

Desenho de Alto Nível



Engenharia de Software 2023/2024
Especificação do Sistema – alto nível

Turma : 3

Grupo nº1

nº202100299 - Rui Barroso

nº202100296 - Gonçalo Vieira

nº202100984 - Francisco Silva

nº201901953 - André Pauli

1			
2			

- SUMARIO EXECUTIVO
- INTRODUÇÃO
- DESENHO DE ALTO NÍVEL
 - ARQUITETURA GERAL
 - ARQUITETURA LÓGICA
 - DIAGRAMA DE CLASSES DE DESENHO
 - PROCESOS DE NEGÓCIO
 - Identificação dos processos
 - Diagramas dos processos de negócio
 - INTERFACE COM O UTILIZADOR
 - Introdução
 - Interfaces (Mockup)
 - Normas
 - Diagrama Geral de Navegação
 - Matriz de acessos
 - PERSISTÊNCIA
 - Introdução
 - Modelo Relacional
 - ARQUITETURA FÍSICA
 - Introdução
 - Diagrama de Componentes
 - Diagrama de Instalação
 - NORMAS DE CODIFICAÇÃO DA APLICAÇÃO

SUMARIO EXECUTIVO

A EasyFitHub é uma aplicação para smart-phones e computadores que oferece ferramentas para a inscrição em ginásios e a gestão de ginásios. Os gym-enthusiasts terão acesso a novas formas de interagir com os diferentes aspectos de um ginásio e, os gestores/donos terão uma plataforma centralizada que facilitará a manutenção do ginásio. Principais Características: Inscrição Simples: Com apenas alguns cliques, os usuários podem encontrar academias, visualizar os seus detalhes e fazer uma inscrição de forma fácil e rápida. Perfil

Personalizado: Os usuários podem criar perfis personalizados a seu gosto. Agendamento de Treinos: A Easy Fit Hub oferece opções para o agendamento, permitindo que os usuários marquem os seus treinos consoante a sua disponibilidade. Acompanhamento de Desempenho: Os usuários podem monitorar seu progresso, registrando os seus treinos e acompanhando as suas estatísticas de condicionamento físico ao longo do tempo. Pagamentos Online: A aplicação oferece um sistema seguro de pagamentos mensais online, eliminando a necessidade de transações presenciais e promovendo a conveniência de todos os participantes. Gestão de Academia Simplificada: Para os gestores de academias, a Easy Fit Hub oferece ferramentas para gerir o equipamento do ginásio, agendamentos e eventos, inscrições, pagamentos e a comunicação com assinantes, otimizando a eficiência nas operações. Notificações: A aplicação facilita a comunicação direta entre os gestores e donos das academias e os membros, permitindo o envio de notificações, promoções e lembretes de forma instantânea. Relatórios Analíticos: Os gestores e os donos das academias têm acesso a relatórios detalhados sobre o desempenho da academia, inscrições, frequência dos membros e outros dados relevantes.

INTRODUÇÃO

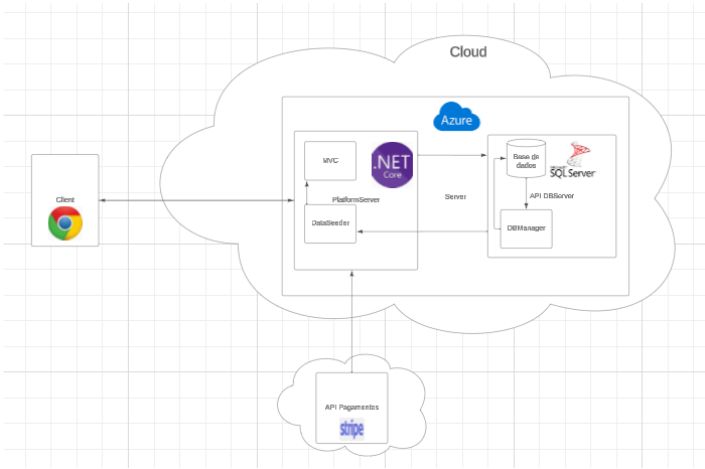
O desenho de alto nível desempenha um papel crucial na arquitetura e no desenvolvimento de sistemas, proporcionando uma visão abrangente e detalhada da estrutura e funcionamento de uma solução tecnológica. Este relatório aborda o processo de desenho de alto nível de EasyFitHub, fornecendo uma análise profunda das escolhas arquiteturais, padrões de design e componentes chave que compõem a infraestrutura da solução.

A elaboração de um desenho de alto nível é um estágio essencial no ciclo de vida do desenvolvimento de software, pois serve como um guia fundamental para os desenvolvedores, arquitetos e demais partes interessadas envolvidas no projeto. Este relatório destaca os principais aspectos relacionados ao desenho de alto nível, incluindo a estrutura geral da aplicação, estrutura lógica, diagrama de classes, processos de negocio, mockups, diagrama de navegação, modelo relacional, diagrama de componentes, diagrama de instalação e normas.

Em última análise, este relatório de desenho de alto nível visa fornecer uma documentação abrangente e compreensível do arcabouço arquitetural da aplicação EasyFitHub, garantindo uma base sólida para o desenvolvimento eficaz, a manutenção contínua e a evolução do sistema ao longo do tempo

DESENHO DE ALTO NÍVEL

ARQUITETURA GERAL



Descrição da arquitetura geral do sistema.

O aplicativo Cliente MVC .NET Core é a interface de utilizador principal da aplicação, construída usando o MVC .NET Core. Ele envia solicitações para o servidor e exibe os resultados para o utilizador.

O aplicativo Servidor é a aplicação backend que gerencia operações como inserção de dados e gerenciamento da base de dados. Ele atua como um mediador entre o cliente e outros componentes do sistema.

O Seeder de Dados é responsável por inserir dados no banco de dados, podendo ser dados iniciais, dados de teste ou dados migrados de uma versão anterior da aplicação.

O Cliente da API Pagamentos interage com uma API externa de Pagamentos, como o Stripe. Ele envia solicitações para a API, recebe respostas e as repassa para a aplicação servidor.

O Servidor da Base de Dados, SQL server, gerencia a base de dados. Ele armazena e recupera dados com base em solicitações do servidor e outros componentes.

A infraestrutura de nuvem, fornecida pelo Azure Cloud, é usada para hospedar o servidor do banco de dados.

Em resumo, o sistema possui um aplicativo cliente que se comunica com um servidor, o qual coordena tarefas com seeder de dados, APIs externas e um servidor de base de dados hospedado na nuvem Azure. Essa arquitetura permite o gerenciamento eficiente de dados e a integração com serviços externos no aplicativo.

ARQUITETURA LÓGICA

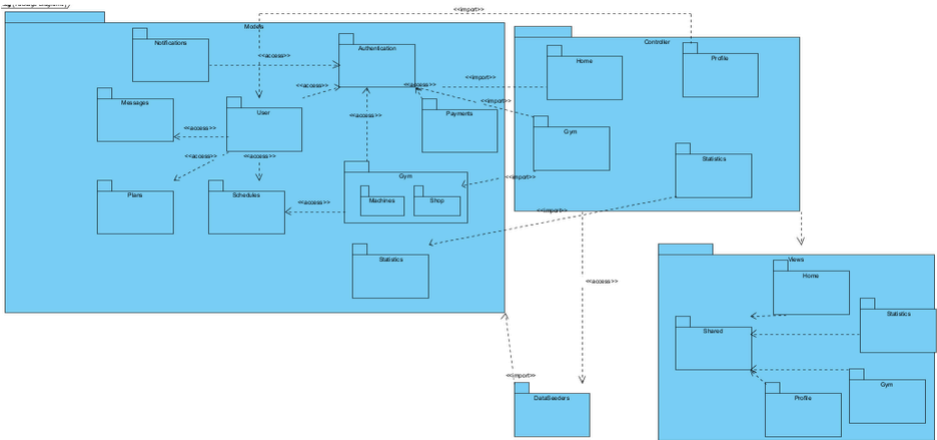
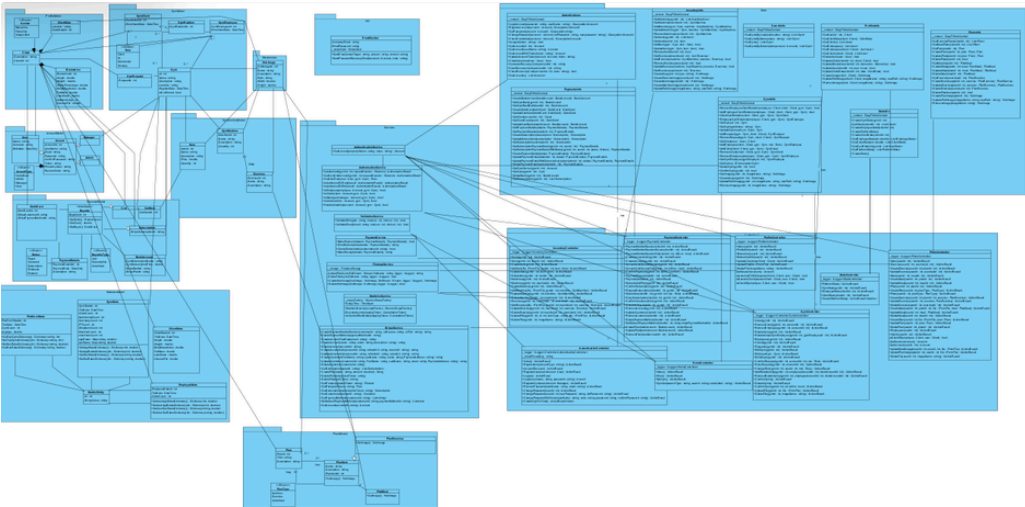


DIAGRAMA DE CLASSES DE DESENHO



PROCESOS DE NEGÓCIO

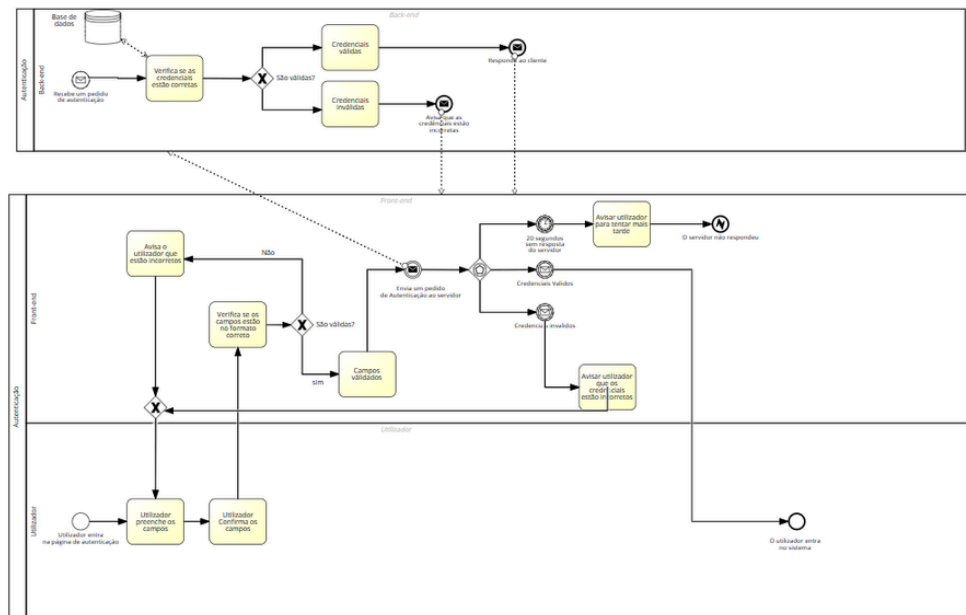
Elaboração e identificação de todos os processos de negócio “to-be”

Identificação dos processos

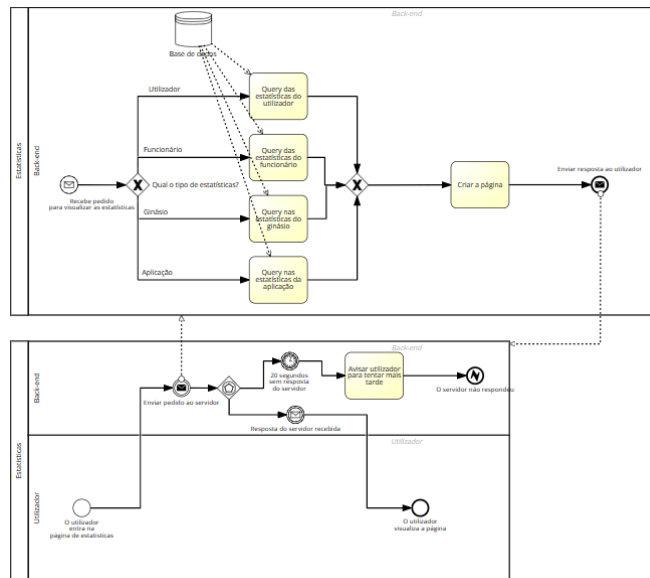
- Autenticação
 - Registo
 - Recuperação de conta
 - Pesquisar Ginasio
 - Gerir perfil
 - Gerir dados biometricos
 - Gerir planos
 - Gerir horario
 - Pagamento mensal
 - Notificações
 - Estatísticas
 - Maquinas
 - Loja
 - Membros ginasio
 - Funcionarios

Diagramas dos processos de negócio

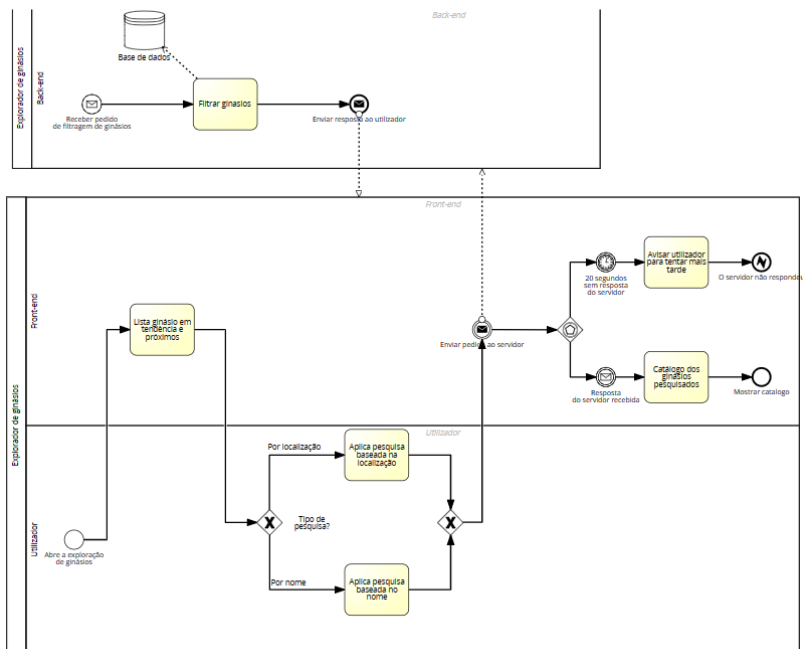
- Autenticação



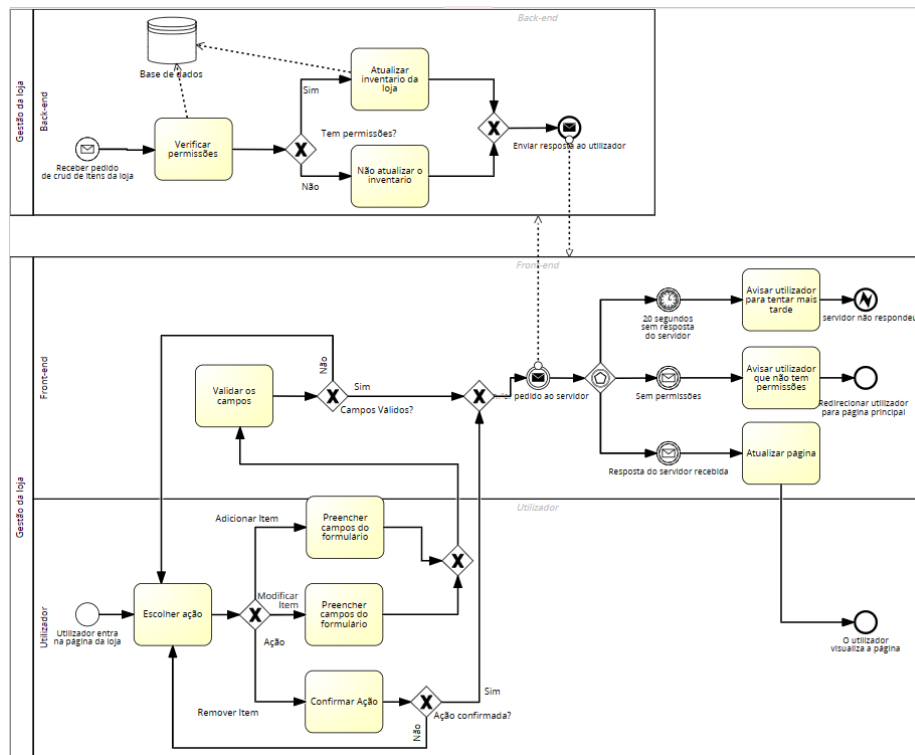
- Estatísticas



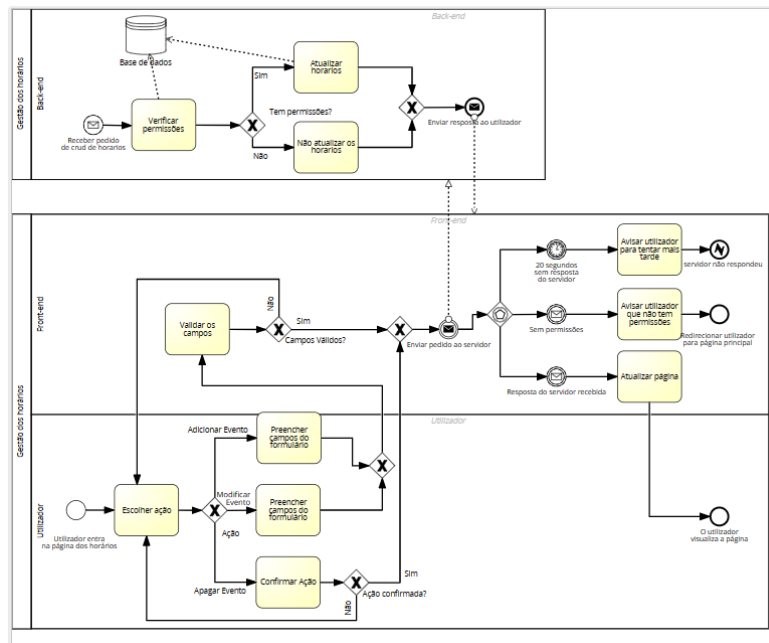
- Explorador de ginásios



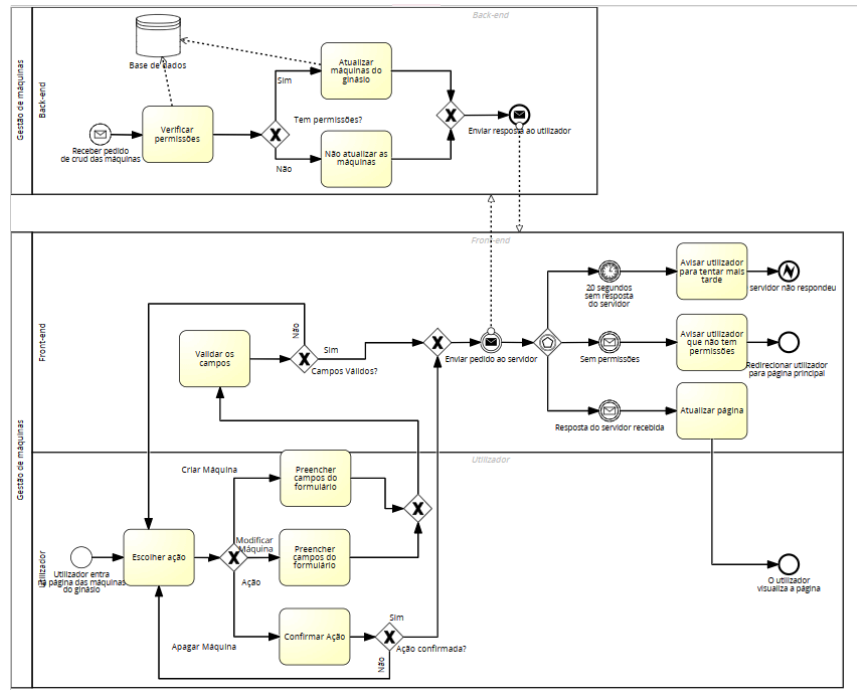
- Gestão da loja



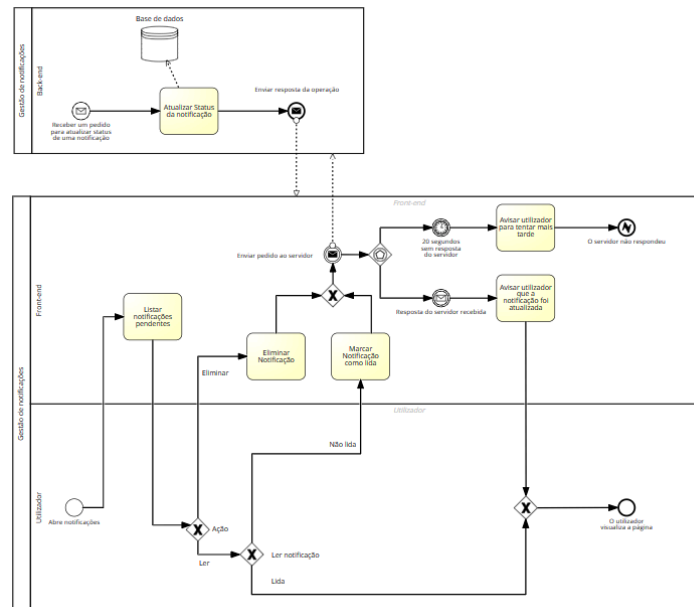
- Gestão de horários



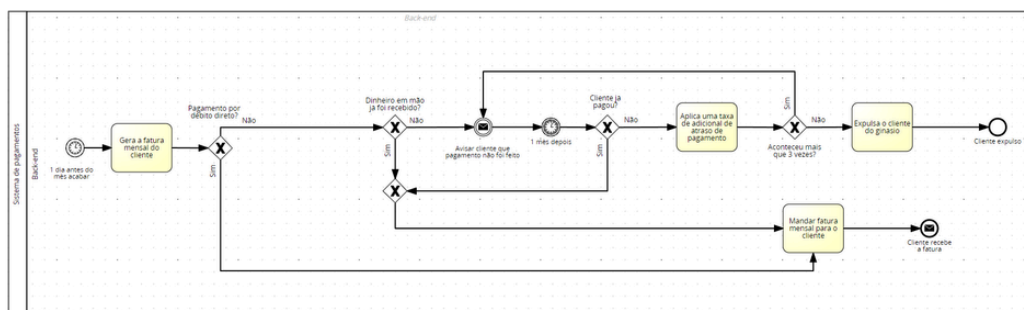
- Gestão de máquinas



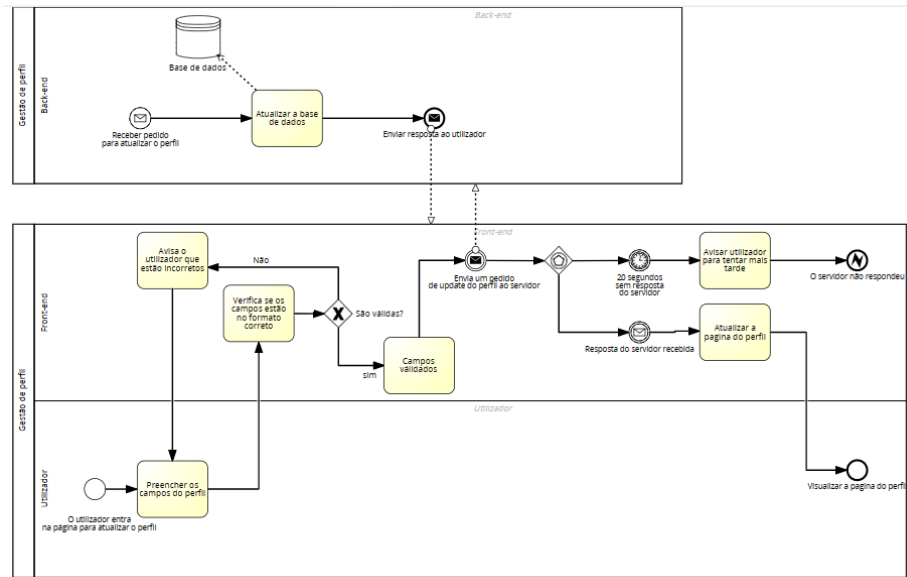
- Gestão de notificações



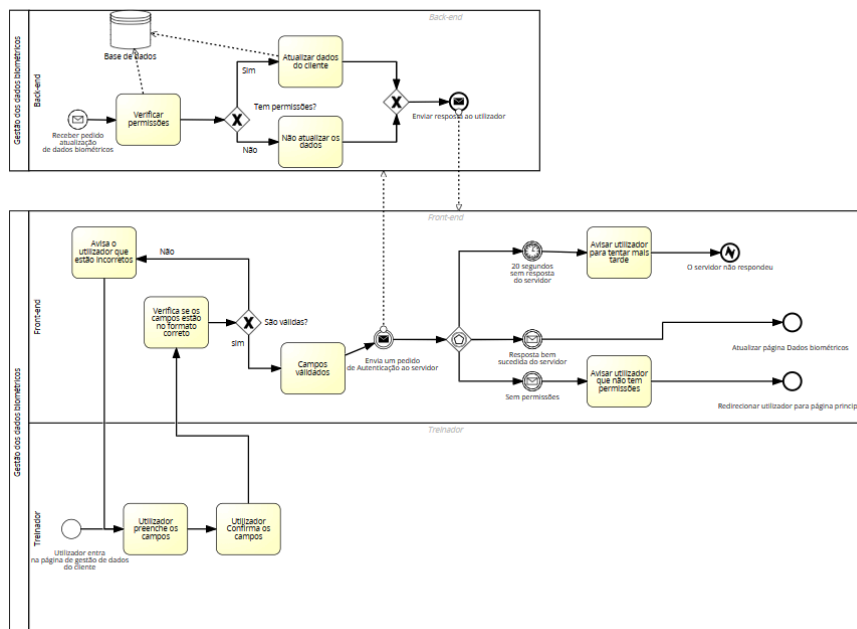
- Sistema de pagamentos



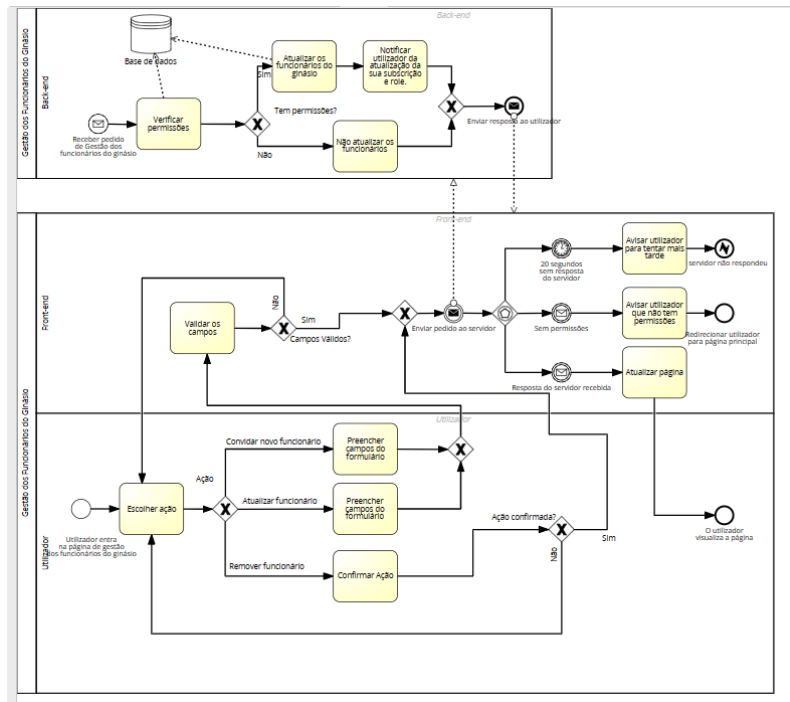
- Gestão de perfil



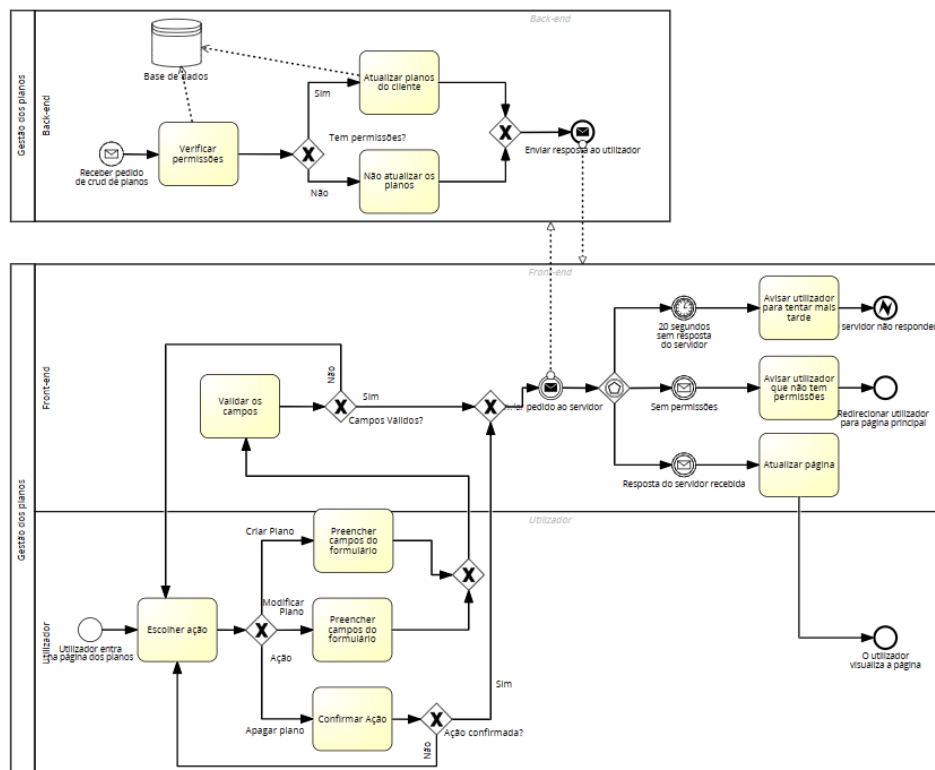
- Gestão dos dados biométricos



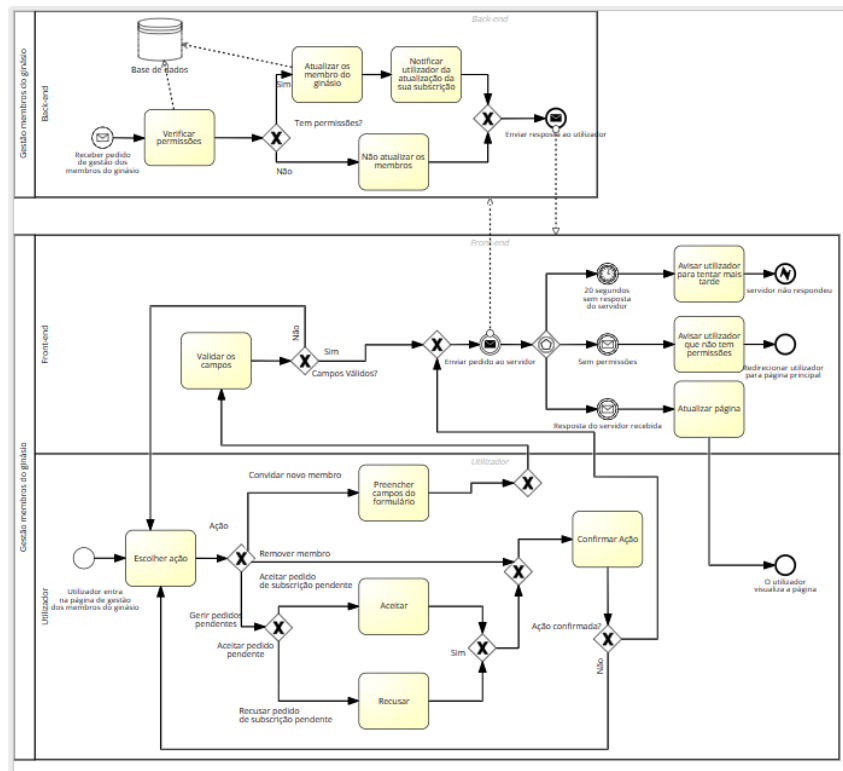
- Gestão dos funcionários do ginásio



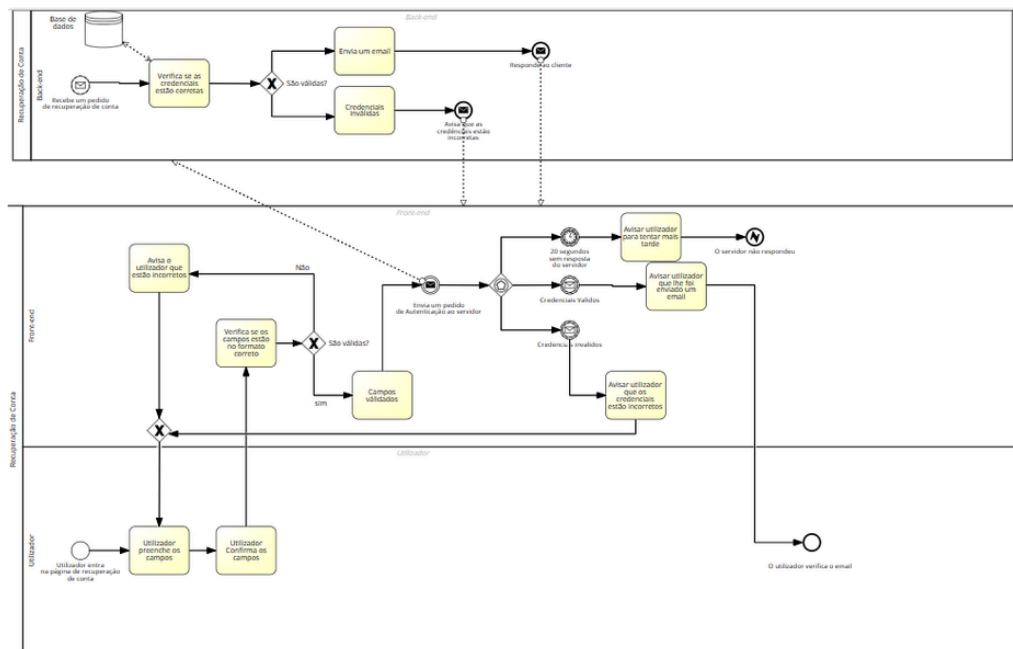
- Gestão de planos



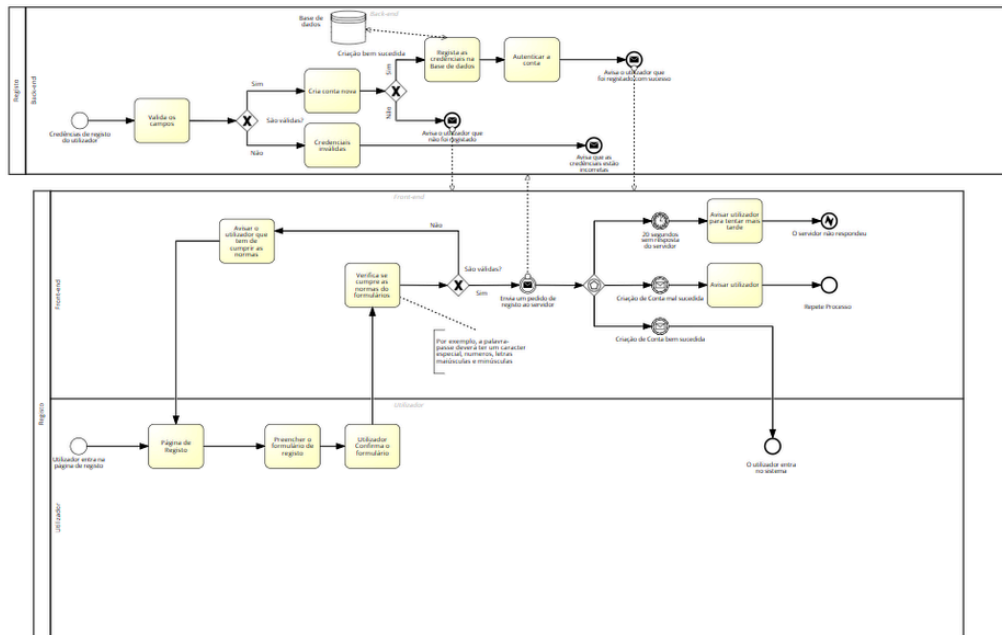
- Gestão de membros do ginásio



- Recuperação de conta



- Registo



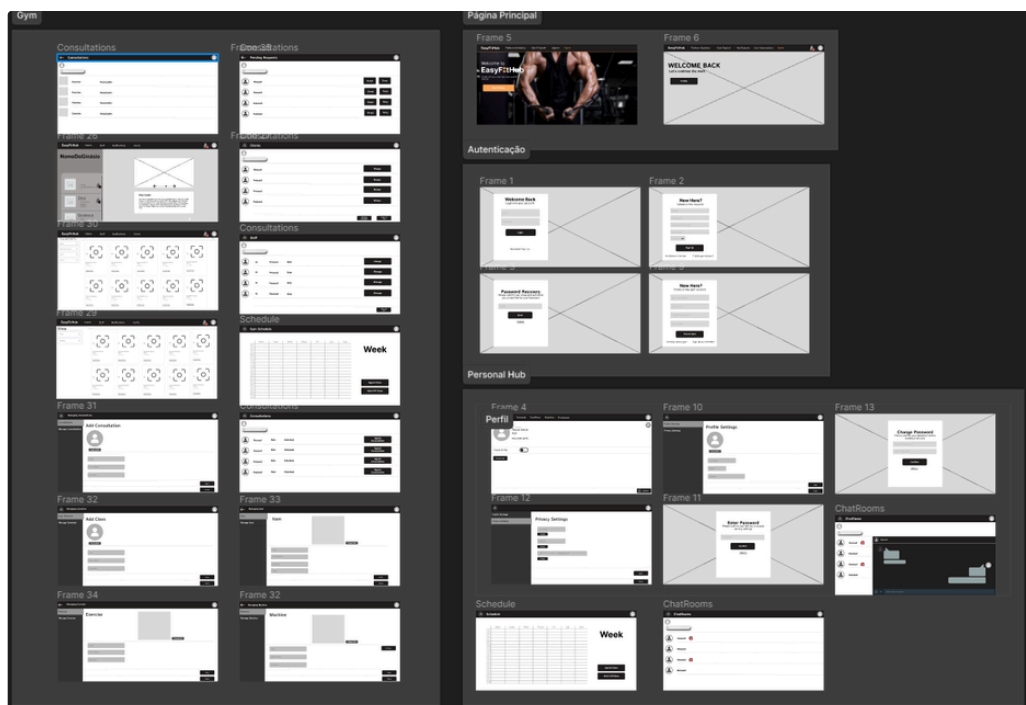
INTERFACE COM O UTILIZADOR

Introdução

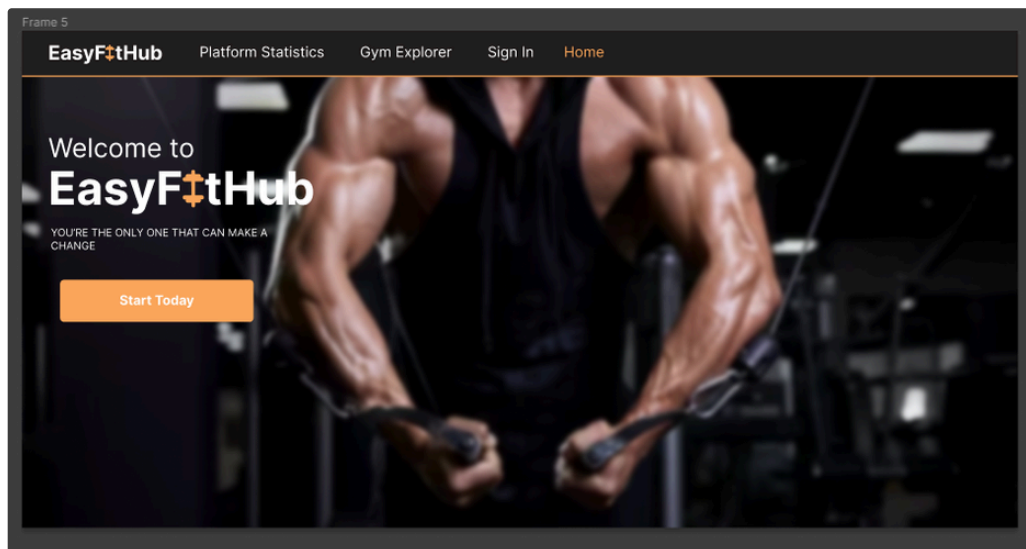
Interfaces (Mockup)

link mockups: [EasyFitHub](#)

Mockups:



Master Page:



Normas

Código de cores:

Laranja: FAA75C

Branco: FFFFFFFF

Cinzento: 353535

Cinzento claro: C1C1C1

Cinzento escuro: 211E1E

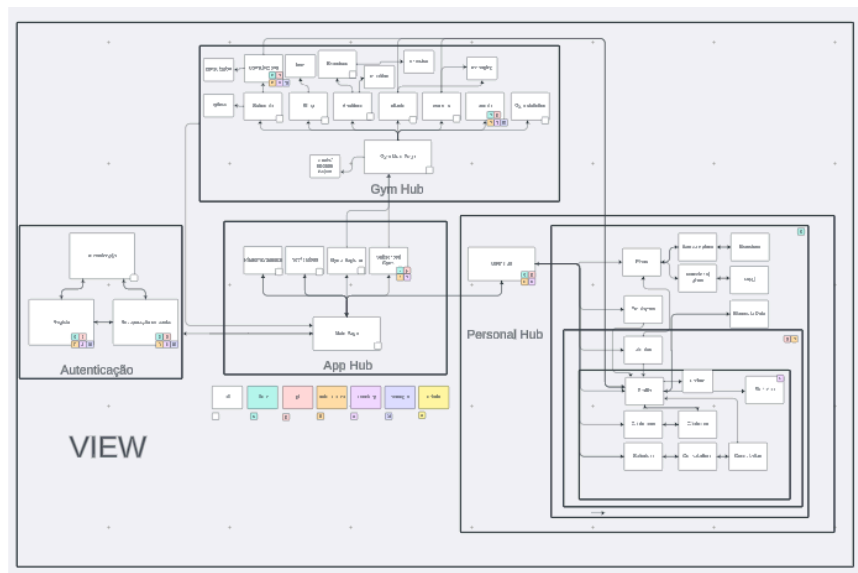
Fontes:

Inter

Logotipos:



Diagrama Geral de Navegação



Matriz de acessos

Páginas	Admin	Manager	Utilizador (roles: Cliente, Pt, Nutricionista e Secretário)
Autenticação	x	x	x
Registo		x	x
Recuperação de conta	x	x	x
Página inicial	x	x	x
Notificações	x	x	x
Explorador de ginásios	x	x	x
Perfil			x
Dados biométricos			x
Gestão Dados biométricos			Pt e Nutricionista
Chat			x
Horários			x
Gestão Horários			x
Planos			Cliente, Pt e Nutricionista
Gestão Planos			Cliente, Pt e Nutricionista
Ginásio	x	x	x

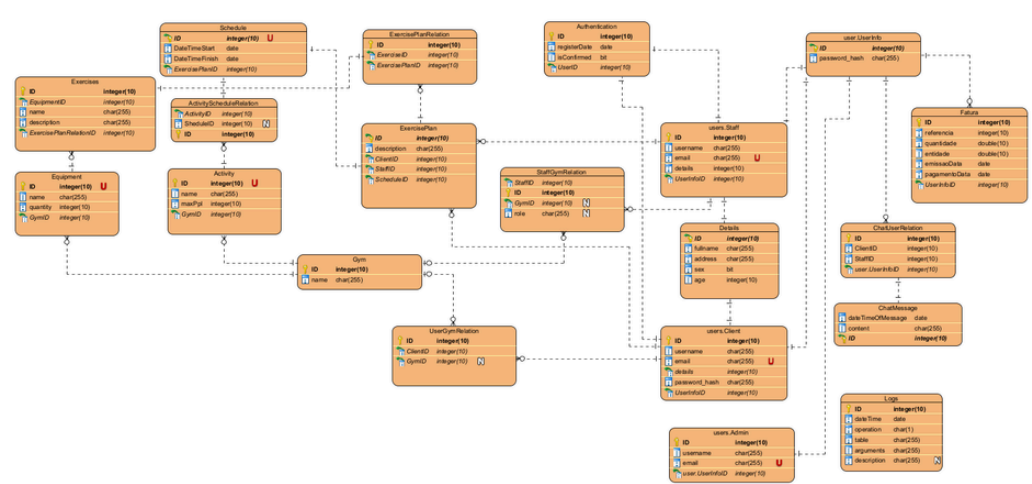
Horários Ginásio	x	x	x
Gestão Horários Ginásio		x	secretário
Loja	x	x	x
Gestão Loja		x	secretário
Maquinas	x	x	x
Gestão Maquinas		x	secretário
Membros	x	x	x
Gestão Membros		x	secretário
Funcionários	x	x	x
Gestão Funcionários		x	secretário

PERSISTÊNCIA

Introdução

A Base de Dados está centrada numa única tabela Gym. A tabela pretende identificar cada ginásio no Sistema unicamente. A tabela users.Admin guarda os dados de utilizadores Admin. A tabela user.UserInfo guarda o identificador dos usuários e as suas palavras-passe encriptada. O Gym terá ligações a tabelas de dados pertinentes ao ginásio: Staff, Client, Activity e Equipment. Staff possui detalhes de um utilizador Staff. Client possui detalhes de um utilizador Client como os seus detalhes pessoais (Details). Activity indica as diferentes atividades/aulas disponíveis para usufruimento no ginásio e os horários (Schedule) para os mesmos. Equipment possui dados que sobre as diferentes máquinas e exercícios disponíveis no ginásio. A tabela ChatMessage visa a guardar dados das mensagens partilhadas entre Users. A tabela ExercisePlan permite que um membro da Staff consiga definir e delegar um plano de exercícios com horários definidos a um Client. Authentication é a tabela usada no registo de um novo User para garantir que o mesmo foi confirmado para inserção no Sistema. Adicionalmente, a Base de Dados possui funcionalidades de Logging com o objetivo de armazenar as diferentes mudanças feitas no Sistema, guardando a data, a operação em si e os seus argumentos (se houverem), uma breve descrição da operação e a tabela em questão.

Modelo Relacional



ARQUITETURA FÍSICA

Introdução

O "Main Server" serve de Hub para a gestão e processamento de dados. O Server possui uma estrutura MVC que serve para organizar as diversas operações do Sistema como o tratamento dos dados, visualização de uma View e operações do tipo CRUD na Base de Dados. O Servidor de Base de dados serve para garantir persistência dos dados e a disponibilidade dos mesmos. É criada uma Base de Dados Relacional para funcionar no Servidor. A Base de Dados do Sistema possui mecanismos de Logging e Backup usando um Servidor de Base de Dados auxiliar para o efeito. No lado do Cliente, o computador interage com o Main Server usando o padrão MVC, podendo visualizar as views das quais possui acesso. Esta arquitetura de instalação promove escalabilidade, robustez e eficiência para qualquer tipo de operação sofisticada no Sistema.

Diagrama de Componentes

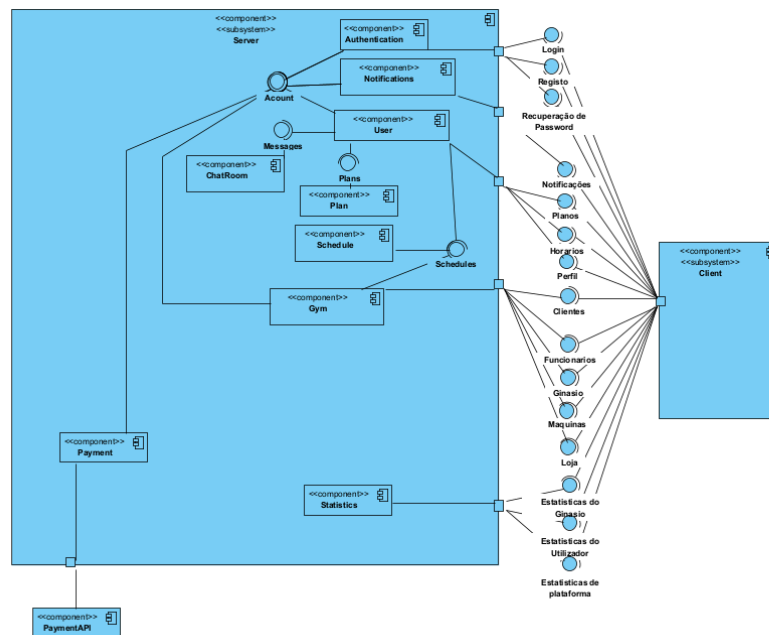
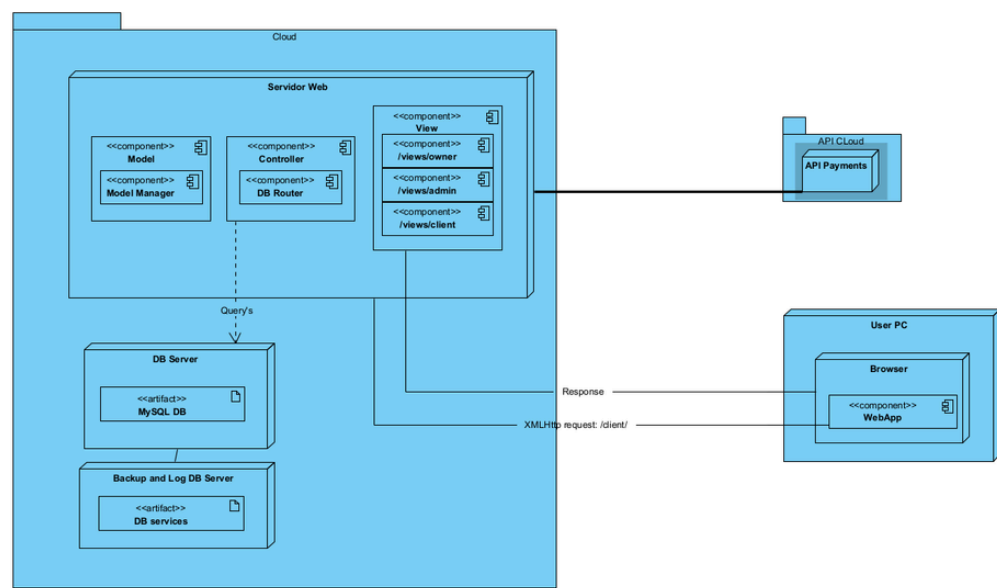


Diagrama de Instalação



NORMAS DE CODIFICAÇÃO DA APLICAÇÃO

1. **Consistência Visual:**

- Garantir que o layout e o estilo das páginas mestras sejam consistentes em todo o aplicativo.
- Adotar uma abordagem modular para facilitar a manutenção e atualização do design.

2. **Nomenclatura Descritiva:**

- Utilizar nomes descritivos para as páginas mestras, de modo que seja fácil compreender sua finalidade e conteúdo.

3. **Hierarquia de Arquivos:**

- Organizar as páginas mestras em uma hierarquia de pastas lógica para facilitar a navegação e a manutenção.

4. **Reusabilidade:**

- Projetar as páginas mestras para serem reutilizáveis em diferentes partes do aplicativo, promovendo eficiência no desenvolvimento.

5. **Comentários Adequados:**

- Incluir comentários explicativos no código para descrever o propósito e o funcionamento de seções críticas das páginas mestras.

6. **Performance:**

- Otimizar o desempenho das páginas mestras, minimizando o uso de recursos e garantindo tempos de carregamento rápidos.

7. **Segurança:**

- Implementar práticas de segurança, especialmente se as páginas mestras que manipulam informações confidenciais ou executam operações críticas.

8. **Testabilidade:**

- Projetar as páginas mestras de forma que seja fácil criar testes para garantir a estabilidade do sistema.

9. **Compatibilidade com Dispositivos:**

- Garantir que as páginas mestras sejam responsivas e compatíveis com diferentes dispositivos e tamanhos de tela.

10. **Documentação:**

- Manter uma documentação atualizada que explique a estrutura, os componentes e as práticas recomendadas associadas às páginas mestras.