



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**GABRIEL FERNANDES DA COSTA**

**ACEV+: UMA SOLUÇÃO WEB EFICIENTE PARA O  
GERENCIAMENTO E INTEGRAÇÃO DE UMA INSTITUIÇÃO  
RELIGIOSA**

**CAMPINA GRANDE - PB**

**2024**

**GABRIEL FERNANDES DA COSTA**

**ACEV+: UMA SOLUÇÃO WEB EFICIENTE PARA O  
GERENCIAMENTO E INTEGRAÇÃO DE UMA INSTITUIÇÃO  
RELIGIOSA**

**Trabalho de Conclusão Curso  
apresentado ao Curso Bacharelado em  
Ciência da Computação do Centro de  
Engenharia Elétrica e Informática da  
Universidade Federal de Campina  
Grande, como requisito parcial para  
obtenção do título de Bacharel em  
Ciência da Computação.**

**Orientador : Adalberto Cajueiro de Farias**

**CAMPINA GRANDE - PB**

**2024**

**GABRIEL FERNANDES DA COSTA**

**ACEV+: UMA SOLUÇÃO WEB EFICIENTE PARA O  
GERENCIAMENTO E INTEGRAÇÃO DE UMA INSTITUIÇÃO  
RELIGIOSA**

**Trabalho de Conclusão Curso  
apresentado ao Curso Bacharelado em  
Ciência da Computação do Centro de  
Engenharia Elétrica e Informática da  
Universidade Federal de Campina  
Grande, como requisito parcial para  
obtenção do título de Bacharel em  
Ciência da Computação.**

**BANCA EXAMINADORA:**

**Adalberto Cajueiro de Farias**

**Orientador – UASC/CEEI/UFCG**

**José Antão Beltrão Moura**

**Examinador – UASC/CEEI/UFCG**

**Francisco Vilar Brasileiro**

**Professor da Disciplina TCC – UASC/CEEI/UFCG**

**Trabalho aprovado em: 15 de maio de 2024.**

## **CAMPINA GRANDE - PB**

### **RESUMO**

As instituições religiosas desempenham um papel vital na vida comunitária de muitos indivíduos. No entanto, à medida que essas comunidades crescem em tamanho e complexidade, a necessidade de ferramentas de gestão eficientes torna-se ainda mais urgente. Este trabalho tem como objetivo apresentar um sistema que atenda a essa necessidade por meio de uma plataforma centralizada para gerenciar as entidades presentes no contexto da instituição ACEV em Campina Grande - PB. O sistema engloba uma variedade de funcionalidades, incluindo registro e gerenciamento de dados dos membros, grupos, ministérios, acompanhamento de reuniões, bem como gestão de séries de estudo e lições associadas. Esta solução visa facilitar a organização, colaboração e crescimento dentro da instituição, além de trazer uma abordagem mais automatizada e eficaz para a criação de planejamentos estratégicos.

# **ACEV+: AN EFFICIENT WEB SOLUTION FOR THE MANAGEMENT AND INTEGRATION OF A RELIGIOUS INSTITUTION**

## **ABSTRACT**

Religious institutions play a vital role in the community life of many individuals. However, as these communities grow in size and complexity, the need for efficient management tools becomes even more urgent. This work aims to present a system that meets this need through a centralized platform for managing the entities within the context of the ACEV institution in Campina Grande - PB. The system encompasses a variety of functionalities, including member data registration and management, groups, ministries, meeting tracking, as well as study series and associated lessons management. This solution aims to facilitate organization, collaboration, and growth within the institution, as well as to bring a more automated and effective approach to strategic planning.

# ACEV+: Uma solução web eficiente para o gerenciamento e integração de uma instituição religiosa

Gabriel Fernandes da Costa  
Universidade Federal de Campina Grande  
Campina Grande, Paraíba, Brasil.  
gabriel.costa@ccc.ufcg.edu.br

Orientador: Adalberto Cajueiro de Farias  
Universidade Federal de Campina Grande  
Campina Grande, Paraíba, Brasil.  
adalberto@computacao.ufcg.edu.br

## RESUMO

As instituições religiosas desempenham um papel vital na vida comunitária de muitos indivíduos. No entanto, à medida que essas comunidades crescem em tamanho e complexidade, a necessidade de ferramentas de gestão eficientes torna-se ainda mais urgente. Este trabalho tem como objetivo apresentar um sistema que atenda a essa necessidade por meio de uma plataforma centralizada para gerenciar as entidades presentes no contexto da instituição ACEV em Campina Grande - PB. O sistema engloba uma variedade de funcionalidades, incluindo registro e gerenciamento de dados dos membros, grupos, ministérios, acompanhamento de reuniões, bem como gestão de séries de estudo e lições associadas. Esta solução visa facilitar a organização, colaboração e crescimento dentro da instituição, além de trazer uma abordagem mais automatizada e eficaz para a criação de planejamentos estratégicos.

## Palavras-chave

Igreja, Gestão, Integração, Desenvolvimento Web.

## 1. INTRODUÇÃO

A ascensão das instituições religiosas é um fato amplamente conhecido no território brasileiro. De acordo com a matéria de Marco Grillo [1], uma nova organização religiosa surge a cada hora, o que exemplifica a magnitude desse crescimento. Segundo Verba, Schlozman e Brady [2], essa realidade tem grandes impactos na sociedade, uma vez que a participação da população em instituições religiosas gera alguns efeitos sociais, como o aumento do potencial para atividade política por meio da geração de oportunidades para praticar habilidades cívicas. Junto desse crescimento, surgem também desafios no que diz respeito ao gerenciamento e integração das pessoas no âmbito das instituições.

No contexto da ACEV, dificuldades significativas são enfrentadas em termos de gestão, especialmente quando se trata da falta de um sistema específico para esse fim. Atualmente, as tarefas administrativas e organizacionais são realizadas de forma descentralizada e muitas vezes fragmentada, principalmente através de gerenciamento de grupos no WhatsApp [3]. Embora essa plataforma ofereça uma maneira rápida e conveniente de compartilhar informações e administrar pessoas em contextos específicos, ela não é adequada para lidar com todas as complexidades envolvidas na gestão abrangente de uma instituição.

Essa abordagem fragmentada resulta em diversos problemas, tais como:

- **Descentralização:** As informações relevantes, como registros de membros, ministérios, estudos e reuniões, muitas vezes se perdem ou ficam dispersas em meio a múltiplas conversas pessoais e grupos variados. A falta de um sistema centralizado dificulta o acesso rápido e eficiente a dados importantes, além de comprometer a rastreabilidade das informações, o que pode levar a erros e redundâncias.
- **Limitações na análise de dados:** A ausência de ferramentas adequadas para análise de dados dificulta a compreensão do desempenho da ACEV em diferentes áreas, como crescimento da membresia, participação em grupos de estudo, ministérios e assiduidade em reuniões.
- **Incapacidade de uso dos membros:** A instituição possui diversos tipos de faixa etária e grupos sociais e alguns destes carecem da possibilidade de utilizar o WhatsApp, como crianças e idosos. Este fato dificulta a contagem exata destas pessoas na participação de grupos específicos, exigindo um trabalho de memória pessoal para a inserção posterior no censo.

Diante desse cenário, torna-se evidente a necessidade de desenvolver um sistema dedicado de gerenciamento para a ACEV que seja capaz de centralizar e organizar as atividades administrativas, melhorar a comunicação interna e facilitar a análise de dados para tomadas de decisão mais embasadas e eficientes. Este trabalho propõe o desenvolvimento e implementação do ACEV+, uma plataforma de gestão específica, visando suprir essas necessidades e promover um gerenciamento mais eficaz e transparente na instituição, além de servir como modelo simples e de fácil implementação para possíveis projetos de mesma natureza.

## 2. SOLUÇÃO

Atualmente, é possível encontrar no mercado soluções já disponíveis para a resolução dos problemas citados, como por exemplo o InChurch [4]. No entanto, essas ferramentas possuem, muitas vezes, preços pré-definidos que fogem da realidade financeira das instituições, ademais, as soluções apresentam limitações no que diz respeito às particularidades e necessidades específicas dessas organizações, que acabam impactando com restrições na personalização, dificuldades na integração de processos existentes, além de custos extra para manutenção e melhorias pontuais.

Diante disso, este trabalho visa apresentar a implementação de um sistema que propõe superar essas limitações, oferecendo uma solução personalizada e adaptada às necessidades específicas

desta instituição, mas que pode servir como base para outros projetos. A solução possui como público alvo as lideranças existentes, além de pastores e secretários da instituição. Os grupos (chamados internamente de agrupês) referem-se a reuniões menores entre os membros da igreja que ocorrem regularmente. Os ministérios, por sua vez, representam áreas específicas de serviço e atividade. Para o levantamento dos requisitos, líderes dessas entidades e secretários da instituição foram consultados para contribuir com ideias de funcionalidades que seriam importantes para o uso do sistema.

## 2.1 Funcionalidades

### 2.1.1 Autenticação

A plataforma possui uma tela de autenticação padrão com email e senha (representação na figura 1) que são pré-definidos e enviados aos líderes que devem ter acesso ao sistema, todas as rotas internas são guardadas e necessitam do *token* recebido na requisição de autenticação para serem validadas.



Figura 1: Tela de autenticação

### 2.1.2 Gerenciamento de usuários

O sistema possui um gerenciamento de usuários controlado via API (Application Programming Interface). O sistema possui dois tipos de usuários: o comum (USER) e o administrador (ADMIN). O administrador, composto pelos pastores, secretários e pessoas que estarão envolvidas no processo de manutenção e desenvolvimento da plataforma, detém todos os direitos de acesso e utilização da aplicação, enquanto os usuários comuns, representados pelos líderes de agrupês e ministérios, possuem acesso limitado a determinadas funcionalidades.

### 2.1.3 Gerenciamento de membros

A plataforma disponibiliza também o gerenciamento de membros (representação na figura 2 e 3). Os membros representam as pessoas que participam ativamente das atividades que a instituição oferece. Algumas informações pessoais básicas relevantes são armazenadas para fins de análise. Os membros podem participar de *agrupês* e ministérios, além de estarem atrelados a relatórios que os líderes cadastram. Tal entidade possui atributos filtráveis que podem facilitar a análise, como o tipo (membro ou congregado) e status de arquivamento, uma vez que os membros podem ser arquivados. Apenas usuários do tipo ADMIN podem realizar cadastro e atualização de membros.

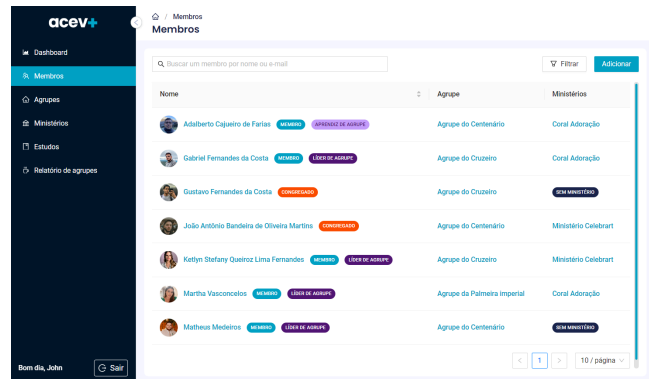


Figura 2: Módulo de listagem de membros

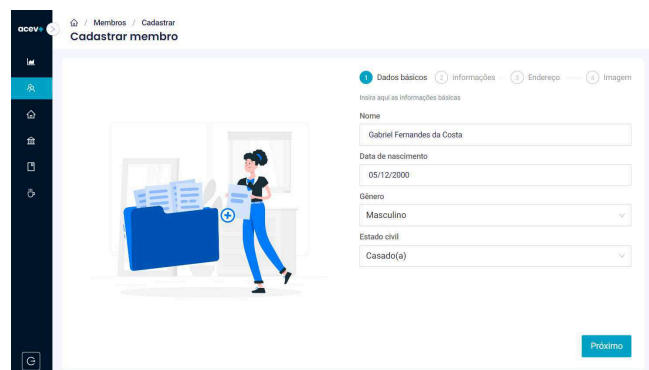


Figura 3: Módulo de gerenciamento de membros

### 2.1.4 Gerenciamento de agrupês

Os líderes e aprendizes possuem a capacidade de gerenciar seus próprios agrupês e os membros pertencentes a eles (representação na figura 4). Cada agrupe possui uma lista de membros associados, um endereço e relatórios atrelados. Os administradores, por sua vez, possuem acesso a funcionalidades avançadas para gerenciar com profundidade cada agrupe. Isso inclui operações básicas como criar, editar e arquivar, bem como atribuir ou remover lideranças de cada grupo.

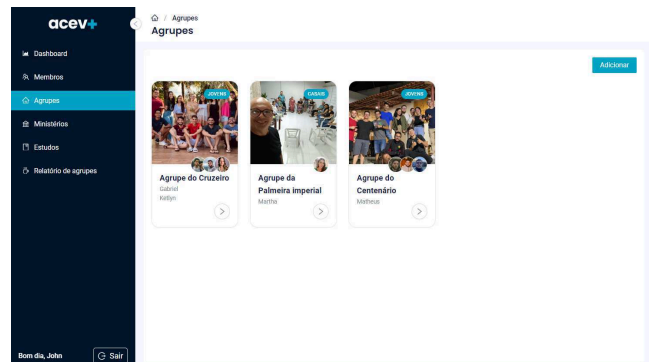


Figura 4: Módulo de gerenciamento de agrupês

### 2.1.5 Gerenciamento de ministérios

Os ministérios também são entidades presentes no contexto do sistema. No entanto, eles possuem, a priori, um escopo apenas para gerenciamento (representação na figura 5), sendo este de responsabilidade dos administradores. Os líderes podem, no entanto, gerenciar as pessoas que participam, assim como seus devidos cargos. A presença desta entidade possibilitará melhorias futuras no que diz respeito à coleta de métricas de participação da comunidade.

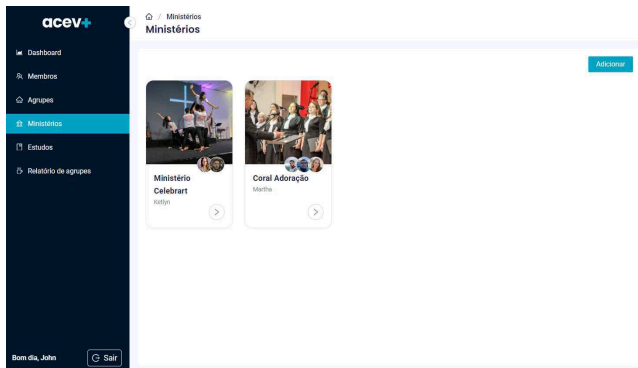


Figura 5: Módulo de gerenciamento de ministérios

### 2.1.6 Gerenciamento de séries e lições

O sistema permite o gerenciamento de séries de estudos e lições (representação na figura 6). Cada série possui uma lista de lições atreladas. Cada um dos estudos possuirá um documento vinculado que pode ser acessado e baixado pelo usuário. As séries e estudos poderão ser manipulados apenas por usuários administradores.

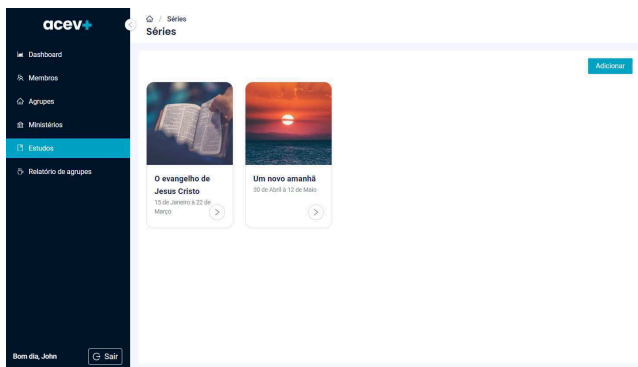


Figura 6: Módulo de gerenciamento de séries e lições

### 2.1.7 Gerenciamento de relatórios

O sistema disponibiliza o gerenciamento de relatórios (representação na figura 7) que significa um encontro presencial de um agrupe. O relatório deve possuir um agrupe vinculado, uma lição, os membros que participaram e possíveis convidados, além de informações básicas, como a data e uma foto retirada no local.

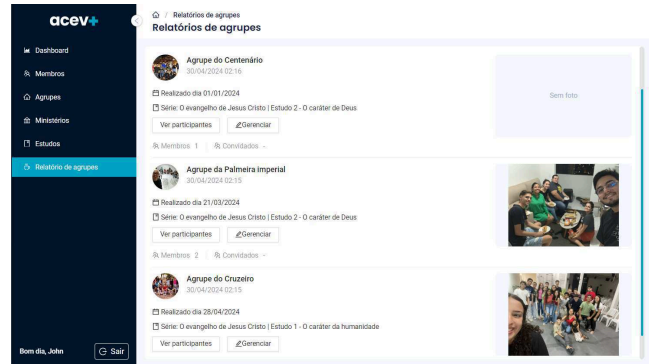


Figura 7: Módulo de gerenciamento de relatórios

### 2.1.8 Dashboard

Para fins de visualização, o sistema possui em sua página inicial um dashboard que contém fácil acesso a informações de censo da instituição, como representado na figura 8. Tal tela possui alta margem para crescimento e personalização futura, uma vez que o sistema possui muitas entidades metrificáveis e passíveis de análise que serão decididas ao longo da utilização da plataforma.



Figura 8: Tela de dashboard

## 3. ARQUITETURA

O sistema segue uma arquitetura cliente-servidor simplificada, utilizando tecnologias modernas, escaláveis e seguras, conforme representado na Figura 9. Essa abordagem facilita a comunicação eficiente entre o cliente (interface de usuário) e o servidor (responsável pelo processamento e armazenamento de dados), garantindo uma experiência de usuário fluida e segura.



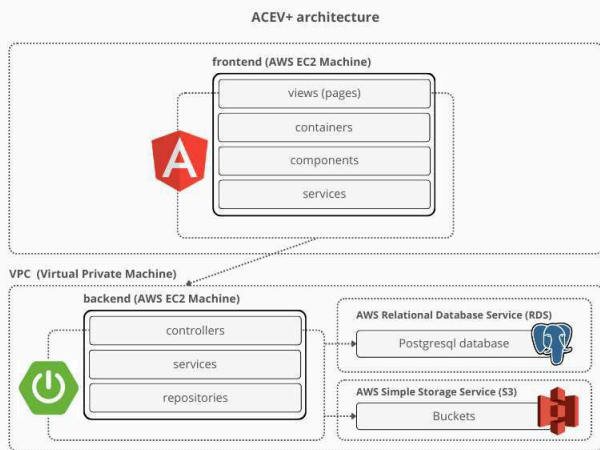


Figura 9: Representação da arquitetura do sistema

### 3.1 Front-end

O front-end da aplicação foi desenvolvido utilizando o framework Angular [5] na versão 16. A estrutura de código visa oferecer maior organização e facilidade de manutenção implementando o conceito de módulos, conforme representado na Figura 10. Além disso, foi realizada uma separação entre *containers* (componentes inteligentes) e *pages* (componentes mudos). Os componentes inteligentes fazem conexão aos serviços e detêm a lógica de negócio da aplicação, enquanto os componentes mudos são simples e dependentes dos dados que recebem, tendo como função apenas a exibição dos dados e interação direta com o usuário. Os benefícios incluem separação de preocupações, reusabilidade, testabilidade e manutenibilidade do código.

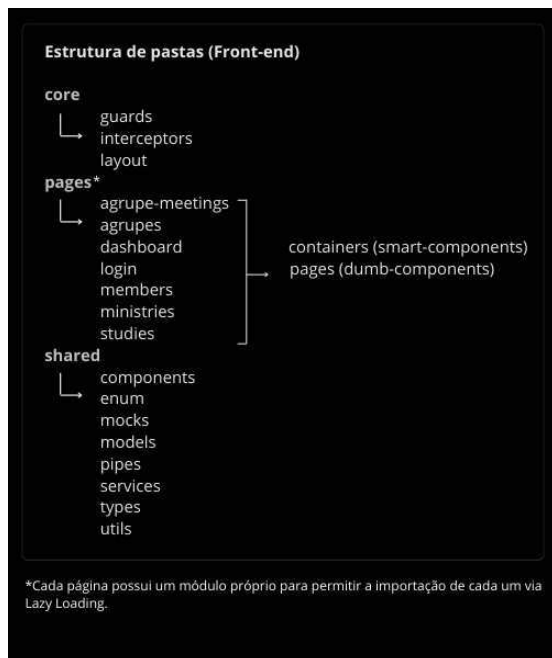


Figura 10: Organização de pastas do Front-end

Para o uso de componentes, foi utilizada a biblioteca Ant Design [6] que oferece uma ampla gama de componentes prontos para uso, como botões, formulários, tabelas e muito mais, esta biblioteca é conhecida por sua simplicidade e elegância, além disso, foram criados componentes próprios reutilizáveis que atendem as especificações do projeto. Para a plotagem de gráficos, foi utilizada a biblioteca Chart.js [7], ela oferece uma série de atributos personalizáveis e de fácil integração para frameworks como o Angular.

### 3.2 Back-end

O servidor foi desenvolvido utilizando Java 17 e o framework Spring Boot [8] na versão 3.2.1. Ele desempenha um papel crucial como intermediário entre os dados armazenados no banco de dados e a interface do usuário. A escolha do Spring Boot foi motivada pela sua facilidade de implementação de serviços web, o que agilizou significativamente o processo de desenvolvimento.

Para a parte de autenticação e autorização, foi adotado o Spring Security [9] juntamente com a utilização do JWT (JSON Web Token) [10], o que proporcionou uma camada adicional de segurança para a transmissão e autenticação de informações sensíveis.

A comunicação via API seguiu o padrão REST API (Representational State Transfer), permitindo uma troca eficiente de informações entre cliente e servidor. O emprego do padrão de projeto DTO (Data Transfer Object) possibilitou a otimização dessa comunicação, garantindo a eficácia na transmissão de dados entre as partes envolvidas.



Figura 11: Organização de pastas do Back-end

Todo o desenvolvimento foi orientado pelo padrão de arquitetura MVC (Model-View-Controller), proporcionando uma estrutura organizada e modular, como pode ser visualizado na figura 11.

### 3.3 Banco de dados

Na camada de persistência, o sistema adotou o PostgreSQL [11] como banco de dados relacional. Esta tecnologia foi escolhida

devido à sua robustez, confiabilidade e capacidade de lidar eficientemente com grandes volumes de dados. Para facilitar a interação entre a aplicação Java e o banco de dados PostgreSQL, foi adotado o JPA (Java Persistence API). O JPA é uma especificação do Java que fornece uma interface de programação para gerenciar dados relacionais no ambiente Java de maneira simplificada e eficiente. Ele simplifica a tarefa de persistir objetos no banco de dados, permitindo que o desenvolvimento ocorra com uma camada de abstração de alto nível, em vez de lidar diretamente com a linguagem de banco de dados.

Ademais, no sistema foi integrado um *bucket* para armazenar imagens e documentos de forma escalável e durável. Isso proporciona aos usuários uma maneira rápida e segura de fazer upload e download de arquivos, sem sobrecarregar o banco de dados.

### 3.4 Infraestrutura

Na infraestrutura do sistema, foram utilizados integralmente os serviços oferecidos pela Amazon Web Services (AWS) [12]. O frontend e o backend foram hospedados em instâncias EC2 (Elastic Compute Cloud), enquanto o banco de dados é gerenciado pelo RDS (Relational Database Service), ao passo que a persistência de imagens e arquivos em *buckets* é realizada no S3. Todos os componentes que fazem parte da camada de servidor foram configurados dentro de uma VPC (Virtual Private Cloud), garantindo um ambiente isolado e seguro para as operações do sistema. Essa VPC oferece controle total sobre a rede virtual, permitindo configurar sub-redes, tabelas de roteamento e gateways de internet, garantindo assim um alto nível de segurança e privacidade para os recursos hospedados na AWS.

## 4. EXPERIÊNCIA

Esta seção tem como intuito destacar os processos e ritos que foram envolvidos no desenvolvimento da aplicação, assim como a avaliação de uso e trabalhos futuros para a plataforma.

### 4.1 Processo de desenvolvimento

A primeira etapa do processo de desenvolvimento partiu do levantamento de requisitos, para isto, foi realizada uma reunião pontual com o líder da instituição, nela foram alinhadas as necessidades e dores que o sistema iria agir em resolução. Após isso, foram feitas análises e testes de potenciais tecnologias que poderiam ser utilizadas, considerando alguns aspectos importantes, como facilidade de manutenção, afinidade e custos, o que resultou na escolha dos frameworks que foram evidenciados. Como uma das ações para a redução de custos, a todo o momento foi utilizado apenas um ambiente de desenvolvimento com a versão menos custosa disponível na AWS. Ao longo do desenvolvimento foi realizada uma outra reunião, desta vez com alguns líderes, pastores e secretária da instituição para uma apresentação de escopo geral da plataforma e levantamento de novos requisitos.

Tratando-se de metodologia, foi adotado o método KANBAN [13], juntamente com um quadro de tarefas digital, utilizando o Jira [14] para gerenciar o fluxo de trabalho. Quanto aos repositórios de código, todos foram armazenados na plataforma Github [15]<sup>1</sup>.

<sup>1</sup> <https://github.com/gabrielfernans/acev-mais>  
<https://github.com/gabrielfernans/acev-mais-api>

## 4.2 Desafios

Durante o desenvolvimento do projeto, algumas dificuldades foram levantadas, exigindo uma certa adaptabilidade e dedicação extra, dentre elas se destacam:

- **Custos de infraestrutura:** O gerenciamento dos custos associados aos servidores e ao banco de dados representou o maior desafio, exigindo estratégias eficientes de otimização de recursos para manter a viabilidade financeira do projeto. Em síntese, as versões gratuitas dos bancos de dados RDS se mostraram satisfatórias nesse sentido após sua configuração.
- **Conhecimento raso sobre AWS:** A escolha da AWS como plataforma de oferta de serviços de infraestrutura demandou uma curva de aprendizado acentuada para compreender os diferentes serviços oferecidos e configurá-los de acordo com as necessidades do projeto. Como apoio, foi utilizada a própria documentação da AWS, o que foi crucial no processo de aprendizado.
- **Desatualização das tecnologias** A evolução contínua das tecnologias utilizadas no projeto exigiu uma adaptação constante do desenvolvimento às últimas atualizações e melhores práticas, uma vez que o último contato prático com o Spring Boot e o JPA fazia um tempo considerável, o que resultou numa dessincronia com as novidades presentes nas versões mais recentes. Nesse quesito, a documentação da própria tecnologia também foi fator determinante para a familiarização.

No entanto, tendo em vista a limitação de tempo por parte do autor, as entregas foram consideradas satisfatórias, tendo em vista a grande margem de crescimento da plataforma no futuro e a base sólida construída com tecnologias seguras e escaláveis.

## 4.3 Avaliação

Algumas avaliações da plataforma foram realizadas com 2 potenciais usuários (líderes), assim como com a secretária da instituição, que deram *feedbacks* interessantes e construtivos. Segundo os líderes, o sistema apresenta uma interface simples e intuitiva e explicitaram que, tratando-se de uma primeira entrega, iria solucionar bem as dores a respeito da descentralização de informações e de busca das lições que são utilizadas como estudos. Por outro lado, foi informado que a plataforma ainda é muito básica, mas que com as propostas futuras de melhoria o uso se tornaria rotineiro. A secretária, por sua vez, realizou alguns testes de usabilidade, envolvendo visualização das features, tentativas de cadastros, edições e exclusões, o que acarretou na identificação de alguns *bugs* de simples resolução no sistema que foram imediatamente corrigidos. No entanto, destacou a beleza da plataforma e a exponencial possibilidade de resolução de problemas sistemáticos da instituição.

## 4.4 Trabalhos futuros

O sistema ACEV+ foi desenvolvido com o intuito de ser utilizado e mantido por tempo indeterminado, sendo este trabalho apenas um produto mínimo viável (MVP). A plataforma possui grandes perspectivas de crescimento, tendo em vista que se trata de um sistema desenvolvido para o uso ativo da instituição mencionada. Neste sentido, como possíveis trabalhos futuros, destacam-se as seguintes *features*:

- Disponibilização da plataforma para dispositivos móveis via *Web App* [16].

- Funcionalidade de exportação de relatórios e mais possibilidades de filtragem dos dados.
- Utilização de mecanismos de “cacheamento” para economia e velocidade do sistema..
- Expansão do sistema para igrejas parceiras da instituição.

## 5. AGRADECIMENTOS

Expresso a minha profunda gratidão ao meu Deus, dono de toda a existência, por tamanha e multiforme graça. A minha amada esposa Ketlyn Stefany, por ter sido um pilar imensurável em minha jornada. Aos meus pais Jamile e Leonardo, por todo o cuidado e incentivo, aos meus irmãos Gustavo e Benjamim, pela força e companheirismo diário. Agradeço de todo o coração a Henrique Lemos, Erick Sena, Matheus Medeiros e em especial o meu querido amigo João Antônio, pela amizade e apoio durante a minha graduação. Ademais, agradeço a ACEV do Catolé na pessoa do Pr. Wostenes e da Miss. Gleydice, por todo o cuidado, pastoreio e permissão de uso do contexto eclesial para este trabalho. Deixo explícito, ainda, meu agradecimento aos professores Carlos Pires e Adalberto Cajueiro por terem depositado sua confiança em mim e terem sido decisivos na minha vida profissional. Por fim, agradeço a todos os demais familiares e amigos que, de alguma forma, contribuíram para o meu crescimento.

## 6. REFERÊNCIAS

- [1] GRILLO, Marco. 2017. Desde 2010, uma nova organização religiosa surge por hora.. Retrieved 2024-04-26 from <https://oglobo.globo.com/politica/desde-2010-uma-nova-organizacao-religiosa-surge-por-hora-21114799>
- [2] VERBA, Sidney; SCHLOZMAN, Kay L.; BRADY, Henry E. Voice and Equality: Civic Volunteerism in American Politics. Cambridge, USA: Harvard University Press, 1995
- [3] WhatsApp LLC. 2024. Retrieved 2024-04-27 from <https://www.whatsapp.com/about>
- [4] InChurch - Plataforma de sistemas para igrejas. retrieved 2024-04-15 from <https://inchurch.com.br>
- [5] Google. 2010. Angular 16+. Retrieved 2024-04-30 from <https://angular.io>
- [6] Ant Design. 2024. Retrieved 2024-04-30 from <https://ng.ant.design/docs/introduce/en>
- [7] Chart.js. Retrieved 2024-04-30 from <https://www.chartjs.org/>
- [8] VMware Inc. 2014. Spring Boot. Retrieved 2024-04-30 from <https://spring.io/projects/spring-boot>
- [9] VMware Inc. 2004. Spring Security. Retrieved 2024-04-30 from <https://spring.io/projects/spring-security>
- [10] Open Source. 2015. JSON Web Token (JWT). Retrieved 2024-04-30 from <https://www.rfc-editor.org/rfc/rfc7519>
- [11] Open Source. 1986. PostgreSQL: The World's Most Advanced Open Source Relational Database. Retrieved 2024-04-30 from <https://www.postgresql.org/>
- [12] Amazon Web Services. 2024. Retrieved 2024-04-30 from <https://aws.amazon.com/>
- [13] O que é KANBAN?. 2024. Retrieved 2024-05-01 from <https://www.atlassian.com/br/agile/kanban>
- [14] Jira. Retrieved 2024-05-01 from <https://www.atlassian.com/br/software/jira>
- [15] Microsoft. 2008. Github. Retrieved 2023-01-26 from <https://github.com/>
- [16] O que é Web App?. 2020. Retrieved 2024-04-30 from <https://madeinweb.com.br/o-que-e-web-app>