

# Solly: Plataforma de gestão de Doações e Conexão Solidária

Ana Caroline T. Trajano, Ingrid Prado dos Santos,  
Rafael Henrique Dalegrave Zottesso

<sup>1</sup>Instituto Federal do Paraná (IFPR)- Campus Paranavaí  
Rua José Felipe Tequinha,  
1400 – Jardim das Nações — Paranavaí– PR– Brasil

{20223018747, 20223025625}@estudantes.ifpr.edu.br

rafael.zottesso@ifpr.edu.br

**Abstract.** *This article presents Solly, a digital platform developed to bring donors and social institutions closer together in a simple and accessible way. More than just a technological tool, the project seeks to create a reliable and transparent space where solidarity can take place in a practical and inspiring manner. The system was built using Django, Python, and modern web technologies, offering features such as donor and institution registration, donation management, and the sharing of inspiring stories. Its main purpose is to make donations more efficient, transparent, and human, fostering a more connected and impactful solidarity ecosystem.*

**Resumo.** *Este artigo apresenta o Solly, uma plataforma digital desenvolvida para aproximar doadores e instituições sociais de forma simples e acessível. Mais do que uma ferramenta tecnológica, o projeto busca criar um espaço confiável e transparente, onde a solidariedade possa acontecer de maneira prática e inspiradora. O sistema foi construído com Django, Python e tecnologias web modernas, oferecendo funcionalidades como o cadastro de doadores e instituições, a gestão das doações e o compartilhamento de histórias inspiradoras. O propósito central é tornar a doação mais eficiente, transparente e humana, estimulando um ecossistema de solidariedade mais conectado e impactante.*

## 1. Introdução

A desigualdade social e a vulnerabilidade de comunidades continuam crescendo e, com elas, o número de instituições que se dedicam a ajudar quem mais precisa. Em meio a tantas iniciativas solidárias, surge uma preocupação comum: criar formas eficazes de conectar pessoas dispostas a ajudar com aquelas que realmente necessitam de apoio [Portulhak et al. 2017]. O problema é que, diante de tantos casos de golpes *online* e uso indevido de recursos, muitos doadores acabam se sentindo inseguros e desmotivados. Falta transparência, falta confiança, e isso acaba, muitas vezes, reduzindo a efetividade das ações sociais que poderiam fazer a diferença. Pesquisas recentes mostram que a desconfiança quanto ao uso dos recursos e a percepção de falta de transparência são fatores decisivos para que muitos brasileiros deixem de doar. Nesse cenário, as plataformas digitais de doação aparecem como soluções inovadoras, pois ampliam o alcance das campanhas, permitem arrecadações em nível nacional e internacional e oferecem maior clareza em todo o processo.

Diante desse contexto, foi desenvolvido o Solly, uma plataforma digital que busca ampliar a visibilidade das organizações sociais e superar a barreira comunicativa entre instituições e doadores. O sistema propõe uma experiência simples, acessível e rastreável, estimulando a confiança e a cidadania digital por meio do uso de tecnologias modernas, como Django, Python e *frameworks* de interface responsiva. Mais do que uma ferramenta de doação, o Solly constitui um ambiente de transparência, onde os usuários podem acompanhar o impacto de suas contribuições e fortalecer a cultura da solidariedade [Bliiv 2025; Instituto Mano Down 2023].

Este trabalho tem como objetivo descrever a arquitetura, o processo de desenvolvimento e as decisões de engenharia adotadas na construção da plataforma Solly, um sistema *web* voltado à conexão transparente entre doadores e instituições sociais. O estudo caracteriza-se como um relato técnico, com ênfase na documentação das práticas de engenharia de software e das soluções implementadas para garantir segurança, acessibilidade e transparência no fluxo de doações, em conformidade com os princípios de boas práticas de desenvolvimento e modelagem de sistemas propostos por [Sommerville 2019, Costa and Barbosa 2020].

## 2. Fundamentação teórica

Esta seção articula os conceitos teóricos que sustentam o problema de falta de transparência em doações, o objetivo de desenvolver uma plataforma rastreável e as escolhas metodológicas em Django. O desenvolvimento *web* é essencial para aplicações interativas, e *frameworks* como o Django adotam a arquitetura MTV (*Model-Template-View*), separando responsabilidades para facilitar manutenção e segurança, alinhando-se ao objetivo de transparência. Isso contrasta com problemas de confiança identificados em estudos sobre LGPD e antifraude, que destacam a importância de práticas de segurança e conformidade para fortalecer a confiança do usuário em ambientes digitais [Pereira and Martins 2023].

Um *framework* consiste em um conjunto de bibliotecas, padrões de código e componentes já prontos, permitindo que desenvolvedores foquem nas particularidades do projeto sem reescrever funcionalidades comuns. No contexto *web*, *frameworks* organizam o fluxo de dados entre o cliente (*browser*) e o servidor, fornecendo suporte para autenticação, tratamento de requisições, integração com banco de dados e rotinas de segurança. Assim, o uso de *frameworks* garante não só agilidade, mas também robustez e escalabilidade às aplicações [Costa and Barbosa 2020].

Python e Django: o Python é uma linguagem popular devido à sua sintaxe simples e ecossistema. O Django adota MTV, com administração automática e segurança, ideal para aplicações como o Solly, sustentando o objetivo de rastreabilidade [Alura 2023].

Bootstrap e jQuery: O Bootstrap facilita interfaces responsivas, e o jQuery simplifica interações, articulando-se com a metodologia de usabilidade [Cardoso and Bispo 2019].

Engenharia de Software Aplicada: Princípios como UML e testes automatizados garantem qualidade, alinhando-se às métricas propostas para validação [de Pieri Bernhardt 2023, Sommerville 2019].

### 3. Trabalhos Correlatos

Nos últimos anos, surgiram diversas iniciativas para facilitar e legitimar doações online, conectando doadores e organizações do terceiro setor. Entre as principais plataformas brasileiras, destacam-se Doare, Link Nacional, Benfeitoria, Catarse e Solidarize, que oferecem soluções robustas para captação de recursos, campanhas solidárias e integração com múltiplos métodos de pagamento digital [Gorgulho et al. 2021]. Em geral, esses sistemas contam com funcionalidades como cadastro de organizações, histórico de contribuições, painéis de gestão e emissão de relatórios, além do recebimento de doações recorrentes ou avulsas em diversas modalidades [Cardoso and Bispo 2019].

Tecnologicamente, muitas dessas plataformas são desenvolvidas com base em WordPress com *plugins* especializados, *frameworks* PHP (como *GiveWP* e *Charitable*) ou sistemas customizados. Estudos recentes, como os de [Jiang et al. 2021] e [Mane 2022], evidenciam a adoção crescente de *frameworks* em Python, como o Django, destacando-se por sua segurança, flexibilidade, manutenção facilitada, arquitetura clara e área administrativa embutida [Portulhak et al. 2017].

Apesar do sucesso, revisões práticas apontam lacunas significativas: muitas plataformas focam apenas em doações financeiras, ignorando o gerenciamento de doações de itens, agendamento de coletas, auto-gestão do histórico de doações e a ausência de taxas sobre os valores recebidos [Portulhak et al. 2017]. Outros pontos frágeis envolvem transparência e comunicação de resultados – raramente o uso dos recursos doados é apresentado de forma clara aos apoiadores [Gorgulho et al. 2021].

Para superar essas limitações, o Solly foi projetado como uma plataforma completa, integrando doações financeiras e de itens, agendamento, painel de transparência, depoimentos e autonomia para perfis de doadores e instituições, tudo sem cobrança de taxas [Portulhak et al. 2017]. Ao optar por Django, Python e *frameworks* modernos de interface, o projeto alia robustez técnica, flexibilidade e alto impacto social, destacando-se na engenharia de software e no compromisso com a transformação social baseada na confiança [Costa and Barbosa 2020].

O Solly diferencia-se das plataformas analisadas por integrar doações financeiras e de itens, permitir o agendamento de coletas e gerar comprovantes digitais verificáveis. Além disso, adota mecanismos de transparência e verificação institucional que fortalecem a confiança dos doadores, superando as lacunas identificadas nas soluções existentes.

### 4. Metodologia de Desenvolvimento e Arquitetura

O presente trabalho caracteriza-se como um relato técnico de engenharia de software, cuja metodologia adota uma abordagem incremental e iterativa voltada ao desenvolvimento e documentação da plataforma Solly. O foco está nas decisões arquiteturais, na modelagem de componentes e nas práticas aplicadas para garantir segurança, escalabilidade e acessibilidade no sistema, conforme as boas práticas descritas por [Sommerville 2019, Costa and Barbosa 2020].

O processo de desenvolvimento foi estruturado em ciclos sucessivos de construção, revisão e aprimoramento, permitindo incorporar melhorias contínuas com base em testes funcionais e validações internas. A arquitetura adotada segue integralmente o padrão MTV (*Model–Template–View*), proposto pelo *framework* Django, que

promove a separação de responsabilidades entre as camadas de dados, controle e interface. Essa estrutura assegura modularidade, reuso e manutenção simplificada, fatores essenciais para a evolução do sistema [Foundation 2024a].

Na camada de *back-end*, utilizou-se Python e Django, responsáveis pela lógica de controle, autenticação e gerenciamento de dados. Na camada de *front-end*, adotou-se o Bootstrap, que fornece uma base sólida para o desenvolvimento de interfaces responsivas, com componentes visuais reutilizáveis e compatíveis com diversos dispositivos. Complementarmente, HTML5, CSS3 e JavaScript/jQuery foram empregados para aprimorar a estrutura, o estilo e a interatividade das páginas [Brasil 2023a, Network 2024].

O banco de dados foi implementado em PostgreSQL, escolhido por sua confiabilidade, integridade transacional e compatibilidade com o ORM do Django, o que simplifica a manipulação e o mapeamento dos dados. O uso combinado dessas ferramentas consolida uma base tecnológica robusta e alinhada às boas práticas de engenharia de software e segurança da informação [Brasil 2023b, Foundation 2024b].

Além dos aspectos técnicos, a metodologia considerou princípios de engenharia segura, incorporando autenticação por perfis, armazenamento criptografado de senhas e controle de acesso baseado em permissões. Essa abordagem busca assegurar conformidade com a LGPD e promover transparência no fluxo de dados, reforçando a confiança entre os usuários da plataforma [Brasil 2018, Foundation 2024b].

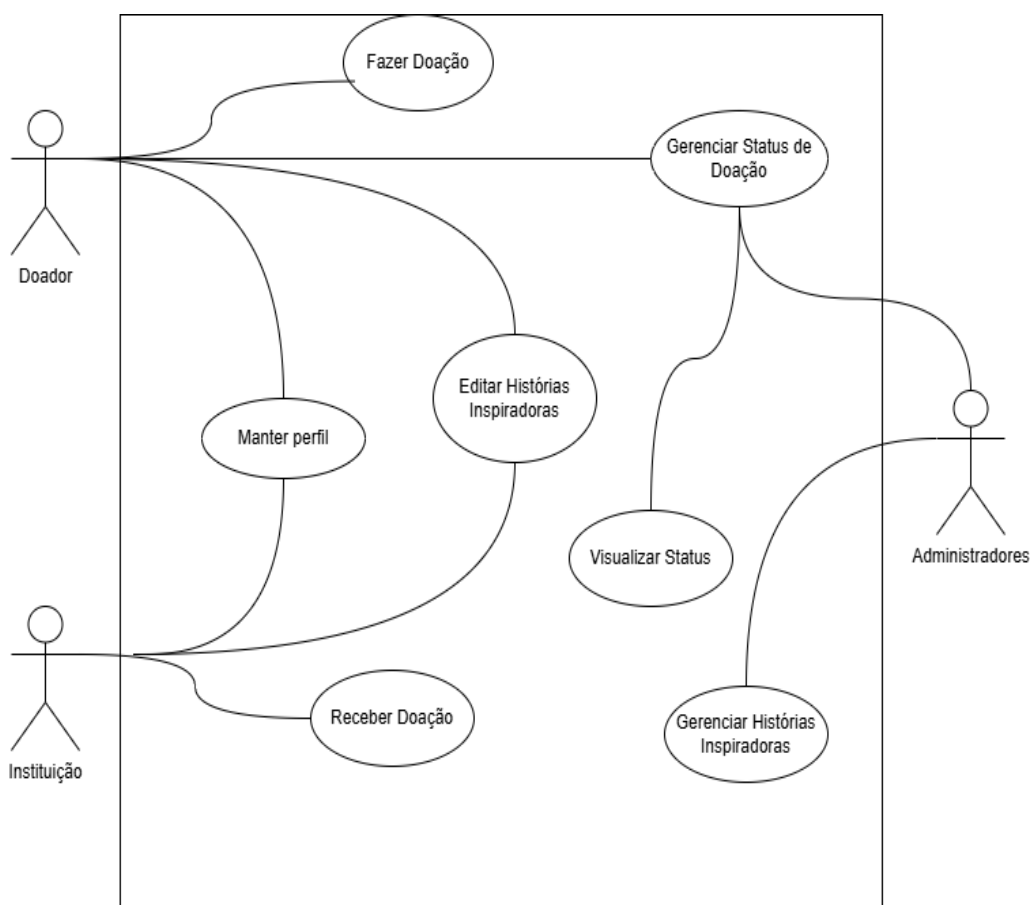
## 5. Modelagem do Sistema

Para estruturar o sistema antes da codificação, foram utilizados diagramas UML (*Unified Modeling Language*). A UML é uma linguagem padrão voltada para modelagem de sistemas, permitindo representar aspectos estruturais e comportamentais de forma unificada [Booch 2006]. Foram elaborados dois tipos principais de diagramas:

### 5.1. Diagrama de Casos de Uso

O diagrama de casos de uso do sistema Solly descreve as interações entre três principais atores (Doador, Instituição e Administrador) e as funcionalidades oferecidas pela plataforma. Doador: pode realizar doações, manter seu perfil atualizado, editar histórias inspiradoras e gerenciar o *status* das doações. Instituição: recebe as doações realizadas, manter seu perfil atualizado e também pode editar histórias inspiradoras. Administrador: atua como um gerenciador central, garantindo a consistência e a confiabilidade dos dados entregues e recebidos. É responsável por gerenciar as histórias inspiradoras e o *status* das doações.

Na Figura 1, é apresentado o diagrama de caso de uso, que ilustra a interação dos três principais atores (doador, instituição e administrador) com as funcionalidades centrais do sistema Solly. Esses diagramas suportam futuras validações de usabilidade.

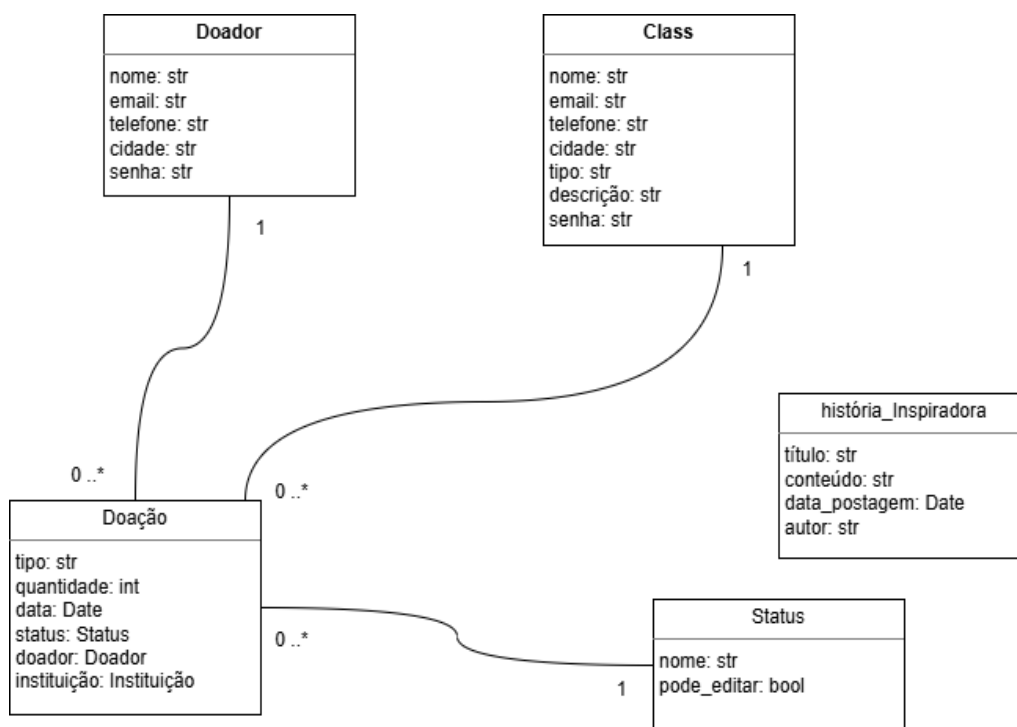


**Figura 1. Diagrama de Casos de Uso do sistema Solly**

## 5.2. Diagrama de Classes

Doador e Instituição: possuem atributos fundamentais para autenticação e identificação, como nome, *e-mail*, telefone, cidade, senha, tipo e descrição. Doação: relaciona-se tanto com o Doador quanto com a Instituição e o *Status*, armazenando dados como tipo, quantidade e data. *Status*: se relaciona com Doações e é responsável por controlar a situação da doação (se foi concluída, finalizada, em andamento, etc.). História Inspiradora: permite o registro de experiências positivas na plataforma e possui informações como título, conteúdo, data da postagem e autor.

A Figura 2 apresenta o diagrama de classes do sistema, evidenciando a modelagem das principais entidades (Doador, Instituição, Administrador e Doação) e seus relacionamentos. Essa representação visual complementa a arquitetura MTV descrita anteriormente e mostra como os dados são estruturados internamente no Solly.



**Figura 2. Diagrama de Classe do sistema Solly**

Os diagramas representaram uma etapa essencial do projeto, pois estabeleceram a ligação entre o planejamento teórico e a fase de implementação. Eles possibilitaram a verificação prévia de que os requisitos levantados estavam devidamente modelados antes do início da codificação, além de servirem como base para a estruturação do banco de dados e das classes do Django. Dessa forma, contribuíram para evitar retrabalhos decorrentes de inconsistências entre requisitos e código.

## 6. Desenvolvimento do Software

O desenvolvimento do sistema Solly envolveu a aplicação prática dos conceitos de engenharia de software apresentados nas seções anteriores, consolidando-os em um produto funcional e escalável. Nesta etapa, descrevem-se as decisões técnicas, a arquitetura adotada e as principais funcionalidades implementadas durante a construção da plataforma. O foco esteve na criação de um ambiente modular, seguro e de fácil manutenção, fundamentado em tecnologias modernas e boas práticas de desenvolvimento web.

### 6.1. Estrutura do Projeto em Django

O desenvolvimento do Solly seguiu a arquitetura padrão do Django, estruturando o projeto em múltiplos aplicativos (*apps*), o que favorece modularidade e manutenção. Os principais *apps* do sistema incluem: *usuarios*, para gestão de autenticação e perfis; *instituicoes*, para cadastro e gerenciamento de instituições; *doacoes*, responsável pelo fluxo de doações de itens e dinheiro; e *historias*, que reúne relatos inspiradores do ecossistema solidário. O Django é fundamentado na separação entre modelos (*models*), visões (*views*) e *templates*:

- **Models**: definem o esquema do banco de dados, representando entidades como Doador, Instituição, Doação e História Inspiradora.

- *Views*: implementam a lógica de controle, processando requisições HTTP, validando formulários, controlando permissões e direcionando respostas.
- *Templates*: estruturam a camada visual, utilizando HTML5 integrado ao mecanismo de *templates* do Django para renderizar páginas dinâmicas. A utilização de *templates* genéricos, herança de base (*base.html*) e inclusão de componentes reutilizáveis potencializou a padronização visual e acelerou a entrega de novas telas.

## 6.2. Principais Funcionalidades Implementadas

O Solly foi projetado para facilitar ao máximo as ações solidárias. Entre as principais funcionalidades destacam-se:

- Cadastro detalhado de doadores e instituições, com validação de e-mails e proteção contra duplicidade.
- Autenticação, alteração e recuperação de senha, *logout* seguro. Perfis editáveis/autogerenciáveis para cada ator do sistema.
- Processo completo de doação de itens: busca, filtro por necessidades, agendamento de coleta e histórico detalhado das transações.
- Doação financeira sem taxas para as instituições, com emissão automática de recibo digital ao doador.
- Sistema centralizado de relatos e depoimentos (Histórias Inspiradoras).
- Interface administrativa intuitiva para gestores, otimizando o controle global da plataforma.

## 6.3. Uso do Bootstrap e jQuery

O Bootstrap foi usado como base do *front-end*, garantindo responsividade, acessibilidade e uniformidade visual. O sistema utiliza *grids*, *cards*, caixas de mensagens e formulários estilizados, além de menus adaptativos. O jQuery, presente na interação do usuário, viabiliza validação dinâmica de campos, mensagens instantâneas, *feedback* visual (ex: *alerts* após cadastro/erro) e manipulação de elementos sem recarregar as páginas, tornando a navegação prática.

## 6.4. Exemplos de Telas do Sistema

A seguir, são apresentadas algumas telas emblemáticas que ilustram a experiência do usuário no Solly:

A Figura 3 mostra o formulário de cadastro de instituições, com campos específicos para informações institucionais, contato, escolha de causa e descrição detalhada.

### Cadastre sua Instituição

**Nome da Instituição**

**CNPJ**

**Email**

**Telefone**

**Causa**

Selecione a causa principal...

**Descrição da Instituição**

Descreva o trabalho realizado pela instituição...

Cadastrar Instituição

**Figura 3. Tela de Cadastro de Instituição**

A Figura 4 apresenta a tela para cadastro de doador, simplificada e objetiva, reforçando a usabilidade para novos usuários.

### Cadastre-se como Doador

**Nome Completo**

**Telefone**

**Cidade**

**Email**

Cadastrar Doador

**Figura 4. Tela de Cadastro de Doador**

A Figura 5 exemplifica o mural de histórias inspiradoras, incentivando o engajamento com depoimentos de beneficiários, doadores e voluntários.



**Figura 5. Página de Histórias Inspiradoras**

A Figura 6 destaca a página inicial da plataforma, que transmite imediatamente a missão do Solly de conectar doadores e instituições.



**Figura 6. Página Inicial da Plataforma Solly**

## 6.5. Benefícios do Uso da Solução Desenvolvida

O Solly agrega vantagens relevantes frente a plataformas do mesmo segmento:

- Possibilita doação tanto financeira quanto de itens, com acompanhamento do processo.
- Transparência total: doador e instituição acessam o histórico completo das transações.
- Intuitividade e adaptação a qualquer dispositivo, estimulando o engajamento contínuo de novos usuários.
- Suporte a perfis independentes e auto gerenciáveis.
- Infraestrutura baseada em código aberto (Django, Bootstrap), o que facilita a evolução, integração com novas APIs e manutenção do sistema.

## 6.6. Segurança, Privacidade e Compliance

O Solly incorpora práticas de segurança aderentes à LGPD e princípios de engenharia de software segura. Entre as medidas adotadas estão:

- Autenticação e autorização baseadas em perfis de usuário;
  - Armazenamento seguro de senhas com *hash*;
  - *logs* de auditoria para rastreamento de operações sensíveis;
  - Criptografia de comunicações via HTTPS;
  - Consentimento explícito para coleta e uso de dados pessoais.
- Futuramente, pretende-se incluir mecanismos de verificação institucional automatizada e relatórios de transparência pública sobre doações e repasses.

## 7. Conclusão

A conclusão deste trabalho reafirma o papel essencial da tecnologia na promoção da solidariedade por meio do desenvolvimento do Solly, uma plataforma digital de doação online. Ao longo do estudo, abordou-se um problema real, demonstrando como barreiras de confiança e transparência dificultam a doação. O projeto alcançou seu principal objetivo: oferecer um ambiente seguro, intuitivo e transparente para conectar doadores e instituições, facilitando e rastreando todo o fluxo de doações.

Os resultados evidenciam benefícios relevantes, como o aumento na segurança, facilidade de uso e personalização para diferentes perfis de usuários. O método de desenvolvimento incremental e a aplicação das melhores práticas de usabilidade e arquitetura do Django propiciaram um produto robusto e adaptável. Destaca-se, ainda, o enfoque no controle de acesso, garantindo privacidade e gestão segura dos dados.

As contribuições práticas do Solly vão desde o estímulo à cidadania digital até a ampliação do impacto de causas sociais, servindo de referência para futuras soluções tecnológicas no setor do terceiro setor. Do ponto de vista científico, o projeto proporciona espaço para novas pesquisas em usabilidade, análise de dados de doações e integração com *fintechs*.

Embora o *gateway* de pagamento ainda esteja em fase de integração, o sistema já implementa as demais etapas do fluxo solidário, incluindo cadastro, agendamento e emissão de comprovantes digitais. O recurso de doação sem taxas está planejado para a próxima versão, utilizando integração direta via PIX e validação de recibos digitais verificáveis.

### 7.1. Lições Aprendidas e Perspectivas Futuras

O desenvolvimento do Solly permitiu consolidar conhecimentos em engenharia de software aplicada, integração de *frameworks* e práticas de segurança *web*. Observou-se que o uso do padrão MTV e da modelagem UML favoreceu a clareza arquitetural e reduziu inconsistências entre requisitos e código.

Em trabalhos futuros, pretende-se ampliar os testes automatizados, realizar avaliações de usabilidade com usuários reais e implementar a integração com APIs de pagamento e verificação institucional.

## Referências

- Alura (2023). Django: o que é, para que serve e um guia desse framework. <https://www.alura.com.br/artigos/django-framework>. Acesso em: 27 set. 2025.
- Booch, G. (2006). *UML: guia do usuário*. Elsevier, Rio de Janeiro.
- Brasil (2018). Lei nº 13.709, de 14 de agosto de 2018 (lei geral de proteção de dados pessoais – lgpd). [https://www.planalto.gov.br/ccivil\\_03/\\_ato2015-2018/2018/lei/113709.htm](https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/113709.htm). Acesso em: 21 out. 2025.
- Brasil, B. (2023a). Documentação oficial do bootstrap 5. <https://getbootstrap.com.br/>. Acesso em: 21 out. 2025.
- Brasil, P. (2023b). Banco de dados postgresql – guia e documentação. <https://www.postgresql.org/>. Acesso em: 21 out. 2025.
- Cardoso, J. and Bispo, W. (2019). Um estudo comparativo entre os principais frameworks de desenvolvimento web em linguagem python. Technical report, UFRA. Acesso em: 27 set. 2025.
- Costa, K. and Barbosa, I. (2020). Modelagem de sistemas utilizando a uml: Aplicando a engenharia de software. Technical report, CPS. Acesso em: 26 set. 2025.
- de Pieri Bernhardt, R. (2023). Modelagem de sistemas utilizando a uml: Aplicando a engenharia de software. Technical report, Instituição Desconhecida. Acesso em: 27 set. 2025.
- Foundation, D. S. (2024a). Django web framework. <https://www.djangoproject.com/>. Acesso em: 21 out. 2025.
- Foundation, O. (2024b). Owasp top 10: Security risks for web applications. <https://owasp.org/www-project-top-ten/>. Acesso em: 21 out. 2025.
- Gorgulho, A., Mendes, L., and Santos, F. (2021). Plataformas brasileiras de doação online: análise e funcionalidades. *Revista Brasileira de Tecnologia Social*, 12(2):45–60.
- Jiang, J., Wang, X., Li, L., and Yang, B. (2021). Design and implementation of charity donation system. *Francis Academic Press*.
- Mane, A. (2022). A study of donation-based crowd funding platforms. *International Journal of Multidisciplinary Research Academy (IJMRA)*.
- Network, M. D. (2024). Documentação de html5, css3 e javascript. <https://developer.mozilla.org/pt-BR/>. Acesso em: 21 out. 2025.
- Pereira, L. A. and Martins, C. R. (2023). Implementação da lgpd em plataformas web: Desafios de segurança e confiabilidade. In *Anais do Congresso Brasileiro de Tecnologia da Informação e Comunicação*, pages 112–123, São Paulo, Brasil. SBC. Acesso em: 21 out. 2025.
- Portulhak, H., Vaz, P. V. C., Delay, A. J., and Pacheco, V. (2017). A qualidade da prestação de contas das entidades do terceiro setor: uma análise a partir de sua relação com o comportamento dos doadores individuais. *Enfoque: Reflexão Contábil*, 36(1):45–63. Acesso em: 26 set. 2025.
- Sommerville, I. (2019). *Engenharia de Software*. Pearson, São Paulo, 10 edition.