Brasil 6G

# Projeto Brasil 6G
# Fase III

# Atividade 4.1
# Integração dos componentes da plataforma - Parte 2

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO

GOVERNO FEDERAL BRASIL UNIÃO E RECONSTRUÇÃO

RNP ORGANIZAÇÃO SOCIAL DO MCTI

Inatel Instituto Nacional de Telecomunicações

UNISINOS

INF INSTITUTO DE INFORMÁTICA UFG UNIVERSIDADE FEDERAL DE GOIÁS

UNIVERSIDADE FEDERAL DO CEARÁ

UFSC

UNIVERSIDADE FEDERAL DO PARÁ UFPA

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

UNICAMP

cpqd

ELDORADO INSPIRAÇÃO PELO NOVO

ADVANTECH Enabling an Intelligent Planet

22/07/2026

**Histórico de Atualizações:**

| Versão | Data | Autor(es) | Notas |
|---|---|---|---|
| 1 | 13/07/2026 | Pedro Carletti Soares | Elaboração de conteúdo |
| 2 | 22/07/2026 | Juliano Silveira Ferreira<br>Luciano Leonel Mendes<br>Vitor Rodrigues Di Toro | Revisão de texto<br>Melhorias em imagens<br>Correção em diagramas |

# Lista de Figuras

# Lista de Tabelas

# Acrônimos

**6G** Sexta Geração de Rede Móvel Celular

**API** *Application Programming Interface*

**HTTP** *Hypertext Transfer Protocol*

**i18n** *internationalization*

**Inatel** Instituto Nacional de Telecomunicações

**JWT** *JSON Web Tokens*

**MFA** *Multi-factor Authentication*

**OTP** *One-Time Password*

**OWASP** *Open Web Application Security Project*

**PAARRC** Plataforma de Agendamento e Acesso Remoto a Recursos Compartilhados

**pen-test** *Penetration Test*

**RBAC** *Role-Based Access Control*

**RDP** *Remote Desktop Protocol*

**SSH** *Secure Shell*

**Unix-like** sistemas operacionais derivados do UNIX

**VNC** *Virtual Network Computing*

**VPN** *Virtual Private Network*

# Sumário

# 1 Introdução

O Projeto Brasil 6G contempla, em seu escopo, a construção de uma plataforma de experimentação voltada ao desenvolvimento, validação e demonstração de tecnologias emergentes associadas às redes de próxima geração, com destaque para a Sexta Geração de Rede Móvel Celular (6G). No contexto da execução da atividade 4.1, a primeira parte do relatório apresentou a concepção e a implementação de funcionalidades básicas principais da Plataforma Multiusuário Brasil 6G ou PAARRC, com foco inicial na disponibilização de agendamento de recursos [1]. Essa primeira etapa estabeleceu a base tecnológica necessária para permitir que usuários pudessem consultar recursos disponíveis, submetessem solicitações de uso e tivessem seus pedidos avaliados por administradores do sistema, dentro de um ambiente controlado, rastreável e seguro.

O presente relatório descreve a continuidade do desenvolvimento, relatando as melhorias, correções e evoluções implementadas na plataforma a partir da versão funcional anterior. Diferentemente da etapa anterior, cujo foco principal estava na construção da base operacional do sistema, esta nova etapa concentra-se no amadurecimento da solução como uma plataforma multiusuário mais robusta, segura, usável e preparada para uso em cenários reais de experimentação compartilhada. Dessa forma, os aprimoramentos descritos neste relatório não se limitam à inclusão de novas funcionalidades, mas também envolvem o fortalecimento de fluxos já existentes, a revisão de mecanismos de autenticação, a melhoria da experiência do usuário, a adoção de práticas de qualidade de software e a correção de vulnerabilidades identificadas por análises automatizadas e testes de segurança.

Entre os principais avanços implementados, destacam-se os ajustes no fluxo de uso do perfil regular, a revisão do processo de acesso e autenticação de usuários, a atualização do fluxo de *login*, a criação do mecanismo de recuperação de senha, a proteção contra tentativas de acesso inválidas sucessivas e o bloqueio temporário. Essas funcionalidades são fundamentais para aumentar a segurança da plataforma, reduzir riscos associados ao uso indevido de credenciais e oferecer ao usuário mecanismos mais completos de acesso e recuperação de conta.

Outro conjunto relevante de melhorias refere-se à implantação de notificações automáticas por *e-mail*. Essa evolução permite maior transparência e rastreabilidade nos processos da plataforma, uma vez que usuários e administradores passam a receber comunicações associadas ao cadastro de usuários, confirmação de recebimento de solicitações, avisos de novas demandas, aprovação ou rejeição de cadastros e atualizações relacionadas aos agendamentos. Com isso, a plataforma passa a oferecer um fluxo de comunicação mais estruturado, reduzindo a dependência de interações manuais externas ao sistema.

Além das funcionalidades operacionais, esta etapa contemplou melhorias de usabilidade e interface, incluindo o aprimoramento de mensagens de erro e exceção, ajustes visuais da aplicação e suporte à internacionalização. Esses aspectos são relevantes para ampliar a clareza das interações com o sistema, reduzir ambiguidades durante o uso e preparar a plataforma para atender diferentes perfis de usuários, incluindo pesquisadores, administradores e potenciais usuários externos.

O presente relatório detalha também melhorias orientadas por ferramentas de análise de qualidade de código e segurança, com destaque para o uso do SonarQube e para os ajustes de problemas encontrados com o auxílio de testes de vulnerabilidade e penetração, também conhecido como *Penetration Test* (pen-test). Foram realizadas correções relacionadas à exposição de informações sensíveis, padronização de respostas de erro, reforço dos mecanismos de autenticação, revisão da política de validação de senhas, aprimoramento no padrão do código

de acesso de uso único, também conhecido como *One-Time Password* (OTP), implementação de mecanismos de limitação de requisições por intervalo de tempo (*rate limiting*) e redução da exposição de informações do servidor web. Essas ações são particularmente importantes porque a PAARRC, cujo objetivo é viabilizar o acesso e compartilhamento remoto de recursos laboratoriais, deve operar com elevados critérios de segurança.

Dessa forma, o presente relatório documenta a evolução da PAARRC em direção a uma plataforma mais madura, segura e adequada ao uso operacional. Visando atender seus objetivos, este relatório está organizado da seguinte forma: a Seção 2 apresenta uma visão geral da plataforma, que foi detalhada no relatório [1]; a Seção 3 detalha os principais aprimoramentos implementados na plataforma; a Seção 4, por fim, apresenta os comentários finais e conclusões do relatório.

# 2 Visão geral da plataforma

A Plataforma Multiusuário Brasil 6G ou PAARRC é um ambiente multiusuário desenvolvido para possibilitar o agendamento, o gerenciamento e o acesso remoto a recursos computacionais e laboratoriais compartilhados, com intuito de viabilizar diferentes atividades de pesquisa e desenvolvimento, especialmente no contexto de redes móveis e tecnologias relacionadas ao Projeto Brasil 6G.

O objetivo principal da plataforma é permitir que pesquisadores, colaboradores e instituições parceiras utilizem parte da infraestrutura laboratorial de forma segura, organizada e controlada. A concepção da plataforma, seus requisitos funcionais e não funcionais, suas interfaces e sua arquitetura de persistência de dados foram especificados na Parte 1 desta atividade, conforme detalhado em [1]. Esta seção apresenta uma visão geral desses elementos, oferecendo o contexto necessário para a compreensão dos aprimoramentos descritos nas seções seguintes.

O acesso à plataforma é restrito a uma rede interna do Instituto Nacional de Telecomunicações (Inatel), isolada e dedicada a este fim, e ocorre por meio de uma *Virtual Private Network* (VPN). Em termos simplificados, a VPN funciona como um canal seguro entre o usuário e a rede interna dos equipamentos. Primeiro, o usuário realiza a autenticação na VPN institucional e, somente depois disso, é possível acessar a interface web da plataforma. Essa abordagem mantém a aplicação e os recursos laboratoriais protegidos em uma rede privada, reduzindo a exposição a acessos indevidos pela internet pública; portanto, a VPN age como uma camada adicional de proteção para a comunicação entre o usuário e a infraestrutura interna, incluindo elementos de redes e os recursos compartilhados [2].

Após estabelecer a conexão segura, o usuário interage com uma interface web responsiva, acessada diretamente pelo navegador. Essa interface se comunica com os serviços de *backend*, componentes responsáveis por executar as regras internas da plataforma. Entre essas responsabilidades estão a autenticação, por meio do controle de acesso, a autorização de acesso, com base no perfil do usuário, o agendamento e a liberação do uso de recursos.

A Figura 1 apresenta uma visão geral dessa arquitetura. Nela, é possível observar que a VPN é a camada mais externa, sendo a interface entre a Internet e a rede interna. Na sequência, as solicitações passam pela interface web e pelos serviços internos da PAARRC, que verificam permissões, processam agendamentos e encaminham as operações necessárias até os recursos laboratoriais. Essa separação de responsabilidades facilita a manutenção do sistema e torna mais claro o caminho percorrido entre o usuário e os equipamentos do laboratório.

## 2.1 Protocolos de conexão

O acesso remoto aos equipamentos ocorre sempre de maneira indireta, por meio de um *gateway* especializado para o encaminhamento das conexões remotas. Na PAARRC, o *gateway* foi desenvolvido com base no Apache Guacamole, que permite abrir sessões remotas diretamente no navegador, sem que o usuário precise instalar algum programa adicional em seu dispositivo, tendo como única dependência a existência de um web *browser* [3].

A plataforma suporta atualmente três dos protocolos de conexão remota mais amplamente utilizados, sendo eles:

- *Secure Shell* (SSH), utilizado principalmente para o acesso seguro entre equipamentos que executam sistemas operacionais derivados do UNIX (Unix-like), apesar de ser possível utilizar também no Microsoft Windows [4];

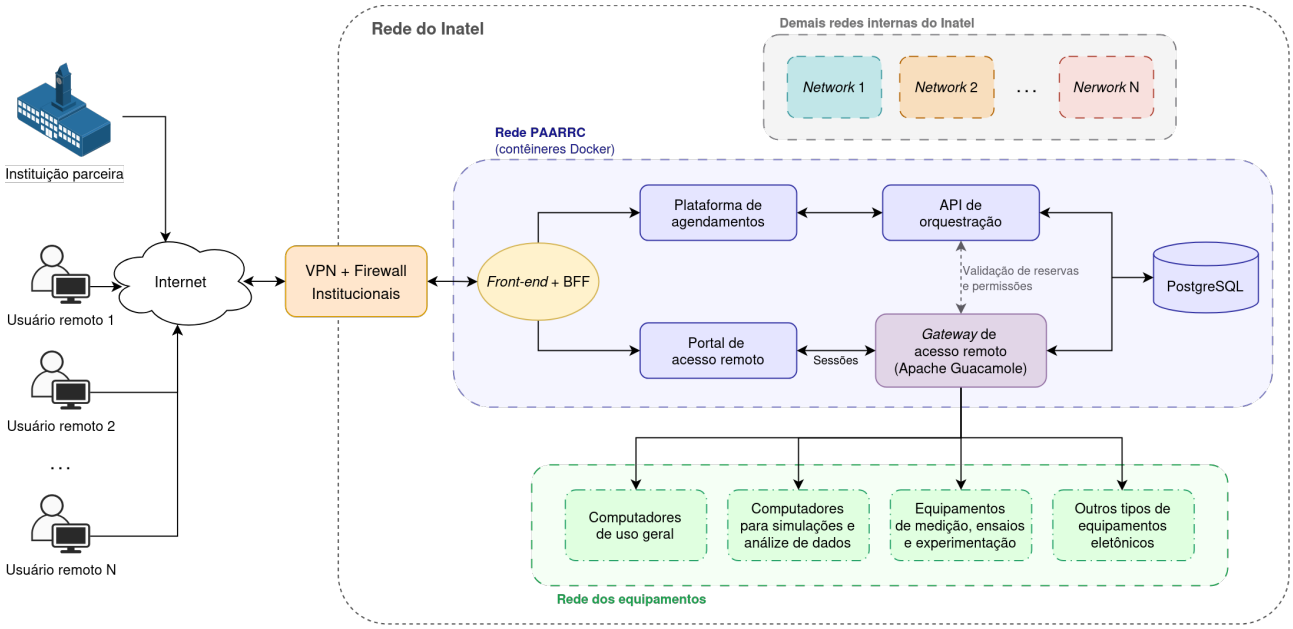



Figura 1: Visão geral da arquitetura da Plataforma Multiusuário Brasil 6G - PAARRC.

- *Remote Desktop Protocol* (RDP), criado pela Microsoft para permitir o acesso remoto, com interface gráfica, entre máquinas Windows que possuam o *Microsoft Terminal Services* instalado [5];
- *Virtual Network Computing* (VNC), protocolo de código aberto baseado em *Remote Framebuffer* e agnóstico quanto ao sistema operacional, que permite realizar o acesso remoto com interface gráfica [6].

A mediação realizada pelo *gateway* padroniza a forma de acesso aos recursos e aumenta o controle sobre as sessões. Desta maneira, a plataforma consegue aplicar regras de autenticação, autorização e auditoria independentemente do protocolo utilizado. Além disso, esse modelo contribui para o isolamento entre usuários, evitando que haja conflito entre suas sessões remotas.

## 2.2 Fluxos da plataforma

Do ponto de vista organizacional, a plataforma é estruturada em torno de algumas entidades principais: usuários, organizações, projetos, *setups*, equipamentos e reservas.

Os usuários representam as pessoas que acessam a plataforma; cada usuário possui um perfil com permissões específicas, como administrador do sistema, administrador da organização e usuário regular. Esse modelo de controle baseado em papéis é conhecido como *Role-Based Access Control* (RBAC), pois as permissões são atribuídas de acordo com o papel desempenhado por cada pessoa no sistema, e não individualmente para cada operação [7].

As organizações representam as instituições parceiras que utilizam a plataforma. Elas agrupam usuários sob uma mesma entidade institucional, permitindo que a plataforma atenda diferentes grupos, cada um com sua própria especificidade. Os projetos conectam pessoas e recursos relacionados a uma mesma atividade. Desta forma, a entidade projeto relaciona as entidades usuário, que participam do projeto, e *setups* associados ao projeto. Ainda neste mesmo contexto, um *setup* se trata de uma configuração específica para cada recurso, que permite a

utilização deste. Deste modo, um *setup* representa ao mesmo tempo um recurso, como computador, e a configuração para o acesso deste recurso, como tipo de conexão, endereço, porta, entre outros.

Essa cadeia de associações é essencial para o funcionamento da plataforma. Em termos simples, o sistema verifica a organização à qual o usuário pertence, em quais projetos ele está vinculado e quais *setups* estão associados a esses projetos. Com base nessas informações, a plataforma determina quais recursos o usuário pode visualizar, reservar e acessar remotamente, a fim de garantir que cada usuário utilize apenas os recursos compatíveis com suas atividades e permissões.

A utilização dos recursos é organizada por meio de um sistema de reservas. Cada reserva possui um ciclo de vida composto por estados bem definidos, representando sua situação na plataforma. Em um primeiro momento, a reserva é criada com o *status* de solicitada. Em seguida, ela passa por análise do administrador e pode ser aprovada ou recusada; se aprovada, o *status* vai para provisionado, se recusada, vai para negado. Quando negado, o fluxo termina neste ponto. Dado que a solicitação foi aprovada, ao chegar o horário programado, a reserva torna-se ativa, e a plataforma libera o acesso remoto ao recurso associado. No final do período reservado, a reserva passa para o *status* finalizada, fazendo com que as sessões sejam encerradas automaticamente e as permissões temporárias sejam revogadas. Além destes estados já supracitados, existem estados complementares, como cancelada e expirada. Uma reserva recebe o *status* de cancelada quando o usuário ou o administrador interrompe a reserva já aprovada antes de sua utilização; já o *status* expirada, por sua vez, indica que a reserva não foi utilizada dentro do prazo previsto, perdendo assim sua validade.

As transições entre os estados da reserva são controladas pelo próprio sistema e registradas para auditoria. Esses registros permitem acompanhar o histórico das operações realizadas na plataforma. Além disso, determinadas mudanças de estado podem acionar ações automáticas, como o envio de notificações, a liberação de acesso, o provisionamento de ambientes e o encerramento das sessões de um usuário.

## 2.3 Segurança

A segurança é um dos principais pontos de atenção no projeto. No fluxo de autenticação do usuário, a validação de acesso passou a utilizar *Multi-factor Authentication* (MFA), visto que este modelo exige mais de uma etapa de autenticação para confirmar a identidade do usuário [8]. Essas etapas podem se basear em algo que o usuário sabe, tipicamente a senha, algo que ele tem, como celular, *e-mail* ou *hardware token*, ou, ainda, algo que ele é, como no caso de reconhecimento facial ou biométrico [9]. Essa camada adicional de proteção reduz o risco de acesso indevido, mesmo em situações nas quais uma senha esteja comprometida.

As sessões na plataforma, assim como as conexões remotas, são protegidas por autorizações temporárias, chamadas de *session tokens*. A plataforma utiliza o padrão *JSON Web Tokens* (JWT), que permite representar as informações sobre a sessão, o usuário e o período de validade da autorização em um formato compacto e verificável, além de ainda realizar a gerência destes *tokens*. Com o JWT é possível definir um tempo de expiração e validar determinadas informações da sessão sem repetir todo o processo de autenticação em cada requisição, porém algumas consultas ao banco de dados continuam tendo de ser feitas, como quando necessário verificar informações atualizadas, permissões modificadas ou condições de revogação. A implementação também deve seguir práticas de segurança relacionadas à validação da assinatura, dos campos do *token* e dos algoritmos permitidos [10, 11].

A plataforma conta ainda com mecanismos para a recuperação de senha, o encerramento automático de sessões ociosas e o registro de eventos relevantes em *logs* de auditoria. Esses registros armazenam informações sobre ações importantes realizadas no sistema, como acessos, alterações de dados, solicitações de reserva e decisões administrativas. Esse histórico contribui para a rastreabilidade das operações e está em conformidade com as políticas de segurança de não repúdio, ou irretratabilidade.

## 2.4 Infraestrutura

A persistência das informações é realizada em um banco de dados relacional PostgreSQL. Esse banco centraliza os dados de usuários, organizações, projetos, *setups*, equipamentos, reservas, notificações e registros de auditoria. Também armazena informações necessárias ao funcionamento do *gateway* de acesso remoto. Por utilizar um modelo relacional, o PostgreSQL permite organizar os dados em tabelas conectadas por relacionamentos. Essa estrutura é adequada para representar, por exemplo, quais usuários pertencem a uma organização, quais usuários estão vinculados a um projeto e quais recursos fazem parte deste projeto [12].

Toda a solução é executada em contêineres Docker e organizada de forma declarativa via Docker Compose. Um contêiner reúne uma aplicação e os componentes necessários para sua execução em um ambiente isolado. Já o Docker Compose descreve, em um arquivo de configuração, os serviços, as redes e os volumes que compõem a solução [13]. Essa abordagem facilita a implantação e a manutenção da plataforma, por permitir que os serviços sejam executados de maneira padronizada, garantindo o mesmo comportamento nos ambientes de desenvolvimento, teste e produção. Ainda que cada um destes ambientes rode em infraestruturas distintas, como serviços de nuvem ou servidores locais, os contêineres ajudam a reduzir diferenças de configuração e comportamento e o modelo também facilita a atualização dos componentes e a criação de novos ambientes experimentais de forma mais flexível e ágil.

Além dos aspectos funcionais, a especificação também define requisitos não funcionais, isto é, características que orientam quanto à qualidade, à usabilidade, à manutenção, à segurança e às conformidades que devem estar presentes na solução. Entre esses requisitos estão uma interface responsiva, adaptável a diferentes tamanhos de tela, possibilitando o uso em diferentes tipos de dispositivos; suporte aos temas claro e escuro; internacionalização, com suporte ao português brasileiro, inglês e espanhol; componentes voltados para a acessibilidade, permitindo a navegação pelo teclado; *design* consistente, conservando o padrão por toda a plataforma, dentre outros.

## 2.5 Fase atual

As seções seguintes descrevem a evolução da versão inicial da plataforma, incluindo ajustes no fluxo do usuário regular, aprimoramento do processo de autenticação, a implantação de notificações automáticas, as melhorias de usabilidade e as correções relacionadas à segurança da plataforma e à qualidade do código, com a adição de plataformas auxiliares de monitoramento e verificação.

# 3 Aprimoramentos implementados na plataforma

Esta seção apresenta as principais melhorias realizadas na plataforma multiusuário, ambiente utilizado por diferentes perfis de pessoas: pesquisadores, estudantes e usuários externos de instituições parceiras. O texto a seguir explica as mudanças feitas no fluxo de uso do sistema, para o perfil de usuário regular, na forma de acesso ao sistema, criação de notificações por *e-mail*, aprimoramento na interface e melhorias relacionadas à qualidade e à segurança da plataforma.

Para facilitar a compreensão, a plataforma pode ser comparada ao sistema de reservas de um laboratório físico: existe um catálogo de equipamentos, um responsável que autoriza o uso, uma agenda com o registro de datas e horários de quem vai utilizar cada item. Porém, no caso da PAARRC, todas as operações acontecem pela Internet: a "chave de acesso" é o *login* do usuário, a "autorização do responsável" é a aprovação pelo administrador, a "agenda" é o sistema de reservas e a "utilização do item reservado" é a conexão remota ao equipamento, feita pelo navegador, sem que o usuário precise estar fisicamente no local.

## 3.1 Ajustes no fluxo de uso do perfil de usuário regular

Os ajustes no fluxo de uso do perfil regular tiveram como objetivo facilitar a navegação e deixar mais claro o que o usuário pode fazer dentro da plataforma. O perfil regular representa, em geral, pessoas que precisam solicitar o uso de recursos, como pesquisadores, colaboradores ou usuários externos. Por isso, o fluxo precisa mostrar de forma simples quais equipamentos estão disponíveis, como solicitar um agendamento, *status* dos agendamentos solicitados, quando o acesso remoto poderá ser utilizado e o mecanismo de acesso ao recurso reservado, quando este já estiver liberado para o uso do usuário em questão. As melhorias desta etapa reduzem pontos de confusão durante o uso, evitam operações incorretas e tornam a plataforma mais intuitiva e coerente com o funcionamento esperado de um ambiente compartilhado.

Entre as principais melhorias, destacam-se as correções no uso do sistema por usuários com perfil regular, especialmente na solicitação de reservas e no acesso a *setups* e equipamentos pelo Portal de Acesso Remoto. No contexto deste projeto, *setup* significa o conjunto de equipamentos, configurações e recursos experimentais disponibilizados para uso. O Portal de Acesso Remoto é a área da plataforma que permite acessar esses recursos à distância, quando o usuário possui autorização. Antes dos ajustes, algumas funcionalidades estavam indisponíveis, pois a plataforma ainda não possuía um fluxo completo de cadastro de projetos capaz de relacionar usuários, projetos e *setups*. Essa ausência gerava erros como o ilustrado na Figura 2. O erro se devia à falta de associações entre usuários regulares e projetos, além da ausência de vínculo entre projetos e *setups*. Em termos práticos, a plataforma não tinha informação suficiente para responder a perguntas básicas, como: qual usuário faz parte de qual projeto e quais equipamentos esse projeto pode utilizar. Sem essas relações, o sistema não conseguia determinar quais recursos cada usuário poderia visualizar, reservar ou acessar remotamente. Como consequência, o processo de agendamento e o acesso remoto ficavam indisponíveis para o perfil regular.

### 3.1.1 Soluções aplicadas

O módulo de cadastro de projetos foi revisado e complementado. Com isso, passou a ser possível criar e gerenciar projetos diretamente na aplicação, conforme apresentado nas Figuras 3 e 4. Nesta atual versão, o projeto passa a funcionar como um agrupador, relacionando os usuários participantes do projeto e os *setups* que poderão ser utilizados por esse grupo.

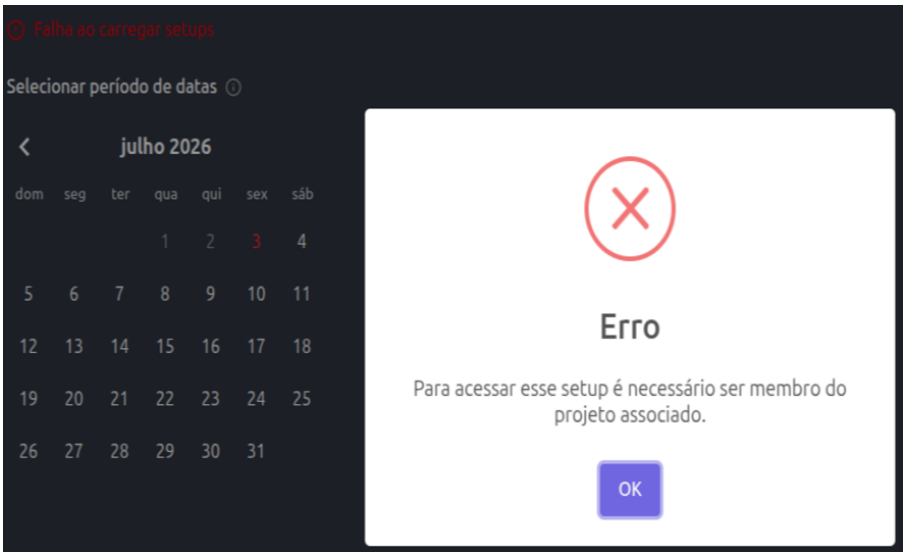



Figura 2: Erro exibido ao usuário regular, sem projeto associado.

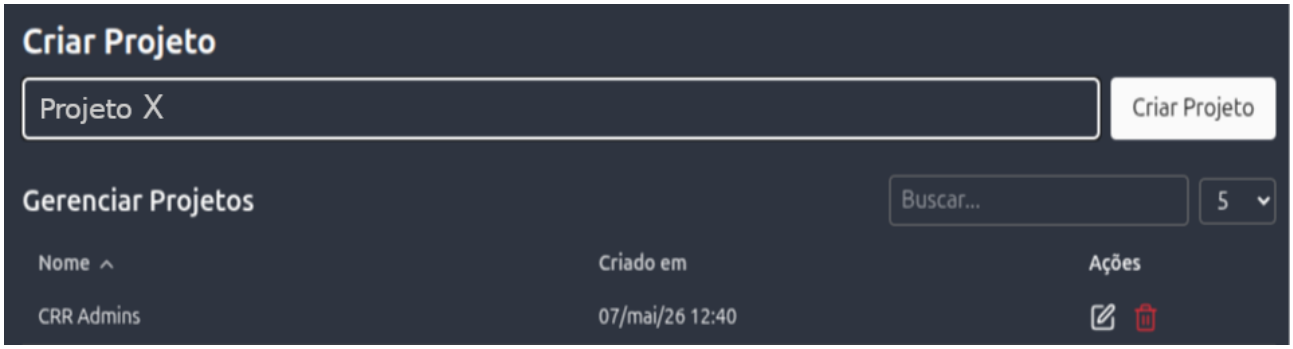



Figura 3: Cadastro de projeto.

Além disso, foi modificada a tela de edição do projeto, apresentada na Figura 5, para permitir a associação de usuários regulares e *setups*. Essa alteração reforça o papel do projeto como ponto central de organização das permissões. Quando um usuário é vinculado a um projeto e esse projeto possui determinados *setups*, a plataforma passa a reconhecer que o usuário pode solicitar o uso desses recursos.

A vinculação de usuários regulares ao projeto passou a ser realizada pela opção Adicionar Usuário, conforme ilustrado na Figura 6.

De maneira complementar, a opção Adicionar *Setup* permite incluir no projeto os *setups* que ficarão disponíveis aos usuários vinculados, conforme apresentado na Figura 7.

A partir das associações projeto → usuário regular e projeto → *setup*, mostradas na Figura 8, a aplicação passa a identificar quais usuários possuem autorização para visualizar e solicitar o uso de cada *setup*. Em outras palavras, o projeto funciona como uma ponte entre pessoas e recursos: quem participa do projeto enxerga apenas os *setups* vinculados a ele.

Uma forma simples de entender essa organização é imaginar o projeto como uma turma de laboratório: a turma tem uma lista de alunos (os usuários vinculados) e uma lista de bancadas que ela pode usar (os *setups* vinculados). Um aluno que não está na lista da turma não enxerga as bancadas daquela turma; da mesma forma, uma bancada que não foi reservada para a turma não aparece como opção. Antes das melhorias, era como se existissem alunos e bancadas

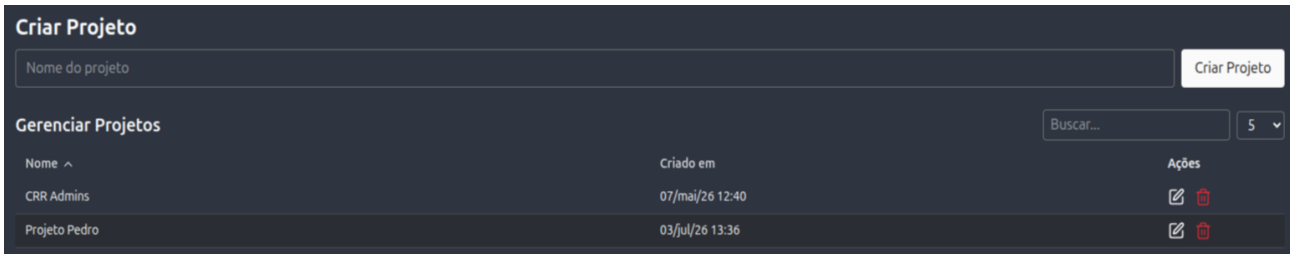



Figura 4: Projeto cadastrado.


Figura 5: Edição de projeto.

cadastrados, mas nenhuma lista de turma interligando uns aos outros — por isso o sistema não sabia o que mostrar a cada pessoa.

Com essa configuração concluída, o usuário regular passa a visualizar os *setups* associados ao seu projeto e pode solicitar agendamentos para utilização dos recursos, conforme ilustrado na Figura 9.

Depois que a solicitação percorre o fluxo de análise e aprovação administrativa, o usuário regular consegue acompanhar seus agendamentos pela plataforma, conforme apresentado na Figura 10.

Com o agendamento aprovado, o acesso aos *setups* vinculados ao projeto é liberado ao usuário por meio do Portal de Acesso Remoto, conforme demonstrado na Figura 11.

A disponibilização do cadastro de projetos com associação de usuários e *setups* foi determinante para completar o fluxo de agendamento e acesso remoto do perfil regular. Antes da alteração, a ausência dessas relações impedia o uso das funcionalidades mais importantes do sistema. Após os ajustes, o processo passou a funcionar de forma integrada; o administrador cria o projeto, vincula os usuários e os *setups*, o usuário solicita o agendamento e, depois da aprovação, acessa remotamente os equipamentos autorizados, conforme apresentado na Figura 12.

Vale esclarecer que, ao acessar o equipamento pelo Portal de Acesso Remoto, o usuário passa


Figura 6: Adição de usuário ao projeto.

a ver, dentro do próprio navegador, a tela do computador fisicamente conectado ao equipamento no laboratório. É como assumir o controle daquele computador à distância. O teclado e o mouse do usuário comandam a máquina remota, e tudo o que acontece na máquina é exibido em tempo real. Esse acesso só é liberado durante o período do agendamento aprovado, o que garante que duas pessoas não disputem o mesmo equipamento ao mesmo tempo.

## 3.2 Aprimoramentos no acesso e na autenticação de usuários

O acesso e a autenticação de usuários são etapas centrais para a segurança da plataforma. A autenticação é o processo que confirma se a pessoa que está tentando entrar realmente possui permissão para usar o sistema. Nesta etapa, foram implementadas melhorias no *login*, na recuperação de senha e na proteção contra tentativas de acesso por força bruta. Essas ações reduzem riscos relacionados ao uso indevido de credenciais, aumentam a confiança no sistema e ajudam a garantir que apenas usuários autorizados utilizem as funcionalidades disponíveis.

A seguir estão apresentados os principais ajustes e melhorias realizados na etapa de acesso e autenticação de usuários.

### 3.2.1 Revisão do mecanismo de autenticação

No modelo anterior, a autenticação dependia apenas da combinação de *e-mail* e senha. Embora esse formato seja clássico, ele pode ser insuficiente quando a senha é fraca, reutilizada em outros serviços, descoberta por terceiros ou exposta em vazamentos de dados. Nesses casos, uma pessoa não autorizada poderia tentar acessar a conta apenas sabendo essas duas informações.

Para elevar o nível de proteção da aplicação, foi incorporado um segundo fator de autenticação, configurando um modelo de autenticação multifator (MFA). Na prática, isso significa


Figura 7: Adição de *setup* ao projeto.

que o usuário precisa comprovar sua identidade em duas etapas, primeiro com *e-mail* e senha e depois com um código temporário enviado ao *e-mail* cadastrado. Assim, mesmo que a senha seja descoberta, o acesso ainda depende do código recebido pelo usuário. Esse tipo de mecanismo acrescenta uma camada de proteção ao exigir mais de uma evidência para confirmar a identidade do usuário [8, 14].

Uma comparação útil é a de um cofre que exige uma senha e uma chave. Conhecer apenas a senha não é suficiente, assim como possuir somente a chave também não permite abrir o cofre. Na plataforma, a senha representa o primeiro fator e o código temporário representa a segunda confirmação. Esse princípio é semelhante ao empregado por serviços bancários, sistemas de *e-mail* e outras aplicações que solicitam um código ou uma confirmação adicional durante o acesso.

### 3.2.2 Fluxo atualizado de *login*

Após a adoção da autenticação em duas etapas, o *login* passou a seguir o fluxo descrito a seguir:

1. Inserção de *e-mail* e senha
   Nessa primeira etapa, o usuário informa suas credenciais de acesso, ou seja, o *e-mail* cadastrado e a senha, na página de *login* da aplicação, conforme mostrado na Figura 13.

2. Validação das credenciais
   Em seguida, a aplicação verifica o *e-mail* e a senha fornecidos. Caso as credenciais sejam válidas, o fluxo prossegue para a etapa complementar de autenticação.

3. Geração e envio do código de uso único (OTP)
   Somente depois da validação das credenciais, isto é, primeiro fator de autenticação realizado com sucesso, é que o sistema gera uma OTP e envia para o *e-mail* cadastrado do usuário. Esse código funciona como uma segunda senha, porém de uso único e que só é válida dentro de uma janela específica de tempo, após essa janela ela expira e passa a ser inválida. Um exemplo de *e-mail* com a OTP recebida é apresentado na Figura 14.

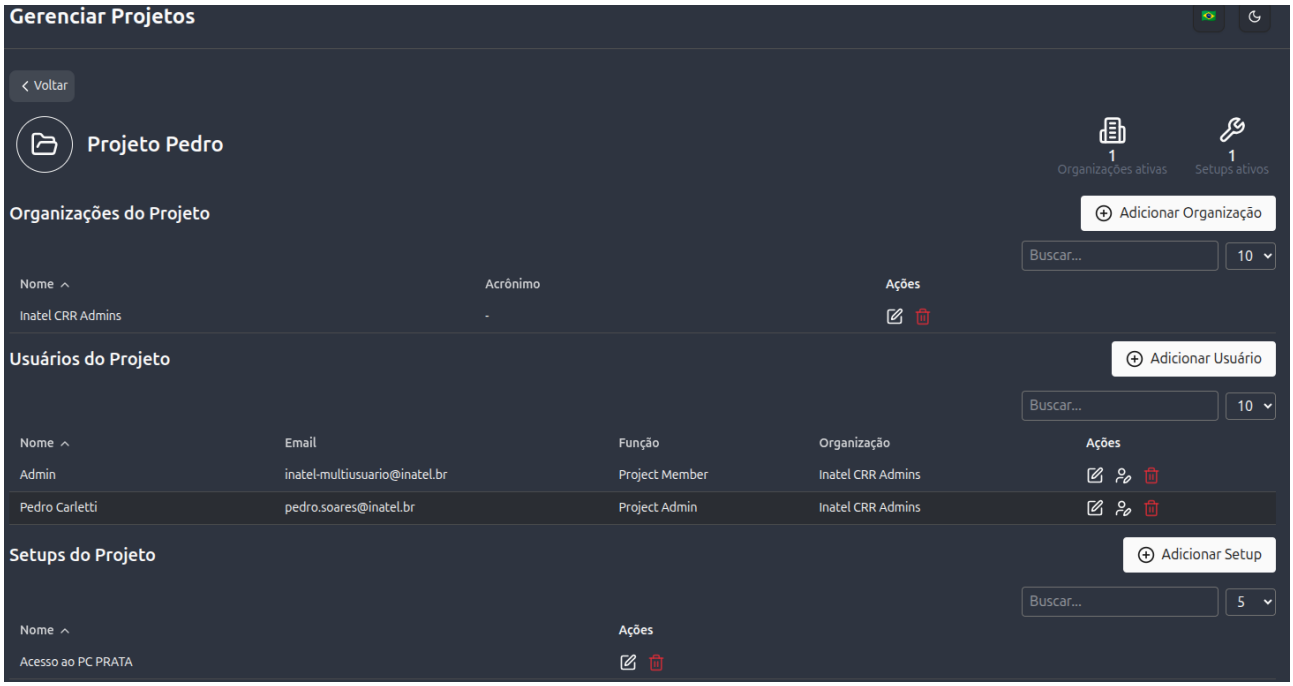



Figura 8: Projeto com usuários e *setups* associados.

4. Validação da OTP
   Na etapa seguinte, o usuário insere o código recebido na tela subsequente, apresentada após a validação da primeira etapa de autenticação, como ilustrado na Figura 15. A aplicação verifica a correspondência do código informado pelo usuário e também se ainda se encontra dentro da janela temporal de validade.

5. Conclusão do *login*
   Com a validação correta da segunda etapa de autenticação, este fluxo se finaliza e ao usuário é concedido acesso à plataforma, com o seu redirecionamento para dentro do sistema.

### 3.2.3 Criação do recurso de recuperação de senha

Antes da implementação desse recurso, a aplicação não oferecia uma forma autônoma para redefinir a senha em caso de esquecimento ou necessidade de troca por motivo de segurança. Nesses cenários, o usuário dependia de apoio externo, como intervenção de um administrador ou suporte técnico, o que aumentava o esforço operacional e prejudicava a experiência de uso.

Em sistemas que utilizam *e-mail* e senha, a recuperação de senha é uma funcionalidade importante para a continuidade do acesso. Para que esse processo seja seguro, a redefinição deve utilizar códigos ou links temporários, limitar tentativas e evitar respostas que revelem se uma conta está ou não cadastrada [15]. Para suprir essa necessidade, foi criado um fluxo de recuperação de senha com utilização da OTP, de modo que apenas o usuário com acesso ao *e-mail* cadastrado consiga realizar a troca da senha.

A recuperação de senha passou a seguir os passos descritos a seguir:

- Passo 1 - Solicitação de redefinição de senha
  Na tela de *login*, o usuário informa seu *e-mail* e seleciona a opção *"Esqueceu a senha?"* para iniciar a recuperação de acesso, conforme mostrado na Figura 16.

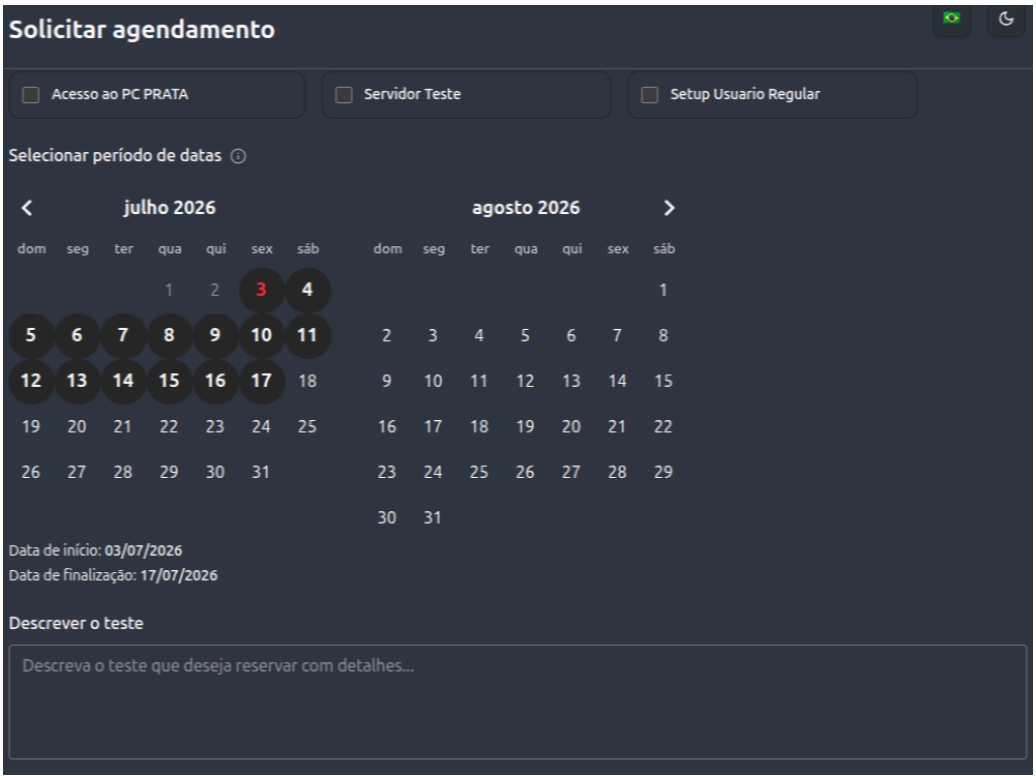



Figura 9: Solicitação de agendamento.

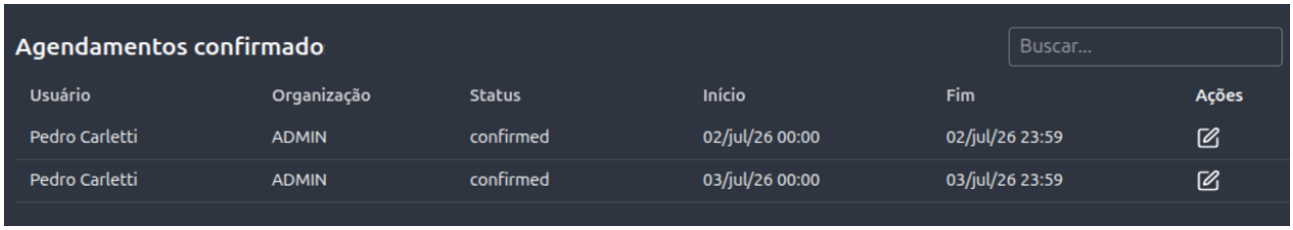



Figura 10: Agendamentos do usuário.

- Passo 2 - Envio do código de verificação
  Com a solicitação registrada, um código temporário de verificação (OTP) é gerado e enviado automaticamente ao *e-mail* associado ao usuário, conforme exemplificado na Figura 17.

- Passo 3 - Validação e definição da nova senha
  Em seguida, o usuário é encaminhado para a tela de recuperação de senha, onde deve preencher o código recebido e definir a nova senha, conforme ilustrado na Figura 18.

- Passo 4 - Finalização do processo
  Quando o código e a nova senha são validados corretamente, a credencial do usuário é atualizada. Logo depois, a aplicação envia um *e-mail* informando que a redefinição da senha foi concluída com sucesso, conforme representado na Figura 19.

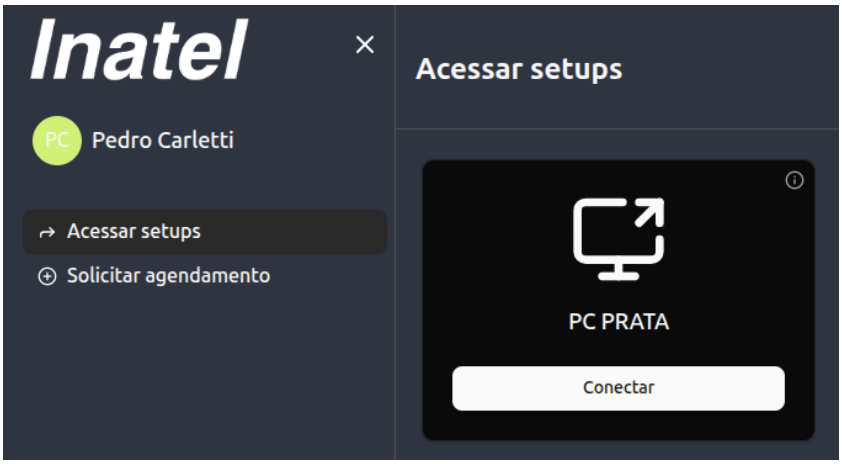



Figura 11: Acesso ao *setup* via portal remoto.

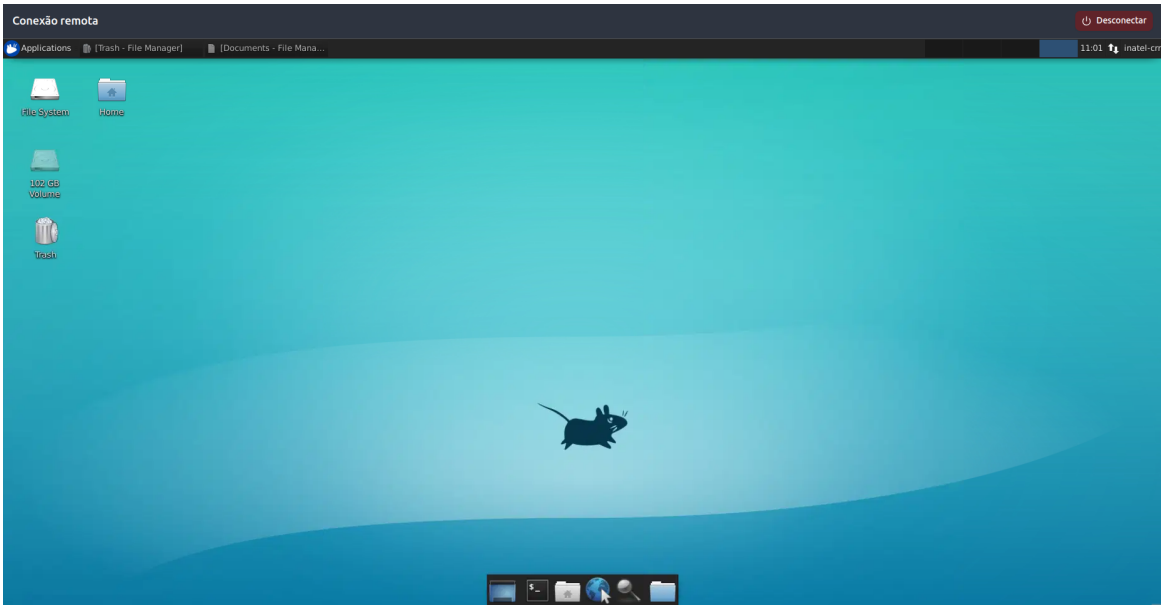



Figura 12: Acesso ao equipamento remoto, por conexão VNC.

### 3.2.4 Proteção contra diversas tentativas de autenticação sucessivas

No cenário anterior, a aplicação não limitava as tentativas de autenticação, podendo um atacante errar infinitas vezes o *e-mail*/senha até acertar a credencial correta, sem bloqueio temporário ou outro tipo de proteção oferecida pela plataforma.

Este tipo de cenário é um risco de segurança, por favorecer ataques do tipo força bruta. Nesse tipo de ataque, diversas opções de *e-mail* e/ou senha são testadas até que uma delas funcione. Sem limite de tentativas, a aplicação fica mais exposta a acessos não autorizados. A limitação de tentativas de autenticação sequencialmente erradas, junto a um tempo de espera progressivo entre cada nova tentativa, é uma das medidas recomendadas para reduzir esse tipo de abuso [8, 16].

Um ataque de força bruta pode ser comparado a alguém testando todas as combinações possíveis de um cadeado numérico: com tempo e tentativas ilimitadas, mais cedo ou mais tarde, a combinação certa é encontrada. O bloqueio temporário quebra essa lógica: após poucas


Figura 13: Interface de *login* da aplicação.

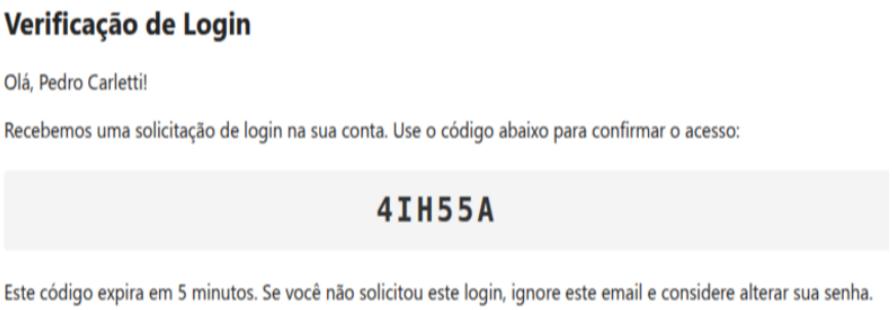



Figura 14: *E-mail* contendo o código de verificação enviado ao usuário.

tentativas erradas, o "cadeado" trava por um período cada vez maior, tornando o ataque de força bruta demasiadamente lento, a ponto de inviabilizá-lo.

Como medida de mitigação, foi implementado o bloqueio temporário da conta após um número definido de falhas consecutivas. O recurso dificulta tentativas repetitivas de acesso e ajuda a proteger a conta do usuário. Também foi incluída uma notificação por *e-mail* sempre que o bloqueio for aplicado, permitindo que o usuário saiba se houve uma tentativa indevida de acesso.

### 3.2.5 Funcionamento do bloqueio temporário

O bloqueio temporário foi estruturado conforme o fluxo a seguir:

- Tentativas de autenticação
  O processo inicia quando o usuário tenta acessar a aplicação informando *e-mail* e senha.


Figura 15: Interface de validação do código de verificação.

- Registro de falhas
  A cada falha consecutiva, a aplicação registra a ocorrência e atualiza um contador de tentativas inválidas vinculado ao usuário.

- Bloqueio automático
  Quando o limite máximo de tentativas inválidas é atingido, atualmente definido como cinco, o acesso do usuário é bloqueado por trinta minutos.

- Notificação por *e-mail*
  No instante em que o bloqueio é aplicado, a aplicação envia automaticamente um *e-mail* ao usuário comunicando a ocorrência e apresentando orientações em caso de tentativa indevida de acesso, conforme mostra a Figura 20.

- Bloqueio de acesso
  Enquanto o bloqueio estiver ativo, novas tentativas de autenticação são recusadas. Nesse período, a interface exibe uma mensagem informativa ao usuário, tal qual mostrada na Figura 21.

- Tempo de bloqueio
  A restrição permanece ativa por 30 minutos, intervalo no qual as tentativas de *login* do usuário são negadas.

- Liberação automática
  Encerrado o período configurado, o bloqueio é removido automaticamente e novas tentativas de autenticação voltam a ser permitidas.

- Liberação via redefinição de senha
  Alternativamente, o usuário pode realizar o procedimento de recuperação de senha. Ao finalizar esse processo com sucesso, o bloqueio é removido e o acesso é restabelecido.

## 3.3 Implantação de notificações automáticas por *e-mail*

A implantação de notificações automáticas por *e-mail* representa uma melhoria importante para a transparência e a organização dos processos da plataforma. Por meio desse recurso,


Figura 16: Interface de *login* com opção de recuperação de senha.

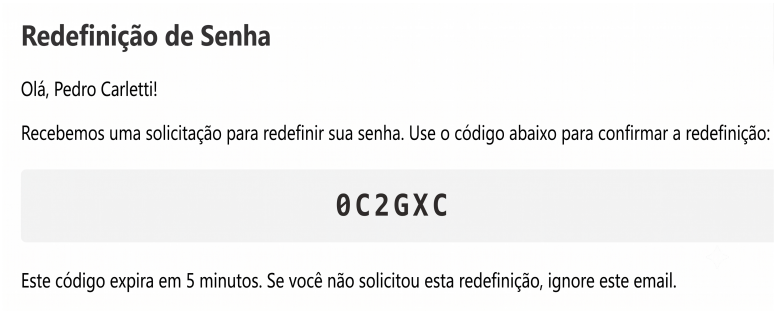



Figura 17: *E-mail* enviado ao usuário contendo o código de verificação.

usuários e administradores recebem avisos sempre que ocorre uma ação relevante, como envio de cadastro, confirmação de solicitação, aprovação ou rejeição de um pedido e atualização de algum agendamento. Essa comunicação automática reduz a necessidade de mensagens manuais fora da plataforma e facilita o acompanhamento de forma mais clara de cada etapa.

As mensagens enviadas pela aplicação atendem principalmente a dois objetivos — informativo e acionável —, cada um com as características descritas a seguir:

- Informativo: comunicar ações realizadas no sistema ou mudanças de situação, como cadastro recebido ou agendamento aprovado.
- Acionável: indicar que alguém precisa tomar uma ação, como analisar, aprovar ou rejeitar uma solicitação.

Além das mensagens vinculadas ao *login*, também foram incorporadas notificações para os cenários de recuperação de senha, bloqueio do usuário, cadastro de usuário e solicitação de agendamento de uso de recurso.

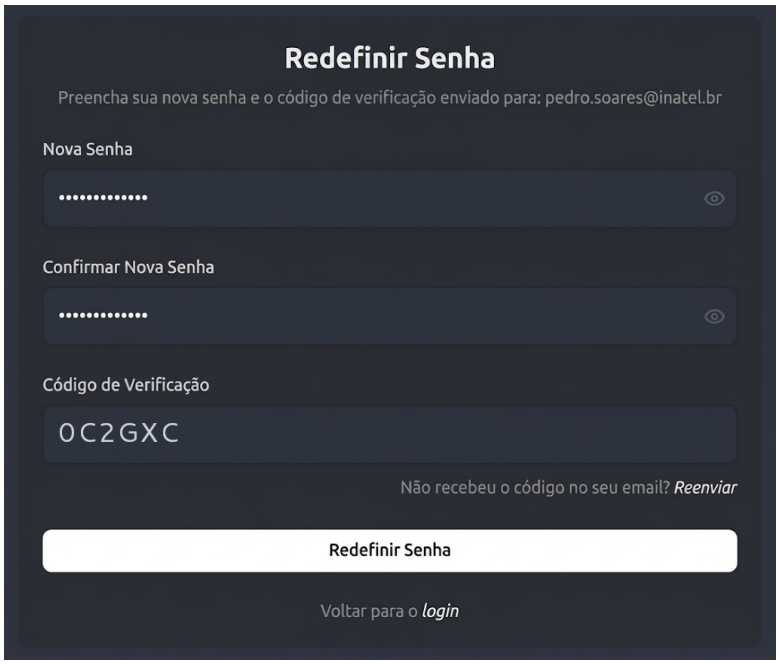



Figura 18: Interface para redefinição de senha com validação por código de verificação.

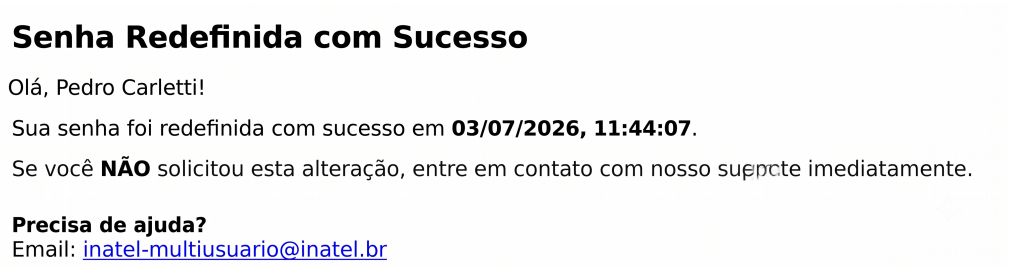



Figura 19: *E-mail* de confirmação da redefinição de senha do usuário.

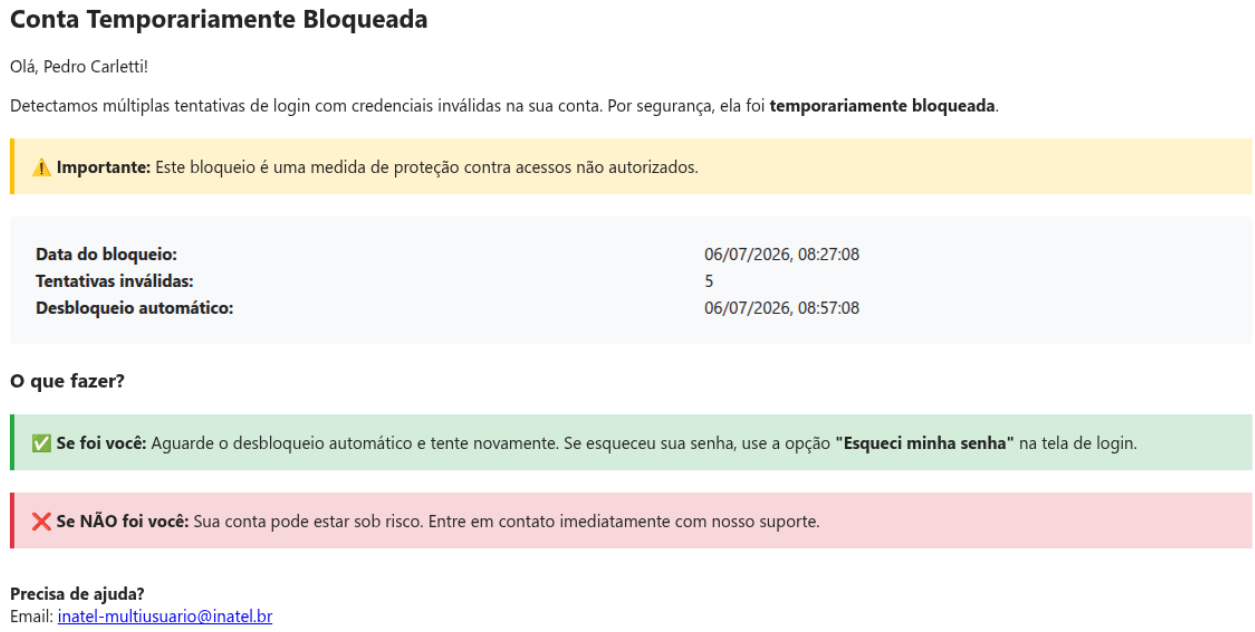
### 3.3.1 Etapas do cadastro de usuário

O cadastro é iniciado pelo usuário na tela de registro da aplicação, onde lhe é solicitado *e-mail*, senha, confirmação de senha, nome, sobrenome e organização à qual pertence, como mostrado na Figura 22.

Depois do envio das informações de registro, a solicitação permanece pendente de avaliação administrativa, isto é, o usuário ainda não poderá acessar o sistema até que um administrador analise o pedido e o aprove. Na sequência, mensagens automáticas por *e-mail* são enviadas ao usuário e ao administrador, ao primeiro, para confirmar o recebimento do cadastro, ao segundo, para alertar sobre a necessidade de análise da nova solicitação de cadastro.

### 3.3.2 Confirmação de recebimento do cadastro ao usuário

A primeira notificação enviada ao usuário confirma o recebimento da solicitação de cadastro e informa que a solicitação será analisada por um administrador antes de ele ter o acesso liberado, conforme mostra a Figura 23.



**Conta Temporariamente Bloqueada**

Olá, Pedro Carletti!

Detectamos múltiplas tentativas de login com credenciais inválidas na sua conta. Por segurança, ela foi **temporariamente bloqueada**.

⚠️ **Importante:** Este bloqueio é uma medida de proteção contra acessos não autorizados.

| | |
|---|---|
| **Data do bloqueio:** | 06/07/2026, 08:27:08 |
| **Tentativas inválidas:** | 5 |
| **Desbloqueio automático:** | 06/07/2026, 08:57:08 |

**O que fazer?**

✅ **Se foi você:** Aguarde o desbloqueio automático e tente novamente. Se esqueceu sua senha, use a opção **"Esqueci minha senha"** na tela de login.

❌ **Se NÃO foi você:** Sua conta pode estar sob risco. Entre em contato imediatamente com nosso suporte.

**Precisa de ajuda?**
Email: inatel-multiusuario@inatel.br

Figura 20: *E-mail* de notificação de bloqueio de conta.


Figura 21: Aviso de bloqueio exibido na aplicação.

### 3.3.3 Aviso de novo cadastro ao administrador

Paralelamente, o administrador é comunicado por *e-mail* sobre a existência de um novo cadastro pendente de aprovação, conforme mostra a Figura 24.

### 3.3.4 Comunicação de aprovação de cadastro ao usuário

Após a aprovação do cadastro, a aplicação envia uma mensagem ao usuário informando que seu acesso foi liberado, conforme mostra a Figura 25.

### 3.3.5 Comunicação de rejeição de cadastro ao usuário

Caso o cadastro seja rejeitado, o usuário recebe a comunicação sobre a rejeição do cadastro por *e-mail*, conforme mostra a Figura 26. Essa mensagem encerra a solicitação e evita dúvidas sobre o resultado da análise.

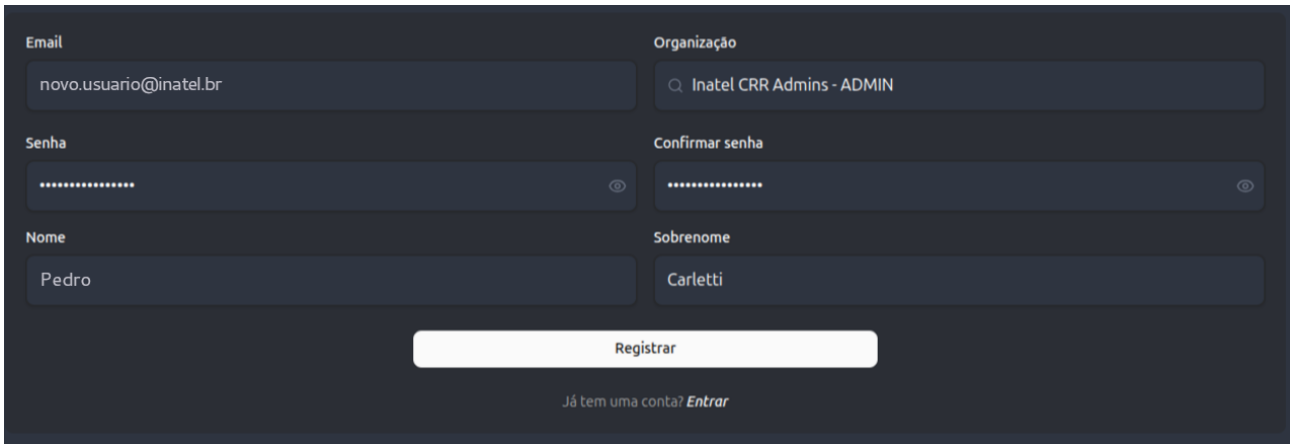



Figura 22: Interface de registro de usuário na aplicação.

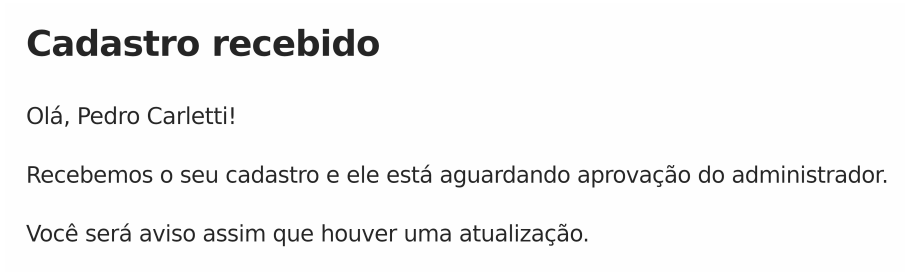



Figura 23: *E-mail* ao usuário, informando o recebimento da solicitação de cadastro.

## 3.4 Fluxo de solicitação de agendamento

O fluxo de solicitação de agendamento é um dos componentes mais relevantes da plataforma, pois organiza o uso compartilhado dos recursos experimentais disponíveis. Como esses recursos podem ser disputados por diferentes usuários ou projetos, o agendamento ajuda a evitar conflitos de horário e uso indevido dos equipamentos. Os aprimoramentos realizados nessa etapa tornaram o processo mais claro e controlado, desde o envio da solicitação pelo usuário até a análise e decisão administrativa.

Com o cadastro aprovado e o vínculo ao projeto configurado pelo administrador, o usuário passa a ter permissão para acessar a área de agendamentos e solicitar o uso dos *setups* associados aos projetos dos quais participa, conforme mostra a Figura 27.

Depois que o agendamento é solicitado, a aplicação envia notificações ao usuário e ao administrador durante o processo de validação. O pedido pode ser aprovado, quando o uso do *setup* é autorizado, ou rejeitado, quando o administrador entende que a solicitação não pode ser atendida naquele momento.

### 3.4.1 Confirmação de recebimento da solicitação ao usuário

Ao registrar a solicitação, o usuário recebe um *e-mail* confirmando que o pedido de agendamento foi recebido e permanece em análise, conforme mostra a Figura 28.

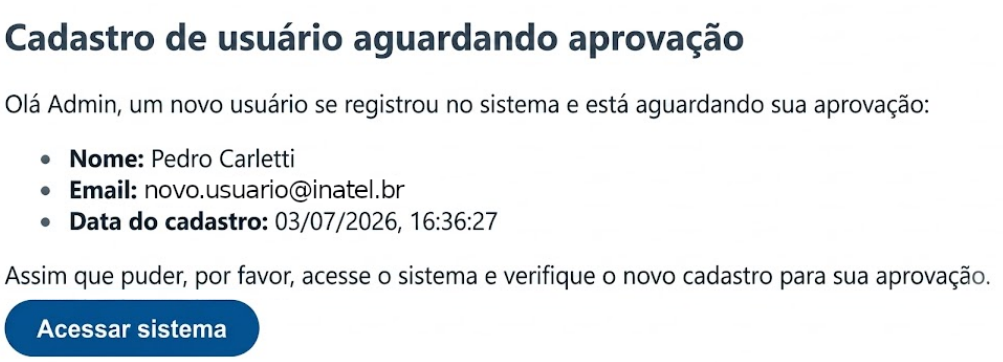



Figura 24: *E-mail* ao administrador, informando sobre o novo cadastro por analisar.

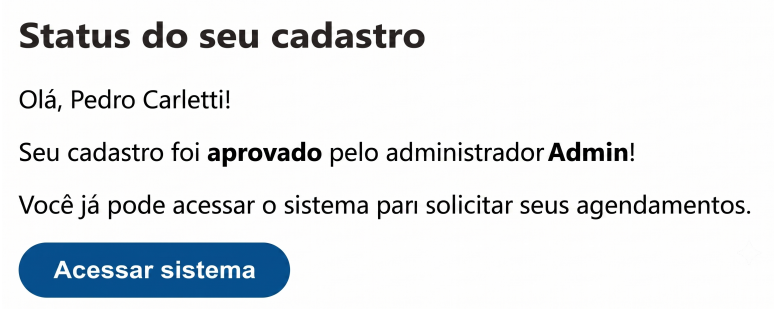



Figura 25: *E-mail* ao usuário, informando a aprovação de seu cadastro.

### 3.4.2 Aviso de nova solicitação ao administrador

O responsável administrativo recebe uma notificação sobre a nova solicitação pendente, conforme mostra a Figura 29. Com isso, ele pode avaliar o pedido e dar continuidade ao processo de aprovação.

### 3.4.3 Comunicação de aprovação do agendamento

Se o agendamento for aprovado, a confirmação é enviada ao usuário por *e-mail*, conforme mostra a Figura 30.

### 3.4.4 Comunicação de rejeição do agendamento

Se o agendamento for rejeitado, a aplicação também informa o usuário por *e-mail*, conforme mostra a Figura 31. Essa comunicação conclui o fluxo de análise e registra o resultado da solicitação.

O conjunto de notificações torna os processos de cadastro e agendamento mais transparentes e melhora a comunicação entre usuários e administradores. Dessa forma, a aplicação oferece maior previsibilidade sobre o andamento das solicitações e mais controle sobre as atividades realizadas na plataforma.

## 3.5 Melhorias de usabilidade e interface

As melhorias de usabilidade e interface foram implementadas para tornar a plataforma mais clara, acessível e adequada a diferentes perfis de usuários. Usabilidade significa a facilidade com

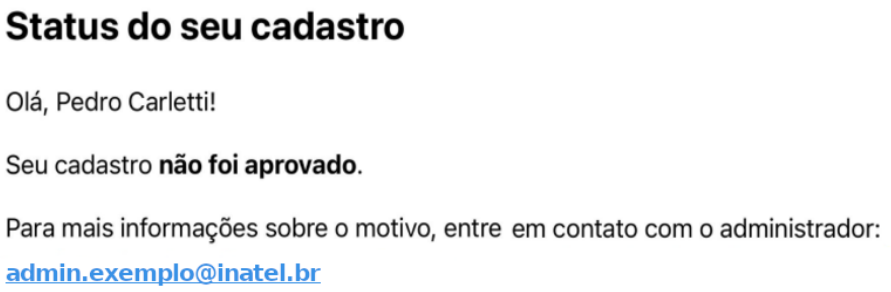



Figura 26: *E-mail* ao usuário, informando a rejeição de seu cadastro.

que uma pessoa consegue entender e utilizar o sistema e uma interface bem organizada reduz dúvidas, diminui erros de preenchimento e ajuda os usuários a executar corretamente suas atividades. Por isso, foram realizados ajustes nas mensagens de erro, nos elementos visuais e no suporte a múltiplos idiomas.

Com esse objetivo, foram aplicadas melhorias na apresentação de mensagens, no tratamento de exceções, na interação com campos sensíveis e na preparação da aplicação para suporte a múltiplos idiomas.

### 3.5.1 Aprimoramento das mensagens de erro e exceção

O tratamento de exceções em requisições *Hypertext Transfer Protocol* (HTTP) é necessário para que o usuário compreenda o que aconteceu quando uma operação falha. Requisições HTTP são as comunicações realizadas entre a interface da aplicação, como um navegador, e o servidor. Os códigos de *status* HTTP indicam o resultado de cada solicitação, distinguindo, por exemplo, operações bem-sucedidas, erros de acesso e falhas internas [17]. Sem um tratamento adequado, o sistema pode exibir mensagens muito técnicas ou pouco claras, dificultando a compreensão do problema pelo usuário, que não saberá se o erro é um problema no sistema ou alguma ação incorreta por ele executada.

A aplicação passou a utilizar uma abordagem de tratamento de exceções baseada em códigos de erro personalizados, associados aos códigos de *status* HTTP e a mensagens mais compreensíveis ao usuário. Esse padrão melhora a comunicação entre as partes internas do sistema e permite apresentar respostas consistentes sem revelar detalhes técnicos desnecessários. Recomendações para *Application Programming Interfaces* (APIs) orientam que as respostas de erro sejam estruturadas e que informações internas sejam mantidas nos registros do servidor, em vez de serem exibidas diretamente ao usuário [18, 19].

Esse tratamento segue em evolução conforme novas funcionalidades são adicionadas e revisões são executadas. A Figura 32 apresenta um exemplo na interface de usuários, demonstrando a captura de uma exceção por um componente da página.

### 3.5.2 Recursos visuais da aplicação

Também foram adicionados ícones com a função de exibir e ocultar o conteúdo de campos sensíveis da interface, como no caso de senhas. O recurso permite que o usuário confira a informação digitada antes de salvar ou enviar o formulário, reduzindo possíveis erros, como o de digitação, que não poderia ser conferido anteriormente pelo usuário.

A Figura 33 ilustra esse recurso aplicado ao campo de senha do formulário de cadastro de equipamento.

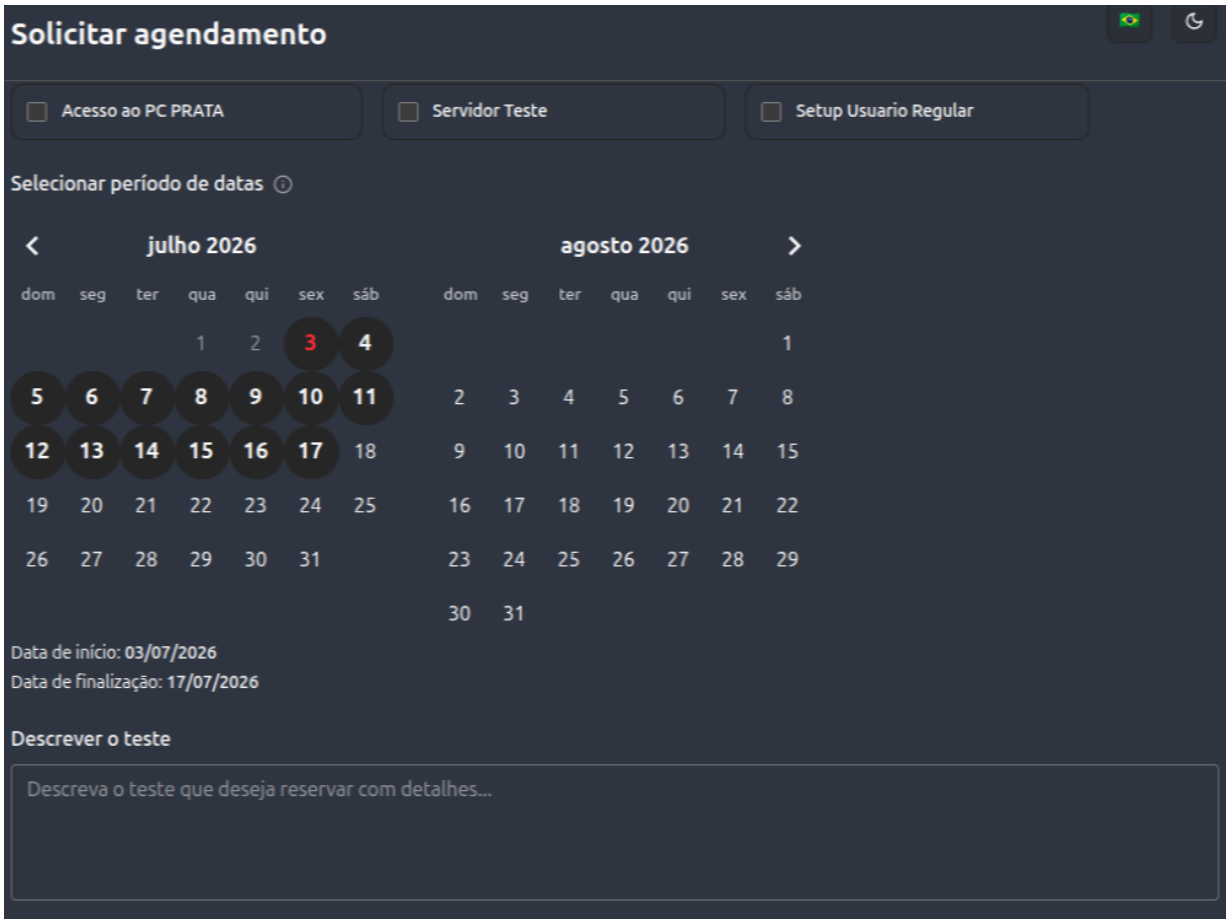



Figura 27: Interface de solicitação de agendamento de *setups*.

### 3.5.3 Suporte à internacionalização

O processo de internacionalização, também conhecido como *internationalization* (i18n), foi considerado um recurso importante para uma aplicação web que pode atender usuários de diferentes regiões e idiomas. Esse recurso consiste em preparar a aplicação para que seus textos, formatos e elementos de interface possam ser adaptados a diferentes idiomas e contextos culturais sem exigir alterações estruturais em seu funcionamento [20].

Além de ampliar o alcance da plataforma, esse recurso melhora a experiência de uso ao apresentar as informações em idioma compatível com o contexto do usuário. Com isso, reduzem-se interpretações incorretas, melhora-se a navegação e amplia-se a acessibilidade para diferentes públicos.

Do ponto de vista de desenvolvimento, a internacionalização favorece a manutenção do sistema, a padronização das mensagens e a escalabilidade da aplicação. Assim, novos idiomas podem ser incorporados organizadamente, sem comprometer a estrutura existente.

A aplicação utiliza uma biblioteca de internacionalização que permite manter os textos da interface separados da lógica do sistema. A cobertura desse recurso é ampliada conforme novas funcionalidades são revisadas ou implementadas, contribuindo para maior consistência da interface. A Figura 34 apresenta as três opções de idioma atualmente disponíveis, enquanto a Figura 35 exemplifica a interface exibida em diferentes idiomas.

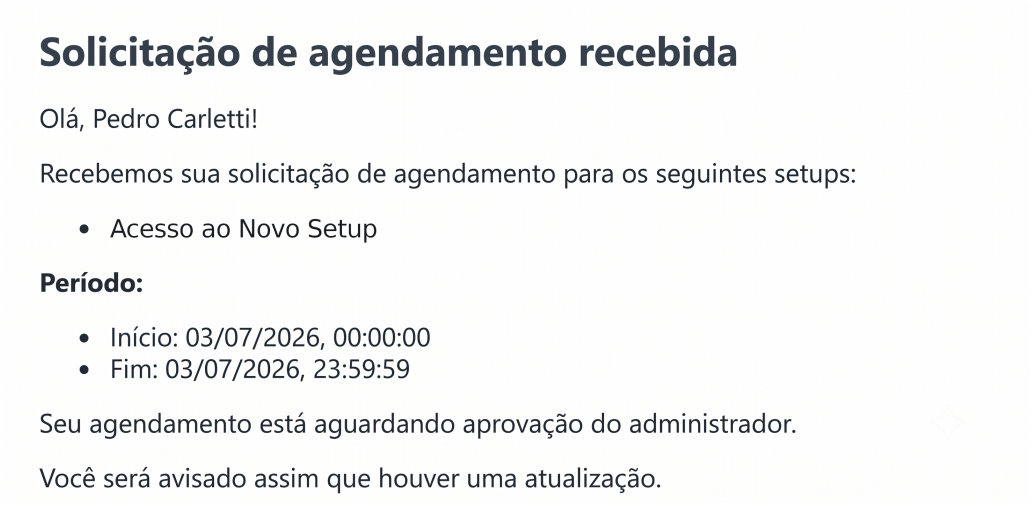



Figura 28: *E-mail* ao usuário, confirmando o recebimento da solicitação de agendamento.

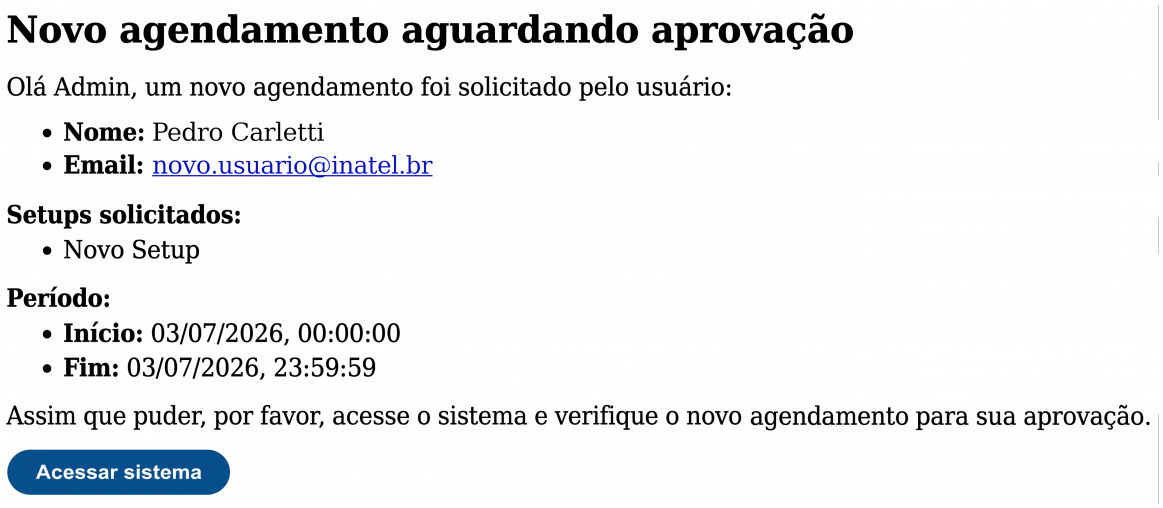



Figura 29: *E-mail* ao administrador, informando sobre a nova solicitação de agendamento.

## 3.6 Melhorias derivadas do *scan* de vulnerabilidades

No âmbito da melhoria contínua de segurança da plataforma desenvolvida, foram realizadas verificações automatizadas no código com ferramentas de *scan* de vulnerabilidades. Esse tipo de verificação funciona como uma varredura técnica da aplicação e da infraestrutura, procurando sinais de fragilidade que possam afetar a segurança ou a estabilidade. Guias de avaliação de segurança classificam a varredura de vulnerabilidades como uma técnica utilizada para identificar sistemas, serviços e possíveis falhas que precisam de análise posterior [21]. A etapa teve como finalidade identificar configurações inadequadas, dependências vulneráveis, exposição desnecessária de informações e outros pontos de atenção.

Esse tipo de verificação complementa revisões manuais e testes funcionais, pois avalia a aplicação e sua infraestrutura de maneira sistemática. Em outras palavras, a ferramenta ajuda a encontrar problemas que poderiam passar despercebidos em uma análise manual. A execução recorrente desses *scans* ajuda a antecipar falhas e orientar correções antes que elas cheguem ao ambiente operacional.

Foram avaliados, entre outros aspectos, a exposição de cabeçalhos HTTP, a configuração dos

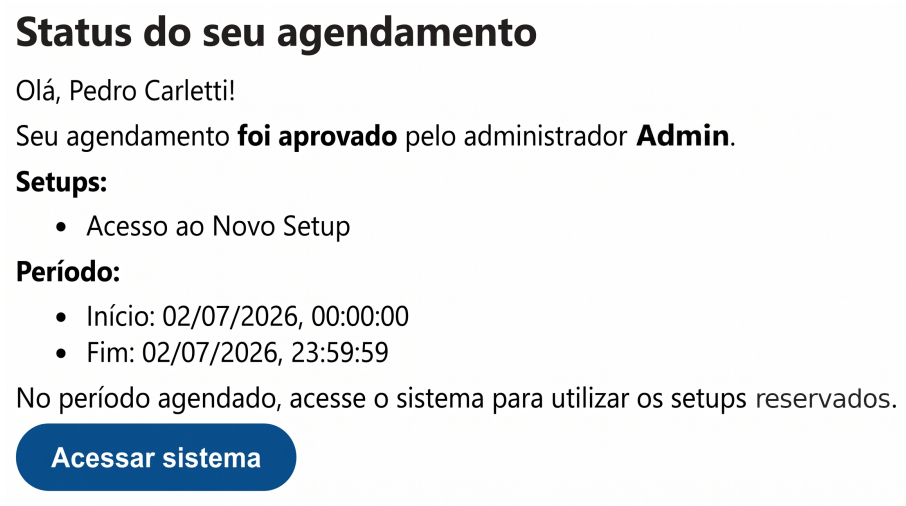



Figura 30: *E-mail* ao usuário, informando a aprovação do agendamento.

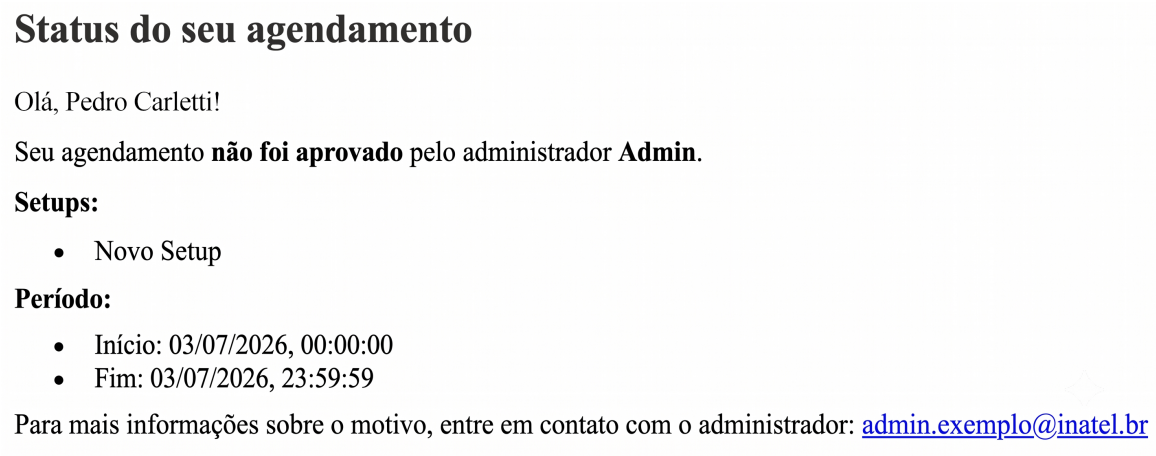



Figura 31: *E-mail* enviado ao usuário informando a rejeição do agendamento.

serviços web, a presença de informações sensíveis nas respostas, o comportamento de *endpoints* públicos e a adequação dos mecanismos de autenticação e autorização. *Endpoints* públicos são rotas ou endereços da aplicação que podem ser acessados externamente. As ocorrências identificadas foram analisadas conforme o impacto e priorizadas de acordo com o risco associado.

Com base nos resultados, foram executados ajustes de configuração, revisão das respostas da aplicação, remoção de informações desnecessárias e reforço dos controles de acesso. Essas ações reduziram a superfície de ataque, isto é, a quantidade de pontos que poderiam ser explorados por uma tentativa indevida de acesso, além de melhorar a proteção das rotas públicas e aumentar a maturidade de segurança da plataforma.

## 3.7 Melhorias orientadas pela análise do SonarQube

A análise realizada com o apoio do SonarQube teve papel importante no aprimoramento da qualidade do código-fonte. O SonarQube realiza análise automatizada e apresenta indicadores relacionados, entre outros aspectos, à confiabilidade, à segurança, à manutenibilidade, à duplicação e à cobertura de testes [22]. Com esse acompanhamento, a equipe consegue identificar pontos de melhoria, corrigir problemas recorrentes e seguir boas práticas de desenvolvimento


Figura 32: Exemplo de captura de erro ao listar usuários.


Figura 33: Campo de senha com a funcionalidade de exibir ou ocultar seu conteúdo.

de forma mais organizada. Assim, as melhorias orientadas pelo SonarQube contribuem para uma plataforma mais estável e mais fácil de evoluir em etapas futuras.

A ferramenta também disponibiliza métricas de manutenibilidade, confiabilidade, segurança e cobertura de testes. Manutenibilidade está relacionada ao esforço necessário para compreender, corrigir e evoluir o código. Confiabilidade considera problemas que podem causar comportamentos incorretos. A cobertura de testes indica a parcela do código que é exercitada durante a execução dos testes automatizados. Essas informações são exibidas em painéis visuais, facilitando o acompanhamento da evolução do projeto e a identificação de pontos que exigem correção ou melhoria [22].

A seguir, estão apresentados detalhes sobre a integração do projeto com a ferramenta SonarQube e os ajustes realizados com base nas métricas geradas.

### 3.7.1 Integração do projeto ao SonarQube

Para melhorar a qualidade do código e ampliar a aderência a boas práticas de desenvolvimento, o projeto foi integrado ao SonarQube disponibilizado internamente pelo Inatel. Com essa integração, o código passou a ser analisado continuamente, permitindo acompanhar indicadores de segurança, confiabilidade, manutenibilidade, duplicação de código e cobertura de testes automatizados. A duplicação de código indica trechos repetidos que podem dificultar futuras manutenções, pois uma mesma correção pode precisar ser feita em vários pontos.

As verificações apontaram ocorrências em diferentes níveis de severidade: baixa (*low*), média (*medium*) e alta (*high*). Esses registros, chamados de *issues*, indicam problemas ou pontos de atenção associados a falhas potenciais, vulnerabilidades ou práticas de implementação que poderiam dificultar a evolução do sistema. No processo de correção, as ocorrências de severidade alta (*high*) foram priorizadas por apresentarem maior potencial de impacto na segurança, estabilidade ou confiabilidade da aplicação. O tratamento desses pontos contribuiu para reduzir

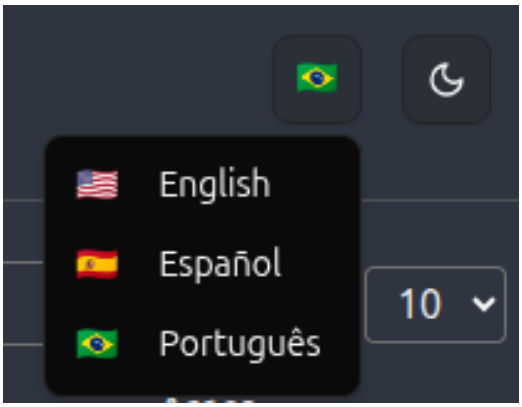



Figura 34: Tela com menu de internacionalização.

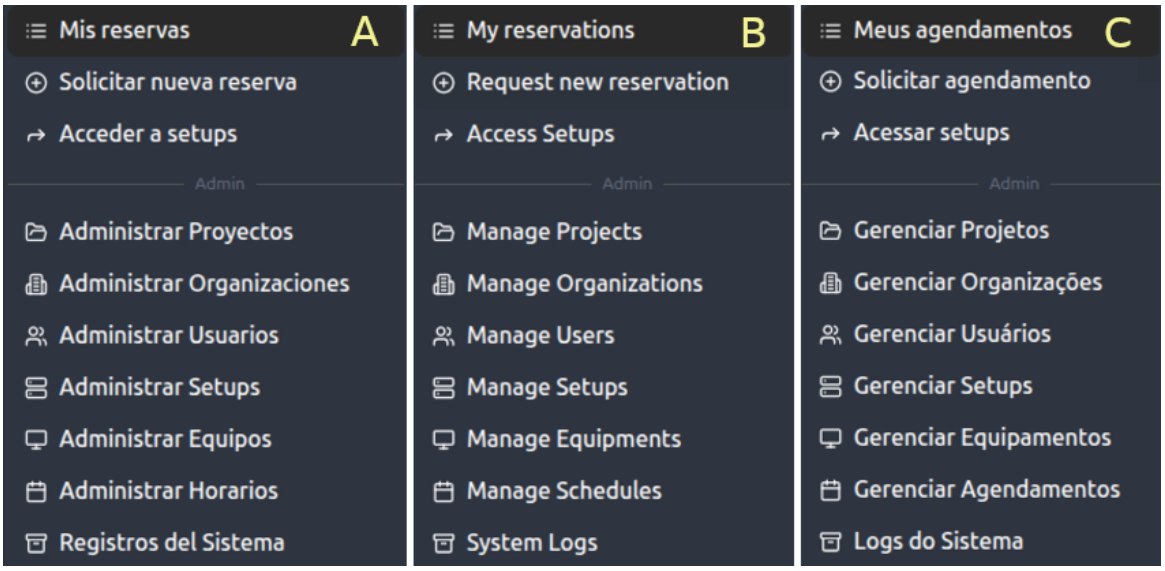



Figura 35: Menu nos três idiomas suportados: (A) Espanhol, (B) Inglês e (C) Português.

riscos técnicos e melhorar a qualidade geral do código.

Como resultado desse acompanhamento contínuo, as métricas do projeto apresentaram evolução positiva, com redução da quantidade de problemas identificados e aumento da cobertura de testes. As figuras a seguir mostram a visão geral do projeto no SonarQube e a evolução dos indicadores monitorados.

### 3.7.2 Panorama do projeto no SonarQube

A Figura 36 mostra o panorama geral do projeto no SonarQube, com destaque para o status no *Quality Gate*, a cobertura de testes, a duplicação de código e outros indicadores de qualidade. O *Quality Gate* reúne condições configuradas para determinar se o código analisado atende aos critérios de qualidade definidos para o projeto [23].

O *Quality Gate* pode ser entendido como uma inspeção de qualidade semelhante a uma vistoria: existe uma lista de requisitos mínimos e, se algum deles não for atendido, o resultado é reprovado até que o problema seja corrigido. A cobertura de testes, por sua vez, indica qual parcela do código é exercitada por verificações automáticas. Uma cobertura maior amplia a parte do sistema submetida aos testes, embora esse indicador, isoladamente, não garanta a

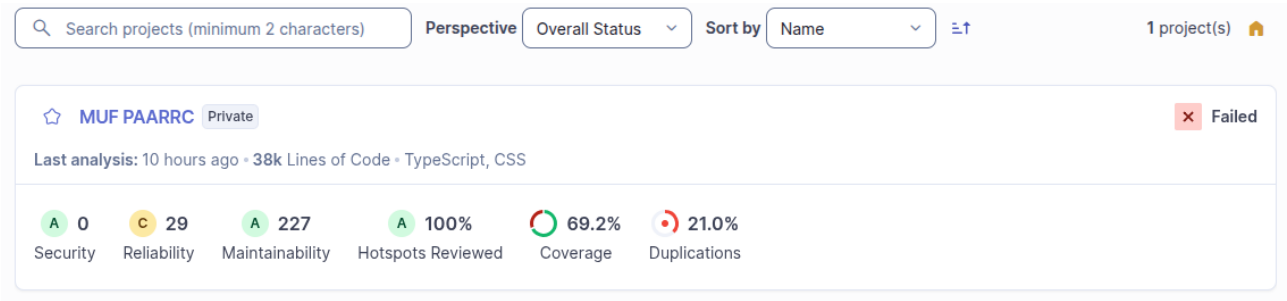



Figura 36: Visão geral do projeto no SonarQube.

ausência de falhas [22, 23].

A Figura 37 detalha as métricas do projeto, incluindo indicadores de segurança, confiabilidade, manutenibilidade, cobertura de testes e duplicação. Esses dados permitem avaliar com mais precisão o estado da aplicação.

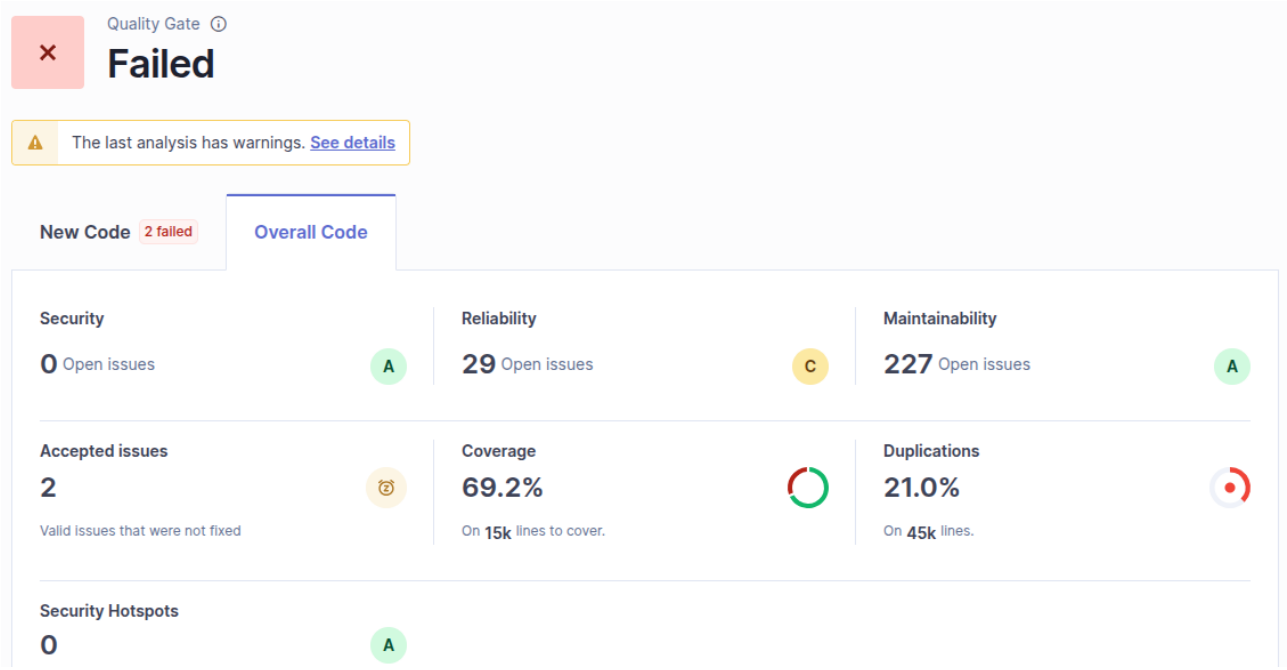



Figura 37: Visão detalhada das métricas de qualidade do projeto no SonarQube.

A Figura 38 apresenta a evolução da quantidade de *issues* ao longo do tempo. Neste contexto, *issues* são problemas, alertas ou pontos de melhoria identificados automaticamente pela ferramenta. A queda gradual dessas ocorrências demonstra o esforço de correção e melhoria contínua, principalmente em relação aos problemas de maior severidade.

A cobertura de testes também apresentou avanço, conforme evidenciado na Figura 39. Isso significa que uma parcela maior do código passou a ser verificada por testes automatizados. Quanto maior essa cobertura, maior a chance de identificar falhas antes que uma alteração chegue ao ambiente de uso.

Essa melhoria está relacionada à criação e expansão da suíte de testes, que passou a oferecer uma verificação mais abrangente da aplicação. Como consequência, houve aumento da confiabilidade do sistema e maior controle sobre impactos gerados por manutenções e novas implementações.


Figura 38: Progresso com as *issues* ao longo do tempo, conforme análise do SonarQube.

A utilização do SonarQube foi relevante para identificar e corrigir problemas no código, apoiando diretamente a melhoria contínua da qualidade do software. A análise recorrente e o tratamento das *issues*, sobretudo as de maior severidade, ajudaram a reduzir riscos e tornar a aplicação mais robusta e sustentável.

## 3.8 Melhorias resultantes do pen-test

As melhorias resultantes do pen-test representam uma etapa importante no fortalecimento da segurança da plataforma. O pen-test, ou teste de invasão controlado, verifica em que medida um sistema resiste a tentativas ativas de comprometimento, dentro de limites previamente autorizados [21]. Diferentemente da análise automática do código, esse teste observa como a aplicação se comporta diante de técnicas de exploração aplicadas aos seus fluxos reais de uso.

A aplicação foi submetida a um teste de invasão (pen-test) conduzido por uma equipe especializada em segurança ofensiva. Esse tipo de avaliação simula cenários reais de ataque, de forma controlada e autorizada, para revelar vulnerabilidades que podem passar despercebidas durante o desenvolvimento ou em testes funcionais convencionais. A seguir estão apresentados os principais pontos otimizados a partir dos resultados iniciais do pen-test. Vale destacar que, nesta fase do projeto, o pen-test envolveu o portal de agendamento, /acAPI e banco de dados. O portal de acesso remoto e o *gateway* serão alvos de análises nas próximas atividades a serem executadas.

### 3.8.1 Escopo do pen-test

O teste de invasão avalia a aplicação a partir da perspectiva de um atacante, utilizando técnicas e ferramentas semelhantes às empregadas em tentativas reais de exploração, mas dentro de um escopo autorizado e com regras de execução previamente estabelecidas. A análise busca identificar falhas técnicas e examinar a postura geral de segurança do sistema, incluindo

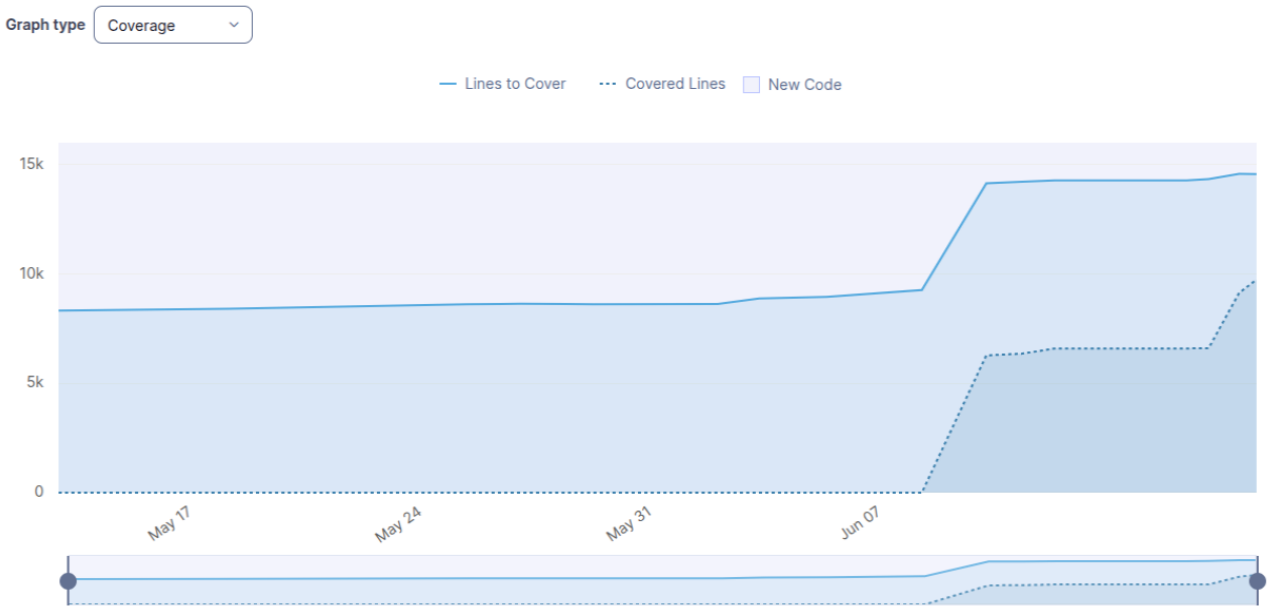



Figura 39: Evolução da cobertura de testes, evidenciando significativa melhoria.

autenticação, autorização, configuração de servidores, tratamento de erros e exposição de informações sensíveis [21].

A avaliação contemplou diferentes categorias de segurança, entre elas:

- Autenticação e autorização: testes para verificar se seria possível contornar o *login*, realizar muitas tentativas de senha, identificar usuários existentes ou acessar recursos sem permissão;
- Exposição de informações: análise de cabeçalhos HTTP, mensagens de erro e respostas da aplicação que pudessem revelar detalhes internos do sistema a pessoas não autorizadas;
- Configuração de servidores: verificação de versões expostas, configurações inadequadas e informações de infraestrutura que não deveriam ficar acessíveis externamente;
- Proteção de dados: avaliação do acesso a dados sensíveis por diferentes perfis de usuário e verificação de possíveis vazamentos entre níveis de permissão.

### 3.8.2 Relevância das correções aplicadas

As vulnerabilidades apontadas no pen-test não devem ser analisadas apenas como ocorrências isoladas. Mesmo quando uma fragilidade não indica risco crítico imediato, ela pode ser combinada com outras técnicas e aumentar o impacto de um ataque. Por isso, a correção preventiva desses pontos é importante para reduzir a superfície de ataque e fortalecer a segurança da aplicação.

As correções aplicadas também se alinham a orientações publicadas pela *Open Web Application Security Project* (OWASP), que reúne práticas para autenticação, controle de acesso, tratamento de erros, proteção de APIs e testes de aplicações web [16, 19, 24]. Além disso, as melhorias contribuem para requisitos frequentemente observados em auditorias, revisões técnicas e processos de conformidade.

### 3.8.3 Principais ajustes de segurança implementados

A partir dos resultados do pen-test, os seguintes ajustes foram incorporados à aplicação, sendo que a maioria das implementações de melhorias estão detalhadas nas seções anteriores:

- Reforço no fluxo de autenticação multifator para eliminar possíveis formas de contornar etapas do *login*;
- Fortalecimento da geração de códigos OTP com uso de letras e números;
- Implementação de política de senhas com tamanho mínimo e máximo definido;
- Ajuste nas configurações de *rate limiting*, que controlam a quantidade de requisições permitidas em determinado período, e remoção de cabeçalhos informativos das respostas HTTP;
- Padronização de mensagens de erro para reduzir a possibilidade de identificar informações internas, como a existência de usuários;
- Restrição da exposição de dados sensíveis de equipamentos para usuários sem permissões administrativas;
- Ocultação de informações de versão do servidor web `nginx`.

As subseções seguintes descrevem cada melhoria, apresentando o contexto identificado, a solução aplicada e os benefícios obtidos com as alterações.

### 3.8.4 Fortalecimento do fluxo de autenticação multifator

Durante a revisão do fluxo de autenticação multifator, identificou-se que o código da OTP podia ser gerado antes da validação completa do *e-mail* e da senha. Esse comportamento poderia permitir que uma pessoa ou sistema externo solicitasse vários códigos sem comprovar previamente que possuía credenciais válidas.

Como correção, o fluxo foi alterado para que a geração e o envio da OTP ocorram apenas depois da validação bem-sucedida do *e-mail* e da senha. Assim, as etapas de autenticação são executadas de forma sequencial, evitando que a emissão do código seja acionada sem a validação prévia das credenciais. Testes de MFA devem verificar justamente se etapas do fluxo podem ser ignoradas, reutilizadas ou acessadas diretamente [25].

### 3.8.5 Aprimoramento da geração dos códigos OTP

Na versão anterior, as OTPs eram compostos apenas por números. Essa abordagem oferece menos combinações possíveis e pode facilitar tentativas de adivinhação, principalmente quando várias tentativas são feitas dentro do período de validade do código.

Para tornar esse mecanismo mais robusto, a geração das OTPs passou a utilizar combinações alfanuméricas, incluindo letras e números. Mantido o mesmo comprimento, a ampliação do conjunto de caracteres aumenta o número de combinações possíveis. A proteção do fluxo também depende de validade curta, uso único e limitação das tentativas de verificação [8, 14].

### 3.8.6 Revisão da política de validação de senhas

As validações anteriores de senha não eram suficientemente rígidas em relação ao tamanho das credenciais definidas pelos usuários. Isso poderia permitir senhas frágeis, mais expostas a tentativas de adivinhação, ataques de força bruta ou reutilização indevida.

Como melhoria, o projeto definiu uma regra de comprimento mínimo de 12 caracteres e máximo de 64 caracteres para as senhas. O valor mínimo adotado é uma decisão de segurança da plataforma. Diretrizes atuais destacam o comprimento como fator relevante, recomendam a limitação das tentativas de autenticação e orientam que os sistemas aceitem senhas com comprimento máximo de pelo menos 64 caracteres [8].

### 3.8.7 Revisão dos mecanismos de rate limiting

A aplicação já contava com limites de requisições por minuto, o *rate limiting*, em seus *endpoints*. Esse controle restringe a quantidade de solicitações que um usuário, endereço ou sistema pode realizar em determinado período, sendo utilizado para reduzir abusos, consumo excessivo de recursos e tentativas automatizadas [16, 24]. Contudo, determinados cabeçalhos das respostas HTTP revelavam informações sobre os limites configurados, permitindo que qualquer cliente identificasse a política de *throttling* aplicada.

Em termos práticos, o *rate limiting* funciona como uma catraca que limita quantas vezes alguém pode passar por minuto. O problema identificado é que a aplicação, além de aplicar o limite, expunha esse limite nas suas respostas: era como se a catraca tivesse um aviso dizendo exatamente quantas passagens ainda restavam antes do bloqueio. Para um usuário comum, essa informação é irrelevante, mas para um atacante, ela é um manual de instruções sobre até onde ele pode insistir sem ser barrado.

Mesmo parecendo uma informação de baixo impacto, esse dado pode ajudar um atacante a estimar quantas requisições podem ser feitas antes do bloqueio. Com isso, seria mais fácil planejar ataques de força bruta ou abuso de rotas públicas da aplicação.

Após os ajustes necessários, a aplicação preserva os mecanismos de proteção contra abuso sem divulgar detalhes internos da política de limitação de requisições.

### 3.8.8 Uniformização das respostas de erro

Anteriormente, algumas respostas de erro apresentavam mensagens detalhadas sobre falhas de autenticação e autorização. Embora esse nível de detalhe seja útil durante o desenvolvimento, ele pode ser explorado em técnicas de enumeração. Nesse contexto, enumeração é a tentativa de descobrir informações internas do sistema, como diferenciar um usuário existente de um inexistente ou identificar se um token está inválido ou expirado.

Como correção, as mensagens passaram a retornar respostas mais genéricas, como “Acesso negado” ou “Requisição inválida”, sem detalhes técnicos que auxiliassem no reconhecimento do ambiente. Em fluxos de autenticação, respostas uniformes ajudam a evitar que diferenças de mensagem revelem a existência de usuários ou detalhes internos da falha [16, 19].

### 3.8.9 Redução da exposição de informações do servidor web

As páginas de erro do servidor web, principalmente respostas do tipo `404 Not Found`, exibiam anteriormente a versão de ferramentas do servidor em uso. A identificação do tipo e da versão de um servidor faz parte das técnicas de reconhecimento utilizadas para buscar vulnerabilidades conhecidas e comportamentos específicos da infraestrutura [26].

A configuração do servidor foi ajustada para ocultar versões de ferramentas nas páginas de erro e nos cabeçalhos em que essa informação era apresentada. O próprio servidor disponibiliza a diretiva `server_tokens` para controlar a exibição da versão nas respostas [27]. Essa alteração não substitui a atualização e a correção do servidor, mas reduz a exposição direta de detalhes utilizados durante o reconhecimento, o que aumenta a dificuldade de um ataque direcionado.

## 3.9 Síntese das melhorias

A Tabela 1 resume as melhorias apresentadas nesta seção e o benefício prático de cada uma para quem utiliza a plataforma.

Tabela 1: Síntese das melhorias e seus benefícios práticos.

| **Melhoria** | **O que muda para o usuário** |
|---|---|
| Cadastro de projetos com vínculo de usuários e *setups* | O usuário regular passa a ver apenas os equipamentos que pode usar e consegue solicitar agendamentos e acessá-los remotamente. |
| Autenticação em duas etapas (MFA) | Mesmo que a senha seja descoberta, a conta continua protegida pelo código enviado ao *e-mail*. |
| Recuperação de senha | O próprio usuário redefine sua senha, sem depender de um administrador. |
| Bloqueio após tentativas inválidas | Tentativas automatizadas de adivinhar a senha são interrompidas, e o usuário é avisado por *e-mail*. |
| Notificações automáticas por *e-mail* | O usuário acompanha cada etapa do cadastro e do agendamento sem precisar perguntar a alguém. |
| Melhorias de interface e idiomas | Mensagens de erro mais claras, campos de senha conferíveis e telas em mais de um idioma. |
| *Scan*, SonarQube e pen-test | Problemas de segurança e de qualidade são encontrados e corrigidos antes de afetar quem usa o sistema. |

# 4 Conclusão

As atividades executadas de desenvolvimento da Plataforma Multiusuário Brasil 6G propiciaram um avanço significativo no amadurecimento técnico da solução. O desenvolvimento teve como foco a evolução da solução por meio da implementação de melhorias funcionais e otimizações de segurança orientadas por ferramentas de testes automatizadas e pen-test.

Entre os principais resultados alcançados, destacam-se os aprimoramentos no fluxo de uso do perfil regular, a revisão dos mecanismos de acesso e autenticação, a atualização do processo de *login*, a criação do recurso de recuperação de senha, a proteção contra tentativas inválidas sucessivas e o funcionamento do bloqueio temporário. Esses avanços contribuem diretamente para o aumento da confiabilidade da plataforma, tornando o acesso mais seguro e reduzindo riscos associados ao uso indevido de credenciais ou a tentativas de acesso não autorizadas.

Também foram implementadas melhorias relevantes nos mecanismos de comunicação da plataforma, com a implantação de notificações automáticas por *e-mail* associadas ao cadastro de usuários, confirmação de recebimento de solicitações, avisos aos administradores, aprovação e rejeição de cadastros e atualização do estado dos agendamentos. Essa funcionalidade fortalece a rastreabilidade dos processos, melhora a comunicação entre usuários e administradores e reduz a dependência de interações manuais externas ao sistema.

No fluxo de solicitação de agendamento, os aprimoramentos realizados permitiram tornar o processo mais claro, organizado e previsível. A confirmação de recebimento da solicitação ao usuário, o aviso de novas solicitações ao administrador e as comunicações de aprovação ou rejeição do agendamento contribuem para uma melhor governança no uso compartilhado dos recursos experimentais. Esses ajustes são especialmente importantes em uma plataforma multiusuário, na qual a transparência e o controle das solicitações são fundamentais para a operação segura e eficiente da infraestrutura.

Além das melhorias funcionais, esta etapa contemplou ajustes de usabilidade e interface, incluindo o aprimoramento de mensagens de erro e exceção, correções visuais e melhorias no suporte à internacionalização. Essas alterações tornam a plataforma mais acessível, compreensível e adequada ao uso por diferentes perfis de usuários, incluindo pesquisadores, administradores, instituições parceiras e potenciais usuários externos.

Outro resultado importante desta etapa foi a incorporação de práticas de análise de qualidade e segurança no processo de desenvolvimento. A integração e utilização do SonarQube permitiram identificar pontos de melhoria no código-fonte, orientar correções e estabelecer uma visão mais estruturada sobre a qualidade da aplicação. De forma complementar, os ajustes resultantes do pen-test contribuíram para o fortalecimento da segurança da plataforma, incluindo melhorias relacionadas à autenticação multifator, geração de códigos OTP, política de validação de senhas, mecanismos de *rate limiting*, padronização de respostas de erro, limitação da exposição de dados de equipamentos e redução da exposição de informações do servidor web.

Com as melhorias implementadas nesta fase do projeto, a Plataforma Multiusuário passa a apresentar maior robustez operacional, melhor experiência de uso e mecanismos de segurança mais consistentes. Ainda assim, por se tratar de uma solução em desenvolvimento contínuo, a plataforma deve permanecer em processo de validação, testes e aprimoramento, especialmente considerando sua futura integração com mecanismos de acesso remoto, recursos laboratoriais reais e ambientes experimentais distribuídos.

Dessa forma, os resultados apresentados demonstram a consolidação progressiva da Plataforma Multiusuário Brasil 6G ou PAARRC como uma plataforma mais segura, organizada e preparada para apoiar o compartilhamento controlado de recursos experimentais. As evoluções

implementadas fortalecem a base tecnológica necessária para o uso remoto da infraestrutura laboratorial e contribuem diretamente para os objetivos do Projeto Brasil 6G, especialmente no que se refere à democratização do acesso a ambientes de experimentação, à colaboração interinstitucional e ao uso eficiente de recursos avançados de pesquisa e desenvolvimento.

# Referências


[1] Centro de Referência em Radiocomunicações (CRR) Inatel, "Atividade 4.1 - Integração dos componentes da plataforma - Parte 1," *Projeto Brasil 6G*, Jul. 2025.

[2] National Institute of Standards and Technology, "Virtual private network (vpn)," 2026, [Acesso em: 20 jul. 2026]. [Online]. Available: https://csrc.nist.gov/glossary/term/virtual_private_network

[3] Apache Software Foundation, *Apache Guacamole Manual, Version 1.6.0*, 2025, [Acesso em: 20 jul. 2026]. [Online]. Available: https://guacamole.apache.org/doc/gug/

[4] T. Ylonen and C. Lonvick, "The secure shell (ssh) protocol architecture," Internet Engineering Task Force, Tech. Rep. RFC 4251, 01 2006, [Acesso em: 20 jul. 2026]. [Online]. Available: https://www.rfc-editor.org/info/rfc4251

[5] Microsoft Corporation, *[MS-RDPBCGR] Remote Desktop Protocol: Basic Connectivity and Graphics Remoting*, 2026, [Acesso em: 20 jul. 2026]. [Online]. Available: https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-rdpbcgr/5073f4ed-1e93-45e1-b039-6e30c385867c

[6] T. Richardson and J. Levine, "The remote framebuffer protocol," Internet Engineering Task Force, Tech. Rep. RFC 6143, 03 2011, [Acesso em: 20 jul. 2026]. [Online]. Available: https://www.rfc-editor.org/info/rfc6143

[7] National Institute of Standards and Technology, "Role based access control," 2016, [Acesso em: 20 jul. 2026]. [Online]. Available: https://csrc.nist.gov/projects/role-based-access-control

[8] D. Temoshok, J. Fenton, Y.-Y. Choong, N. Lefkovitz, A. Regenscheid, R. Galluzzo, and J. Richer, "Digital identity guidelines: Authentication and authenticator management," National Institute of Standards and Technology, Tech. Rep. NIST Special Publication 800-63B-4, 07 2025, [Acesso em: 20 jul. 2026]. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-63b-4

[9] W. Newhouse, B. Johnson, S. Kinling, J. Kuruvilla, B. Mulugeta, and K. Sandlin, "Multifactor authentication for e-commerce: Risk-based, fido universal second factor implementations for purchasers," National Institute of Standards and Technology, Tech. Rep. NIST Special Publication 1800-17, 07 2019, [Acesso em: 20 jul. 2026]. [Online]. Available: https://doi.org/10.6028/NIST.SP.1800-17

[10] M. B. Jones, J. Bradley, and N. Sakimura, "Json web token (jwt)," Internet Engineering Task Force, Tech. Rep. RFC 7519, 05 2015, [Acesso em: 20 jul. 2026]. [Online]. Available: https://www.rfc-editor.org/info/rfc7519

[11] Y. Sheffer, D. Hardt, and M. B. Jones, "Json web token best current practices," Internet Engineering Task Force, Tech. Rep. RFC 8725; BCP 225, 02 2020, [Acesso em: 20 jul. 2026]. [Online]. Available: https://www.rfc-editor.org/info/rfc8725

[12] PostgreSQL Global Development Group, *PostgreSQL 18 Documentation*, 2026, [Acesso em: 20 jul. 2026]. [Online]. Available: https://www.postgresql.org/docs/current/

[13] Docker, Inc., *Docker Compose*, 2026, [Acesso em: 20 jul. 2026]. [Online]. Available: https://docs.docker.com/compose/

[14] OWASP Foundation. (2026) Multifactor authentication cheat sheet. [Online]. Available: https://cheatsheetseries.owasp.org/cheatsheets/Multifactor_Authentication_Cheat_Sheet.html

[15] ——. (2026) Forgot password cheat sheet. [Online]. Available: https://cheatsheetseries.owasp.org/cheatsheets/Forgot_Password_Cheat_Sheet.html

[16] ——. (2026) Authentication cheat sheet. [Online]. Available: https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html

[17] R. T. Fielding, M. Nottingham, and J. Reschke, "Http semantics," Internet Engineering Task Force, Tech. Rep. RFC 9110, 06 2022, [Acesso em: 20 jul. 2026]. [Online]. Available: https://www.rfc-editor.org/info/rfc9110

[18] M. Nottingham, E. Wilde, and S. Dalal, "Problem details for http apis," Internet Engineering Task Force, Tech. Rep. RFC 9457, 07 2023, [Acesso em: 20 jul. 2026]. [Online]. Available: https://www.rfc-editor.org/info/rfc9457

[19] OWASP Foundation. (2026) Error handling cheat sheet. [Online]. Available: https://cheatsheetseries.owasp.org/cheatsheets/Error_Handling_Cheat_Sheet.html

[20] World Wide Web Consortium. (2005) Localization vs. internationalization. [Online]. Available: https://www.w3.org/International/questions/qa-i18n

[21] K. Scarfone, M. Souppaya, A. Cody, and A. Orebaugh, "Technical guide to information security testing and assessment," National Institute of Standards and Technology, Tech. Rep. NIST Special Publication 800-115, 09 2008, [Acesso em: 20 jul. 2026]. [Online]. Available: https://csrc.nist.gov/pubs/sp/800/115/final

[22] SonarSource S.A. (2026) Understanding measures and metrics. [Online]. Available: https://docs.sonarsource.com/sonarqube-server/user-guide/code-metrics/metrics-definition

[23] ——. (2026) Understanding quality gates. [Online]. Available: https://docs.sonarsource.com/sonarqube-server/quality-standards-administration/managing-quality-gates/introduction-to-quality-gates

[24] OWASP Foundation. (2023) Owasp api security top 10 – 2023. [Online]. Available: https://owasp.org/API-Security/editions/2023/en/0x11-t10/

[25] ——. (2026) Testing multi-factor authentication. [Online]. Available: https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/04-Authentication_Testing/11-Testing_Multi-Factor_Authentication

[26] ——. (2026) Fingerprint web server. [Online]. Available: https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/01-Information_Gathering/02-Fingerprint_Web_Server

[27] NGINX, Inc., *Module ngx_http_core_module*, 2026, [Acesso em: 20 jul. 2026]. [Online]. Available: https://nginx.org/en/docs/http/ngx_http_core_module.html