
Documentação Projeto Web Site Experimento Intercâmbio

Desenvolvedor Full Star: Caio Moraes

Desenvolvedor Full Star: Nilson Filho

ÍNDICE DETALHADO

PREFÁCIO.....	4
1. INTRODUÇÃO AO DOCUMENTO	6
1.1. TEMA	6
1.2. OBJETIVO DO PROJETO.....	6
1.3. DELIMITAÇÃO DO PROBLEMA.....	6
1.4. JUSTIFICATIVA DA ESCOLHA DO TEMA.....	6
1.5. MÉTODO DE TRABALHO.....	6
1.6. ORGANIZAÇÃO DO TRABALHO	7
1.7. GLOSSÁRIO	7
2. DESCRIÇÃO GERAL DO SISTEMA.....	8
2.1. DESCRIÇÃO DO PROBLEMA.....	8
2.2. PRINCIPAIS ENVOLVIDOS E SUAS CARACTERÍSTICAS	8
2.3. REGRAS DE NEGÓCIO	8
3. REQUISITOS DO SISTEMA.....	9
3.1. REQUISITOS FUNCIONAIS	9
3.2. REQUISITOS NÃO-FUNCIONAIS.....	10
3.3. PROTÓTIPO	10
3.4. MÉTRICAS E CRONOGRAMA	11
4. ANÁLISE E DESIGN.....	12
4.1. ARQUITETURA DO SISTEMA.....	12
4.2. MODELO DO DOMÍNIO.....	12
4.3. DIAGRAMAS DE INTERAÇÃO.....	13
4.4. DIAGRAMA DE CLASSES	14
4.5. DIAGRAMA DE ATIVIDADES	14
4.6. DIAGRAMA DE ESTADOS.....	14
4.7. DIAGRAMA DE COMPONENTES	15
4.8. MODELO DE DADOS	16
4.8.1. <i>Modelo Lógico da Base de Dados</i>	<i>16</i>
4.8.2. <i>Criação Física do Modelo de Dados</i>	<i>16</i>
4.8.3. <i>Dicionário de Dados.....</i>	<i>16</i>
4.9. AMBIENTE DE DESENVOLVIMENTO.....	16
5. IMPLEMENTAÇÃO.....	17
5.1. LINGUAGEM DE PROGRAMAÇÃO	12
5.2. IMPEMENTAÇÃO DE CLASSES.....	12
5.3. CRONOGRAMA DE IMPLEMENTAÇÃO.....	12
6. TESTES	18
6.1. PLANO DE TESTES.....	18
6.2. EXECUÇÃO DO PLANO DE TESTES	18
7. IMPLANTAÇÃO.....	19
7.1. DIAGRAMA DE IMPLANTAÇÃO.....	19

7.2.	MANUAL DE IMPLANTAÇÃO.....	19
8.	MANUAL DO USUÁRIO	20
8.1.	NAVEGAÇÃO NO SITE	12
8.2.	AGENDAMENTO DE HORÁRIO	120
8.3.	CARRINHO DE COMPRAS.....	12
8.4.	CONSIDERAÇÕES FINAIS.....	12
9.	CONCLUSÕES E CONSIDERAÇÕES FINAIS	21
	BIBLIOGRAFIA	22
	GLOSSÁRIO	28

Prefácio

O objetivo deste documento é fornecer um roteiro para o desenvolvimento de sistemas de software utilizando os princípios da engenharia de software orientada a objetos com notação UML (*Unified Modeling Language*). É destinado a todos os alunos da Universidade Uninassau Olinda do curso de, Sistemas de Informação, apoiando a disciplina de Maturidade em Desenvolvimento de software.

Esta é a versão 3.0 do documento, totalmente revisada para utilizar a notação UML e modelos do RUP (*Rational Unified Process* – Processo Unificado *Rational*). Neste documento são citados alguns modelos do RUP que podem ser utilizados e consultados na ferramenta *Rational Unified Process* (que faz parte da ferramenta *Rational Suite Enterprise*) ou o site da IBM. A escolha da orientação a objetos é devido à tendência de mercado.

1. Introdução ao Documento

Este capítulo tem como propósito introduzir o projeto, destacando seus objetivos, escopo e abordagem metodológica. Através deste documento, apresentaremos um projeto que visa a criação de um website para a loja física de intercâmbios culturais, Experimento Boa Viagem Intercâmbios. O site tem como finalidade facilitar a interação entre clientes e empreendedores, permitindo aos cliente explorar os serviços pela loja e estabelecer contato direto com um consultor de intercâmbio especializado. A escolha desse tema busca não apenas fornecer uma solução prática e funcional, mas também servir como um exemplo de desenvolvimento de website para fins educativos. Utilizando tecnologias como HTML5, CSS, JavaScript e SQLite, o projeto busca fornecer experiência de usuário e informativa.

O capítulo inclui a definição do tema do projeto, seus objetivos, a delimitação do problema que o projeto visa resolver, a justificativa por trás da escolha do tema e a metodologia que será empregada durante o desenvolvimento. Além disso, o capítulo também descreve a organização do restante do documento, delineando como os tópicos subsequentes serão abordados.

1.1. Tema

Site de uma loja física de intercâmbios culturais, com a venda de produtos.

1.2. Objetivo do Projeto

Interação via Web Site entre o cliente e o empreendedor, expandindo o conhecimento dos serviços e contato com o consultor de intercâmbio.

1.3. Delimitação do Problema

Site da loja Experimento Boa Viagem Intercâmbios, que realiza serviços de intercâmbios culturais para diversos países e com diversos serviços como Travel Plus, AU Pair, High School, onde o cliente terá acesso aos dados sobre a loja e seus serviços, podendo selecionar os programas desejados, incluindo-os em um carrinho e quando desejar finalizar a navegação será direcionado para um breve cadastro e após terá seus dados e produtos selecionados direto com o WhatsApp do consultor pessoal.

1.4. Justificativa da Escolha do Tema

Projeto ao qual será desenvolvido para a explanação do conhecimento em desenvolvimento de Web Site.

1.5. Método de Trabalho

Desenvolvimento utilizando a interface do Visual Code Studio, com as linguagens HTML5, CSS, JavaScript, SQLite como banco de dados.

1.6. Organização do Trabalho

O presente documento está estruturado de forma a fornecer uma compreensão abrangente do projeto de criação do website para a loja Experimento Boa Viagem Intercâmbios. As seções subsequentes estão organizadas da seguinte maneira:

- Descrição Geral do Sistema
- Requisitos do Sistema
- Análise e Design
- Implementação
- Testes
- Implantação
- Manual do Usuário
- Conclusões e Considerações Finais

1.7. Glossário

- Website: Página da web que exibe informações e interage com os usuários por meio de um navegador.
- Intercâmbios Culturais: Programas que permitem que indivíduos vivenciem experiências em outros países, promovendo a troca de culturas e conhecimento.
- Cliente: Pessoa interessada em utilizar os serviços oferecidos pelo website.
- Consultor de Intercâmbio: Profissional especializado responsável por auxiliar os clientes na escolha e organização de programas de intercâmbio.
- HTML5: Linguagem de marcação utilizada para estruturar o conteúdo de páginas da web.
- CSS: Linguagem de estilo utilizada para definir a apresentação visual de páginas da web.
- JavaScript: Linguagem de programação usada para adicionar interatividade e dinamismo a páginas da web.
- SQLite: Sistema de gerenciamento de banco de dados leve e incorporado, adequado para aplicações de tamanho moderado.

2. Descrição Geral do Sistema

2.1. Descrição do Problema

A loja "Experimento Boa Viagem Intercâmbios" oferece serviços de intercâmbio cultural para diversos países. O sistema visa proporcionar uma plataforma online para clientes explorarem esses serviços e se conectarem com consultores especializados.

2.2. Principais Envolvidos e suas Características

2.2.1. Usuários do Sistema

- **Cliente:** Interessado em programas de intercâmbio, deseja informações claras e opções de contato com consultores.
- **Consultor de Intercâmbio:** Especialista que fornecerá orientação e informações personalizadas aos clientes.

2.2.2. Desenvolvedores do Sistema

O projeto está sendo conduzido por dois desenvolvedores Full Stack altamente qualificados, que possuem habilidades abrangentes em todas as camadas do desenvolvimento de software. Eles desempenharão papéis críticos ao longo do ciclo de desenvolvimento, assumindo tanto as responsabilidades front-end quanto back-end. Suas funções específicas incluem:

- **Desenvolvedor Full Stack 1:** Analisar e compreender os requisitos funcionais e não funcionais do sistema. Projetar a interface do usuário de forma atraente e intuitiva, utilizando HTML, CSS e JavaScript. Implementar a lógica de negócios e funcionalidades do sistema no back-end. Garantir a interação eficaz com o banco de dados para armazenamento de dados. Realizar testes de unidade e integração para validar a funcionalidade do sistema.
- **Desenvolvedor Full Stack 2:** Colaborar na definição da arquitetura do sistema, garantindo a escalabilidade e eficiência. Implementar componentes de front-end, criando uma experiência de usuário coesa e responsiva. Desenvolver APIs e serviços do back-end para atender às necessidades do front-end. Trabalhar na integração de sistemas e componentes externos quando necessário. Realizar testes abrangentes para garantir a qualidade e a robustez do sistema.

2.3. Regras de Negócio

2.3.1 Restrições de Negócio

- Os usuários devem se cadastrar no sistema antes de realizar qualquer ação.
- Apenas usuários autenticados podem adicionar programas de intercâmbio ao carrinho de compras.
- O sistema não permitirá a conclusão de uma transação se não houver um consultor de intercâmbio disponível para contato.

2.3.2 Restrições de Desempenho

- O sistema deve fornecer um tempo de resposta de menos de 3 segundos para a exibição da lista de programas de intercâmbio.
- A resposta do consultor de intercâmbio às consultas dos clientes deve ocorrer em até 24 horas.

2.3.3 Tolerância à Falhas

- O sistema deve ser projetado para lidar com picos de tráfego e garantir a disponibilidade contínua.
- Em caso de falha do servidor, o sistema deve ser capaz de alternar para um servidor de backup.

2.3.4 Volume de Informação e Estimativa de Crescimento

- O sistema deve ser capaz de armazenar informações sobre pelo menos 1000 programas de intercâmbio inicialmente.
- Espera-se um aumento de 20% no volume de informações anualmente, portanto, o sistema deve ser escalável para acomodar esse crescimento.

3. Requisitos do Sistema

Neste capítulo, serão detalhados os requisitos do sistema, que são elementos essenciais para o funcionamento e desenvolvimento do nosso sistema. O sistema em questão foi projetado com o objetivo de proporcionar uma experiência interativa e eficaz aos usuários, oferecendo uma ampla gama de funcionalidades para atender às necessidades específicas de nossos clientes e colaboradores.

O sistema incorpora diversas características que o tornam uma ferramenta versátil e adaptável, proporcionando uma base sólida para o crescimento e aprimoramento contínuo. A seguir, serão apresentados os requisitos funcionais e não funcionais que definem o escopo do sistema, abrangendo desde as funcionalidades principais até as considerações de desempenho, segurança e usabilidade.

Os requisitos do sistema servirão como diretrizes fundamentais para o desenvolvimento, teste e implementação bem-sucedidos do sistema, garantindo que ele cumpra seu propósito de forma eficiente e confiável. A compreensão e o cumprimento desses requisitos são essenciais para o sucesso do projeto e a satisfação de nossos usuários finais.

3.1. Requisitos Funcionais

Requisitos Funcionais do Sistema - Site Experimento Boa Viagem

Abaixo estão descritos os requisitos funcionais do sistema para o site da Experimento Boa Viagem, que oferece serviços relacionados a viagens e turismo. Esses requisitos abrangem as principais funcionalidades e recursos que o sistema deve oferecer aos usuários.

1. Pesquisa de Serviços:

Os usuários podem pesquisar por serviços de intercâmbios.

2. Visualização da Descrição de Serviços

Os usuários podem visualizar detalhes de pacotes de serviços.

As informações devem ser apresentadas de maneira clara e atrativa.

3. Reservas de Serviços:

Os usuários podem selecionar pacotes de serviços.

O processo de reserva deve incluir opções de pagamento seguro.

4. Contato e Suporte:

Os usuários devem ter acesso a informações de contato para obter suporte e esclarecimento de dúvidas.

Um formulário de contato também pode ser disponibilizado para perguntas específicas.

5. Integração de Rede Sociais:

Os usuários podem compartilhar suas experiências de viagem nas redes sociais diretamente do site.

6. Sistema de Notificações:

O site pode enviar notificações por WhatsApp e/ou e-mail aos usuários, como confirmações de reserva e atualizações de viagem.

7. Filtro de Comentários

Os comentários dos usuários podem ser moderados para evitar conteúdo inadequado ou spam.

8. Gerenciamento de Conteúdo:

A equipe administrativa do site pode atualizar e gerenciar o conteúdo, incluindo pacotes de serviços, artigos de rede social e promoções.

9. Avaliações e Classificações:

Os usuários podem deixar avaliações e classificações para pacotes de viagem e serviços prestados.

Esses requisitos funcionais representam as principais funcionalidades do site da Experimento Boa Viagem, garantindo uma experiência completa e eficaz para seus usuários.

3.2. Requisitos Não-Funcionais

Os requisitos não funcionais são aspectos do sistema que não se relacionam diretamente com funcionalidades específicas, mas que afetam a qualidade, desempenho e usabilidade do site da Experimento Boa Viagem. Abaixo estão os principais requisitos não funcionais para o sistema:

1. Usabilidade:

O site deve ser intuitivo e fácil de usar para usuários de todas as idades.
Deve ser acessível em dispositivos móveis e desktop.

2. Desempenho:

O site deve carregar rapidamente, com tempos de resposta curtos para garantir uma experiência sem problemas.
Deve ser otimizado para minimizar o consumo de largura de banda.

3. Segurança:

As informações dos usuários, como dados pessoais e informações de pagamento, devem ser protegidas por criptografia.
O sistema deve ser protegido contra ameaças de segurança, como ataques de hackers.

4. Disponibilidade:

O site deve estar disponível 24 horas por dia, 7 dias por semana, com tempo de inatividade mínimo planejado para manutenção.

5. Escalabilidade:

O sistema deve ser escalável para lidar com picos de tráfego, especialmente durante promoções e períodos de alta demanda de reservas.

6. Compatibilidade do Navegador:

O site deve ser compatível com os principais navegadores da web, como Google Chrome, Mozilla Firefox, Safari e Microsoft Edge.

7. Responsividade:

O design do site deve ser responsivo, adaptando-se automaticamente a diferentes tamanhos de tela e dispositivos.

8. Backup e Recuperação de Dados:

Deve haver um sistema de backup regular para proteger os dados do usuário e permitir a recuperação em caso de falhas.

9. Política de Privacidade e Conformidade:

O site deve seguir rigorosamente as leis de proteção de dados e privacidade, fornecendo aos usuários informações claras sobre como seus dados são usados.

10. Velocidade de Processamento:

As transações de reserva devem ser processadas com rapidez e eficiência.

11. Manutenção e Atualizações:

O sistema deve ser facilmente mantido e atualizado pela equipe de administração do site.

12. Experiência do Usuário:

O design do site deve ser esteticamente agradável e oferecer uma experiência de usuário atraente.

13. Armazenamento de Dados:

Os dados do usuário devem ser armazenados de forma segura e eficiente em servidores confiáveis.

Estes requisitos não funcionais são fundamentais para garantir que o site da Experimento Boa Viagem ofereça uma experiência de alta qualidade, segurança e desempenho para seus usuários, independentemente de onde estejam ou dos dispositivos que utilizem.

3.3. Protótipo

Neste item deve ser apresentado o protótipo do sistema que consiste na interface preliminar contendo um subconjunto de funcionalidades e telas. O protótipo deve ser incrementalmente evoluído até a concordância completa dos requisitos previstos para o sistema, de comum acordo com o usuário. O protótipo é um recurso que deve ser adotado como estratégia para levantamento, detalhamento, validação de requisitos e modelagem de interface com o usuário (usabilidade).

As telas do sistema podem ser criadas na própria linguagem de desenvolvimento ou em qualquer outra ferramenta de desenho. Cada tela deve possuir uma descrição detalhada do seu funcionamento. Alguns itens importantes na descrição são:

- Objetivo da tela;
- De onde é chamada e que outras telas pode chamar;
- Regras:
 - ✓ Domínio (tamanho de campo, tipo de dados que aceita valor default);
 - ✓ Tipo de usuários que podem acessar;

-
- ✓ Lógica de negócio (campos obrigatórios, validade entre datas, preenchimento anterior de um campo para efetuar uma operação, etc).

A descrição detalhada das telas deve registrar informações que possam ser consultadas na implementação do sistema, facilitando, agilizando e minimizando erros de implementação e na execução de testes.

3.3.1. Diagrama de Navegação

Início: A página inicial apresenta uma introdução ao Experimento Intercâmbio Cultural, juntamente com uma imagem e um botão "Agendar um horário" que leva a uma página externa de agendamento via WhatsApp.

Sobre: A página "Sobre nós" descreve informações sobre a Experimento, incluindo sua história e missão.

Serviços: A página "Serviços" lista três programas oferecidos pela Experimento, cada um com um botão "Adicionar ao Carrinho" que permite que os usuários adicionem esses programas ao carrinho de compras.

Depoimentos: A página "Depoimentos" exibe depoimentos de clientes anteriores, apresentando várias citações e fotos de clientes satisfeitos.

Contato: A página "Contato" inclui informações de contato da Experimento, como número de telefone, endereço e endereço de e-mail. Também há um botão "Entrar em contato" que leva a uma página externa via WhatsApp.

Além disso, há um botão de carrinho de compras visível em todo o site que exibe um resumo do carrinho quando clicado. Os usuários podem adicionar programas ao carrinho de compras clicando nos botões "Adicionar ao Carrinho" nas páginas de serviços.

3.4. Métricas e Cronograma

Estimar os esforços necessários e elaborar um cronograma detalhado para o desenvolvimento de um sistema requer uma compreensão clara dos requisitos, escopo do projeto e da equipe disponível.

Estimativa de Esforço usando Pontos de Caso de Uso:

Uma abordagem comum para estimar o esforço do desenvolvimento de software é usar os Pontos de Caso de Uso (UCP - Use Case Points). Esta técnica envolve a avaliação da complexidade dos casos de uso e a alocação de pontos com base nessa complexidade. Os pontos de caso de uso são então convertidos em horas de trabalho com base na produtividade da equipe.

O sistema "Experimento Intercâmbio Cultural" tenha os seguintes casos de uso principais:

Início

Sobre

Serviços

Depoimentos

Contato

Atribuição de Pontos de Caso de Uso (UCP):

Cada Caso de Uso simples: 5 pontos

Cada Caso de Uso médio: 10 pontos

Cada Caso de Uso complexo: 15 pontos

Suponhamos que, após a análise, atribuímos os seguintes pontos aos casos de uso:

Início: 5 pontos (simples)

Sobre: 5 pontos (simples)

Serviços: 10 pontos (médio)
Depoimentos: 10 pontos (médio)
Contato: 5 pontos (simples)
Carrinho: 10 pontos (médio)
Total de Pontos de Caso de Uso: 45 pontos

Estimativa de Tempo:

A equipe de desenvolvimento possa entregar 20 horas de trabalho em um período de uma semana.
Total de Horas Estimadas: 20 horas/semana

Cronograma Detalhado:

Com base nas estimativas acima, podemos elaborar um cronograma aproximado para o desenvolvimento do sistema. Vamos supor que a equipe esteja disponível para o desenvolvimento em tempo integral.

Planejamento e Análise (1 semana):
Detalhamento dos Requisitos
Definição da Arquitetura
Preparação do Ambiente de Desenvolvimento

Desenvolvimento (2 semanas):
Implementação das Páginas (Home, About, Services, Testimonials, Contact)
Integração com APIs Externas (WhatsApp, Redes Sociais)

Testes e Depuração (1 semana):
Teste de Unidade
Teste de Integração
Depuração e Correções

Design e Estilização (1 semana):
Design Responsivo
Estilização das Páginas

Finalização e Preparação para Implantação (1 semana):
Revisão Geral
Preparação para Implantação

Implantação (1 semana):
Implantação do Site
Configuração do Servidor

Testes Finais e Correções (1 semana):
Testes de Aceitação
Correção de Problemas

Documentação e Treinamento (1 semana):
Documentação do Sistema
Treinamento para a Equipe de Suporte
Lançamento (1 dia):
Lançamento Oficial do Site

Total de Semanas Estimadas: 9 semanas

Observe que esta é uma estimativa aproximada e genérica, e o cronograma real pode variar dependendo de vários fatores, como a complexidade real do desenvolvimento, a disponibilidade da equipe e a necessidade de revisões adicionais.

4. Análise e Design

Este sistema de análise e design tem como objetivo oferecer uma visão detalhada e abrangente de um projeto de software denominado “Experimento Intercâmbio Cultural”. Este sistema visa atender a necessidades específicas de estudante e clientes interessados em explorar oportunidades de intercâmbio internacional oferecidas pelo Experimento.

4.1. Arquitetura do Sistema

A arquitetura do sistema "Experimento Intercâmbio Cultural" é projetada para fornecer uma experiência de usuário eficiente e escalável, permitindo que os usuários acessem informações e serviços relacionados a programas de intercâmbio cultural. A arquitetura é baseada em uma abordagem cliente/servidor de múltiplas camadas, que oferece flexibilidade e capacidade de expansão à medida que o sistema cresce.

Camada de Apresentação (Front-End):

Esta camada é responsável por interagir diretamente com o usuário final. Ela consiste em páginas da web acessadas pelos navegadores dos usuários. O front-end é construído usando tecnologias web modernas, como HTML5, CSS3 e JavaScript. Também inclui bibliotecas e frameworks, para criar uma interface de usuário interativa e responsiva.

Camada de Lógica de Aplicação (Back-End):

Implementada usando tecnologias Node.js para fornecer um servidor web robusto e escalável, esta camada é responsável por gerenciar a autenticação de usuários, a recuperação de dados de programas de intercâmbio e a manipulação das operações do carrinho de compras.

Configuração de Hardware e Rede:

A infraestrutura de hardware é baseada em servidores em nuvem, permitindo escalabilidade e alta disponibilidade. A escolha de provedores de nuvem, como Amazon Web Services (AWS).

Para garantir a disponibilidade contínua do sistema, será implementada uma estratégia de balanceamento de carga para distribuir o tráfego entre vários servidores.

A rede é configurada para suportar conexões seguras por meio do protocolo HTTPS, garantindo a proteção dos dados do usuário durante a transmissão.

Software Utilizado:

Sistema Operacional de Servidores: Linux (por exemplo, Ubuntu Server) para a camada de servidor.

Servidor Web: Node.js e Express.js para a camada de lógica de aplicação.

Banco de Dados: MongoDB como banco de dados NoSQL para armazenar informações relacionadas a programas, usuários e pedidos.

Frameworks e Bibliotecas: React.js e outras bibliotecas JavaScript para criar a interface do usuário.

Dimensionamento Mínimo de Conexões:

O dimensionamento mínimo de conexões será baseado na expectativa de tráfego do site. Como diretriz inicial, o sistema será projetado para suportar centenas de conexões simultâneas. À medida que o tráfego aumentar, a arquitetura escalável em nuvem permitirá a alocação dinâmica de recursos para atender à demanda crescente.

A arquitetura cliente/servidor de três camadas e a configuração em nuvem garantem que o sistema seja flexível, escalável e capaz de lidar com um número crescente de usuários, proporcionando uma experiência de usuário consistente e confiável. Além disso, a infraestrutura em nuvem facilita a manutenção, atualização e expansão contínua do sistema à medida que novos recursos são adicionados e os requisitos evoluem.

4.2. Modelo do Domínio

O modelo do domínio do sistema "Experimento Intercâmbio Cultural" representa um conjunto inicial de classes e seus relacionamentos para melhor compreender a estrutura conceitual do sistema. Esse modelo servirá como base para a criação do diagrama de classes final, que incluirá todos os detalhes de

atributos, métodos e associações entre as classes.

Classes do Modelo do Domínio:

Usuário:

Atributos: Nome, sobrenome, telefone.

Métodos: Registrar, Autenticar.

ProgramaIntercambio:

Atributos: Nome, descrição, destino, duração, preço, imagem.

Métodos: ConsultarDetalhes.

PedidoIntercambio:

Atributos: Data do pedido, status, itens (associação com ItemPedido).

Métodos: AdicionarItem, RemoverItem.

ItemPedido:

Atributos: Programa de intercâmbio (associação com ProgramaIntercambio), quantidade.

Métodos: CalcularSubtotal.

Relacionamentos:

Um Usuário pode fazer Zero ou Muitos PedidoIntercambio.

Um PedidoIntercambio pertence a um único Usuário.

Um PedidoIntercambio contém Zero ou Muitos ItemPedido.

Um ItemPedido pertence a um único PedidoIntercambio.

Um ItemPedido está associado a um único ProgramaIntercambio.

Um ProgramaIntercambio pode estar associado a Zero ou Muitos ItemPedido.

Este modelo de domínio inicial captura as principais entidades e associações do sistema, como usuários, programas de intercâmbio, pedidos e itens de pedido.

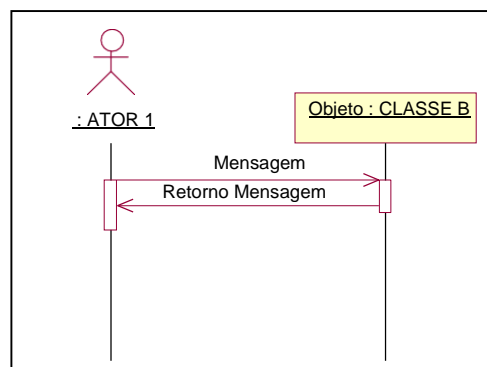
4.3. Diagramas de Interação

O diagrama de interação é composto pelos diagramas de seqüência e colaboração (comunicação, versão 2.0 UML) e modela os aspectos dinâmicos do sistema, mostrando a interação formada por um conjunto de objetos permitindo identificar mensagens que poderão ser enviadas entre eles.

4.3.1. Diagrama de Seqüência

Neste item devem ser apresentados os diagramas de seqüência essenciais ao sistema. Um diagrama de seqüência representa interações de objetos organizadas em uma seqüência temporal, apresentando os objetos que participam da interação e a seqüência das mensagens trocadas. O diagrama de seqüência deve validar o diagrama de classes e vice-versa.

A seguir é apresentada a notação básica de um diagrama de seqüência.



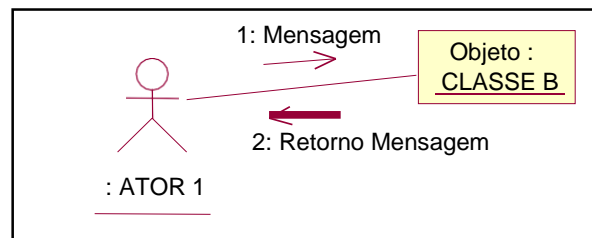
Notação básica do diagrama de seqüência.

4.3.2. Diagrama de Colaboração / Comunicação

Esse diagrama é uma alternativa para o diagrama de seqüência (item 4.3.1). Neste item devem ser apresentados os diagramas de colaboração/comunicação essenciais ao sistema. Um diagrama de colaboração descreve um padrão de interação entre objetos, apresentando os objetos que participam da interação bem como os seus links e mensagens trocadas.

Geralmente as ferramentas CASE geram automaticamente o diagrama de colaboração/comunicação a partir do diagrama de seqüência.

A seguir é apresentada a notação básica de um diagrama de colaboração/comunicação.



Notação básica do diagrama de colaboração.

4.4. Diagrama de Classes

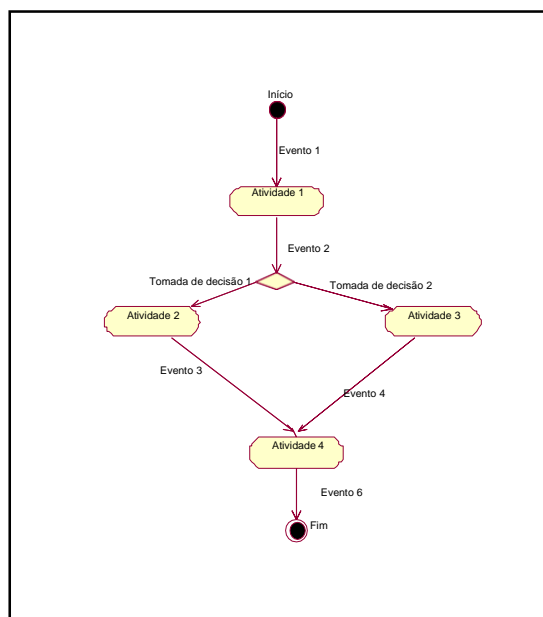
Neste item deve ser apresentado o diagrama de classes completo e validado.

4.5. Diagrama de Atividades

Neste item deve ser apresentado o diagrama de atividades, que representa o detalhamento de tarefas e o fluxo de uma atividade para outra de um sistema.

Nem todos os sistemas necessitam da elaboração do diagrama de atividades, pois nem todas as tarefas do sistema necessitam de um detalhamento. Com isso, deve-se analisar a real necessidade e no que este diagrama irá auxiliar na implementação do sistema, como: detalhamento de *workflow*, de métodos, entre outros.

A seguir é apresentada a notação básica de um diagrama de atividades.



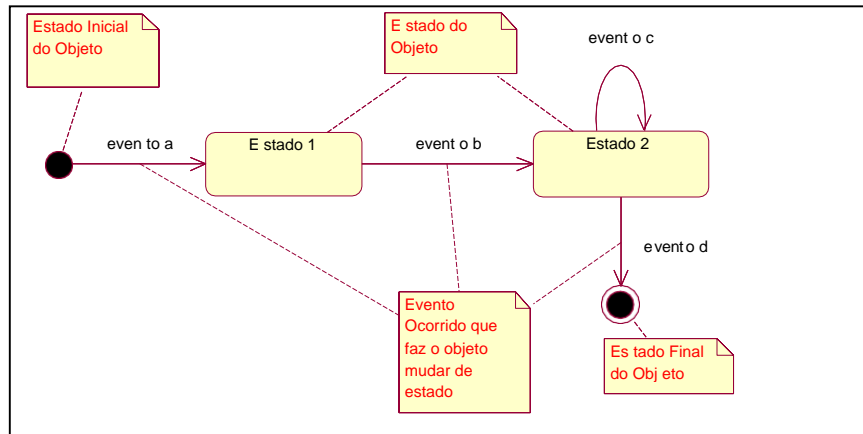
Notação básica do diagrama de atividades.

4.6. Diagrama de Estados

Neste item deve ser apresentado o diagrama de estados, que especifica as seqüências de estados pelas quais o objeto pode passar durante seu ciclo de vida em resposta a eventos.

Nem todas as classes necessitam da elaboração do diagrama de estados, pois nem todas as classes mudam muito de estado no seu ciclo de vida. Com isso, deve-se analisar a real necessidade desse diagrama para o desenvolvimento do sistema.

A seguir é apresentada a notação básica de um diagrama de estados.

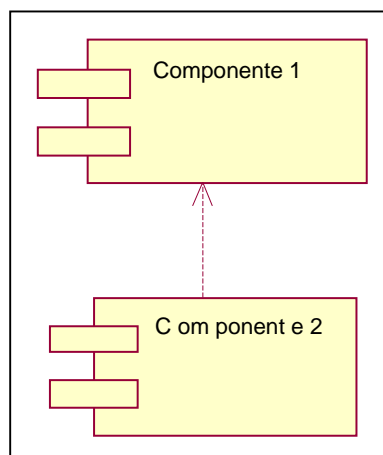


Notação básica do diagrama de estados.

4.7. Diagrama de Componentes

Neste item deve ser apresentado o diagrama de componentes que apresenta a organização e as dependências entre os componentes ⁶.

A seguir é apresentada a notação básica de um diagrama de componentes.



Notação básica do diagrama de componentes.

4.8. Modelo de Dados

4.8.1. Modelo Lógico da Base de Dados

O modelo lógico da base de dados para o sistema "Experimento Intercâmbio Cultural" é derivado do diagrama de classes e das entidades identificadas no modelo de domínio. Para representar o modelo lógico, usaremos o modelo entidade-relacionamento (MER) e aplicaremos as regras de normalização para garantir a integridade e eficiência dos dados.

4.8.2. Criação Física do Modelo de Dados

Tabela Usuário:

Atributos: ID (Chave Primária), nome, sobrenome, email, telefone, endereço.

Tabela ProgramaIntercambio

Atributos: ID (Chave Primária), nome, descrição, destino, duração, preço, imagem.

Tabela PedidoIntercambio

Atributos: ID (Chave Primária), data do pedido, status, ID do usuário (Chave Estrangeira).

Tabela ItemPedido:

Atributos: ID (Chave Primária), quantidade, ID do programa de intercâmbio (Chave Estrangeira), ID do pedido (Chave Estrangeira).

4.8.3. Dicionário de Dados

A seguir, apresentamos o dicionário de dados do banco de dados do sistema "Experimento Intercâmbio Cultural". Este dicionário inclui informações detalhadas sobre as tabelas, atributos e procedimentos armazenados que fazem parte do sistema.

Tabela: Usuário

ID:

Chave primária autoincrementada que identifica de forma única cada usuário.

Nome:

O primeiro nome do usuário.

Sobrenome:

O sobrenome do usuário.

Email

O endereço de e-mail do usuário, que também serve como identificador único.

Telefone:

O número de telefone de contato do usuário.

Endereço:

O endereço residencial do usuário.

Tabela: ProgramaIntercambio

ID:

Chave primária autoincrementada que identifica de forma única cada programa de intercâmbio.

Nome:

O nome do programa de intercâmbio.

Descrição:

Uma descrição detalhada do programa.

Destino

O destino ou país onde o programa será realizado.

Duração:

A duração do programa em semanas ou meses.

Preço:

O preço do programa de intercâmbio.

Imagem:

O caminho para a imagem associada ao programa.

Tabela: PedidoIntercambio

ID:

Chave primária autoincrementada que identifica de forma única cada pedido de intercâmbio.

Data do Pedido:

A data em que o pedido foi feito.

Status:

O status atual do pedido (por exemplo, "pendente", "aprovado", "concluído").

ID do Usuário:

Chave estrangeira que faz referência ao usuário que fez o pedido.

Tabela: ItemPedido

ID:

Chave primária autoincrementada que identifica de forma única cada item de pedido.

Quantidade:

A quantidade de programas de intercâmbio incluídos no item do pedido.

ID do Programa de Intercâmbio:

Chave estrangeira que faz referência ao programa de intercâmbio associado ao item do pedido.

ID do Pedido:

Chave estrangeira que faz referência ao pedido ao qual o item pertence.

Procedimentos Armazenados:

InserirUsuario:

Procedimento para inserir um novo usuário na tabela Usuário.

AtualizarUsuario:

Procedimento para atualizar os dados de um usuário existente na tabela Usuário.

BuscarUsuarioPorEmail:

Procedimento para buscar um usuário pelo seu endereço de e-mail.

InserirPedido:

Procedimento para inserir um novo pedido de intercâmbio na tabela PedidoIntercambio.

AtualizarStatusPedido:

Procedimento para atualizar o status de um pedido na tabela PedidoIntercambio.

InserirItemPedido:

Procedimento para inserir um novo item de pedido na tabela ItemPedido.

TotalPedido:

Função que calcula o valor total de um pedido com base nos itens do pedido.

Este dicionário de dados fornece uma documentação detalhada das estruturas de dados e procedimentos utilizados no banco de dados do sistema "Experimento Intercâmbio Cultural". Essas informações são essenciais para o desenvolvimento, manutenção e compreensão do sistema.

4.9. Ambiente de Desenvolvimento

O ambiente de desenvolvimento para o sistema "Experimento Intercâmbio Cultural" é composto por uma série de componentes, incluindo linguagens de programação, banco de dados, ferramentas e equipamentos de hardware. Abaixo, são apresentados os principais elementos desse ambiente:

Linguagens de Programação:**HTML, CSS e JavaScript:**

Essas linguagens são utilizadas para desenvolver a interface do usuário do site, incluindo páginas web, estilos e interatividade.

Python:

A linguagem Python é utilizada para criar scripts e aplicativos que controlam a lógica de negócios do sistema, incluindo o gerenciamento de pedidos e a interação com o banco de dados.

Banco de Dados:**MySQL:**

O sistema utiliza o banco de dados MySQL para armazenar informações sobre usuários, programas de intercâmbio, pedidos e itens de pedido.

Ferramentas de Desenvolvimento:**Visual Studio Code:**

Um ambiente de desenvolvimento integrado (IDE) amplamente utilizado para escrever código HTML, CSS, JavaScript e Python. Ele oferece recursos avançados de edição e depuração.

Git:

Um sistema de controle de versão que permite o controle e colaboração no desenvolvimento de código-fonte.

GitHub:

Uma plataforma de hospedagem de código-fonte que é usada para hospedar o repositório do projeto e facilitar a colaboração entre desenvolvedores.

Equipamentos de Hardware:**Computadores:**

São utilizados computadores de alto desempenho com sistemas operacionais Windows.

Servidor Web:

Um servidor web é configurado para hospedar o site do Experimento Intercâmbio Cultural e torná-lo acessível aos usuários finais.

Redes:**Internet:**

A conectividade à Internet é essencial para o desenvolvimento, implantação e operação do sistema, permitindo o acesso ao servidor web e ao banco de dados.

O ambiente de desenvolvimento é essencial para a criação, teste e implantação bem-sucedidos do sistema. Ele oferece as ferramentas e recursos necessários para que a equipe de desenvolvimento colabore e crie uma experiência de usuário excepcional para os clientes do Experimento Intercâmbio Cultural.

5. Implementação

Neste capítulo, descreveremos a implementação das classes e componentes do sistema "Experimento Intercâmbio Cultural" de acordo com as definições das fases anteriores do processo de desenvolvimento. A implementação é uma etapa fundamental em que transformamos o modelo conceitual em código executável, utilizando as práticas de programação orientada a objetos e as melhores práticas de desenvolvimento de software.

5.1. Linguagem de Programação

O sistema "Experimento Intercâmbio Cultural" é implementado principalmente usando as seguintes linguagens de programação:

HTML, CSS e JavaScript:

Essas linguagens são usadas para criar a interface do usuário do site, definindo a estrutura das páginas, o estilo visual e a interatividade.

Python:

A linguagem Python é utilizada para desenvolver a lógica de negócios do sistema, manipular dados, controlar pedidos e interagir com o banco de dados.

5.2. Implementação das Classes

As classes do sistema são implementadas com base no diagrama de classes definido nas fases anteriores. Cada classe é transformada em código Python, seguindo as práticas de programação orientada a objetos. Para manter a organização e o entendimento do código, aplicamos as seguintes diretrizes:

Cabeçalho de Funções:

Cada função e método inclui um cabeçalho contendo informações como descrição da função, data de criação, autor e outras informações relevantes.

```
def minha_funcao(parametro1, parametro2):  
    """  
    Descrição: Esta função realiza alguma tarefa.  
    Data de Criação: 10/09/2023  
    Autor: João Silva  
    """  
    # Implementação da função aqui
```

Comentários no Código:

Incluimos comentários explicativos ao longo do código para esclarecer partes complexas, tomar notas importantes e facilitar a compreensão do código por outros desenvolvedores.

```
# Este é um comentário explicativo  
variavel = 42 # Este comentário descreve a variável
```

Padronização de Nomes:

Seguimos convenções de nomenclatura consistentes para variáveis, parâmetros, funções, classes e outros identificadores. Isso ajuda a tornar o código mais legível e consistente.

```
# Exemplo de nomeação de função em snake_case  
def minha_funcao_exemplo(parametro_exemplo):  
    # Implementação da função aqui
```

Verificação de Declaração de Variáveis:

Sempre declaramos variáveis antes de usá-las e evitamos o uso de variáveis globais sempre que possível.

Tratamento de Erros:

Implementamos tratamento de erros adequado para lidar com exceções e situações inesperadas de forma controlada.

Stored Procedures:

Quando necessário, criamos stored procedures no banco de dados para acessar e manipular dados de forma eficiente.

Utilização de Padrões de Projeto:

Aplicamos padrões de projeto quando apropriado para resolver problemas específicos de design de software.

Otimização do Código:

Priorizamos a otimização do código, selecionando os algoritmos e estruturas de dados mais adequados para melhorar o desempenho e a eficiência.

5.3. Cronograma de Implementação

O cronograma de implementação detalhado do sistema "Experimento Intercâmbio Cultural" é elaborado com base nas tarefas definidas nas fases anteriores do projeto. Cada tarefa é atribuída a membros da equipe de desenvolvimento, e são definidas datas de início e término para cada atividade. A sequência das tarefas e a alocação de recursos são organizadas de acordo com o processo de desenvolvimento de software escolhido.

O cronograma de implementação inclui, entre outras atividades:

- Desenvolvimento da interface do usuário (HTML, CSS e JavaScript).
- Implementação da lógica de negócios (Python).
- Integração com o banco de dados (MySQL).
- Testes unitários e de integração.
- Correção de bugs e ajustes.
- Preparação para implantação.

O acompanhamento e o gerenciamento do progresso são realizados de forma regular para garantir que o desenvolvimento siga o cronograma previsto e atenda aos requisitos do sistema.

O processo de implementação é uma etapa crucial que transforma os conceitos em software funcional e operacional, e todas as práticas e diretrizes mencionadas acima são adotadas para garantir a qualidade e a confiabilidade do sistema "Experimento Intercâmbio Cultural".

6. Testes

Este capítulo tem como objetivo identificar defeitos no sistema, validar as funções do sistema, verificar se os requisitos foram implementados de forma adequada e avaliar a qualidade do software. Os testes desempenham um papel fundamental na garantia de que o sistema atenda às expectativas dos usuários e funcione conforme o planejado.

6.1. Plano de Testes

O Plano de Testes é uma documentação essencial que detalha a estratégia de teste a ser seguida para o sistema "Experimento Intercâmbio Cultural". Ele permite a validação do sistema por parte dos desenvolvedores, verificando se os requisitos do sistema foram implementados corretamente. O plano de testes é elaborado com base nos requisitos técnicos e funcionais do sistema e inclui uma lista completa de testes a serem realizados.

Conteúdo do Plano de Testes do Sistema:

Número do Teste:

Cada teste é identificado por um número único para referência.

Descrição do Teste:

Uma descrição detalhada do que o teste visa verificar ou validar.

Resultado Esperado:

O resultado que se espera alcançar ao executar o teste com sucesso.

Os tipos de testes a serem realizados podem incluir:

Teste de Funcionalidades: Para verificar se as funções do sistema estão operando conforme especificado.

Teste de Usabilidade: Para avaliar a facilidade de uso e a experiência do usuário.

Teste de Desempenho: Para medir o desempenho do sistema em condições normais.

Teste de Carga: Para avaliar o desempenho do sistema sob carga máxima.

Teste de Estresse: Para verificar como o sistema se comporta sob condições extremas.

Teste de Volume: Para avaliar o comportamento do sistema com grandes volumes de dados.

Teste de Segurança e Controle de Acesso: Para garantir que o sistema seja seguro contra ameaças.

Teste de Tolerância a Falhas e Recuperação: Para verificar como o sistema lida com falhas e se recupera delas.

Teste de Configuração: Para garantir que o sistema funcione corretamente em diferentes configurações.

Teste de Instalação: Para verificar a instalação do sistema em diferentes ambientes.

O Plano de Testes do Sistema é um documento que pode ser anexado à documentação do sistema e serve como um guia para a equipe de teste durante a execução dos testes.

6.2. Execução do Plano de Testes

Nesta etapa, os testes planejados no Plano de Testes do Sistema são executados. Durante a execução dos testes, são registrados dados relevantes que ajudam a avaliar o desempenho e a conformidade do sistema com os requisitos. As informações registradas incluem:

Identificação do Sistema:

Nome do sistema que está sendo testado.

Nome do Realizador dos Testes:

Nome da pessoa ou equipe responsável pela execução dos testes.

Configuração do Ambiente:

Descrição do ambiente em que os testes estão sendo realizados, incluindo hardware, software e configurações específicas.

Para cada teste executado, os seguintes dados são registrados:

Número do Teste:

O número de identificação do teste.

Resultado Obtido:

O resultado real da execução do teste, indicando se o teste passou ou falhou.

Comentários:

Comentários adicionais, se necessário, para explicar detalhes sobre o teste ou os resultados obtidos.

A execução do Plano de Testes é uma etapa crítica para identificar quaisquer defeitos ou problemas no sistema. Os testes não apenas validam o funcionamento correto do software, mas também ajudam a melhorar a qualidade do sistema, garantindo que ele atenda aos requisitos e expectativas dos usuários.

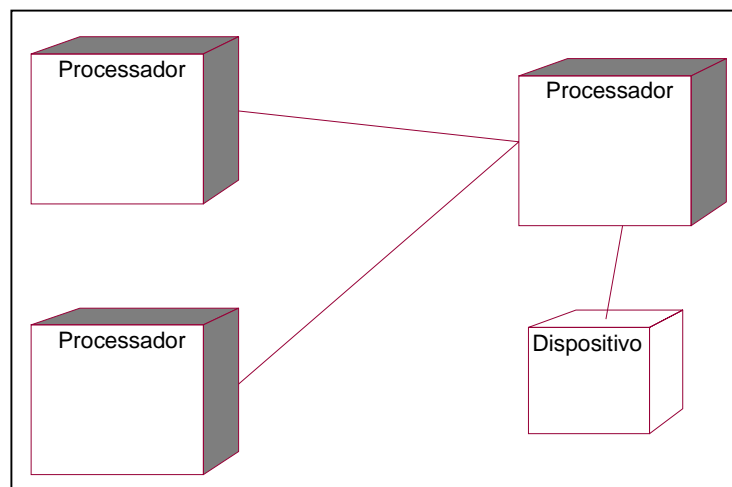
7. Implantação

Este capítulo tem como objetivo apresentar informações relevantes para a implantação e funcionamento do sistema.

7.1. Diagrama de Implantação

Neste item deve ser apresentado o diagrama de implantação que representa a parte física do sistema, exibindo os dispositivos, as máquinas de processamento em tempo de execução e os componentes que nelas serão instalados.

A seguir é apresentada a notação básica de um diagrama de implantação.



Notação básica do diagrama de implantação.

7.2. Manual de Implantação

Neste item deve ser elaborado o manual de instalação. Este manual deve conter a descrição passo a passo de como deve ser realizada a instalação do sistema.

Para maiores detalhes pode-se consultar artefatos do RUP da fase de Instalação.

8. Manual do Usuário

Bem-vindo ao Sistema de Intercâmbio da Experimento Boa Viagem!

Este manual foi criado para ajudar você a navegar e utilizar nosso sistema de forma eficaz. Aqui você encontrará instruções detalhadas sobre como usar os recursos disponíveis no site para obter informações sobre nossos programas de intercâmbio cultural e como agendar um horário conosco. Siga as instruções abaixo para aproveitar ao máximo sua experiência.

8.1. Navegação no Site

Ao acessar o site da Experimento Boa Viagem, você verá diversas seções, cada uma fornecendo informações específicas sobre nossos serviços. Aqui estão as principais seções e como navegá-las:

Início: A seção "Início" apresenta uma breve visão geral da Experimento Boa Viagem e oferece um botão para agendar um horário conosco.

Sobre: Na seção "Sobre", você encontrará informações detalhadas sobre a Experimento, sua história e sua missão.

Serviços: Nesta seção, você pode explorar os diferentes programas de intercâmbio oferecidos pela Experimento. Cada programa é apresentado em um cartão com uma imagem e uma descrição. Clique no botão "Saiba Mais" em um cartão para obter informações adicionais sobre o programa.

Depoimentos: Aqui, você encontrará depoimentos de clientes satisfeitos que compartilharam suas experiências com a Experimento Boa Viagem.

Contato: A seção de "Contato" fornece informações de contato, incluindo um número de telefone, endereço e um link direto para nos contatar pelo WhatsApp.

8.2. Agendamento de Horário

Se você estiver interessado em nossos serviços e desejar agendar um horário para obter informações mais detalhadas, siga estas etapas:

- 1 - Na seção "Início", clique no botão "Agendar um Horário".
- 2 - Isso o levará a uma página onde você pode preencher seu nome e número de telefone.
- 3 - Após preencher as informações, clique em "Enviar Solicitação".
- 4 - Entraremos em contato com você o mais breve possível para agendar um horário adequado para discutir suas necessidades.

8.3. Carrinho de Compras

Você também pode adicionar programas de intercâmbio ao seu carrinho de compras e nos enviar uma solicitação para obter mais informações sobre eles. Veja como fazer isso:

- 1 - Na seção "Serviços", clique no botão "Saiba Mais" no cartão do programa de intercâmbio que você deseja adicionar ao carrinho.
- 2 - Na página de detalhes do programa, role para baixo e encontre a seção "Adicionar ao Carrinho".
- 3 - Selecione a quantidade de programas que você deseja adicionar e clique em "Adicionar ao Carrinho".
- 4 - O programa será adicionado ao seu carrinho de compras, e você verá o número de itens no carrinho no ícone do carrinho na parte superior direita da página.
- 5 - Você pode continuar explorando outros programas e adicionando-os ao carrinho, se desejar.
- 6 - Quando terminar, clique no ícone do carrinho para visualizar os programas que você adicionou. Você verá uma lista com os programas selecionados.
- 7 - Role para baixo e preencha suas informações pessoais no formulário de contato.
- 8 - Clique em "Enviar Solicitação" para nos enviar sua solicitação. Entraremos em contato com você para fornecer informações adicionais sobre os programas e o processo de inscrição.

8.4. Considerações Finais

Esperamos que este manual tenha ajudado você a entender como navegar e usar nosso sistema de forma eficaz. Se tiver alguma dúvida ou precisar de assistência adicional, não hesite em entrar em contato conosco por telefone ou WhatsApp, conforme as informações fornecidas na seção de "Contato".

Agradecemos por escolher a Experimento Boa Viagem como sua opção de intercâmbio cultural. Estamos ansiosos para ajudá-lo a realizar sua jornada de aprendizado internacional!

9. Conclusões e Considerações Finais

Este capítulo tem como objetivo apresentar os resultados obtidos no desenvolvimento do sistema "Experimento Intercâmbio Cultural". Além disso, serão discutidas as conclusões, limitações, inovações, possíveis integrações com outros projetos e perspectivas para o futuro do sistema.

9.1. Resultados Obtidos

Os resultados obtidos no desenvolvimento do sistema são a entrega de um sistema funcional que atende aos requisitos especificados. O sistema "Experimento Intercâmbio Cultural" oferece os seguintes recursos:

Navegação Intuitiva:

O sistema possui uma interface de usuário amigável que permite aos usuários navegar facilmente pelas diferentes seções do site.

Informações Detalhadas:

O sistema fornece informações detalhadas sobre os serviços oferecidos pela Experimento, incluindo programas de intercâmbio, depoimentos de clientes e informações de contato.

Funcionalidade de Carrinho de Compras

Os usuários podem adicionar programas de intercâmbio ao carrinho de compras e preencher informações pessoais para solicitar mais detalhes ou agendar um horário.

Depoimentos de Clientes:

O sistema exibe depoimentos de clientes satisfeitos, adicionando credibilidade à Experimento.

Contato Fácil

Os usuários podem entrar em contato com a Experimento por meio de links para o WhatsApp e outras informações de contato.

9.2. Conclusões

Com base no desenvolvimento do sistema, conclui-se que a Experimento Intercâmbio Cultural agora possui uma presença online eficaz para atrair clientes em potencial e fornecer informações detalhadas sobre seus serviços. O sistema facilita a interação com os usuários e pode contribuir para aumentar o número de clientes e melhorar a experiência do cliente.

9.3. Limitações

Algumas limitações do sistema incluem:

Falta de Recursos Avançados:

Embora o sistema cumpra os requisitos básicos, ele não possui recursos avançados, como integração com sistemas de pagamento online ou recursos de personalização de programas de intercâmbio.

Aprimoramentos de Usabilidade:

Embora a usabilidade seja boa, sempre há espaço para melhorias na experiência do usuário.

9.4. Inovações

O sistema introduz a Experimento Intercâmbio Cultural no ambiente digital, permitindo que a empresa alcance um público mais amplo e forneça informações detalhadas sobre seus serviços. Isso representa uma inovação significativa para a empresa, que anteriormente dependia principalmente de publicidade offline.

9.5. Possíveis Integrações com Outros Projetos

O sistema pode ser integrado a outros projetos futuros, como:

Sistema de Gerenciamento de Cliente:

Um sistema para gerenciar informações de clientes, acompanhar solicitações e interações.

Sistema de Pagamento:

Integração com um sistema de pagamento pelo WhatsApp para permitir que os clientes paguem pelos programas de intercâmbio diretamente com o prestador do serviço.

Sistema de Gerenciamento de Conteúdo:

Para facilitar a atualização de informações no site.

9.6. Continuação do Sistema em Trabalhos Futuros

O sistema pode ser continuado e aprimorado em trabalhos futuros, considerando as seguintes áreas:

Melhorias de Usabilidade:

Realizar estudos de usabilidade para identificar áreas de melhoria na interface do usuário.

Integrações Avançadas:

Integrar o sistema com outros serviços, como sistemas de pagamento online e sistemas de gerenciamento de clientes.

Personalização de Programas de Intercâmbio:

Adicionar recursos que permitam aos usuários personalizar seus programas de intercâmbio com base em suas preferências.

Aprimoramentos de Desempenho:

Otimizar o desempenho do site para garantir tempos de carregamento rápidos e uma experiência fluida para os usuários.

O sistema "Experimento Intercâmbio Cultural" tem o potencial de se tornar uma ferramenta valiosa para a empresa e pode continuar a evoluir para atender às crescentes demandas e expectativas dos clientes.

Bibliografia

Neste item devem-se apresentar todas as obras (livros, artigos, Internet, revistas, etc...) utilizadas na elaboração da documentação e na implementação do projeto.

Glossário

- ✓ **API (Interface de Programação de Aplicativo):** Um conjunto de regras e protocolos que permitem que diferentes softwares se comuniquem entre si. As APIs são usadas para acessar e interagir com recursos ou serviços de outros aplicativos.
- ✓ **Framework:** Uma estrutura de desenvolvimento que fornece uma base para a construção de aplicativos. Os frameworks frequentemente incluem bibliotecas, padrões e convenções que aceleram o processo de desenvolvimento.
- ✓ **IDE (Ambiente de Desenvolvimento Integrado):** Um ambiente de software que combina ferramentas de desenvolvimento, como editor de código, depurador e compilador, em um único aplicativo.
- ✓ **Git:** Um sistema de controle de versão amplamente utilizado para rastrear e gerenciar as alterações no código-fonte durante o desenvolvimento de software.
- ✓ **Branch (Ramificação):** Uma cópia independente do código-fonte em um repositório de controle de versão, geralmente criada para desenvolver recursos ou correções separadamente antes de serem mescladas de volta à linha principal.
- ✓ **Merge (Mesclagem):** O processo de combinar as alterações de uma ramificação em outra no repositório de controle de versão.
- ✓ **Dependência:** Um componente de software que é necessário para que outro componente funcione corretamente. As dependências podem ser bibliotecas, módulos ou pacotes externos.
- ✓ **API RESTful:** Uma arquitetura de design de API que segue os princípios do REST (Representational State Transfer). As APIs RESTful usam métodos HTTP (como GET, POST, PUT e DELETE) para manipular recursos.
- ✓ **SQL (Structured Query Language):** Uma linguagem de programação utilizada para gerenciar e consultar bancos de dados relacionais.
- ✓ **Front-End:** A parte de um aplicativo ou site que os usuários interagem diretamente, incluindo a interface do usuário e a experiência do usuário.
- ✓ **Back-End:** A parte de um aplicativo ou site que lida com o processamento de dados, a lógica de negócios e a comunicação com o banco de dados. É responsável pela funcionalidade "nos bastidores".
- ✓ **DevOps:** Uma cultura e conjunto de práticas que visam unificar o desenvolvimento de software (Dev) e operações de TI (Ops) para melhorar a colaboração e automatizar o ciclo de vida do software.
- ✓ **CI/CD (Integração Contínua/Entrega Contínua):** Um conjunto de práticas e ferramentas que automatizam a construção, testes e implantação de código, permitindo entregas de software frequentes e confiáveis.
- ✓ **Scrum:** Um framework ágil de gerenciamento de projetos que se concentra na colaboração, na entrega iterativa e na adaptação contínua.
- ✓ **MVC (Model-View-Controller):** Um padrão de design de software que separa um aplicativo em três componentes principais: Model (modelo de dados), View (interface do usuário) e Controller (controlador de lógica).
- ✓ **Bug:** Um erro ou defeito no código de software que leva a comportamento indesejado ou a problemas de funcionamento.
- ✓ **IDEAL (Identificar, Definir, Explorar, Agir, Aprender):** Um modelo de ciclo de vida de desenvolvimento que enfatiza a melhoria contínua e a adaptação com base no feedback.
- ✓ **Scripting:** A escrita de pequenos programas ou scripts para automação de tarefas ou processamento de dados.
- ✓ **Sprint:** Um período de tempo definido durante o qual uma equipe de desenvolvimento trabalha em um conjunto específico de tarefas ou funcionalidades, com o objetivo de entregar um incremento funcional.
- ✓ **Deployment (Implantação):** O processo de disponibilizar uma versão de software para uso em um ambiente de produção.