

Evolução do Projeto PetShop sob a Metodologia RUP

Projeto: Sistema de Gerenciamento de PetShop (Frontend React e Backend Serverless AWS) **Metodologia:** Rational Unified Process (RUP) - Adaptado **Período Principal de Desenvolvimento (Conforme Plano de Projeto):** 23/05/2025 - 27/05/2025

1. Introdução

Este documento descreve a evolução do projeto de desenvolvimento do Sistema de PetShop, desde sua concepção inicial até a fase de construção, seguindo os princípios da metodologia Rational Unified Process (RUP). Dado o cronograma intensivo do projeto, as fases e disciplinas do RUP foram aplicadas de forma adaptada e iterativa, com foco na entrega de valor e na mitigação de riscos. O objetivo é rastrear o progresso, os artefatos gerados e as principais decisões tomadas em cada fase, refletindo o trabalho realizado pela equipe e o suporte no desenvolvimento do backend.

2. Fase de Concepção (Inception)

Período Estimado (conforme Plano de Projeto): 23/05/2025

Objetivos Principais (RUP):

- Definir o escopo do projeto e seus limites.
- Identificar os principais stakeholders e seus interesses.
- Realizar uma análise de viabilidade inicial.
- Estabelecer os objetivos de alto nível do sistema.
- Identificar riscos críticos iniciais.

Artefatos Gerados e Atividades Realizadas:

- **Documento de Visão:**
 - **Referência:** Seções "Propósito", "Escopo" e "Stakeholders" do arquivo [Requisitos - Sistema Petshop.pdf](#).

- **Evolução:** Foi estabelecido que o sistema visa automatizar processos de agendamento, cadastro de clientes/pets e gerenciamento de relatórios. O escopo inicial incluiu funcionalidades de cadastro (clientes, pets, funcionários), agendamento e relatórios. Os stakeholders (Dono do Petshop, Funcionário, Cliente Final, Desenvolvedores) e seus interesses foram identificados.
- **Plano de Projeto Inicial:**
 - **Referência:** [Plano_Projeto_Petshop.pdf](#).
 - **Evolução:** Um cronograma detalhado foi estabelecido, com datas específicas para cada etapa (Levantamento e Análise: 23/05, Modelagem e Arquitetura: 24/05, Desenvolvimento Frontend/Backend: 25-26/05, Testes: 26/05 noite, Entrega: 27/05). A equipe foi definida, os recursos tecnológicos (React, AWS Lambda, API Gateway, DynamoDB, GitHub) foram escolhidos, e uma estimativa inicial de carga de trabalho foi feita.
- **Lista de Riscos Iniciais:**
 - **Referência:** Seção "Risco do Escopo Funcional" no [Plano_Projeto_Petshop.pdf](#).
 - **Evolução:** Foram identificados riscos como complexidade na configuração AWS, problemas de integração frontend-backend e erros de configuração do banco de dados.
- **Requisitos Iniciais de Alto Nível:**
 - **Referência:** Seção "Escopo" do [Requisitos - Sistema Petshop.pdf](#).
 - **Evolução:** Os requisitos funcionais macro (cadastros, agendamento, relatórios) e os papéis de usuário (Administrador, Funcionário) foram definidos.

Conclusão da Fase de Concepção: A fase de Concepção estabeleceu uma visão clara do projeto, seu escopo, os principais envolvidos e um plano inicial para sua execução, reconhecendo os desafios tecnológicos e de cronograma.

3. Fase de Elaboração (Elaboration)

Período Estimado (conforme Plano de Projeto): 24/05/2025 (para Modelagem e Arquitetura, mas se estende com o detalhamento de requisitos)

Objetivos Principais (RUP):

- Analisar detalhadamente o domínio do problema e os requisitos.
- Definir uma arquitetura de linha de base estável.
- Mitigar os riscos técnicos mais significativos.
- Refinar o plano do projeto.

Artefatos Gerados/Refinados e Atividades Realizadas:

- **Modelo de Casos de Uso Detalhado:**
 - **Referência:** Descrições textuais dos Requisitos Funcionais (RF 001 a RF 014) no [Requisitos - Sistema Petshop.pdf](#) e os diagramas gráficos [Diagrama-caso-de-uso-administrador.jpg](#), [Diagrama-caso-de-uso-funcionario.jpg](#), [Diagrama-caso-de-uso-funcionario-2.jpg](#), [Diagrama-caso-de-uso-funcionario-3.jpg](#).
 - **Evolução:** Os requisitos funcionais foram detalhados em fluxos principais e secundários, especificando as interações dos atores (Administrador, Funcionário) com o sistema para cada funcionalidade. Os diagramas visuais complementam a compreensão dessas interações.
- **Modelo de Domínio/Classes:**
 - **Referência:** [diagrama_v2.png](#) (Diagrama de Classes).
 - **Evolução:** As principais entidades do sistema (Usuario, Administrador, Funcionario, Cliente, Pet, Servico, Agendamento) e seus relacionamentos foram modelados, servindo como base para o design do banco de dados e da API.
- **Definição da Arquitetura do Backend:**
 - **Evolução:** Foi definida uma arquitetura serverless na AWS, compreendendo:
 - **Amazon API Gateway:** Para expor os endpoints RESTful.

- **AWS Lambda:** Para a lógica de negócio de cada entidade (inicialmente uma Lambda por entidade/recurso principal).
- **Amazon DynamoDB:** Como banco de dados NoSQL, com definição de tabelas, chaves primárias (ID UUID String) e Índices Secundários Globais (GSIs) para otimizar consultas específicas (ex: pets por cliente, agendamentos por diversos critérios).
- **Protótipos Executáveis (Backend):**
 - **Evolução:** O desenvolvimento iterativo dos endpoints da API para Usuários, Serviços, Clientes, Pets e Agendamentos serviu como protótipo para validar a arquitetura do backend, a interação com o DynamoDB e a estrutura das requisições/respostas. Testes foram realizados via Postman.
- **Requisitos Não Funcionais Detalhados:**
 - **Referência:** Seção "Requisitos Não Funcionais" (RNF 001 a RNF 003) no [Requisitos - Sistema Petshop.pdf](#).
 - **Evolução:** Foram especificados requisitos de segurança (criptografia), documentação (GitHub) e usabilidade/acessibilidade (responsividade).
- **Mitigação de Riscos:**
 - **Evolução:** O risco de "Complexidade na configuração dos serviços AWS" foi abordado através do desenvolvimento incremental e assistido do backend, validando cada componente. A decisão de usar DynamoDB (NoSQL) e a modelagem cuidadosa dos GSIs visaram otimizar o desempenho e a escalabilidade (RNF relacionado).

Conclusão da Fase de Elaboração: A fase de Elaboração solidificou o entendimento dos requisitos e estabeleceu uma arquitetura de backend viável. A prototipagem dos endpoints da API ajudou a mitigar riscos técnicos e a validar as escolhas de design do banco de dados. O foco foi em criar uma base estável para a fase de Construção.

4. Fase de Construção (Construction)

Período Estimado (conforme Plano de Projeto): 25/05/2025 - 26/05/2025 (Desenvolvimento) e 26/05/2025 noite (Integração e Testes)

Objetivos Principais (RUP):

- Desenvolver iterativamente o software completo.
- Minimizar os custos de desenvolvimento otimizando recursos.
- Alcançar qualidade adequada o mais rápido possível.

Artefatos Gerados/Em Andamento e Atividades Realizadas:

- **Código-fonte (Backend):**

- **Evolução:** Foram implementadas as funções Lambda em Python para gerenciar as entidades:
 - `gerenciarUsuarios`
 - `gerenciarServicos`
 - `gerenciarClientes`
 - `gerenciarPets` (incluindo lógica para `clienteId` e GSI)
 - `gerenciarAgendamentos` (incluindo lógica para múltiplas referências de IDs, `dataHora` ISO 8601, `dataAgendamento` derivada e múltiplos GSIs para consultas complexas).
 - Foram resolvidos desafios como a serialização de `Decimal` e a correta configuração de variáveis de ambiente e GSIs.

- **Código-fonte (Frontend):**

- **Evolução (Conforme Plano):** Desenvolvimento das interfaces em React para cadastro, agendamento e relatórios. (Detalhes específicos desta parte não foram o foco da nossa interação, mas são parte crucial do projeto).

- **Plano e Casos de Teste:**

- **Referência:** `Plano e Casos de Testes.pdf`.
- **Evolução:** O plano definiu a estratégia e casos de teste funcionais para diversas funcionalidades.

- **Relatórios de Teste:**

- **Referência:** `Relatório de Testes.pdf`.
- **Evolução:** Os resultados dos testes foram documentados, indicando sucessos, falhas e observações importantes (ex: "Sistema não aceita

valores negativos", "Verificação de e-mail duplicado em falta"). Este feedback é crucial para iterações de correção e melhoria.

- **Documentação da API (Backend):**

- **Evolução:** Para cada conjunto de endpoints do backend (Usuários, Serviços, Clientes, Pets, Agendamentos), foi gerada documentação detalhada em formato Markdown, especificando URLs, métodos, corpos de requisição/resposta, parâmetros e exemplos.

- **Integração e Testes (Backend):**

- **Evolução:** Os endpoints da API foram configurados no API Gateway e testados extensivamente via Postman, validando a integração entre API Gateway, Lambda e DynamoDB para cada operação CRUD.

Conclusão da Fase de Construção: A fase de Construção resultou na implementação da lógica principal do backend, com APIs funcionais para todas as entidades centrais do sistema. A documentação da API foi criada para facilitar a integração com o frontend e o entendimento futuro. Os testes iniciais forneceram feedback valioso. O desenvolvimento do frontend ocorreu em paralelo.

5. Fase de Transição (Transition) - Perspectiva

Período Estimado (conforme Plano de Projeto): Entrega final em 27/05/2025

Objetivos Principais (RUP):

- Entregar o sistema aos usuários finais.
- Realizar treinamento e garantir que os usuários possam usar o sistema.
- Coletar feedback final.

Evolução (Projetada): Dado o contexto acadêmico e o cronograma, esta fase se concentrará na entrega do projeto funcional, incluindo o frontend React integrado ao backend AWS, e toda a documentação gerada. A "implantação" é o deploy na AWS (backend) e a disponibilização do frontend (ex: Vercel, Netlify, Amplify). O "treinamento" pode ser substituído por uma boa documentação (incluindo o README do repositório GitHub) e uma apresentação clara do sistema.

6. Conclusão Geral

O projeto PetShop demonstrou uma progressão notável através das fases iniciais e intermediárias do RUP, adaptando a metodologia a um cronograma desafiador. A produção consistente de artefatos de documentação, alinhada com o desenvolvimento iterativo do backend e (presumivelmente) do frontend, reflete uma aplicação prática dos princípios do RUP. Os principais marcos, como a definição da arquitetura, a implementação dos CRUDs essenciais e a realização de testes iniciais, foram alcançados, pavimentando o caminho para a entrega final do sistema.