

Sprint Documentation

Sprint 7 Implementar Validações para classes Religion	Start Date: 02/06/2025	Final Date: 10/08/2025
Projetc Name: Senior Stock Manager- Frontend		
Ano: 5º	Análise e Desenvolvimento de Sistemas - AMS	
Team Members		
Kétlyn Christini Tonholi de Moraes		

Sprint Backlog

Task#	Description	Start Date	Final date
38	Sprint 07 – Implementar Validações para classes Religion	02/06/2025	10/08/2025

Task Description

Task #	Description	Assigned To	Status	Estimated Hours	Logged Hours
38	Implementadas validações para a entidade Religion no Front-End React com a criação do service e type de Religion, implementação provisória do service e type de Resident para futura validação de vínculo, aplicação de validações no campo Name (obrigatoriedade, tamanho entre 3 e 100 caracteres, apenas letras e espaços), bloqueio de criação de nomes duplicados, prevenção de exclusão quando houver Resident vinculado, e realização de testes unitários, de integração e práticos para garantir o funcionamento do frontend integrado ao backend.	Kétlyn Christini Tonholi de Moraes.	Concluído	8h	8h

Sprint Description

A Sprint 07 teve como foco principal a implementação das validações para a entidade Religion, com o objetivo de reforçar a consistência dos dados e preparar o sistema para futuras integrações com outras entidades. Esta entrega representou um passo importante na evolução do projeto, ao garantir regras claras para cadastro, edição e exclusão de registros, promovendo maior confiabilidade e padronização no gerenciamento dessas informações.

Durante o desenvolvimento, foram aplicadas validações robustas no frontend, assegurando que o campo Name seja obrigatório, contenha entre 3 e 100 caracteres, aceite apenas letras e espaços e não permita a duplicação de nomes. Além disso, foi criada uma estrutura preliminar de service e type para Resident, permitindo desde já a preparação do sistema para impedir a exclusão de uma Religion que esteja vinculada a algum Resident, exibindo mensagens claras ao usuário sempre que a restrição for acionada.

No aspecto técnico, além das validações, foram criados o service e o type referentes a Religion, respeitando a arquitetura já definida na branch principal. A sprint também avançou na integração entre frontend e backend, realizando ajustes necessários para garantir o fluxo correto das operações, mesmo com a ausência temporária da tabela Resident na base de dados. Essa abordagem preparatória permitirá uma implementação futura mais simples e eficiente do vínculo entre as entidades.

A entrega foi acompanhada de testes unitários, de integração e práticos, que validaram o comportamento da aplicação, a exibição das mensagens de erro e sucesso e a consistência das regras de negócio. Ainda que a exclusão de uma Religion vinculada a Resident não possa ser efetivamente testada nesta fase devido à ausência da entidade Resident no banco, toda a estrutura necessária para essa funcionalidade já foi planejada e parcialmente implementada.

Esta sprint reforçou o compromisso com boas práticas de desenvolvimento, documentação clara no código-fonte e padronização de arquitetura. Ao final do ciclo, concluiu-se a implementação das validações da entidade Religion, preparando o terreno para a evolução do sistema e garantindo maior segurança e qualidade nos dados.

Sprint Results

Telas produzidas



Cadastro de Religião

Início > Cadastros > Cadastro de Religião

Buscar religião...



+ Adicionar

Nome	Ações
<input type="checkbox"/> Católico	
<input type="checkbox"/> Evangelico	
<input type="checkbox"/> Ateu	
<input type="checkbox"/> Espirita	
<input type="checkbox"/> Testemunha de Jeová	
Linhas selecionadas: 0	
Página 1 de 1	

Durante a Sprint 07, foi desenvolvida e validada a tela de **Cadastro de Religião**, uma interface voltada ao gerenciamento de registros de religiões no sistema. O objetivo principal desta entrega foi permitir o cadastro, edição, exclusão e consulta dessas informações, garantindo a aplicação das regras de negócio estabelecidas e proporcionando uma experiência de uso simples e eficiente.

A imagem anexada ilustra a tela implementada, que apresenta uma listagem organizada das religiões cadastradas, acompanhada de campo de busca para facilitar a localização de registros, além de botões de ação para edição e exclusão. O botão de adição de novas religiões está destacado no canto superior direito, priorizando a acessibilidade e a usabilidade. Foram aplicadas validações no campo Nome para garantir integridade dos dados, impedir duplicidade de registros e bloquear exclusões quando houver vínculo com a entidade Resident, exibindo mensagens informativas ao usuário nesses casos.

Principais características da tela produzida:

- **Listagem organizada de dados:** A tabela exibe de forma clara as religiões cadastradas, priorizando a leitura e a navegação intuitiva.
- **Funcionalidades CRUD:** A tela permite criar, editar e excluir registros, com confirmações em ações sensíveis e feedback visual ao usuário.
- **Campo de busca:** Implementação de funcionalidade de pesquisa para facilitar a localização de registros e melhorar a experiência de uso.
- **Validações avançadas:** Garantia de que o campo Nome seja obrigatório, tenha entre 3 e 100 caracteres, aceite apenas letras e espaços, e impeça duplicações.
- **Restrições de exclusão:** Bloqueio da remoção de religiões vinculadas a Resident, com mensagens claras ao usuário.
- **Interface responsiva:** A tela foi projetada para adaptação a diferentes dispositivos, mantendo a consistência visual.
- **Padronização visual:** O layout segue o padrão de identidade visual do sistema, mantendo coesão e usabilidade.

Importância da tela para o projeto:

A implementação da tela de **Cadastro de Religião** representa um avanço significativo no sistema, pois introduz regras de negócio fundamentais e reforça a integridade dos dados cadastrados. A funcionalidade integra-se diretamente ao backend, mesmo que parcialmente devido à ausência temporária da tabela **Resident**, e estabelece uma base sólida para futuras evoluções do sistema.

Esta entrega reforça boas práticas de desenvolvimento, como padronização de código, documentação clara e arquitetura modular, servindo como modelo para o desenvolvimento de novas funcionalidades administrativas. A tela agrega valor ao projeto ao oferecer aos usuários uma ferramenta confiável e funcional de gerenciamento de dados sensíveis, consolidando mais um passo importante na evolução da aplicação.

O que foi bem durante a sprint – Aprendizados

A Sprint 07 representou um avanço significativo no projeto, ao implementar as validações da entidade **Religion**, garantindo a integridade e consistência dos dados desde o frontend até a comunicação com o backend. Diferente das sprints iniciais, que priorizavam estruturação do ambiente, esta entrega trouxe regras de negócio concretas, aumentando a confiabilidade do sistema e aprimorando a experiência do usuário.



Pontos que se destacaram:

- **Validações completas no frontend:**
Foram aplicadas regras detalhadas para o campo **Name**, incluindo obrigatoriedade, restrição de tamanho (3 a 100 caracteres), bloqueio de nomes duplicados e limitação a apenas letras e espaços. Essas validações previnem dados incorretos ou inconsistentes na base, garantindo maior qualidade da informação desde o momento do cadastro ou edição.
- **Preparação para integridade relacional:**
Embora a tabela **Resident** ainda não esteja implementada, foi criado de forma provisória o service e o type correspondente. Isso prepara o sistema para impedir a exclusão de uma **Religion** caso haja residentes vinculados, garantindo a integridade referencial futura e evitando inconsistências no banco de dados.
- **Integração inicial com backend:**
A tela de Religion foi testada de ponta a ponta com o backend, garantindo que a validação, criação e edição de registros funcionem corretamente. Apesar da limitação atual quanto à exclusão de registros vinculados, os testes confirmaram o correto funcionamento do restante das regras.
- **Código modular e escalável:**
Foram criados e organizados **services** e **types** para a entidade **Religion**, mantendo padrões de código consistentes com a branch **main**. Essa abordagem facilita a manutenção futura e a adição de novas funcionalidades.
- **Documentação clara no código:**
Cada validação e regra implementada foi detalhadamente documentada com comentários, permitindo que outros desenvolvedores compreendam facilmente a lógica aplicada e possam dar continuidade ao desenvolvimento.
- **Testes abrangentes:**
Foram realizados testes unitários, de integração e práticos, garantindo o correto funcionamento das validações no frontend e a comunicação com o backend. Esse cuidado reforça a confiabilidade das entregas e diminui a possibilidade de falhas no uso real do sistema.

Conclusão dos aprendizados

A Sprint 07 consolidou a capacidade do time em entregar funcionalidades com regras de negócio completas e consistentes, reforçando a importância de validações detalhadas e integração inicial com o backend. A experiência destacou a necessidade de planejar antecipadamente a integridade referencial e o fluxo de feedback ao usuário, elevando a qualidade do sistema e prevenindo problemas futuros.

Essa sprint estabeleceu uma base sólida para as próximas etapas do projeto, permitindo que a exclusão de registros vinculados e outras funcionalidades mais complexas sejam implementadas de forma segura e escalável.

Funcionalidades não implementadas – Justificativas técnicas

Embora a Sprint 07 tenha cumprido integralmente sua proposta de implementar as validações da entidade **Religion**, algumas funcionalidades relacionadas à gestão completa de vínculos e à tabela **Resident** ainda não puderam ser contempladas nesta etapa.

A não implementação desses elementos não se deu por falhas operacionais, mas sim por limitações técnicas externas à sprint, que exigem a existência da tabela **Resident** e de uma coluna de chave estrangeira vinculada à **Religion** para que a exclusão de registros possa ser corretamente validada.

Funcionalidades não implementadas:

- **Exclusão com verificação de vínculo com Resident**
Justificativa: A tabela **Resident** ainda não foi criada, portanto, a lógica de bloqueio de exclusão de uma **Religion** vinculada não pode ser efetivamente testada. A implementação completa dependerá da criação da coluna `residentReligionId` como chave estrangeira para assegurar integridade relacional.
- **Validações de backend complementares para duplicidade**
Justificativa: Atualmente, o frontend impede a criação de nomes duplicados, mas o backend ainda precisará reforçar essa regra para cenários simultâneos ou de chamadas diretas à API, garantindo consistência em todas as camadas do sistema.
- **Integração de mensagens globais de erro ou alerta**
Justificativa: Embora mensagens de erro locais estejam funcionando corretamente, a implementação de notificações globais (toast/alert) foi adiada para alinhar com o design final da interface e padronizar feedbacks do sistema.

Conclusão

A Sprint 07 teve caráter técnico e funcional, entregando validações robustas no frontend e preparando o backend para futuras integrações com a tabela **Resident**. A decisão de adiar funcionalidades dependentes de estruturas ainda não implementadas foi estratégica, garantindo a estabilidade do sistema e permitindo que as próximas sprints adicionem essas regras de forma segura e escalável.

Com essa base consolidada, é possível garantir que os registros de **Religion** sejam consistentes, que os dados inseridos sejam válidos e que o usuário receba feedbacks claros e intuitivos, mesmo antes da implementação completa da verificação de vínculos.

Testes realizados

Foram conduzidos testes detalhados para validar o funcionamento das regras de negócio implementadas:

Testes de integração com backend:

- **Objetivo:** Verificar a comunicação correta entre frontend e backend durante criação, edição e listagem de registros de Religion.
- **Resultado:** Fluxo de cadastro e edição funcionando conforme especificado; retorno de mensagens de erro consistente.

Testes de validação de campos:

- **Objetivo:** Garantir que o campo **Name** respeite obrigatoriedade, limite de caracteres e formato permitido (apenas letras e espaços), além de impedir duplicidade no frontend.
- **Resultado:** Validações acionadas corretamente, exibindo mensagens claras ao usuário.

Testes de exclusão com vínculo (provisório):

- **Objetivo:** Preparar a funcionalidade de bloqueio de exclusão caso exista um **Resident** vinculado.
- **Resultado:** Serviço e type do Resident criados provisoriamente; bloqueio real só será possível após a criação da tabela Resident.

Testes práticos e exploratórios:

- **Objetivo:** Avaliar experiência do usuário e interface em diferentes resoluções.
- **Resultado:** Layout responsivo, ações intuitivas e feedbacks claros; nenhuma inconsistência crítica identificada.

Tipos de testes executados:

1. Testes de execução do ambiente local

- **Objetivo:** Confirmar inicialização correta do frontend e backend.
- **Resultado obtido:** Sucesso na execução do servidor e carregamento da aplicação sem erros.

2. Testes de renderização de componente

- **Objetivo:** Validar que componentes de formulário e mensagens de erro renderizam corretamente.
- **Resultado obtido:** Componentes exibidos conforme esperado, com layout limpo e legível.

3. Testes de responsividade

- **Objetivo:** Analisar comportamento da interface em mobile, tablet e desktop.
- **Resultado obtido:** Layout adaptável e consistente, garantindo boa experiência em diferentes dispositivos.

4. Testes práticos de fluxo de criação e edição

- **Objetivo:** Avaliar experiência do usuário e fluxo de validações.
- **Resultado obtido:** Operações de cadastro e edição funcionando corretamente; feedbacks claros exibidos em tempo real.

Observações:

Testes automatizados ainda não foram aplicados, pois a lógica de negócio depende de tabelas e integrações não totalmente implementadas.

A Sprint estabeleceu uma base sólida de validações, garantindo que futuras sprints possam adicionar regras de exclusão e backend de forma segura e escalável.

Assinaturas

Kétlyn Christini Tonholi de Moraes
Orientador: Jefferson Antonio Ribeiro Passerini