

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

VINICIUS GABRIEL CIUNEK

**CAMPUS ALERTA - PLATAFORMA PARA GESTÃO DE PROBLEMAS NA
INFRAESTRUTURA DA UTFPR**

GUARAPUAVA

2025

VINICIUS GABRIEL CIUNEK

**CAMPUS ALERTA - PLATAFORMA PARA GESTÃO DE PROBLEMAS NA
INFRAESTRUTURA DA UTFPR**

**CAMPUS ALERTA - PLATFORM FOR MANAGING INFRASTRUCTURE
PROBLEMS AT UTFPR**

Projeto de Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Tecnólogo em Tecnologia em Sistemas para Internet do Curso Superior de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Luciano Ogiboski

GUARAPUAVA

2025



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

RESUMO

A gestão da infraestrutura física na Universidade Tecnológica Federal do Paraná (UTFPR) enfrenta desafios relacionados à descentralização e à rastreabilidade das solicitações de manutenção, muitas vezes dependentes de processos informais ou sistemas legados restritivos. Este trabalho propõe o desenvolvimento do sistema “Campus Alerta”, uma plataforma digital para a gestão colaborativa e eficiente de chamados de manutenção predial. A solução adota uma abordagem de desenvolvimento *Low-Code* utilizando a plataforma OutSystems, visando agilidade na entrega de um Produto Mínimo Viável (MVP) no formato de *Progressive Web App* (PWA). A principal inovação do projeto reside na integração entre ambientes físicos e digitais através de códigos QR fixados nos locais, permitindo a identificação inequívoca e georreferenciada das ocorrências. A metodologia é de natureza aplicada e exploratória, utilizando o *framework* ágil Scrum para o desenvolvimento iterativo e a técnica MoSCoW para a priorização de requisitos. Os resultados parciais incluem o levantamento de requisitos junto aos *stakeholders*, a modelagem da arquitetura de dados com suporte a múltiplas instituições (*multitenancy*) e a prototipação de alta fidelidade das interfaces, validando a viabilidade técnica da leitura de QR Codes. Conclui-se que a arquitetura proposta oferece uma alternativa viável, robusta e escalável, não apenas para o contexto universitário, mas com potencial de expansão futura para outros órgãos públicos, promovendo o conceito de *Smart Campus* e cidades inteligentes. Como resultados esperados para a próxima etapa, projeta-se a entrega do MVP funcional e a execução de testes de campo com a comunidade acadêmica. Desta forma, espera-se consolidar métricas de desempenho alinhadas ao órgão gestor, permitindo mensurar com consistência a eficiência da manutenção e a qualidade da infraestrutura do campus.

Palavras-chave: gestão de infraestrutura; low-code; outsystems; qr code; smart campus.

Palavras-chave: palavra-chave 1; palavra-chave 2; palavra-chave 3; palavra-chave 4; palavra-chave 5.

The management of physical infrastructure at the Federal University of Technology – Paraná (UTFPR) faces challenges regarding the centralization and traceability of maintenance requests, often relying on informal processes or restrictive legacy systems. This work proposes the development of the “Campus Alerta” system, a digital platform for the collaborative and efficient management of building maintenance tickets. The solution adopts a Low-Code development approach using the OutSystems platform, aiming for agility in delivering a Minimum Viable Product (MVP) as a Progressive Web App (PWA). The main innovation lies in the integration between physical and digital environments through QR Codes fixed in locations, allowing for unequivocal and georeferenced identification of occurrences. The methodology is applied and exploratory in nature, using the Scrum agile framework for iterative development and the MoSCoW technique for requirements prioritization. Partial results include requirements gathering with stakeholders, data architecture modeling supporting multiple institutions (multitenancy), and high-fidelity interface prototyping, validating the technical feasibility of QR Code scanning. It is concluded that the proposed architecture offers a viable, robust, and scalable alternative, not only for the university context but with potential for future expansion to other public bodies, promoting the concept of Smart Campus and smart cities.

Keywords: infrastructure management; low-code; outsystems; qr code; smart campus.

LISTA DE ABREVIATURAS E SIGLAS

Siglas

ITSM	IT Service Management (Gerenciamento de Serviços de TI)
LCDP	Plataforma de Desenvolvimento Low-Code, do inglês <i>Low-Code Development Platform</i>
MVP	Produto Mínimo Viável, do inglês <i>Minimum Viable Product</i>
RAD	Desenvolvimento Rápido de Aplicação, do inglês <i>Rapid Application Development</i>
SGBD	Sistema Gerenciador de Banco de Dados

SUMÁRIO

1	INTRODUÇÃO	6
1.1	Considerações iniciais	6
1.2	Objetivos	7
1.2.1	Objetivo Geral	7
1.2.2	Objetivos Específicos	7
1.3	Justificativa	8
1.4	Estrutura do trabalho	8
2	TRABALHOS RELACIONADOS	10
2.1	Contextualização Teórica	10
2.1.1	Gestão de Facilidades e Smart Campus	10
2.1.2	Desenvolvimento Low-Code	10
2.2	Análise de Soluções Existentes	11
2.2.1	Jira Service Management	11
2.2.2	Zammad	11
2.2.3	Síntese Comparativa	11
3	DETALHAMENTO DA SOLUÇÃO	13
3.1	Fluxo de Funcionamento	13
3.2	Perfis de Usuários	14
3.2.1	Comunidade (Alunos e Servidores)	14
3.2.2	Equipe de Manutenção (Técnicos)	14
3.2.3	Gestor do Campus (Administrador)	14
4	MATERIAIS E MÉTODOS	15
4.1	Materiais	15
4.1.1	Plataforma de Desenvolvimento: OutSystems	15
4.1.2	Infraestrutura de Dados	15
4.1.3	Ferramentas de Apoio	16
4.2	Métodos	16
4.2.1	Estudo e Adaptação Tecnológica	16
4.2.2	Engenharia de Requisitos e Priorização	16
4.2.3	Ciclo de Desenvolvimento	17

4.2.4	Procedimentos de Validação e Métricas	17
5	RESULTADOS PARCIAIS	19
5.1	Levantamento e Priorização de Requisitos	19
5.2	Prototipação de Interfaces	19
5.3	Arquitetura de Dados Preliminar	20
6	CONSIDERAÇÕES FINAIS	23
	REFERÊNCIAS	24

1 INTRODUÇÃO

A gestão eficiente da infraestrutura física é um dos pilares fundamentais para a garantia da qualidade do ensino e do bem-estar em instituições educacionais. Na Universidade Tecnológica Federal do Paraná (UTFPR), a manutenção de um campus complexo, composto por diversos blocos, laboratórios e áreas de convivência, impõe desafios logísticos significativos. Atualmente, a comunicação de problemas estruturais — como falhas elétricas, hidráulicas ou danos em equipamentos — ocorre de maneira descentralizada e, por vezes, informal, o que dificulta a rastreabilidade das solicitações e o planejamento da equipe de manutenção.

Neste contexto, o conceito de *Smart Campus* (Campus Inteligente) surge como uma alternativa viável, utilizando tecnologias da informação para otimizar processos e recursos. A proposta deste trabalho é o desenvolvimento do sistema “Campus Alerta”, uma solução tecnológica que visa modernizar esse fluxo. Diferente das abordagens tradicionais de desenvolvimento de software, este projeto propõe a utilização de uma plataforma de desenvolvimento *Low-Code* (OutSystems) para a criação de um Produto Mínimo Viável, do inglês *Minimum Viable Product* (MVP).

A escolha pelo *Low-Code* justifica-se pela necessidade de agilidade na entrega de soluções digitais e pela facilidade de manutenção e escalabilidade. Além disso, o sistema inova ao integrar o mundo físico e digital por meio do uso de QR Codes fixados nos ambientes, permitindo a identificação inequívoca do local da ocorrência. Embora concebido inicialmente para o cenário da UTFPR, a arquitetura da solução é projetada para ser genérica e escalável, visando sua futura expansão para atender a outras instituições públicas da cidade, como escolas estaduais e postos de saúde, promovendo uma gestão colaborativa da infraestrutura pública.

Considerando a inexistência de métricas consolidadas e a dificuldade de padronização das demandas junto ao órgão responsável, define-se o seguinte problema de pesquisa: **Como uma solução digital integrada pode estruturar e automatizar a coleta de requisitos de manutenção predial, transformando solicitações informais e descentralizadas em dados técnicos acionáveis para a gestão do campus?**

1.1 Considerações iniciais

A qualidade da infraestrutura física impacta diretamente a experiência universitária. O bem-estar da comunidade acadêmica, composta por alunos e servidores, depende da funcionalidade de salas, laboratórios e espaços de convivência (LOMAS; JONES, 2006). No entanto, a manutenção desta grande estrutura ainda representa um desafio logístico.

No cenário atual, embora existam mecanismos institucionais para a gestão de ocorrências, estes apresentam limitações críticas de acessibilidade e usabilidade. O sistema vigente encontra-se restrito a perfis específicos (docentes e servidores administrativos), excluindo a maior parte da comunidade acadêmica — os alunos — do processo de monitoramento e re-

porte. Além disso, trata-se de uma ferramenta baseada em tecnologia legada, com interfaces pouco intuitivas e baixa capacidade de rastreabilidade do fluxo de atendimento.

Essa restrição de acesso e a obsolescência tecnológica forçam o uso recorrente de canais informais (corredor, e-mail, mensagens instantâneas), gerando a dispersão das informações e dificultando a alocação eficiente da equipe de manutenção. O sistema Campus Alerta surge com o propósito de unificar esses canais e democratizar o acesso, permitindo que qualquer membro da comunidade acadêmica reporte problemas de forma ágil, substituindo a burocracia atual por um fluxo transparente e rastreável.

1.2 Objetivos

Esta seção apresenta os objetivos traçados para a pesquisa e o desenvolvimento da ferramenta.

1.2.1 Objetivo Geral

Desenvolver uma plataforma digital multiplataforma utilizando tecnologia *Low-Code* (OutSystems) para a gestão colaborativa de chamados de manutenção predial, validando o conceito na UTFPR com uma arquitetura preparada para escalabilidade interinstitucional.

1.2.2 Objetivos Específicos

Os objetivos específicos para alcançar o propósito do trabalho são:

- **Mapear** os fluxos de abertura e atendimento de chamados para definir os requisitos do MVP.
- **Investigar** a viabilidade técnica e a produtividade do uso de ferramentas *Low-Code* no contexto acadêmico e na gestão pública.
- **Desenvolver** a aplicação Web e Mobile na plataforma OutSystems, implementando a leitura de QR Codes para georreferenciamento interno dos ambientes.
- **Definir** um modelo de autenticação flexível que permita tanto o acesso institucional quanto o acesso público controlado, visando a expansão futura para o conceito de *Smart City*.
- **Validar** o protótipo junto à comunidade acadêmica, coletando métricas para a evolução do produto.

1.3 Justificativa

A realização deste trabalho justifica-se pela necessidade de modernizar um processo administrativo crucial para a qualidade da experiência na UTFPR. Atualmente, a restrição de acesso e a interface desatualizada do sistema legado geram ineficiências que afetam diretamente a agilidade dos reparos. A implementação de uma plataforma única e acessível a alunos e servidores não apenas agiliza a comunicação, mas também cria um ambiente mais transparente e confiável.

A escolha pelo desenvolvimento utilizando a plataforma **OutSystems** é estratégica e acadêmica. Em um mercado de TI que exige cada vez mais velocidade (Time-to-Market), o domínio de plataformas *Low-Code* torna-se um diferencial competitivo. Este projeto justifica-se, portanto, pela investigação dessa tecnologia emergente, avaliando seus ganhos de produtividade em comparação ao desenvolvimento tradicional (High-Code).

Além disso, a implementação da identificação via **QR Code** resolve um problema prático recorrente: a imprecisão na descrição do local do problema por parte dos usuários. Ao escanear o código, o sistema garante que a equipe de manutenção saiba exatamente onde atuar.

Por fim, o projeto possui uma forte justificativa social e de extensão. Ao projetar a arquitetura para ser *multitenant* (multi-instituição), o Campus Alerta não se limita aos muros da universidade, podendo ser ofertado futuramente como uma ferramenta de gestão de zeladoria para escolas públicas e órgãos municipais, retornando o investimento de conhecimento à sociedade.

Ressalta-se que, devido à natureza descentralizada e muitas vezes informal dos canais de comunicação atuais (solicitações verbais, e-mails dispersos e mensagens instantâneas), torna-se inviável mensurar com precisão o volume exato de chamados mensais ou o tempo médio de atendimento (TMA) vigente. Essa lacuna de dados quantitativos históricos representa, por si só, uma justificativa crítica para este projeto: a gestão atual opera baseada em percepções subjetivas, sem indicadores auditáveis. A implementação do Campus Alerta visa sanar essa deficiência, criando pela primeira vez uma base de dados estruturada que permitirá à UTFPR quantificar a demanda reprimida e aplicar inteligência na gestão dos recursos.

1.4 Estrutura do trabalho

O presente trabalho está organizado em capítulos que refletem o desenvolvimento do projeto. O **Capítulo 1** (esta introdução) contextualiza o problema e define os objetivos. O **Capítulo 2** apresenta a Fundamentação Teórica e os Trabalhos Relacionados, abordando conceitos de Gestão de Facilidades e sistemas similares. O **Capítulo 3** descreve a Proposta e o Detalhamento da Solução, explicando o funcionamento do sistema via QR Code e OutSystems. O **Capítulo 4** detalha a Metodologia e os Materiais utilizados, bem como os Resultados Parciais

alcançados (requisitos e protótipos). Por fim, o **Capítulo 5** traz as Considerações Finais e o cronograma para a próxima etapa.

2 TRABALHOS RELACIONADOS

Neste capítulo, são apresentados os conceitos teóricos que fundamentam a proposta do Campus Alerta, bem como uma análise de soluções de mercado existentes, a fim de posicionar o projeto e destacar seus diferenciais. A primeira seção aborda os conceitos de Gestão de Facilidades, Smart Campus e Desenvolvimento Low-Code, enquanto a segunda se aprofunda na análise de sistemas de gestão de chamados já consolidados.

2.1 Contextualização Teórica

Para compreender a relevância e o impacto potencial do Campus Alerta, é fundamental inseri-lo em contextos mais amplos de gestão e inovação tecnológica.

2.1.1 Gestão de Facilidades e Smart Campus

A Gestão de Facilidades (*Facility Management*) é uma disciplina focada em garantir a funcionalidade, o conforto, a segurança e a eficiência do ambiente construído, integrando pessoas, lugares, processos e tecnologias (International Facility Management Association (IFMA), 2025). Em uma instituição como a UTFPR, isso se traduz na administração de toda a infraestrutura física — desde a manutenção de salas e laboratórios até a gestão de recursos. O Campus Alerta se posiciona como uma ferramenta de apoio a essa gestão, otimizando o fluxo de detecção e correção de falhas.

Paralelamente, o conceito de *Smart Campus* descreve a aplicação de tecnologias digitais para criar um ecossistema universitário conectado e eficiente (Gartner IT Glossary, 2025). Ao transformar um processo manual em um fluxo de dados digital, o sistema permite que a gestão deixe de ser apenas reativa e passe a ser baseada em dados, identificando padrões de desgaste na infraestrutura.

2.1.2 Desenvolvimento Low-Code

Diferente das abordagens tradicionais de desenvolvimento de software (conhecidas como *High-Code*), este projeto adota o paradigma *Low-Code*. Plataformas de Desenvolvimento Low-Code, do inglês *Low-Code Development Platform* (LCDP) permitem a criação de aplicações através de interfaces gráficas e configurações, reduzindo drasticamente a necessidade de codificação manual linha a linha.

Segundo pesquisas recentes da área de Engenharia de Software, o uso de Low-Code pode acelerar o ciclo de entrega de software em até 10 vezes, permitindo que desenvolvedores foquem mais na regra de negócio e na experiência do usuário do que na infraestrutura de

código. A escolha da plataforma **OutSystems** para este projeto visa validar essa agilidade no contexto de soluções para o setor público, onde a escassez de desenvolvedores especialistas é um gargalo frequente.

2.2 Análise de Soluções Existentes

Embora não se tenha conhecimento de uma solução desenvolvida especificamente para a gestão de infraestrutura na UTFPR que utilize georreferenciação interna via QR Codes, existem no mercado ferramentas consolidadas para gestão de chamados. A análise a seguir posiciona o Campus Alerta em relação a algumas dessas soluções.

2.2.1 Jira Service Management

O Jira Service Management, da Atlassian, é uma das plataformas de IT Service Management (Gerenciamento de Serviços de TI) (ITSM) mais robustas do mercado (ATLASSIAN, 2025). Ele é projetado para gerenciar solicitações de TI e fluxos complexos. **Limitações frente ao Campus Alerta:** Apesar de sua potência, o Jira possui uma curva de aprendizado elevada e uma interface complexa para o usuário final (aluno), o que pode desestimular o reporte rápido de problemas simples. Além disso, a implementação de um fluxo de "leitura de QR Code para abertura automática de ticket" exigiria customizações avançadas e plugins caros, enquanto no Campus Alerta essa é uma funcionalidade nativa e central.

2.2.2 Zammad

Zammad é um sistema de *help desk* de código aberto, flexível e com custo acessível (ZAMMAD, 2025). Ele permite a criação de tickets via múltiplos canais. **Limitações frente ao Campus Alerta:** Sua natureza genérica é sua principal limitação para o cenário de manutenção predial. A interface não é otimizada para uso em campo (mobile-first) por equipes de manutenção que precisam anexar fotos e atualizar status em tempo real. Diferente do Campus Alerta, que está sendo desenhado sob medida para a realidade física do campus (blocos e salas), o Zammad exigiria um esforço técnico considerável de adaptação para oferecer a mesma agilidade de localização das ocorrências.

2.2.3 Síntese Comparativa

A principal lacuna deixada pelas soluções de mercado analisadas é a falta de integração entre o mundo físico e o digital. Enquanto Jira e Zammad focam no gerenciamento do ticket após sua criação, o **Campus Alerta** inova na etapa de **captura da ocorrência**, utilizando QR

Codes e uma interface simplificada em Low-Code para garantir que o dado chegue à equipe de manutenção de forma rápida, precisa e georreferenciada.

3 DETALHAMENTO DA SOLUÇÃO

A solução proposta, denominada **Campus Alerta**, consiste em uma plataforma digital de gestão de manutenção predial desenvolvida sobre a infraestrutura *Low-Code* da OutSystems. O sistema atua como um intermediário ágil entre a comunidade acadêmica (usuários dos espaços físicos) e as equipes de gestão de infraestrutura da UTFPR.

O diferencial técnico e funcional da solução reside na integração entre o ambiente físico e digital (*phygital*) através do uso de códigos QR. A seguir, são detalhados o fluxo de funcionamento e os perfis de usuários envolvidos.

3.1 Fluxo de Funcionamento

O funcionamento do sistema baseia-se na interação contextualizada com o local da ocorrência. O fluxo principal foi desenhado para minimizar o esforço cognitivo do usuário e garantir a precisão dos dados coletados:

1. **Identificação do Local via QR Code:** Etiquetas contendo códigos QR únicos serão fixadas em locais estratégicos (portas de salas, laboratórios, corredores e equipamentos). Ao identificar um problema, o usuário escaneia o código com a câmera do smartphone. O sistema decodifica o identificador e preenche automaticamente o local da ocorrência (Ex: Bloco B - Sala 102), eliminando erros de digitação e ambiguidades.
2. **Autenticação e Acesso:** Para garantir a segurança e a rastreabilidade, o sistema exigirá autenticação. No escopo do MVP, serão avaliados dois modelos de acesso:
 - *Integração Institucional:* Login via credenciais da UTFPR (RA ou e-mail institucional), garantindo que apenas membros da comunidade acadêmica possam abrir chamados internos.
 - *Acesso Público Identificado:* Um módulo para cadastro de cidadãos externos à universidade. Este recurso é fundamental para a escalabilidade do projeto, permitindo que, no futuro, a mesma plataforma seja contratada por Prefeituras para que municípios reportem problemas em escolas, postos de saúde e praças, utilizando apenas o CPF ou login social (Google/Gov.br), com mecanismos de verificação (como CAPTCHA ou confirmação por e-mail) para mitigar o registro de ocorrências falsas (spam).
3. **Registro da Ocorrência:** Após a identificação do local e do usuário, o sistema apresenta um formulário simplificado onde é possível categorizar o problema (ex: Elétrica, Hidráulica, Mobiliário) e anexar uma foto como evidência.

4. **Gestão e Atendimento:** Os dados são enviados para a nuvem da OutSystems e disponibilizados em um painel administrativo para a equipe de manutenção, que pode triar, priorizar e atualizar o status do chamado (Pendente, Em Análise, Resolvido).

3.2 Perfis de Usuários

Para atender às necessidades de governança e operação do sistema, foram definidos três perfis de acesso distintos, cada um com permissões específicas.

3.2.1 Comunidade (Alunos e Servidores)

Este é o perfil do solicitante final. Alunos e servidores são os principais responsáveis por identificar e reportar os problemas de infraestrutura encontrados no dia a dia. Para este perfil, a interface é projetada para ser minimalista e focada na rapidez: leitura do QR Code, seleção do problema, foto e envio. O usuário também pode acompanhar o histórico e o status de suas solicitações, promovendo transparência.

3.2.2 Equipe de Manutenção (Técnicos)

Os técnicos de manutenção recebem as ordens de serviço geradas a partir dos chamados. Eles utilizam o sistema para visualizar a fila de tarefas, consultar detalhes (como a foto e o local exato) e registrar a conclusão do serviço. Para este perfil, a mobilidade é essencial, permitindo que o técnico atualize o status diretamente do local do reparo.

3.2.3 Gestor do Campus (Administrador)

O perfil administrativo é voltado para a Direção de Planejamento e Administração (DIR-PLAD) ou órgão equivalente. O gestor tem acesso a um painel de controle (*Dashboard*) com indicadores-chave de desempenho (KPIs), tais como: Tempo Médio de Atendimento (TMA), ranking de equipamentos com maior índice de defeitos e mapa de calor das áreas mais críticas do campus. Esses dados transformam a manutenção corretiva em inteligência para o planejamento de manutenções preventivas e alocação de orçamento.

4 MATERIAIS E MÉTODOS

Este capítulo descreve as ferramentas tecnológicas e os procedimentos metodológicos que serão adotados para a execução do projeto. A escolha dos materiais e métodos fundamenta-se na necessidade de alta produtividade, robustez e na natureza aplicada do trabalho, visando a validação de uma solução inovadora para a gestão de infraestrutura.

4.1 Materiais

Para a implementação da solução *Campus Alerta*, foi selecionado um conjunto de tecnologias focado em Desenvolvimento Rápido de Aplicação, do inglês *Rapid Application Development* (RAD), garantindo a entrega do MVP dentro do cronograma acadêmico.

4.1.1 Plataforma de Desenvolvimento: OutSystems

A principal ferramenta utilizada neste projeto é a plataforma **OutSystems** (OUTSYSTEMS, 2025) (através do ambiente de desenvolvimento *Service Studio*). Trata-se de uma plataforma de desenvolvimento *Low-Code* líder de mercado, que permite a construção visual de aplicações corporativas.

A justificativa para esta escolha é estratégica e acadêmica:

- **Agilidade:** O desenvolvimento visual reduz drasticamente a quantidade de código manual (boilerplate), permitindo que o foco do trabalho seja a regra de negócio (gestão de manutenção) e a experiência do usuário, e não a configuração de servidores ou APIs básicas.
- **Capacidade PWA:** A plataforma oferece recursos nativos para a geração de *Progressive Web Apps* (PWA). Isso garante que o sistema funcione como um aplicativo nativo em dispositivos móveis (Android e iOS) dos alunos e servidores, com acesso à câmera (para ler QR Codes) e armazenamento local, sem a barreira de entrada que seria exigir o download em uma loja de aplicativos (Google for Developers, 2025). Além disso, o uso de Low-Code viabiliza a entrega de um sistema complexo (com Painel Web, App Mobile e Banco de Dados) por um único desenvolvedor dentro do cronograma acadêmico, algo que seria inviável com o desenvolvimento tradicional (High-Code).

4.1.2 Infraestrutura de Dados

O armazenamento e gerenciamento dos dados serão realizados utilizando o Sistema Gerenciador de Banco de Dados (SGBD) relacional integrado à nuvem da OutSystems (base-

ado em SQL Server). Esta arquitetura *cloud-native* garante integridade referencial, segurança automática e, principalmente, escalabilidade, suportando a modelagem de entidades complexas necessárias para a visão de futuro do projeto (múltiplas instituições utilizando a mesma base).

4.1.3 Ferramentas de Apoio

Para complementar o desenvolvimento, serão utilizados:

- **Figma:** Utilizado para a prototipação de alta fidelidade das interfaces (UI/UX) e validação dos fluxos de navegação antes da implementação.
- **Gerador de QR Codes:** Ferramentas de codificação para a criação das etiquetas bi-dimensionais únicas que serão fixadas fisicamente nos locais de teste (laboratórios e salas).
- **Dispositivos Móveis:** Smartphones com sistemas operacionais Android e iOS para a validação de campo da leitura dos códigos QR e responsividade da interface em diferentes resoluções.

4.2 Métodos

A metodologia de trabalho adotada é de natureza aplicada e exploratória. O processo de desenvolvimento seguirá os ritos da metodologia ágil **Scrum** (SCHWABER; SUTHERLAND, 2020), adaptados para a execução individual do Trabalho de Conclusão de Curso.

4.2.1 Estudo e Adaptação Tecnológica

Considerando que a plataforma OutSystems e o paradigma *Low-Code* não compõem a grade curricular tradicional do curso de Tecnologia em Sistemas para Internet, a primeira etapa metodológica consiste no domínio da ferramenta. Serão realizados estudos técnicos sobre a lógica de fluxos visuais, arquitetura de módulos e ciclo de vida de aplicações na plataforma. A curva de aprendizado e os desafios de adaptação de um desenvolvedor tradicional para o ambiente *Low-Code* serão documentados como parte dos resultados da pesquisa.

4.2.2 Engenharia de Requisitos e Priorização

Para definir o escopo do MVP (*Minimum Viable Product*), utiliza-se a técnica de priorização **MoSCoW** (SEBRAE, 2024) (*Must have, Should have, Could have, Won't have*). Esta técnica permite classificar as funcionalidades de acordo com sua criticidade para o negócio, garantindo

que o núcleo do sistema — a leitura de QR Code e a abertura de chamado — seja entregue prioritariamente, enquanto funcionalidades secundárias são planejadas para etapas futuras.

4.2.3 Ciclo de Desenvolvimento

O projeto será dividido em ciclos quinzenais (*Sprints*), estruturados da seguinte forma:

1. **Planejamento e Modelagem:** Definição das *user stories* da quinzena e modelagem das entidades no banco de dados da OutSystems.
2. **Implementação (Front e Back):** Construção das telas e da lógica de negócios utilizando os aceleradores visuais da plataforma.
3. **Testes de Campo e Feedback:** Validação física da leitura dos QR Codes em diferentes condições de iluminação e conectividade nas dependências do campus. Nesta etapa, membros voluntários da comunidade acadêmica serão convidados a utilizar o sistema para reportar problemas reais ou simulados, respondendo a um breve questionário de usabilidade (SUS - System Usability Scale) para validar a eficácia da interface.

4.2.4 Procedimentos de Validação e Métricas

Para garantir a eficácia da solução, a validação do MVP no Trabalho de Conclusão de Curso 2 será realizada através de testes de usabilidade com três grupos distintos de usuários, cobrindo todo o ciclo de vida de um chamado de manutenção:

- **Grupo 1 - Solicitantes (Alunos e Servidores):**

- **Participantes:** Uma amostra de 10 a 15 voluntários da comunidade acadêmica.
- **Atividade:** Simulação de abertura de chamados em locais reais do campus, utilizando os recursos de identificação de local (QR Code ou Geolocalização).
- **Métricas:** Tempo médio para registro da ocorrência e taxa de sucesso na conclusão da tarefa sem auxílio externo.
- **Instrumento:** Aplicação do questionário SUS (*System Usability Scale*) ao final do teste para mensurar o nível de satisfação.

- **Grupo 2 - Equipe Técnica (Manutenção):**

- **Participantes:** 2 a 3 técnicos responsáveis pelos reparos.
- **Atividade:** Utilização do **Painel do Técnico** para recebimento, triagem e alteração de status (de "Pendente" para "Concluído") das solicitações abertas pelo Grupo 1.

- **Objetivo:** Validar se as informações capturadas (foto, local e descrição) são suficientes para a execução do reparo sem necessidade de deslocamento prévio para averiguação.

- **Grupo 3 - Gestão (Administração do Campus):**

- **Participantes:** Gestores do setor de infraestrutura DIRSEG.
- **Atividade:** Análise do **Dashboard Administrativo** populado com os dados gerados durante os testes.
- **Objetivo:** Confirmar se as métricas e indicadores apresentados (locais com mais defeitos, tipos de problemas recorrentes) atendem às necessidades de tomada de decisão do órgão.

5 RESULTADOS PARCIAIS

Até o presente momento, as atividades de desenvolvimento do projeto concentraram-se no levantamento e priorização de requisitos, na definição da arquitetura da solução baseada na plataforma OutSystems e na prototipação das interfaces principais do aplicativo. Adicionalmente, estudos preliminares na documentação da plataforma OutSystems confirmaram a viabilidade técnica da integração com a câmera do dispositivo para leitura de códigos QR em ambiente PWA, mitigando o principal risco técnico do projeto.

5.1 Levantamento e Priorização de Requisitos

Com base na metodologia ágil adotada, os requisitos foram detalhados no formato de Histórias de Usuário (*User Stories*), descrevendo as funcionalidades sob a perspectiva de valor para quem utiliza o sistema. O Quadro 1 apresenta as principais histórias mapeadas para o MVP.

Tabela 1 – Histórias de Usuário do Campus Alerta

ID	Perfil	História
US01	Comunidade (Aluno/-Servidor)	Quero escanear um QR Code na sala para que o sistema identifique automaticamente onde estou sem eu precisar digitar.
US02	Comunidade (Aluno/-Servidor)	Quero registrar um problema anexando uma foto e selecionando a categoria para que a manutenção entenda exatamente o que precisa ser consertado.
US03	Comunidade (Aluno/-Servidor)	Quero visualizar o histórico dos meus chamados para que eu possa saber se o problema que reporte já foi resolvido.
US04	Equipe de Manutenção	Quero visualizar uma lista de chamados pendentes com a localização exata para que eu possa organizar meu roteiro de trabalho.
US05	Equipe de Manutenção	Quero alterar o status do chamado para "Em andamento"ou "Concluído" para que o solicitante seja informado do progresso.
US06	Gestor do Campus	Quero ver um painel com indicadores de locais com mais problemas para que eu possa planejar reformas preventivas.
US07	Cidadão Externo	Quero realizar um cadastro simplificado (Gov.br ou Social) para que eu possa reportar problemas em áreas públicas sem ter vínculo com a universidade.

Fonte: Autoria própria (2025).

A partir destas histórias, foi realizada a priorização técnica utilizando o método MoSCoW, conforme apresentado no Quadro 2 a seguir.

5.2 Prototipação de Interfaces

Foram desenvolvidos protótipos de alta fidelidade das principais telas do aplicativo móvel, utilizando a ferramenta Figma. O objetivo desta etapa foi validar o fluxo do usuário antes da implementação na plataforma OutSystems.

Tabela 2 – Priorização de Requisitos Funcionais (Método MoSCoW)

ID	Descrição do Requisito	Prioridade
RF01	O sistema deve permitir a leitura de QR Codes para identificação automática do local (Bloco/Sala).	Must Have
RF02	O usuário deve poder registrar um chamado contendo categoria, descrição e foto do problema.	Must Have
RF03	O sistema deve possuir um painel administrativo para gestão e alteração de status dos chamados.	Must Have
RF04	O sistema deve suportar múltiplos perfis de acesso (Usuário Comum e Gestor).	Must Have
RF05	O sistema deve permitir o cadastro de múltiplas instituições (Multitenancy) para escalabilidade futura.	Should Have
RF06	O usuário deve visualizar o histórico e o status das suas solicitações.	Should Have
RF07	Integração automática com o login institucional da UTFPR.	Could Have

Fonte: Autoria própria (2025)..

As figuras abaixo apresentam os protótipos desenvolvidos. As Figuras 1 e 2 exibem as opções de acesso (institucional e público), demonstrando a flexibilidade da solução, enquanto a Figura 3 detalha o fluxo de abertura de chamado automatizado via QR Code.

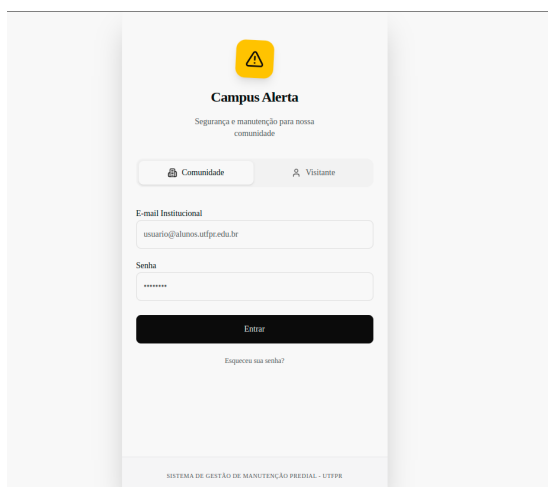


Figura 1 – Tela de Login Institucional

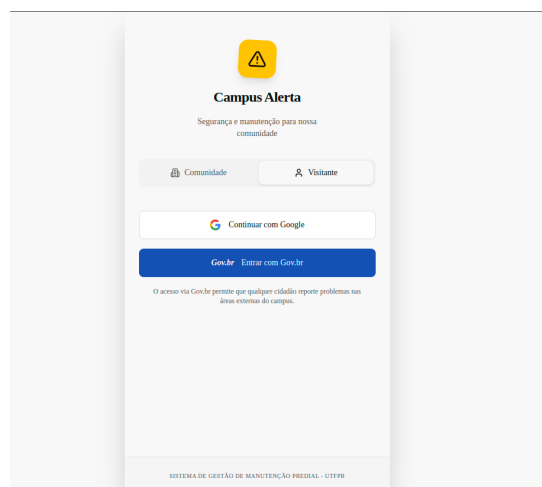


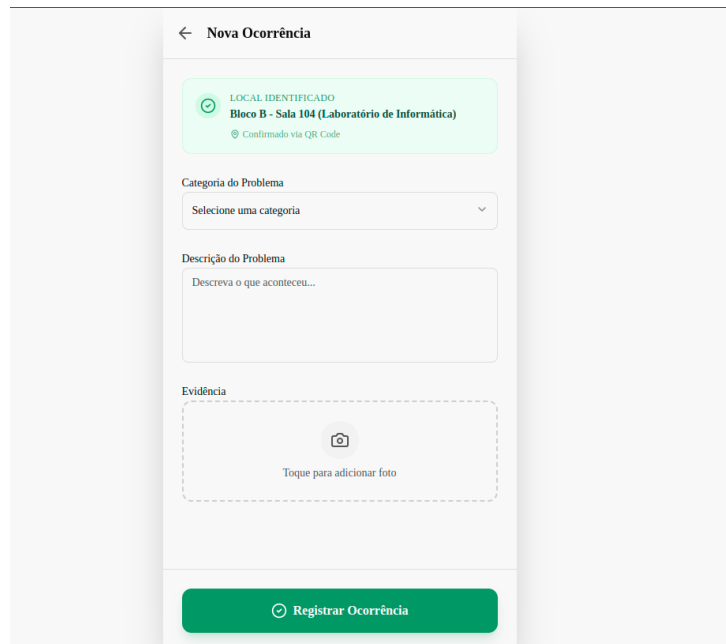
Figura 2 – Tela de Login Público

Fonte: Autoria própria (2025).

5.3 Arquitetura de Dados Preliminar

Para suportar as funcionalidades do MVP, foi definida uma modelagem de dados inicial focada na agilidade e na performance da aplicação mobile (PWA). A estrutura armazena as informações essenciais para o ciclo de vida da manutenção, sem complexidade desnecessária.

A Figura 4 apresenta o Diagrama Entidade-Relacionamento (DER), destacando as duas entidades principais:



← Nova Ocorrência

LOCAL IDENTIFICADO
Bloco B - Sala 104 (Laboratório de Informática)
Confirmado via QR Code

Categoria do Problema
Selecione uma categoria

Descrição do Problema
Descreva o que aconteceu...

Evidência
Toque para adicionar foto

Registrar Ocorrência

Figura 3 – Protótipo da Tela de Registro de Ocorrência via QR Code

Fonte: Autoria própria (2025).

- **Usuário:** Centraliza todos os perfis de acesso (Aluno, Servidor, Técnico e Gestor), diferenciados pelo atributo *Role*.
- **Chamado:** Armazena os dados da ocorrência (descrição, foto e local) e mantém o vínculo tanto com quem abriu o chamado quanto com o técnico responsável pelo atendimento.

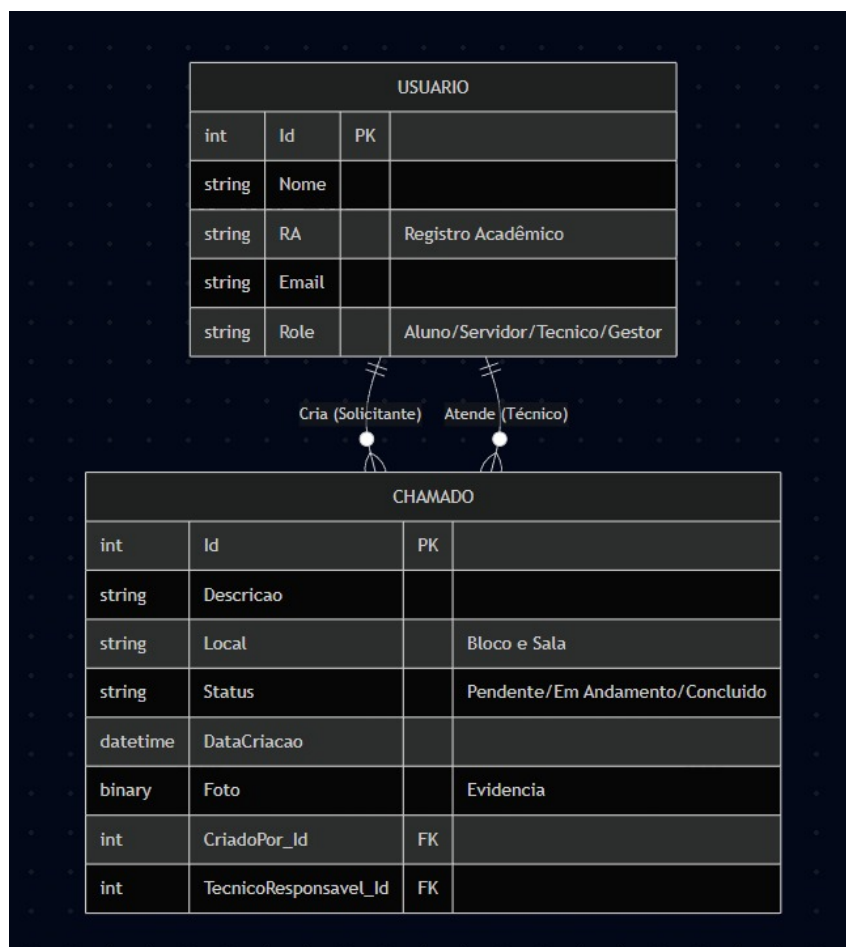


Figura 4 – Diagrama de Entidade-Relacionamento (DER) Inicial
Fonte: Autoria própria (2025).

6 CONSIDERAÇÕES FINAIS

Este trabalho apresentou a estruturação do projeto para o desenvolvimento da plataforma *Campus Alerta*, uma solução voltada à otimização da gestão de manutenção predial na UTFPR. Ao longo deste documento, foi detalhado o problema da descentralização das ocorrências e a proposta de uma solução tecnológica baseada em *Low-Code* e georreferenciação interna via códigos QR.

A escolha pela plataforma OutSystems apresenta-se como uma estratégia acadêmica relevante, permitindo a investigação de paradigmas de desenvolvimento ágil que são tendências no mercado corporativo. Acredita-se que essa tecnologia oferece o equilíbrio ideal entre a velocidade necessária para o desenvolvimento de um MVP e a robustez exigida para uma futura expansão do sistema para outras instituições públicas (*multitenancy*).

Os resultados parciais indicam que a arquitetura proposta é viável e atende aos objetivos de agilidade e facilidade de uso. A principal dificuldade prevista para a próxima etapa reside na curva de aprendizado da plataforma OutSystems e na modelagem correta da base de dados para suportar múltiplos inquilinos (instituições).

Para a etapa de TCC 2, estabelecem-se como entregas obrigatórias do MVP as seguintes funcionalidades e atividades:

1. **Módulo de Abertura de Chamado:** Implementação do fluxo completo para o usuário final, incluindo a identificação do local (via geolocalização ou QR Code), seleção de categoria e anexo de evidência fotográfica.
2. **Painel de Atendimento Técnico:** Interface para que a equipe de manutenção visualize a fila de chamados e realize a alteração de status (Triagem → Execução → Conclusão).
3. **Dashboard de Gestão:** Painel administrativo contendo métricas básicas (indicadores de volume de chamados por bloco e tipo de defeito) para apoiar a tomada de decisão do órgão responsável (DIRSEG/DIRPLAD).
4. **Validação de Usabilidade:** Realização de testes de campo com voluntários da comunidade acadêmica para validar a eficácia da interface e do fluxo de registro.

Desta forma, o projeto garante a entrega de uma ferramenta funcional que poderá ser submetida ao órgão gestor como um piloto de modernização da infraestrutura do campus.

REFERÊNCIAS

ATLASSIAN. **Jira Service Management**. 2025. <https://www.atlassian.com/software/jira/service-management>. Acesso em: 11 set. 2025.

Gartner IT Glossary. **Smart Campus**. 2025. <https://www.gartner.com/en/information-technology/glossary/smart-campus>. Acesso em: 11 set. 2025.

Google for Developers. **What are Progressive Web Apps?** 2025. <https://web.dev/progressive-web-apps/>. Acesso em: 11 set. 2025.

International Facility Management Association (IFMA). **What is Facility Management?** 2025. <https://www.ifma.org/what-is-facility-management>. Acesso em: 11 set. 2025.

LOMAS, L.; JONES, R. The practice of facility management in higher education. **New Directions for Institutional Research**, v. 2006, n. 131, p. 5–15, 2006.

OUTSYSTEMS. **High-Performance Low-Code Platform**. 2025. <https://www.outsystems.com/>. Acesso em: 18 nov. 2025.

SCHWABER, K.; SUTHERLAND, J. **The Scrum Guide: The Definitive Guide to Scrum**. 2020. <https://scrumguides.org/scrum-guide.html>. Acesso em: 19 nov. 2025.

SEBRAE. **O método MoSCoW para definição de prioridades**. 2024. https://sebrae.com.br/Sebrae/Portal%20Sebrae/Arquivos/ebook_sebrae_metodologia_moscow.pdf. Acesso em: 19 nov. 2025.

ZAMMAD. **Zammad: The Helpdesk & Support System**. 2025. <https://zammad.org/>. Acesso em: 11 set. 2025.