CENTRO UNIVERSITÁRIO SENAC

ANÁLISE E DESENVOLVIMENTO DE SISTEMA
PROJETO INTEGRADOR: DESENVOLVIMENTO DE SISTEMAS ORIENTADO A OBJETOS

PROTÓTIPO DE SOFTWARE PARA ORIENTAÇÃO FORMALIZADA COM A LINGUAGEM UNIFICADA DE MODELAGEM (UML)

ALEXSANDRA OLIVEIRA DE JESUS
ANA KAROLINI BLOEMER NOTH
GUSTAVO FERNANDES COSTA
KAUE OSMAR BORNHOFEN
MARCELO DE OLIVEIRA SILVA
TACIO DE PAULO CAVALCANTI FARIAS
THALITA PEREIRA ALFONSO

OSASCO, SÃO PAULO

2024

CENTRO UNIVERSITÁRIO SENAC

ANÁLISE E DESENVOLVIMENTO DE SISTEMA
PROJETO INTEGRADOR: DESENVOLVIMENTO DE SISTEMAS ORIENTADO A OBJETOS

PROTÓTIPO DE SOFTWARE PARA ORIENTAÇÃO FORMALIZADA COM A LINGUAGEM UNIFICADA DE MODELAGEM (UML)

Trabalho de Projeto Integrador, desenvolvido como exigência para a obtenção de nota parcial para o 3º semestre do curso de Análise e Desenvolvimento de Sistemas – Centro Universitário SENAC,

OSASCO, SÃO PAULO

2024

SUMÁRIO

1 INTRODUÇÃO

No contexto contemporâneo do desenvolvimento de software, a formalização e a visualização dos requisitos, arquitetura e funcionamento de sistemas tornaram-se cruciais para o sucesso de projetos. Nesse cenário, a Linguagem Unificada de Modelagem (UML) emerge como uma ferramenta fundamental, oferecendo uma notação padronizada para a representação gráfica e textual de sistemas orientados a objetos.

Este protótipo de software foi concebido com o propósito de facilitar e aprimorar o processo de orientação formalizada utilizando a UML. Combinando a flexibilidade e a praticidade das tecnologias contemporâneas, o protótipo visa fornecer uma plataforma intuitiva e eficiente para a criação, edição e análise de modelos UML.

Este documento apresentará uma visão geral do protótipo, delineando suas principais funcionalidades, abordagens de design e potenciais benefícios para os profissionais envolvidos no ciclo de vida do desenvolvimento de software.

Ao explorar este protótipo, os usuários poderão experimentar uma abordagem simplificada e integrada para a criação de diagramas UML, promovendo uma comunicação clara e consistente entre as equipes de desenvolvimento, analistas de negócios e stakeholders. Além disso, a plataforma busca fornecer recursos avançados de análise e validação, contribuindo para a detecção precoce de inconsistências e erros de projeto.

Por meio deste protótipo, espera-se que os usuários possam otimizar o processo de modelagem UML, reduzindo o tempo gasto na documentação e na compreensão dos requisitos do sistema. Ademais, a integração com ferramentas de desenvolvimento de software existentes possibilitará uma transição suave do design conceitual para a implementação prática, promovendo a coesão e a eficiência em todo o ciclo de vida do desenvolvimento de software.

Ao final desta apresentação, os usuários serão convidados a explorar o protótipo de software, fornecendo feedback valioso para aprimorar ainda mais suas funcionalidades e usabilidade. Com a colaboração contínua dos usuários e o compromisso com a excelência técnica, este protótipo tem o potencial de se tornar uma ferramenta indispensável para profissionais envolvidos na criação e no gerenciamento de sistemas de software complexos.

2 VISÃO GERAL DA PROPOSTA

2.1 Contextualização e motivação:

A crescente complexidade dos sistemas de software modernos demanda uma abordagem rigorosa e estruturada para sua concepção e desenvolvimento. Nesse contexto, a Linguagem Unificada de Modelagem (UML) tem sido amplamente reconhecida como uma ferramenta vital para a representação visual e formal de sistemas de software.

A motivação por trás deste protótipo de software reside na necessidade de preencher essa lacuna, oferecendo uma solução abrangente e amigável para a orientação formalizada com a UML. O objetivo é fornecer uma plataforma que simplifique o processo de modelagem UML, promovendo uma compreensão mais clara e precisa dos requisitos e da arquitetura do sistema.

2.2 Objetivos:

Os objetivos principais deste protótipo de software são os seguintes:

1. Desenvolver uma interface de usuário intuitiva e amigável que permita aos usuários criar, editar e visualizar facilmente diagramas UML de diversos tipos, como diagramas de classe, diagramas de sequência e diagramas de atividade.

2. Implementar funcionalidades avançadas de validação e verificação para garantir a consistência e a precisão dos modelos UML, identificando automaticamente possíveis erros, inconsistências ou violações das regras da UML.

3. Integrar o protótipo com outras ferramentas e ambientes de desenvolvimento de software, facilitando a transição do design conceitual para a implementação prática e garantindo a sincronização entre os modelos UML e o código-fonte correspondente.

4. Fornecer recursos de colaboração em equipe que permitam a múltiplos usuários trabalhar simultaneamente em um modelo UML, visualizar alterações em tempo real e registrar comentários e discussões relacionadas ao design do sistema.

5. Oferecer capacidades de exportação e importação para facilitar a integração com outras ferramentas e permitir a reutilização de modelos UML em diferentes contextos e plataformas.

Ao alcançar esses objetivos, o protótipo de software pretende se tornar uma ferramenta indispensável para profissionais envolvidos no desenvolvimento de software, fornecendo uma abordagem eficaz e eficiente para a orientação formalizada com a Linguagem Unificada de Modelagem (UML).

3 PLANEJAMENTO PARA O DESENVOLVIMENTO DA SOLUÇÃO PROPOSTA

3.1 Ciclo de vida de desenvolvimento, com a devida justificativa baseada na visão geral da solução proposta.

O ciclo de vida de desenvolvimento escolhido para o Gerenciamento de Bibliotecas será através do modelo iterativo e incremental. Esse modelo foi selecionado devido à sua flexibilidade para lidar com mudanças de requisitos ao longo do tempo e à sua capacidade de fornecer entregas incrementais que podem ser rapidamente avaliadas e adaptadas.
O modelo iterativo e incremental permitirá que a equipe de desenvolvimento entregue funcionalidades essenciais do sistema em ciclos curtos de desenvolvimento. Isso possibilitará que a solução seja continuamente

aprimorada com base no feedback do usuário, garantindo maior alinhamento com as necessidades reais dos usuários da biblioteca.

3.2 Premissas.

A equipe de desenvolvimento terá acesso aos recursos necessários, como hardware, software e ambiente de desenvolvimento, os stakeholders estarão disponíveis para fornecer feedback e aprovar as entregas do sistema, o orçamento e o prazo para o desenvolvimento do projeto serão respeitados, as tecnologias e ferramentas selecionadas para o desenvolvimento da solução serão adequadas para atender aos requisitos do sistema de gerenciamento de biblioteca.

3.3 Requisitos (descrição textual e classificação) / Histórias do usuário.

Requisito 1: Cadastro de Livros

- Descrição: O sistema deve permitir o cadastro de novos livros, incluindo título, autor, editora e categoria.
- Classificação: Funcionalidade Essencial
- História de usuário:
- Como um bibliotecário ou administrador do sistema,
- Eu gostaria de poder cadastrar novos livros na biblioteca, para que seja possível manter um registro atualizado do acervo e facilitar o acesso dos usuários às informações sobre os livros disponíveis.

Requisito 2: Empréstimo de Livros

- Descrição: Os usuários devem ser capazes de realizar empréstimos de livros, com registro da data de empréstimo e data de devolução.
- Classificação: Funcionalidade Essencial
- História do Usuário:
- Como um usuário da biblioteca,
- Eu gostaria de poder realizar empréstimos de livros, para que eu possa desfrutar dos recursos disponíveis na biblioteca e ampliar meu conhecimento através da leitura.

Requisito 3: Reserva de Livros

- Descrição: Os usuários devem poder reservar livros que estão atualmente emprestados por outros usuários.
- Classificação: Funcionalidade Desejável
- História do Usuário:
- Como um usuário interessado em um livro específico,
- Eu gostaria de poder reservar um livro que está atualmente emprestado por outro usuário, para que eu possa garantir que o livro estará disponível para mim assim que for devolvido e não o perder para outro usuário.

Requisito 4: Relatórios de Desempenho

- Descrição: O sistema deve fornecer relatórios sobre o desempenho da biblioteca, incluindo estatísticas de empréstimos, devoluções e estoque de livros.
- Classificação: Funcionalidade Opcional
- História do Usuário:
- Como um bibliotecário ou administrador da biblioteca,
- Eu gostaria de ter acesso a relatórios detalhados sobre o desempenho da biblioteca, para que eu possa monitorar o fluxo de empréstimos, avaliar a demanda por diferentes livros e tomar decisões informadas para melhorar a gestão do acervo.

Como parte do planejamento, temos:

As etapas:

Levantamento de Requisitos: Licitação de requisitos detalhados do sistema com base nas necessidades dos usuários e stakeholders.
Projeto e Design: Elaboração da arquitetura do sistema e design de interfaces de usuário.
Implementação: Desenvolvimento das funcionalidades do sistema de acordo com os requisitos levantados.
Testes: Realização de testes funcionais, de usabilidade e de desempenho para garantir a qualidade do sistema.

Entrega Incremental: Implementação de ciclos de desenvolvimento iterativos, entregando funcionalidades essenciais em incrementos sucessivos.

As tarefas:

Condução de entrevistas com os stakeholders para entender as necessidades e requisitos do sistema.
Elaboração de documentos de requisitos e histórias do usuário.
Desenvolvimento de protótipos para validar conceitos de design e usabilidade.
Implementação das funcionalidades do sistema, seguindo as práticas recomendadas de codificação.
Realização de testes de unidade, integração e aceitação para garantir a qualidade do software.

Ciclo de Desenvolvimento:

Os ciclos de desenvolvimento serão planejados com duração de duas semanas, com entrega de incrementos funcionais ao final de cada ciclo. Após cada ciclo, haverá a review com os stakeholders para avaliar o progresso do projeto e identificar possíveis ajustes no planejamento.

4 PROTÓTIPO FUNCIONAL E EXPERIMENTOS DE USABILIDADE

4.1 Protótipo funcional com interface humano-computador desejada, justificando o devido fluxo, elementos visuais e outros recursos á visão geral da solução proposta

Um protótipo funcional de um software de gerenciamento de biblioteca pode incluir vários módulos e funcionalidades. Aqui estão alguns dos principais componentes que você pode considerar:

1 Catalogação e Classificação: O sistema deve facilitar a catalogação e classificação eficientes dos recursos da biblioteca, incluindo livros, periódicos, multimídia e outros materiais.

2 Pesquisa e Recuperação: Uma interface de pesquisa amigável é crucial para permitir que os usuários encontrem e acessem recursos rapidamente.

3 Gerenciamento de Circulação: O sistema deve lidar com a circulação de materiais da biblioteca, incluindo check-in, checkout, renovações e reservas.

4 Gerenciamento de Usuários: As contas e perfis dos usuários devem ser gerenciados de forma eficaz, permitindo que a equipe da biblioteca registre novos usuários, atualize informações e mantenha registros do histórico de empréstimos.

5 Aquisição e Gerenciamento de Orçamento: O sistema deve auxiliar no processo de aquisição, gerenciando pedidos de compra, faturas e informações do fornecedor.

6 Gerenciamento de Recursos Digitais: Com a proliferação de conteúdo digital, o sistema deve gerenciar recursos eletrônicos, como e-books, e-jornais e bancos de dados.

7 Relatórios e Análises: Recursos de relatórios abrangentes são essenciais para analisar o uso da biblioteca, tendências de coleta e comportamento do usuário, auxiliando na tomada de decisões baseada em dados.

Esses são apenas alguns exemplos de funcionalidades que um software de gerenciamento de biblioteca pode ter. A implementação específica pode variar dependendo das necessidades específicas da biblioteca. Além disso, o desenvolvimento de um protótipo funcional é um processo iterativo que envolve várias fases, como formação de equipe, seleção de tópicos, coleta de requisitos, codificação ou implementação, teste, apresentação do projeto e redação de um artigo de pesquisa.

A prototipação de software é uma técnica de desenvolvimento para criar versões incompletas de um sistema, onde o objetivo é validar uma ideia ou demonstrar uma funcionalidade. No contexto de um software de gerenciamento de biblioteca, a prototipação pode ajudar a demonstrar e validar o fluxo do usuário, os elementos visuais e outros recursos antes da implementação completa.

Protótipos de Alta Fidelidade: São representações muito próximas do produto, incluindo interatividade e detalhes finos do design. Eles são úteis para testar a experiência do usuário em um ambiente que se assemelha ao produto.
Para um software de gerenciamento de biblioteca, o protótipo pode incluir elementos como:
Página inicial: Com opções para pesquisar livros, visualizar livros em destaque, fazer login ou se registrar.
Página de resultados da pesquisa: Mostrando os livros que correspondem aos critérios de pesquisa do usuário.
Página de detalhes do livro: Com informações detalhadas sobre um livro específico, como título, autor, resumo, disponibilidade e opções para empréstimo.
Página de gerenciamento de conta: Permitindo que os usuários vejam e gerenciem seus empréstimos de livros.
Cada uma dessas páginas seria projetada com a experiência do usuário em mente, garantindo que o fluxo do usuário seja intuitivo e a interface seja fácil de usar4. A prototipação é uma parte crucial do processo de desenvolvimento, pois permite que a equipe de desenvolvimento e os stakeholders visualizem a solução proposta e forneçam feedback antes que o desenvolvimento completo seja realizado5.

A usabilidade em sistemas de gerenciamento de biblioteca é um aspecto crucial para garantir a eficácia e satisfação dos usuários. Vários estudos têm sido realizados nesse campo, avaliando a facilidade de uso e a eficiência de softwares específicos. Alguns exemplos incluem:

SIGA e SophiA:

Um estudo avaliou a usabilidade dos sistemas de gerenciamento de biblioteca SIGA e SophiA1. Os testes de usabilidade envolveram usuários experientes, intermediários e básicos, utilizando observação, entrevistas e realização de tarefas.

Esses sistemas são amplamente utilizados em bibliotecas universitárias, onde a rapidez e precisão nas ações de consulta, empréstimo e devolução de obras são essenciais para o bom funcionamento da biblioteca e satisfação dos usuários.

Sistema PERGAMUM:

Outro exemplo é o Sistema PERGAMUM, utilizado em várias bibliotecas2. Avaliações de usabilidade foram conduzidas por especialistas da área para verificar se o software atende às necessidades e expectativas dos bibliotecários e usuários.

A usabilidade é um atributo fundamental para garantir que o sistema seja eficaz, eficiente e agradável de usar.

Em resumo, a usabilidade desses softwares é crucial para otimizar a gestão de bibliotecas, proporcionando uma experiência positiva para os usuários e facilitando o acesso à informação.

4.2 Experimento de usabilidade que valide a interação e experiência de uso a ser alcançada

Um experimento de usabilidade para validar a interação e experiência de uso de um software de gerenciamento de biblioteca pode envolver várias etapas. Aqui estão algumas sugestões:

Identificação dos usuários: Identifique quem são os usuários do software. Eles podem ser bibliotecários, estudantes, professores, pesquisadores etc.

Definição de tarefas: Defina uma lista de tarefas que os usuários precisam realizar no software. Por exemplo, buscar um livro, reservar um livro, renovar um empréstimo etc.

Seleção de participantes: Selecione um grupo diversificado de participantes que representem seus usuários. Tente incluir pessoas com diferentes níveis de familiaridade com o software.

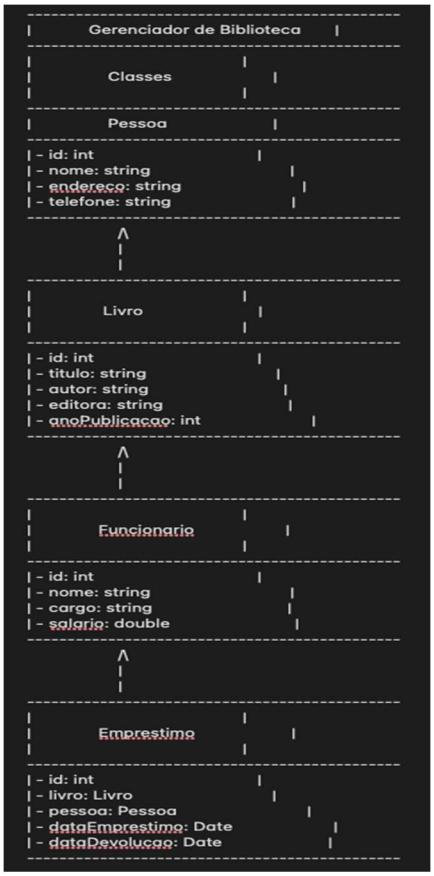
Realização de testes de usabilidade: Peça aos participantes que realizem as tarefas definidas enquanto observa e registra suas ações, comportamentos e feedbacks1. Você pode usar gravações das sessões de teste para identificar áreas de dificuldade dos usuários.

Análise dos resultados: Analise os resultados para identificar problemas de usabilidade. Calcule taxas de erro/acerto para cada tarefa realizada pelos usuários2. Priorize os problemas críticos identificados nos testes.

Iteração: Use os resultados para fazer melhorias no software e repita o processo até alcançar a experiência de usuário desejada.

5 MODELO DE PERSISTÊNCIA DE DADOS ORIENTADO A OBJETOS

5.1 Diagrama de Classes UML



5.2 Descrição das Classes:

1. Pessoa:
   - A classe base que contém informações comuns a todas as pessoas, como nome, endereço etc.
   - Métodos: getters e setters para os atributos.

2. Pessoa Física:
   - Subclasse de Pessoa que representa uma pessoa física.
   - Atributos adicionais: CPF, data de nascimento etc.
   - Métodos: getters e setters específicos.

3. Pessoa Jurídica:
   - Subclasse de Pessoa que representa uma pessoa jurídica.
   - Atributos adicionais: CNPJ, razão social etc.
   - Métodos: getters e setters específicos.

4. Professor:
   - Representa um professor na escola.
   - Atributos adicionais: formação acadêmica, disciplinas lecionadas etc.
   - Métodos: getters e setters específicos.

5. Fornecedor:
   - Representa um fornecedor de serviços ou produtos para a escola.
   - Atributos adicionais: tipo de serviço/produto fornecido, informações de contato etc.
   - Métodos: getters e setters específicos.

6. Aluno:
   - Representa um aluno matriculado na escola.
   - Atributos adicionais: número de matrícula, disciplinas matriculadas etc.
   - Métodos: getters e setters específicos.

Relacionamentos:

- Herança: As classes Pessoa Física e Pessoa Jurídica herdam da classe Pessoa, pois compartilham atributos e comportamentos comuns a todas as pessoas.
- Associação: A classe Aluno

6 REFERÊNCIAS

BIBLIOTECA PUC MINAS. **Desenvolvido pela Pontifícia Universidade Católica de Minas Gerais. Apresenta o Sistema de Bibliotecas da PUCMINAS e o sistema Pergamum**. Disponível em: Acesso em: 21 mar. 2005.

GUSMÃO, Alexandre Oliveira de Meira. **Avaliação da Qualidade e Determinantes de Desempenho do Aleph 500 em Bibliotecas Universitárias Brasileiras.** 2001. 213f. Dissertação (Mestrado em Ciência da Informação) ñ Universidade Federal da Paraíba, João Pessoa.

OLIVEIRA, Ulysses de. **Interação homem-máquina:** material básico. Disponível em: Acesso: em 29 abr. 2005.

PERGAMUM. **Sistema de gerenciamento de bibliotecas. Apresenta o Sistema Pergamum com todas as instituíeis que participam da Rede PERGAMUM**. Disponível em. Acesso: em 21 mar. 2005.

PRESSMAN, Roger S. **Engenharia de software.** Tradução Mônica Maria G. Travieso. 5. ed. Rio de Janeiro: McGraw Hill, 2002. 843p. ROBREDO, Jaime; CUNHA, Murilo Bastos da. Documentação.

USER EXPERIENCE E DESIGNER THINKING, Artigo: Entenda a Importância do teste de Usabilidade no Desenvolvimento de Produtos digitais, 2023.