilustrada na imagem 6.2 na

seção de Anexos.

4.2 Análise e Design

Com base nos requisitos, criar wireframes e especificações detalhadas de design de interface do usuário. O protótipo navegável está disponível no figma (acesso por:

<https://www.figma.com/proto/OhZnJurR49NrVAEmvbk0Zt/Wireframe-API-2sem?type=design&node-id=52-2&t=cp3tF9aUk2F9STFi-1&scaling=min-zoom&page-id=0%3A1&starting-point-node-id=52%3A2&show-proto-sidebar=1&mode=design>)

4.4 Desenvolvimento

Implementar o sistema utilizando linguagens de programação orientadas a objetos, como TypeScript e Node.js, com atenção ao mapeamento objeto-relacional para garantir a integração eficaz com um banco de dados relacional.

4.4.1 Modelo de dados

Elaboração do modelo de dados para representação abstrata e estruturada de como os dados serão organizados, armazenados, processados e acessados no sistema. Ele desempenha um papel fundamental no projeto de bancos de dados e sistemas de informação, pois define como os dados serão manipulados e estruturados para atender às necessidades de uma aplicação específica.

O modelo lógico está representado no anexo 6.3 da seção de anexos.

4.5 Testes e Validação

Realizar testes de funcionalidade e desempenho, além de verificar a conformidade com os requisitos estabelecidos.

4.6 Documentação Completa

Preparar documentação abrangente, abordando todos os aspectos do projeto.

4.7 Usabilidade e UX/UI

Garantir que a interface do usuário seja projetada com foco na usabilidade e na experiência do usuário.

4.8 Implantação e Homologação

Implementar a solução em um ambiente de homologação com capacidade limitada para avaliação e validação pelo cliente.

4.9 Avaliação e Ajustes

Coletar feedback do cliente e realizar ajustes conforme necessário para garantir que a solução atenda plenamente às expectativas.

5 CONCLUSÃO

Após aprovação do cliente, implantar o sistema em ambiente de produção e oferecer suporte contínuo.

6 ANEXOS

6.1 Product Backlog

PRODUCT BACKLOG				
User Story ID	User Story	Prioridade	Sprint	Critério de Aceitação
US001	Eu como cliente desejo visualizar o protótipo do projeto para ter noção de como o projeto ficará	Alta	1	Desenvolver um protótipo do projeto utilizando o Figma contendo a estrutura do projeto
US002	Eu como cliente desejo visualizar um modelo de formulário para ter uma ideia de quais serão as informações fornecidas na página de abertura de chamados	Alta	1	Elaborar um modelo de formulário com as informações que serão preenchidas pelo usuário como nome, título do problema, descrição, categoria e prioridade
US003	Eu como usuário desejo poder solicitar suporte técnico através de um formulário com título e descrição do problema para que possa obter ajuda	Alta	2	Desenvolver um formulário em código com opções de inserção de informações por categoria e descritivas por parte do usuário
US004	Eu como usuário desejo poder realizar login no sistema de acordo com a minha função afim de obter acesso às opções pertinentes a mim no sistema	Alta	2	Desenvolver uma página com a opção de login para os três tipos de usuários separadamente
US005	Eu como administrador desejo registrar os usuários no sistema para poder identificar cada acesso de usuário	Alta	2	Criar uma página de registros onde será possível cadastrar cada usuário de acordo com sua função no sistema contendo as informações para identificação de cada um
US006	Eu como cliente desejo poder informar a prioridade e a categoria dos tickets para poder indicar a gravidade do problema	Média	2	Desenvolver dropdowns para selecionar a prioridade e a categoria dos tickets abertos pelo usuário
US007	Eu enquanto administrador desejo desenvolver o banco de dados físico para poder visualizar o armazenamento dos dados e seus relacionamentos	Alta	2	Com base no modelo conceitual e lógico de dados, criar o modelo físico do banco de dados
US008	Eu como cliente desejo a integração do Banco de dados a fim de poder armazenar e acessar dados inseridos no sistema	Alta	3	Realizar a integração do banco de dados ao projeto conectando o back-end ao front-end
US009	Eu como técnico desejo poder visualizar o histórico de chamados para poder acompanhar o andamento dos processos	Média	3	Desenvolver a opção de visualizar os chamados abertos, seu andamento, por quem foi aberto, quem é o responsável pela sua solução, data de abertura e data de fechamento
US010	Eu como técnico desejo poder adicionar comentários aos chamados para poder me comunicar com o cliente caso necessário	Baixa	3	Mostrar uma opção onde é possível adicionar comentários para o cliente
US011	Eu como técnico desejo rastrear o status de cada ticket, incluindo etapas como "aberto", "atribuído", "em andamento" e "concluído" para ter um melhor controle dos chamados	Média	4	Desenvolver uma função de filtro onde seja possível buscar chamados de acordo com o seu status de andamento
US012	Eu como administrador quero poder gerar e visualizar relatórios para monitorar estatísticas, incluindo métricas como tempo de resolução de chamados, taxa de resolução, prazos etc.	Média	4	Desenvolver uma função que possibilite gerar automaticamente relatórios com bases nos dados do sistema referente aos chamados
US013	Eu como usuário desejo poder acessar uma área com perguntas frequentes afim de evitar abertura de chamados desnecessários e resolver rapidamente um problema sempre que possível	Média	4	Criar um tópico com perguntas frequentes dos usuários do sistema para exibição na página inicial do sistema
US014	Eu como programador desejo resolver todos os bugs que possam aparecer no sistema para poder garantir o bom desempenho e excelência do projeto	Alta	4	Fazer a checagem e teste de todo o sistema para verificar o seu funcionamento

6.2 Sprint Backlog

BACKLOG - Sprint 1

Tarefa	Esforço
Criação do mockup do projeto no Figma	8 horas
Conversão do mockup em protótipo navegável	5 horas
Estudo da estrutura do sistema para elaboração do modelo de dados	10 horas
Criação do modelo de dados conceitual	5 horas
Conversão do modelo conceitual para modelo lógico	3 horas
Estudo sobre banco de dados MySQL	13 horas
Estudo sobre o funcionamento de sistemas de gerenciamento de chamados	13 horas
Estudo sobre TypeScript	10 horas
Estudo sobre Java Script	10 horas
Estudo sobre NodeJS	10 horas
Estudo sobre React	12 horas
Desenvolver um modelo do formulário a ser preenchido para abertura do chamado	3 horas

6.3 Modelo de dados

