



INSTITUTO DE INFORMÁTICA

ARQUITETURA DE SOFTWARE

LEONARDO RIBEIRO OLIVEIRA PALMEIRA

THÂMARA CORDEIRO DE CASTRO

LUIZ FELIPE PIRES CARVALHO

**Protótipo de documentação de
engenharia de software**

**LEONARDO RIBEIRO OLIVEIRA PALMEIRA
THÂMARA CORDEIRO DE CASTRO
LUIZ FELIPE PIRES CARVALHO**

**Protótipo de documentação de
engenharia de software**

Documento de arquitetura de software
apresentado à disciplina de arquitetura de
software.

Professor: JACSON RODRIGUES BARBOSA

Especificação e Visão Geral de Arquitetura e Tecnologias:

O software de gestão de academia é projetado para fornecer uma solução completa e eficiente para a administração de academias, visando melhorar a experiência dos usuários e otimizar as operações internas. O programa será desenvolvido como um aplicativo web, oferecendo uma interface intuitiva e amigável para os administradores da academia, instrutores e usuários finais.

A arquitetura do software seguirá uma abordagem modular e escalável, permitindo que as funcionalidades sejam adicionadas e atualizadas conforme necessário. A aplicação será baseada em um modelo cliente-servidor, onde o servidor será responsável pelo processamento das solicitações e armazenamento dos dados no banco de dados.

A aplicação utiliza a linguagem de programação Elixir e o framework Phoenix para desenvolvimento web.

- Módulos de Controle de Treinamentos (Wabanex.Trainings.Create):
 - Responsabilidade: Esse módulo é responsável por criar um novo treinamento no sistema.
 - Tecnologias utilizadas: Elixir, Phoenix, Ecto (ORM para banco de dados).
- Módulos de Controle de Usuários (Wabanex.Users.Create, Wabanex.Users.Get):
 - Responsabilidade: Esses módulos são responsáveis por criar e buscar usuários no sistema.
 - Tecnologias utilizadas: Elixir, Phoenix, Ecto.
- Módulo para cálculo de IMC (Wabanex.IMCTest, Wabanex.IMC):
 - Responsabilidade: Esse módulo realiza o cálculo do Índice de Massa Corporal (IMC) com base em um arquivo CSV fornecido.
 - Tecnologias utilizadas: Elixir.
- Módulo de Teste de Usuário (Wabanex.UserTest):
 - Responsabilidade: Esse módulo realiza testes de unidade para verificar se a função changeset/1 retorna um conjunto de alterações válido para criar um usuário.
 - Tecnologias utilizadas: Elixir, ExUnit (framework de testes).
- Módulo da Aplicação Principal (Wabanex.Application):
 - Responsabilidade: Esse módulo é responsável por iniciar a aplicação Phoenix, definindo os processos supervisionados e as configurações iniciais.
 - Tecnologias utilizadas: Elixir, Phoenix.
- Módulo de Exercício (Wabanex.Exercise):
 - Responsabilidade: Esse módulo define o schema e as alterações possíveis para o objeto "Exercício" no sistema.
 - Tecnologias utilizadas: Elixir, Ecto.
- Módulo de Repositório (Wabanex.Repo):
 - Responsabilidade: Esse módulo define o repositório de dados, especificando a conexão com o banco de dados PostgreSQL.
 - Tecnologias utilizadas: Elixir, Ecto, PostgreSQL.

- Módulo de Treinamento (Wabanex.Training):
 - Responsabilidade: Esse módulo define o schema e as alterações possíveis para o objeto "Treinamento" no sistema, incluindo relacionamentos com usuários e exercícios.
 - Tecnologias utilizadas: Elixir, Ecto.
- Módulo de Usuário (Wabanex.User):
 - Responsabilidade: Esse módulo define o schema e as alterações possíveis para o objeto "Usuário" no sistema, incluindo relacionamentos com treinamentos.
 - Tecnologias utilizadas: Elixir, Ecto.

Em resumo, a aplicação utiliza a linguagem Elixir juntamente com o framework Phoenix para desenvolver uma aplicação web de gestão de academia. Ela se baseia em um modelo de banco de dados relacional (utilizando o Ecto como ORM) e segue a arquitetura MVC (Model-View-Controller) para organizar os componentes.

Identificação de Requisitos e Atributos de Qualidade:

Durante o levantamento de requisitos, foram identificados os seguintes requisitos:

- Criação de treinamentos personalizados: O software deve permitir que os administradores criem treinamentos personalizados, definindo parâmetros e validando os dados fornecidos.
- Criação de novos usuários: Deve ser possível criar novos usuários, com validação de dados e inserção no banco de dados.
- Informações detalhadas dos usuários: O software deve fornecer informações detalhadas sobre cada usuário, incluindo treinamentos associados.
- Cálculo automatizado do IMC: Deve ser capaz de calcular o índice de massa corporal (IMC) a partir de um arquivo CSV, fornecendo uma análise precisa e rápida.
- Listagem de treinamentos disponíveis: Deve permitir a listagem de todos os treinamentos disponíveis, facilitando o acesso e seleção por parte dos usuários.
- Agendamento de treinamentos: Deve ser possível agendar treinamentos para usuários específicos, garantindo a organização e otimização do uso dos recursos da academia.
- Registro de histórico de treinamentos: O software deve registrar o histórico de treinamentos de cada usuário, facilitando o acompanhamento do progresso e desempenho individual.
- Geração de relatórios de desempenho: Deve ser capaz de gerar relatórios de desempenho personalizados para cada usuário, fornecendo métricas relevantes para avaliação e motivação.

- Gerenciamento de planos de pagamento: O software deve gerenciar os planos de pagamento oferecidos pela academia, incluindo criação, atualização e exclusão de planos.
- Atualização de informações dos usuários: Deve permitir a atualização fácil e segura das informações dos usuários, garantindo a precisão dos dados registrados.

Além dos requisitos funcionais, também foram identificados atributos de qualidade importantes para o software:

- Desempenho: O software deve ser capaz de lidar com um grande volume de dados e ter um desempenho adequado ao recuperar informações dos usuários.
- Segurança: Medidas de segurança robustas devem ser implementadas para proteger os dados do usuário, incluindo criptografia, autenticação segura e proteção contra acessos não autorizados.
- Usabilidade: A usabilidade do software deve ser priorizada, com uma interface de usuário intuitiva, fluxos de trabalho bem projetados e documentação disponível para orientar os usuários.
- Escalabilidade: A arquitetura do software deve ser escalável, permitindo o aumento do número de usuários e volume de dados sem comprometer o desempenho.

Identificação dos Stakeholders:

- Administradores da academia: São responsáveis pela configuração e gerenciamento do software, incluindo a criação de treinamentos, gestão de usuários e planos de pagamento.
- Instrutores: São responsáveis por ministrar os treinamentos e utilizar o software para acompanhar o desempenho dos usuários.
- Usuários da academia: Utilizam o software para agendar treinamentos, acompanhar seu progresso e receber relatórios de desempenho personalizados.
- Equipe de desenvolvimento: Responsável por projetar, desenvolver, testar e manter o software de gestão de academia.

Administradores da academia:

- Experiência: Os administradores terão acesso a recursos avançados de configuração e gerenciamento do software. Eles poderão criar treinamentos personalizados, definir parâmetros, configurar planos de pagamento, gerenciar usuários e acessar relatórios de desempenho. A interface do administrador será intuitiva e fornecerá ferramentas eficientes para facilitar suas tarefas diárias.

- Benefícios: Com o software, os administradores terão uma visão abrangente das operações da academia. Eles poderão otimizar a oferta de treinamentos, acompanhar o desempenho dos usuários, gerenciar planos de pagamento de forma eficiente e tomar decisões informadas com base nos relatórios de desempenho.

Instrutores:

- Experiência: Os instrutores terão acesso a um painel dedicado no software, onde poderão visualizar os treinamentos atribuídos a eles, monitorar o progresso dos usuários, registrar informações relevantes e fornecer feedback. Eles também poderão interagir com os usuários através do sistema de mensagens integrado.
- Benefícios: O software permitirá que os instrutores tenham uma visão abrangente do desempenho dos usuários sob sua responsabilidade. Eles poderão acompanhar o progresso, identificar áreas de melhoria, fornecer orientações personalizadas e ajudar os usuários a alcançarem seus objetivos de forma mais eficaz.

Usuários da academia:

- Experiência: Os usuários terão acesso a uma interface amigável onde poderão visualizar os treinamentos disponíveis, agendar suas sessões, acompanhar seu progresso, receber relatórios de desempenho e interagir com os instrutores e outros usuários através do sistema de mensagens. A interface será personalizada para atender às necessidades individuais de cada usuário.
- Benefícios: O software proporcionará uma experiência aprimorada para os usuários, permitindo que eles acessem treinamentos personalizados, acompanhem seu progresso em tempo real, recebam feedback dos instrutores e recebam relatórios de desempenho detalhados. Isso os motivará a se envolverem mais nas atividades da academia e alcançarem seus objetivos de forma mais eficiente.

Equipe de desenvolvimento:

- Experiência: A equipe de desenvolvimento será responsável por projetar, desenvolver, testar e manter o software. Eles utilizarão as melhores práticas de desenvolvimento de software e colaborarão em um ambiente de desenvolvimento integrado. Eles também terão acesso a ferramentas e recursos para facilitar seu trabalho e garantir a qualidade do produto final.