

TecHug

Escopo do Projeto

Nosso sistema tem como objetivo, criar uma rede de apoio nas horas necessárias. Ligando interessados em serviços aos prestadores desse tipo de serviço, nesse primeiro momento nosso foco é em quem necessita de profissionais para executar serviços de tecnologia.

Partes Interessadas

Profissional, perfil de usuário que deseja ofertar um serviço no sistema, como uma das partes interessadas do sistema desejo:

Atributos de Qualidade

1. **Usabilidade** - Desejo que o sistema seja fácil de utilizar
2. **Segurança** - Apresentar níveis aceitáveis de risco aos usuários
3. **Disponibilidade** - Suporta vários usuários que desejam ofertar serviços no sistema
4. **Manutenibilidade** - Preciso que o sistema suporte novas categorias de serviços, que possam ser implementadas futuramente.
5. **Tolerância a Falhas** - Capacidade do produto de software de manter um nível de desempenho especificado em casos de defeitos no software ou de violação de sua interface especificada.

Usuário, perfil do usuário que contrata um serviço oferecido por um usuário com perfil Profissional, como uma das partes interessadas do sistema desejo:

Atributos de Qualidade

1. **Usabilidade** - Preciso que o sistema seja fácil de se utilizar.
2. **Segurança** - Apresentar níveis aceitáveis de risco aos usuários
3. **Disponibilidade** - Que o sistema suporte vários acessos ao mesmo.

4. **Tolerância a Falhas** - Capacidade do produto de software de manter um nível de desempenho especificado em casos de defeitos no software ou de violação de sua interface especificada.

Arquitetura Usada

Para atender os requisitos que nossos stakeholders necessitam optamos por uma arquitetura MVC. Olhando para os requisitos disponíveis o estilo de arquitetura MVC é simples e suporta bem os atributos de qualidade propostos já que separa bem os componentes do sistema e por ser um padrão bastante popular acreditamos que a implementação seja facilitada também.

Decisões Arquiteturais

Usabilidade: O sistema vai ser web para que o sistema seja possível de ser acessado por qualquer dispositivo que tenha acesso à internet e navegador.

Segurança: E em conjunto com a camada de autenticação temos outra de validação que opera quando pode checar se a requisição recebida tem os dados e os campos válidos e seguros para podermos realizarmos as ações desejadas pelos usuários.

Disponibilidade: Para contemplarmos este requisito vamos usar containerização o que vai facilitar para que escalemos o sistema horizontalmente com bastante facilidade e talvez futuramente usando alguns orquestradores mais complexos o que possibilitaria deixar o sistema em diversas máquinas.

Manutenibilidade: Como nosso sistema é dividido em partes devido a usar o padrão MVC isso facilita a manutenção pois ao alterarmos uma camada a outra não deve ser afetada a menos que tenha adição de uma nova feature.

Tolerância a falhas: Como usaremos containerização junto com um esquema de réplicas reestabelecer o sistema em casos de falha será simples, pois caso um falhe outro poderá assumir e em casos que será necessário o rollback é possível o fazer sem que o sistema fique offline.

Ponto de vistas e visões da Arquitetura

Ponto de vista	Visão
Projetista	Desenvolvimento/Implementação
Desenvolvedor	Lógica

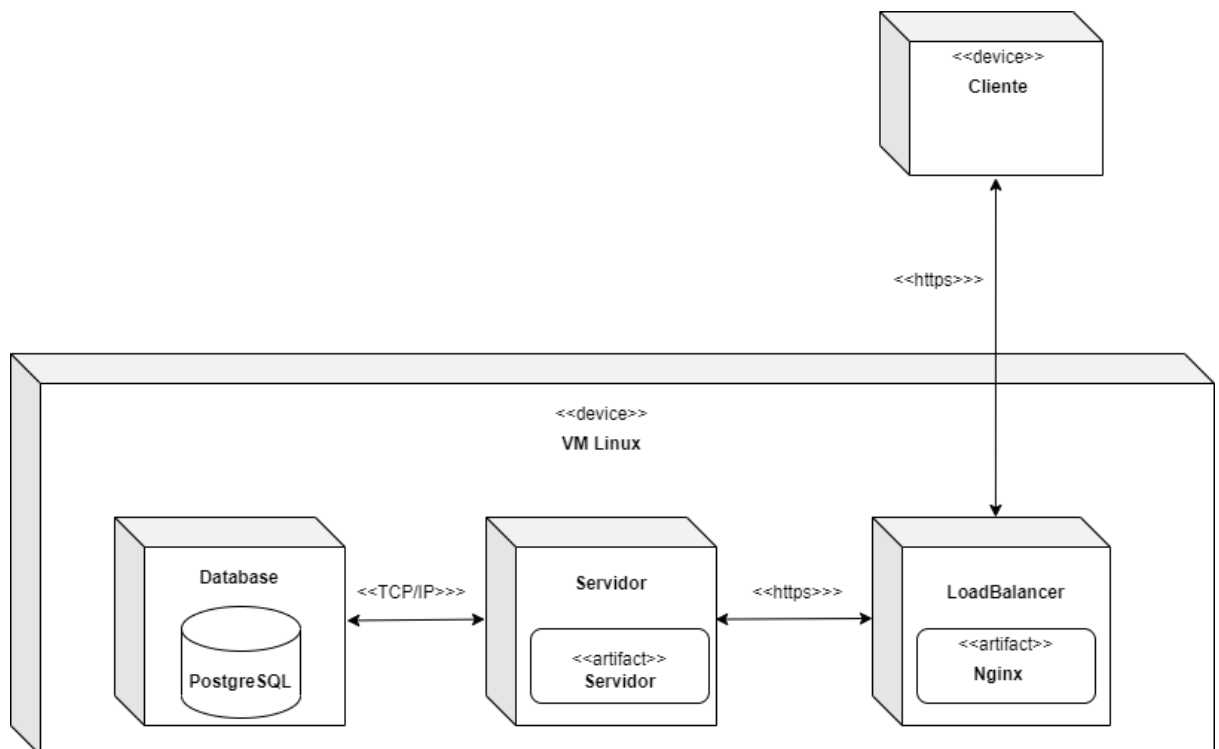
Implantação	Física
Usuário Final/Negócio	Processo
Junção de todos	Cenários ou caso de uso

- **Ponto de vista de Projetista:** Preocupação com as partes principais.
- **Ponto de vista de Desenvolvedor:** Preocupação com a decomposição dos componentes.
- **Ponto de vista de Implantação:** Preocupação com o mapeamento do hardware necessário para a arquitetura.
- **Ponto de vista de Usuário Final/Negócio:** Preocupação com as atividades realizadas pelo usuário final.
- **Ponto de vista de Projetista:** Preocupação com a interação entre as partes e os processos

Visão Física

A visão física é a visão responsável por apresentar o mapeamento entre unidades de software e hardware. Traz uma visão estática do aparato necessário de software e hardware subjacente para permitir que a arquitetura seja funcional.

Diagrama de Implantação

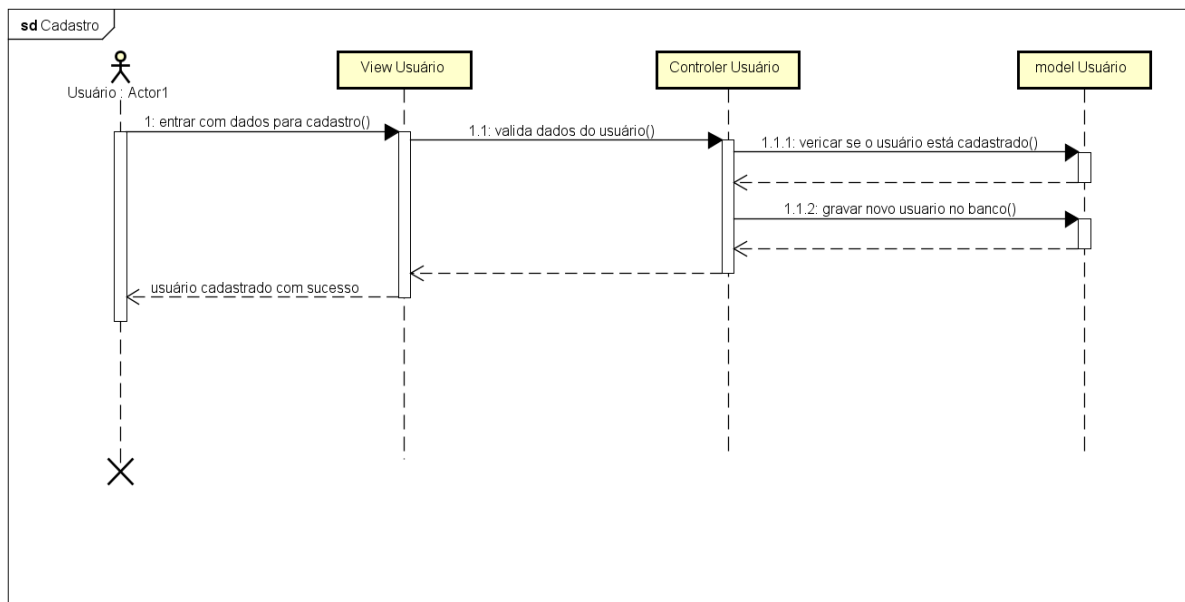


- Cliente: representa o usuário que acessa o sistema TechHug de algum dispositivo com acesso a internet. O acesso se dá através do protocolo HTTPS.
- Database: representa a camada de persistência do sistema, onde os dados serão persistidos. O acesso do banco de dados com o servidor se dá através do protocolo TCP/IP.
- Servidor: servidor que executa a aplicação TechHug.
- LoadBalancer: que tem como função manter a estabilidade de um servidor quando o tráfego e volume de dados é muito grande.

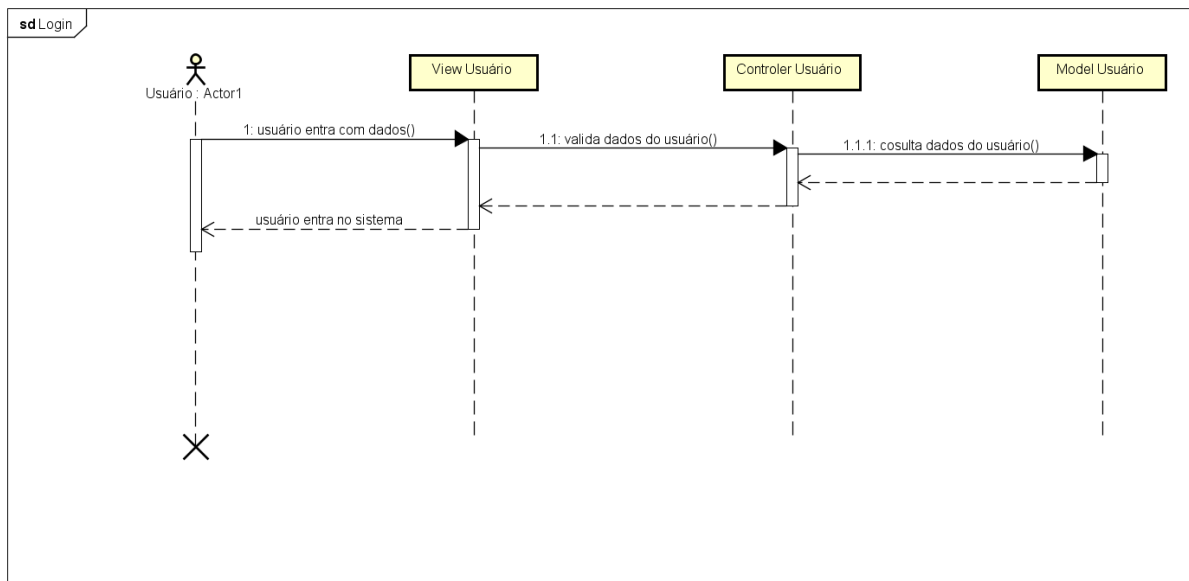
Visão de Processos

A Visão de Processos vem mostrar um pouco de como será o fluxo de comunicação entre processos, passagem de mensagens, atividades entre componentes e sequência de mensagens. Aqui iremos descrever os principais casos de uso em seu fluxo principal sem exceções ou fluxos alternativos;

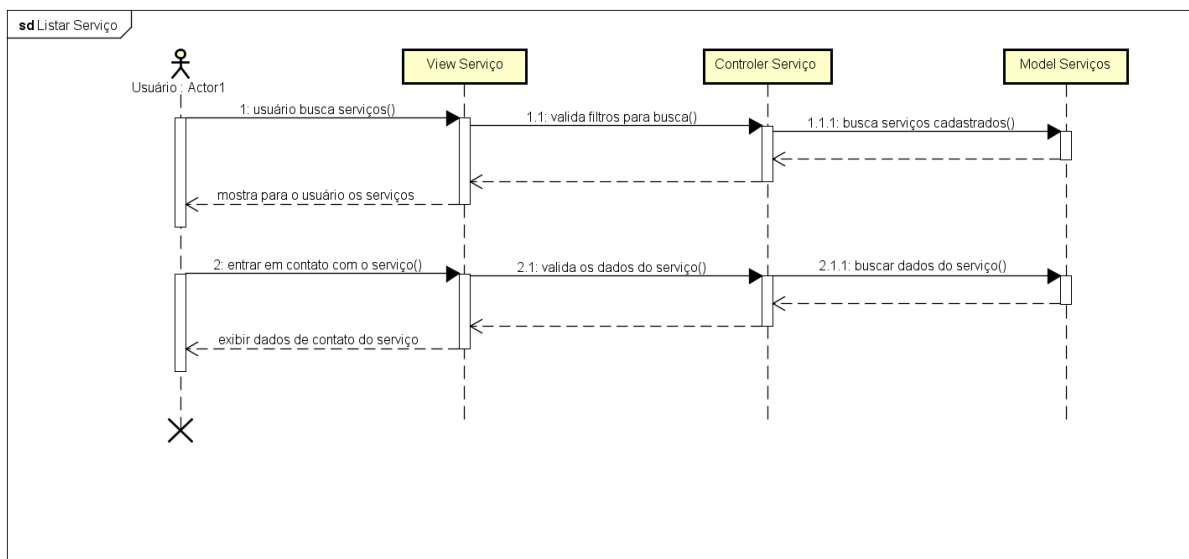
Cadastro



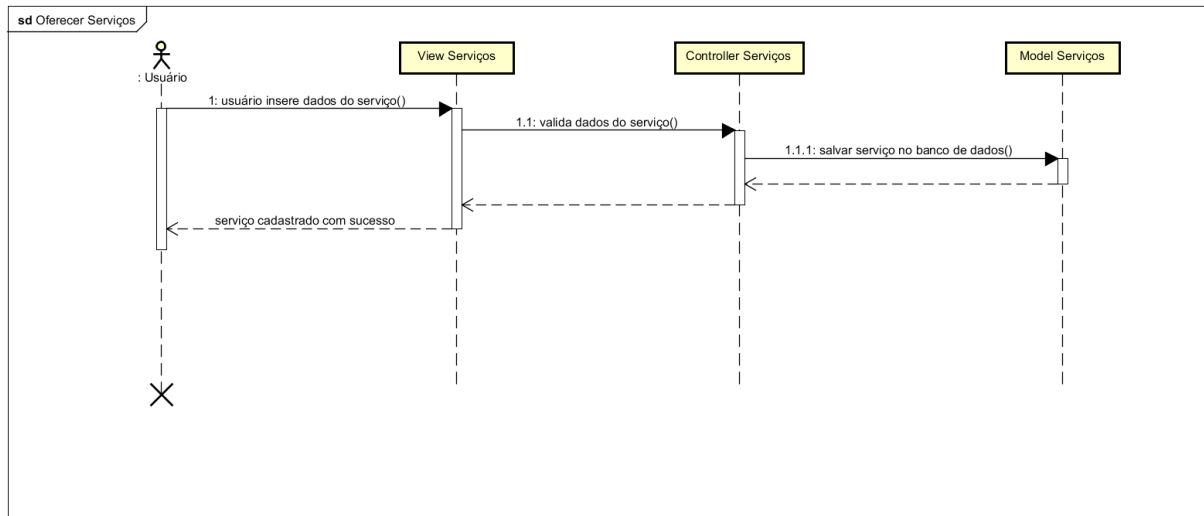
Realizar Login



Listar Serviços



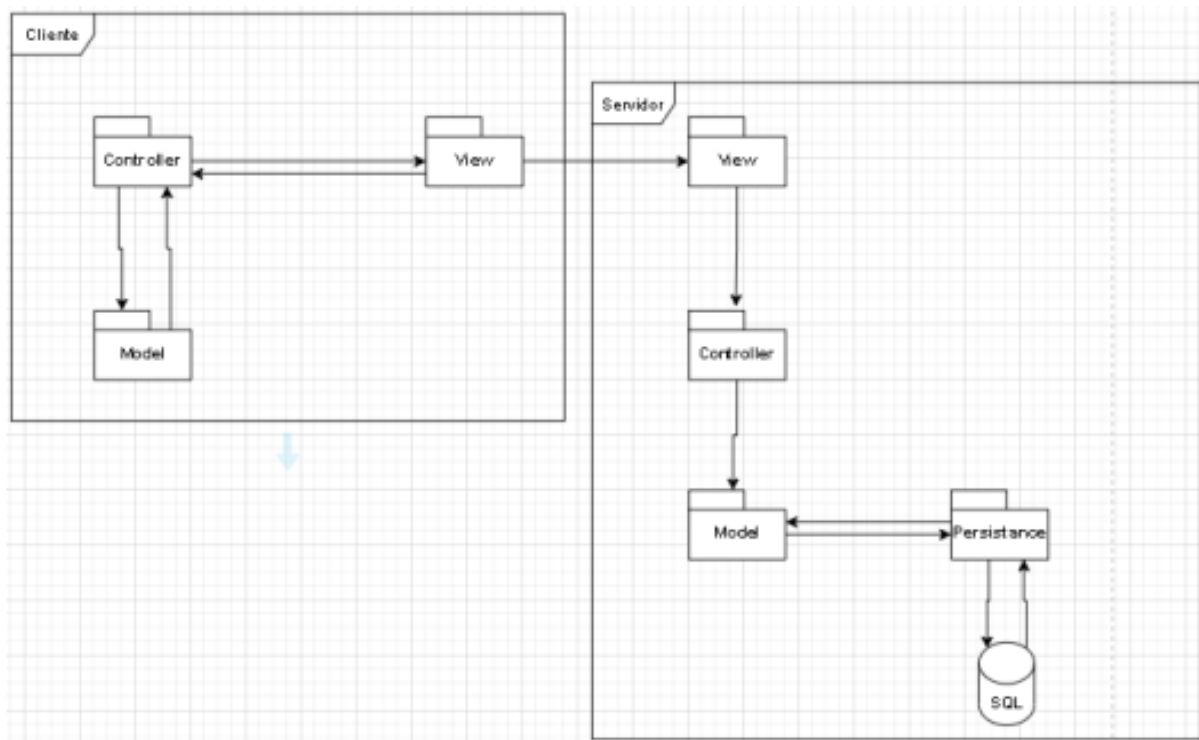
Oferecer Serviços



Essas visões foram pensadas para contemplar questões de usabilidade também, tentando simplificar o processo de uso das funcionalidades do sistema.

Visão de desenvolvimento

A visão lógica define a estrutura da arquitetura. Abaixo será especificado o padrão utilizado para o desenvolvimento do sistema, no caso, MVC.



- *View*: Componente que contém as visões do projeto, ou seja, as interfaces, formulários e etc, além de trocar as informações com a View entre Cliente e Servidor;
- *Controller*: Componente que recebe as informações e requisições do pacote *view* e os despacha para devida classe de controle, o *controller* se comunica com classes *Model* que contém as regras de negócio do sistema;
- *Model*: pacote que recebe as informações e requisições do pacote *controller* e os atribui de acordo com as regras de negócio , às classes *Persistence* para devida troca de informações.
- *Persistence*: pacote que recebe as requisições e realiza as operações relacionadas ao Banco de Dados, utilizando as informações das classes *Model*. O pacote *persistence* também pode enviar dados para o pacote *Model*. Para repasse da informação para as demais classes do lado Cliente.

Classes de Design Significativos do Ponto de Vista da Arquitetura

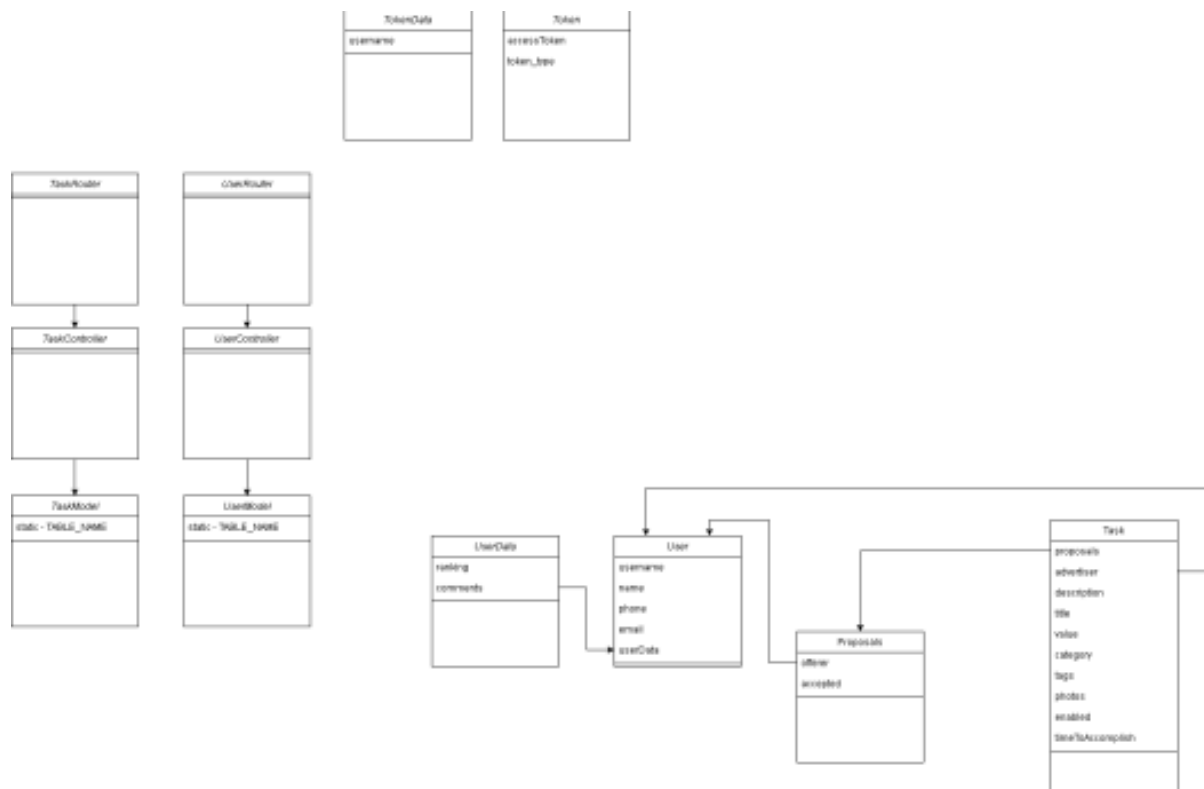
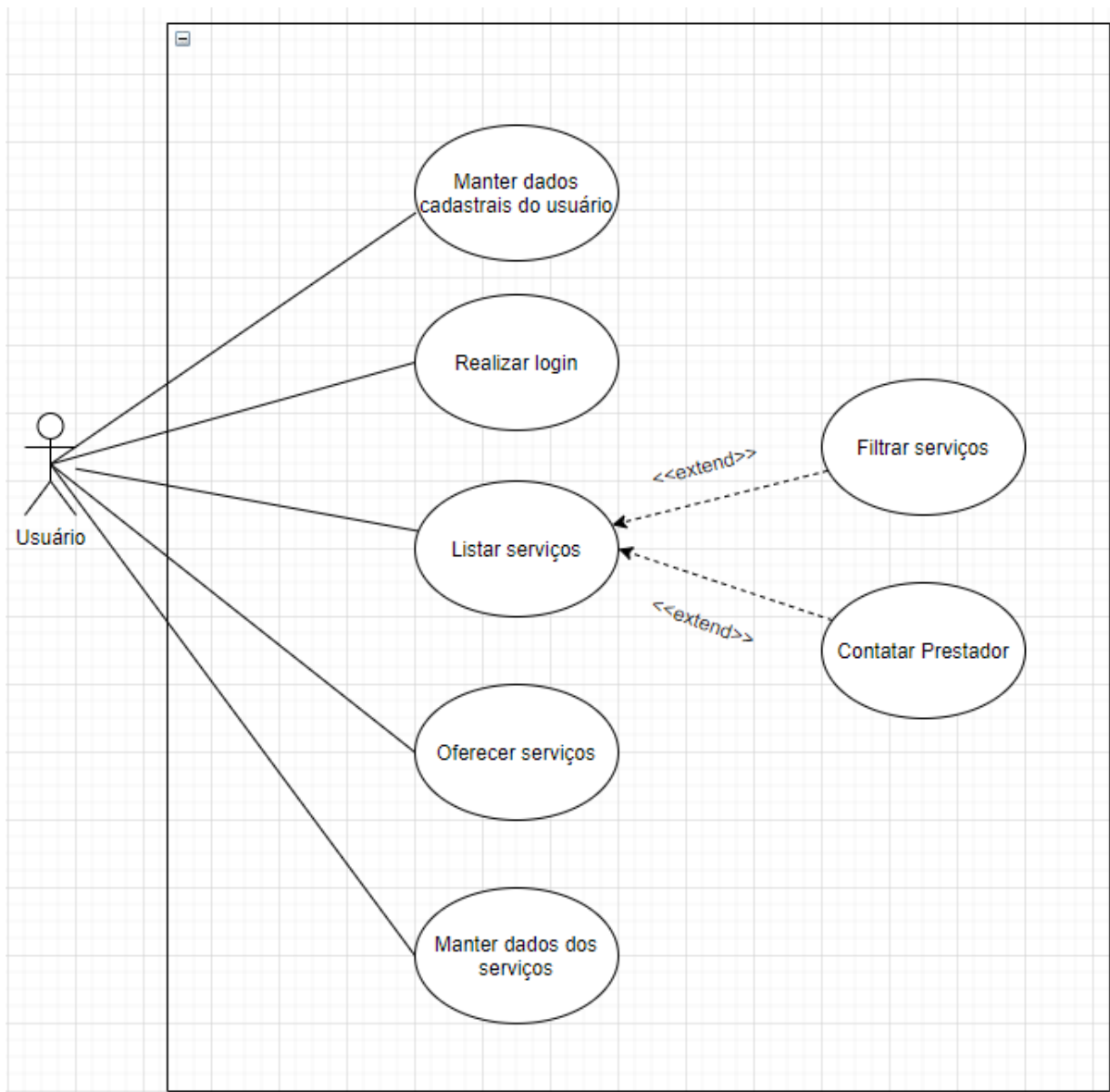


Diagrama de Classe geral do sistema.

Visão Lógica

Para a visão lógica foi utilizado um diagrama de classes UML com um padrão MVC com as respectivas classes que precisaremos para fazer a lógica de negócio e a autenticação usando tokens.



O diagrama de casos de uso representado acima, descreve quem são os atores do sistema TechHug (neste caso, existe apenas 1 (um) ator) e quais são suas principais interações com o sistema: Manter dados cadastrais do usuário, Realizar Login, Listar serviços (em que o ator pode Filtrar Serviços ou Contratar Prestador), Oferecer serviços e Manter dados dos serviços. Estas interações/funcionalidades, norteiam todas as outras visões arquiteturais.