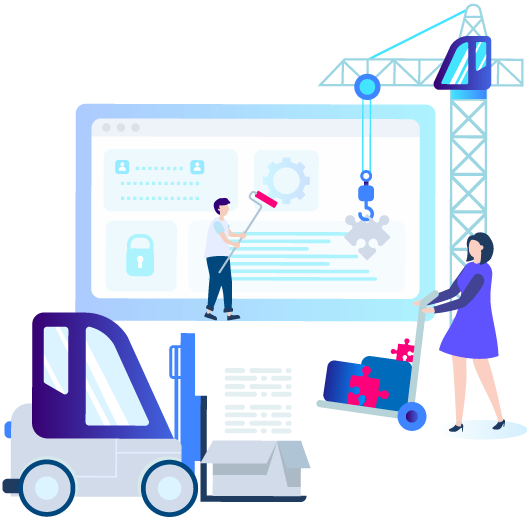


Relatório do trabalho prático

Projeto de Desenvolvimento de Software

Tema: Simulação de orçamentos de obras



Licenciatura de Engenharia de Sistemas Informáticos (pós-laboral) – 2ºAno

Luís Esteves – 16960

Luís Silva – 14289

Pedro Costa – 13747

Luís Oliveira – 18851

Gonçalo dos Santos - 11359

# Índice

[Índice 2](#_Toc134810999)

[1. Atividade 4](#_Toc134811000)

[1.1. Empresa 4](#_Toc134811001)

[1.2. Projeto 4](#_Toc134811002)

[1.3. Objetivos de negócio 4](#_Toc134811003)

[1.4. Mais-valias de fazer uma plataforma web 5](#_Toc134811004)

[2. Requisitos Funcionais 6](#_Toc134811005)

[3. Requisitos Não Funcionais 7](#_Toc134811006)

[4. Regras de Negócio e Restrições 8](#_Toc134811007)

[4.1. Regras de Negócio 8](#_Toc134811008)

[4.2. Pressupostos e Restrições 9](#_Toc134811009)

[5. Product Backlog 9](#_Toc134811010)

[6. Diagramas UML 11](#_Toc134811011)

[6.1. Diagramas de Casos de Uso 11](#_Toc134811012)

[6.2. Diagrama de Classes 14](#_Toc134811013)

[6.3. Diagrama Entidade-Relação 15](#_Toc134811014)

[6.4. Diagrama BPMN 16](#_Toc134811015)

[6.5. Diagramas de Contexto 17](#_Toc134811016)

[6.6. Arquitetura técnica 18](#_Toc134811017)

[6.7. Diagramas de Sequência 19](#_Toc134811018)

[7. Mockups 20](#_Toc134811019)

**Índice de imagens**

[Figura 1 - Caso de Uso – Login 12](#_Toc129551241)

[Figura 2 - Caso de Uso – Trabalho 12](#_Toc129551242)

[Figura 3 - Caso de Uso - Reunião 13](#_Toc129551243)

[Figura 4 - Diagrama de Classes 14](#_Toc129551244)

[Figura 5 - Diagrama Entidade-Relação 15](#_Toc129551245)

[Figura 6 – Simulador 16](#_Toc129551246)

[Figura 7 – Orçamentação 16](#_Toc129551247)

[Figura 8 - Estatísticas 17](#_Toc129551248)

[Figura 9 - Diagrama de Contexto 17](#_Toc129551249)

[Figura 10: Arquitetura técnica 18](#_Toc129551250)

Figura 11,12,13: Diagramas de Sequência ………………………………………………………………………………19

# Atividade

## Empresa

O objetivo deste negócio insere-se na área de construção, reconstrução e orçamentação de imóveis. A empresa com o nome *Building Bug* quer inovar nesta área, desenvolvendo um projeto que consiste numa página web com um sistema mais rápido, prático e menos burocrático, fazendo com que este novo serviço se torne a principal plataforma online para clientes que procuram ajuda para construir o seu novo lar.

O sistema vai permitir que os utilizadores possam simular e/ou pedir orçamentos para pequenas e médias obras que pretendem realizar em casa.

Através da mesma plataforma, profissionais técnicos relacionados com o tema, como empreiteiros, eletricistas, canalizadores, decoradores, encontram oportunidades de obras para realizar, pois vão ser os mesmos a trabalhar para que os lares dos nossos clientes sejam criados.

Assim *a Building Bug* vai fornecer um serviço conveniente e eficiente para utilizadores de todo o país, facultando orçamentos de alta qualidade e soluções personalizadas para atender às necessidades específicas de cada utilizador.

## Projeto

O projeto apelida-se de “Build.Me”. Este vai consistir numa página web que fornece orçamentos para construções/reconstruções de casas em pequena e média escala, unindo o cliente e as empresas especializadas na área numa única plataforma, fornecendo serviços de alta qualidade e altamente personalizados a cada cliente, conseguindo empregar pessoas especializadas na área da construção civil e construindo a casa dos sonhos dos nossos clientes.

## Objetivos de negócio

* Fornecer serviços de alta qualidade - O principal objetivo é oferecer serviços de construção e reconstrução de alta qualidade que correspondam às expectativas e necessidades dos utilizadores.
* Aumentar a visibilidade online - A plataforma deve trabalhar para aumentar a sua visibilidade online, através de publicidade e marketing digital (*Meta, discord, twitter* "anúncios") para atrair um grande número de utilizadores.
* Atrair novos clientes - A plataforma deve trabalhar para atrair novos utilizadores, fornecendo informações úteis e claras e objetivas sobre os serviços oferecidos e também criar parcerias com outras empresas relacionadas.
* Manter relacionamentos com utilizadores existentes - A plataforma deve manter contato com os utilizadores existentes para entender suas necessidades, e garantir que o utilizador tenha uma boa experiência.
* Expandir serviços- A plataforma deve considerar a possibilidade de expandir a sua oferta de serviços para atender a novas necessidades e espectativas dos utilizadores e assim poder aumentar o valor do negócio.
* Eficiência- A plataforma deve aperfeiçoar continuamente todos os seus processos para garantir a eficiência na entrega dos projetos, bem como a redução de custos.
* Adotar práticas sustentáveis - A plataforma deve adotar práticas sustentáveis em todos os seus projetos, reduzindo o impacto ambiental e contribuindo para a construção de um futuro mais sustentável.

## Mais-valias de fazer uma plataforma web

* Acesso conveniente- A plataforma tem de fornecer um serviço conveniente e fácil de usar para os clientes, permitindo que façam orçamentos online a qualquer momento e em qualquer lugar (fácil e rápido).
* Suporte- A plataforma web pode fornecer um serviço de suporte em tempo real rápido e eficaz.
* Redução de custos- A plataforma pode ajudar a reduzir os custos, pois não há necessidade de deslocamento para fazer orçamentos ou reuniões com os clientes.
* Aumento da eficiência- A plataforma permite que a empresa forneça um serviço mais eficiente e rápido para os clientes, reduzindo o tempo de espera e aumentando a satisfação do cliente.
* Alcance a nível nacional- A plataforma permite que a empresa alcance clientes em todo o país, expandindo a base de utilizadores e aumentando o potencial de negócios.
* Personalização dos serviços- A plataforma pode fornecer soluções personalizadas para atender às necessidades específicas dos utilizadores, aumentando a satisfação e fidelidade dos mesmos.
* Sustentabilidade- A plataforma pode ajudar a empresa a adotar práticas sustentáveis em seus projetos, reduzindo o impacto ambiental

.

* Segurança: A plataforma web pode ser protegida com medidas de segurança, como autenticação e criptografia de dados, para garantir a privacidade dos utilizadors e a segurança dos dados.

# Requisitos Funcionais

Os requisitos funcionais são uma especificação das funcionalidades e comportamentos esperados do sistema em questão. Eles descrevem as ações que o sistema deve executar, como ele se comporta e como deve responder a diferentes entradas de dados ou solicitações dos utilizadores.

Estes requisitos estão diretamente relacionados às funcionalidades principais do sistema e ao seu propósito principal. Eles são importantes para garantir que o sistema atenda às necessidades do utilizador e cumpra seus objetivos principais de forma eficiente e eficaz.

Em seguida mencionamos os Requisitos Funcionais deste trabalho:

* **Registo de Utilizador**: O sistema deve permitir que os utilizadores criem uma conta facilmente, inserindo informações pessoais básicas.
* **Registo de Especialista/Equipa**: O sistema deve permitir que os especialistas ou equipas se registem e criem um perfil detalhado, incluindo informações de contato, especialidades e experiência profissional.
* **Seleção de Zona de Construção**: O sistema deve fornecer um formulário de seleção de zona de construção que permita ao utilizador especificar a área geográfica em que deseja realizar o projeto assim como a zona que pretende trabalhar (Ex: cozinha, sala, etc.).
* **Orçamento Aproximado**: O sistema deve permitir que o utilizador faça um orçamento aproximado do projeto, inserir informações básicas sobre o propósito do trabalho.
* **Agendamento de Reuniões**: O sistema deve permitir que o utilizador agende reuniões entre o cliente e o especialista/equipe, facilitando a comunicação e a colaboração entre as partes.
* **Atas de Reuniões**: O sistema deve gerar atas de reuniões detalhadas, incluindo todas as decisões e acordos discutidos durante o encontro.
* **Formalização de Contrato**: O sistema deve fornecer um formulário para formalização do contrato entre as partes, com uma opção de assinatura digital validada para maior segurança.
* **Suporte por Formulário**: O sistema deve fornecer um canal de suporte por formulário para que os utilizadores possam enviar suas dúvidas e problemas e receber uma resposta adequada.

Todos estes requisitos estão relacionados com as funções que o software deve executar para atender às necessidades do utilizador.

# Requisitos Não Funcionais

Requisitos não funcionais, estão relacionados à qualidade do sistema como um todo. Estes requisitos incluem aspetos como usabilidade, confiabilidade, desempenho, segurança, escalabilidade, manutenção, portabilidade e integração.

Eles são importantes para garantir que o sistema seja eficiente, seguro, fácil de usar e manter, sendo assim capaz de atender às necessidades dos utilizadores de forma flexível e adaptável. Portanto, é fundamental considerar ambos os tipos de requisitos durante a análise e desenvolvimento do projeto para garantir a qualidade do produto final.

Mencionamos em seguida os Requisitos não Funcionais deste projeto:

**1-Usabilidade:** O sistema tem de ser fácil de usar e intuitivo para que qualquer pessoa que tenha tido um valor de 80% num survey que foi disponibilizado ao utilizador de forma que possa utilizá-lo sem dificuldades.

**2-Confiabilidade:** O sistema deve ser capaz de funcionar com uma percentagem de down time de 2% anualmente (previsto para atualizações ou qualquer falha no sistema “bug” que precise de ser reparada).

**3-Desempenho:** O sistema deve ser rápido e eficiente, permitindo que o utilizador faça o login, navegue e utilize as funções sem ser lento e de uma maneira intuitiva.

**4-Segurança**: O sistema deve ser seguro e proteger as informações do utilizador contra possíveis ameaças, desvio de dados pessoais, etc.

**5-Escalabilidade:** O sistema deve ser capaz de se adaptar a um aumento na quantidade de utilizadores sem afetar o seu desempenho nem as suas funcionalidades.

**6-Manutenção:** O sistema deve ser fácil de manter e atualizar, permitindo correções de erros e adição de novas funcionalidades de forma simples e prática.

**7-Portabilidade:** O sistema deve ser compatível com diferentes plataformas e dispositivos, (pelo menos Android 11 ou superior em telemóvel e no computador os requisitos mínimos Celeron, 8Gb RAM e Windows 10 ou superior) permitindo que este seja acessado de forma flexível pelos utilizadores.

**8-Integração:** O sistema deve ser capaz de se integrar com outros sistemas e serviços através de uma API, permitindo que os utilizadores acessem outras funcionalidades relacionadas de forma fácil e rápida e eficaz.

# Regras de Negócio e Restrições

Neste capítulo é explicado em detalhe as regras de negócio e restrições que tencionamos implementar neste projeto.

## Regras de Negócio

As regras de negócio são um conjunto de instruções que definem como uma organização deve operar. Estabelecem as políticas, práticas e procedimentos que orientam as atividades de uma empresa, desde as operações diárias até as decisões estratégicas de longo prazo.

Estas regras são criadas para garantir que a empresa opere de forma consistente e eficiente, seguindo padrões éticos e sempre dentro da lei. Elas são criadas com base na experiência e conhecimento da empresa, bem como em práticas e regulamentações do mercado.

As regras de negócio podem abranger uma ampla variedade de áreas, como produção, finanças, recursos humanos, marketing, vendas, atendimento a clientes e muito mais. Estas podem ser simples ou complexas, dependendo das necessidades da empresa e dos desafios enfrentados.

Ao implementar regras de negócio bem definidas e estruturadas, as empresas podem garantir uma operação mais eficiente e consistente, reduzir erros e retrabalho, melhorar a qualidade dos produtos e serviços, aumentar a satisfação do cliente e até mesmo aumentar a rentabilidade do negócio.

Demonstramos em seguida as regras de negócio que pretendemos implementar no projeto:

* Apenas utilizadores registados podem criar projetos de construção e solicitar serviços de especialistas/equipes.
* Especialistas/equipas registados/as devem ter experiência e especialização adequadas para os projetos que se candidatarem.
* Os utilizadores devem selecionar a zona de construção correta para garantir que os especialistas/equipas disponíveis tenham conhecimento e experiência adequados na área.
* O orçamento aproximado fornecido pelo utilizador deve ser realista e baseado em informações precisas para evitar problemas futuros.
* As reuniões agendadas devem ser realizadas no horário e local combinados para garantir a eficiência e a satisfação do cliente.
* As atas de reuniões devem ser completas e precisas, registando todas as decisões e acordos acordados entre as partes envolvidas.
* O contrato formal deve ser elaborado e assinado por ambas as partes antes do início do projeto para garantir a clareza das expectativas e responsabilidades de cada parte.
* O suporte por formulário deve ser respondido dentro de um prazo razoável para garantir a satisfação do cliente e a resolução de quaisquer problemas ou dúvidas.

## Pressupostos e Restrições

Neste subcapítulo mencionamos os pressupostos e restrições que vamos aplicar neste projeto.

* **PR01 — *Open Source*** — A equipa usará tecnologia de código aberto, tanto quanto possível.
* **PR02 — Idiomas** — A aplicação suportará os idiomas português e inglês.
* **PR03 — Proteção de Dados** — Toda a informação pessoal divulgada tem de ser autorizada.

# Product Backlog

Neste capítulo iremos mencionar a lista de tarefas, de serviços ou de funcionalidades que temos de implementar, ou seja, *o Product Backlog* deste projeto.

Em seguida, temos uma tabela a estabelecer quais as funcionalidades que queremos então implementar e a sua prioridade perante este trabalho.

|  |  |  |
| --- | --- | --- |
| ID | Nome da Tarefa | Prioridade |
| 0. | Criar conta como cliente | 1 |
|  | Criar conta como prestador de serviços | 1 |
|  | Fazer login com email e password | 1 |
|  | Recuperar password | 2 |
|  | Alterar dados da conta | 2 |
|  | Registar um pedido de orçamento | 1 |
|  | Visualizar estado do pedido | 2 |
|  | Editar o pedido de orçamento | 2 |
|  | Visualizar dashboard com detalhes do pedido | 3 |
|  | Visualizar os dados da conta | 2 |
|  | Selecionar um ou vários serviços | 1 |
|  | Acompanhar em tempo real o progresso/estado do serviço | 2 |
|  | Subscrever novos serviços | 1 |
|  | Avaliar o trabalho e publicar resultado | 3 |
|  | Receber notificações do progresso | 2 |
|  | Marcar empresa que realizou o serviço como favorita | 4 |
|  | Adicionar fotos e projetos que já realizou | 4 |
|  | Aceitar ou rejeitar serviços | 2 |
|  | Visualizar pedidos aceites | 2 |
|  | Visualizar pedidos rejeitados | 2 |
|  | Visualizar pedidos concluídos | 2 |
|  | Visualizar progresso do seu serviço | 3 |
|  | Alterar estado do serviço | 3 |
|  | Gerir contas de clientes/outsourcing | 3 |
|  | Visualizar dashboard com todos os serviços (estado, progresso, …) | 3 |
|  | Receber mensagens suporte | 4 |

*Tabela 1: Product Backlog*

Nota: As avaliações das prioridades das tarefas do *Product Backlog* são avaliadas de 1 a 5.

# Diagramas UML

## Diagramas de Casos de Uso

Os casos de uso são uma forma de entender como um sistema funciona. Eles ajudam a identificar o que o sistema precisa fazer e como ele deve ser utilizado pelos utilizadores. São como um guião que descreve diferentes situações em que o sistema pode ser usado.

Um caso de uso geralmente tem uma história que conta como o sistema pode ser usado numa situação específica. Existem também fluxos de eventos que explicam o que o sistema fará em cada etapa da história, e diagramas que mostram a interação entre o sistema e os utilizadores.

Usar casos de uso é importante porque garante que o sistema seja construído para atender às necessidades dos utilizadores. Eles ajudam a comunicar as ideias do sistema entre as equipas de desenvolvimento, clientes e utilizadores finais. Também ajuda a documentar como o sistema funciona ao longo do tempo, o que é muito útil para manutenção e atualizações.

Nos casos de uso identificamos 3 atores principais, sendo eles o *User*, a *Company* (empresa) e o *System*. O *User* é utilizador de sistema que pretende algum serviço feito dentro do nosso sistema, a empresa pode tomar ações sobre o sistema (como por exemplo enviar propostas para um determinado pedido de um utilizador) e, por último, o *system* que é responsável por validações e notificações para estes dois atores anteriormente referidos.

Em seguida demonstramos os Diagramas de Casos de Uso deste trabalho:

Diagram

Description automatically generated

Figura - Caso de Uso – Login

Diagram

Description automatically generated

Figura - Caso de Uso – Trabalho

Diagram

Description automatically generated

Figura - Caso de Uso - Reunião

## Diagrama de Classes

Os diagramas de classe são fundamentais para o processo de modelagem de objetos e modelam a estrutura estática de um sistema. Dependendo da complexidade de um sistema, é possível utilizar um único diagrama de classe para modelar um sistema inteiro ou vários diagramas de classe para modelar os componentes de um sistema.

Um diagrama de classe é uma representação visual de objetos de classe em um sistema de modelo, categorizado por tipos de classe. Cada tipo de classe é representado como um retângulo com três compartimentos para o nome da classe, atributos e operações.

Em seguida demonstramos o Diagrama de Classes utilizado:

Diagram

Description automatically generated

Figura - Diagrama de Classes

## Diagrama Entidade-Relação

Um diagrama Entidade-Relação (ER) é um diagrama que mostra como as entidades interagem entre si, sendo um dos diagramas mais importantes porque irá ser a base para a criação da base de dados que terá toda a informação do sistema contida dentro de si.

É um diagrama que mostra como os dados fluem de uma entidade para a outra, sendo um formato fácil para qualquer *developer* ou designer para acompanhar. Este diagrama é também responsável por referir qual é o campo principal em cada tabela, assim como identificar quais são os campos referentes a tabelas externas.

Em seguida demonstramos o Diagrama de Entidade-Relação deste trabalho:

Diagram

Description automatically generated

Figura - Diagrama Entidade-Relação

## Diagrama BPMN

Os diagramas BPMN, são uma ferramenta útil para exibir um processo de negócios interno. A meta do BPMN é apresentar uma representação visual de um procedimento de negócios específico para que seja compreensível a todas as partes interessadas. Desta forma, diferentes áreas da empresa podem visualizar as colaborações e transações que ocorrem dentro da organização.

Em seguida demonstramos os Diagramas BPMN deste trabalho:

Diagram

Description automatically generated

Figura – Simulador

Diagram

Description automatically generated with medium confidence

Figura – Orçamentação

Shape

Description automatically generated with medium confidence

Figura - Estatísticas

## Diagramas de Contexto

O Diagrama de contexto é composto por um fluxo de dados que mostra as diferentes interfaces entre o projeto e como eles vão interagir com o sistema que vai ser elaborado. Este diagrama pode ser considerado como uma forma mais aprofundada do diagrama de fluxo de dados.

Em geral, este diagrama pode ser uma movimentação gráfica dos movimentos dos dados através de um sistema, sendo que também pode capturar e comunicar as interações e os fluxos desses dados.

Em seguida demonstramos o Diagrama de Contexto deste trabalho:

A picture containing logo

Description automatically generated

Figura - Diagrama de Contexto

## Arquitetura técnica

O nosso grupo decidiu seguir a seguinte arquitetura para este trabalho:

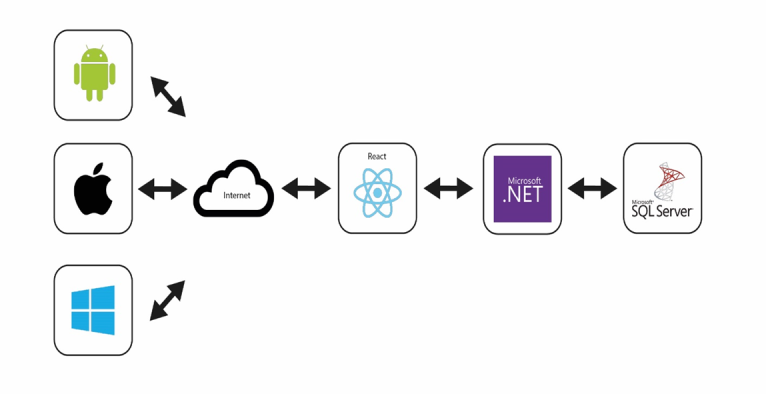


Figura : Arquitetura técnica

O serviço a desenvolver vai estar disponível em formato página web. O *front-end* vai ser desenvolvido em *Angular*. O *back-end* com Microsoft .Net e a base de dados vai ser desenvolvida em SQL Server.

## Diagramas de Sequência

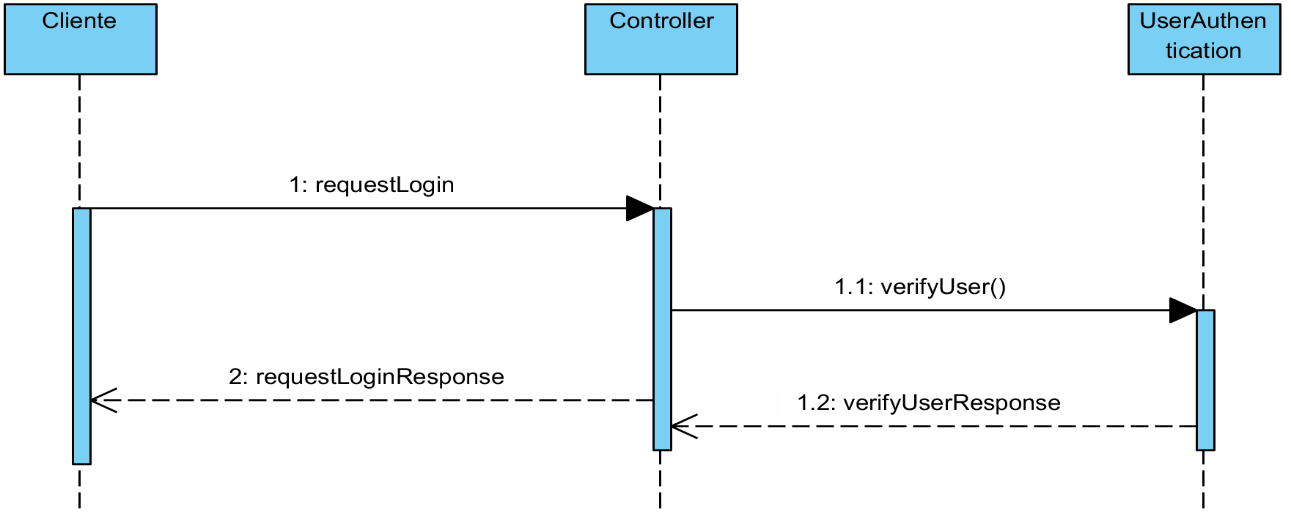


Figura - Diagrama de Sequência para Caso de Uso 1

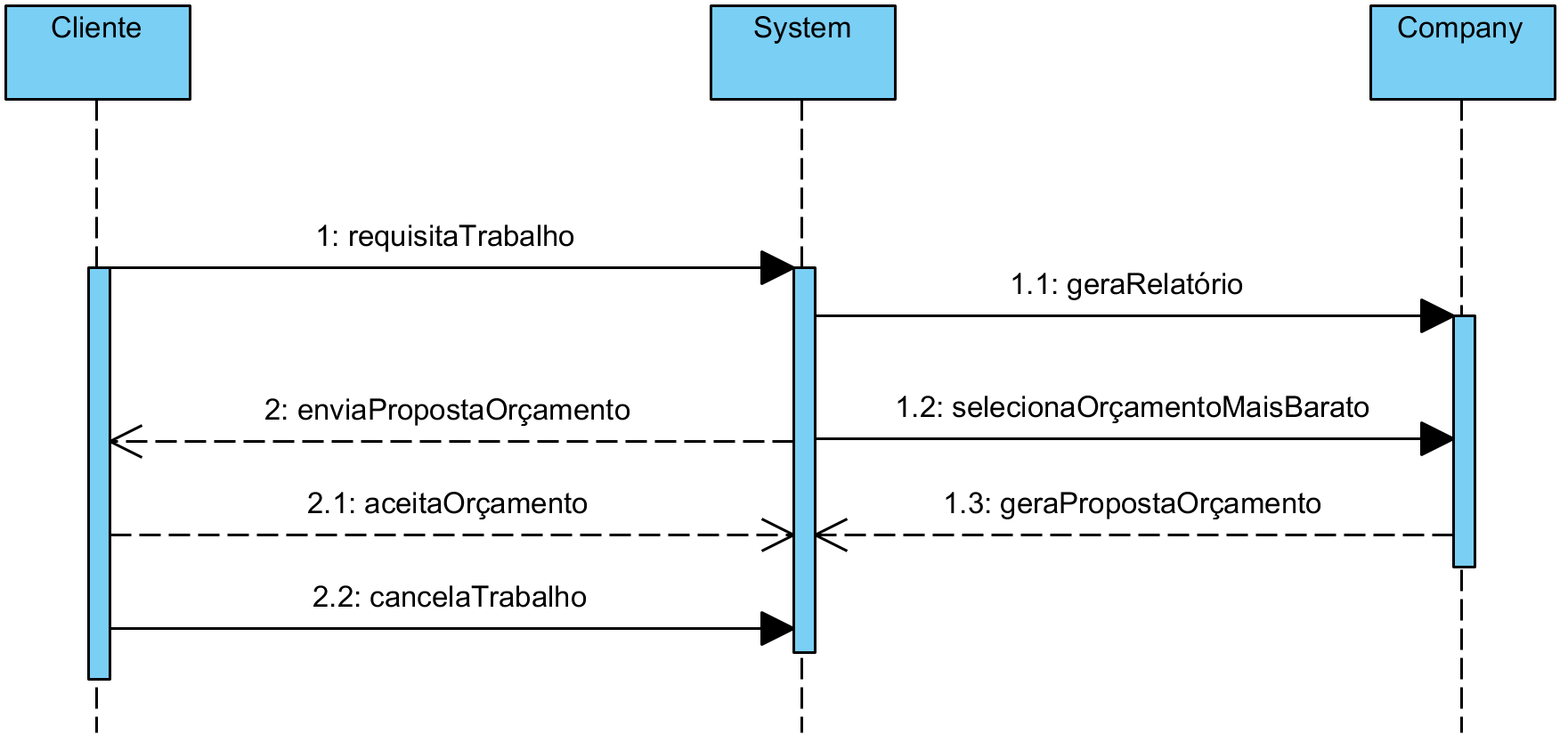


Figura - Diagrama de Sequência para Caso de Uso 2

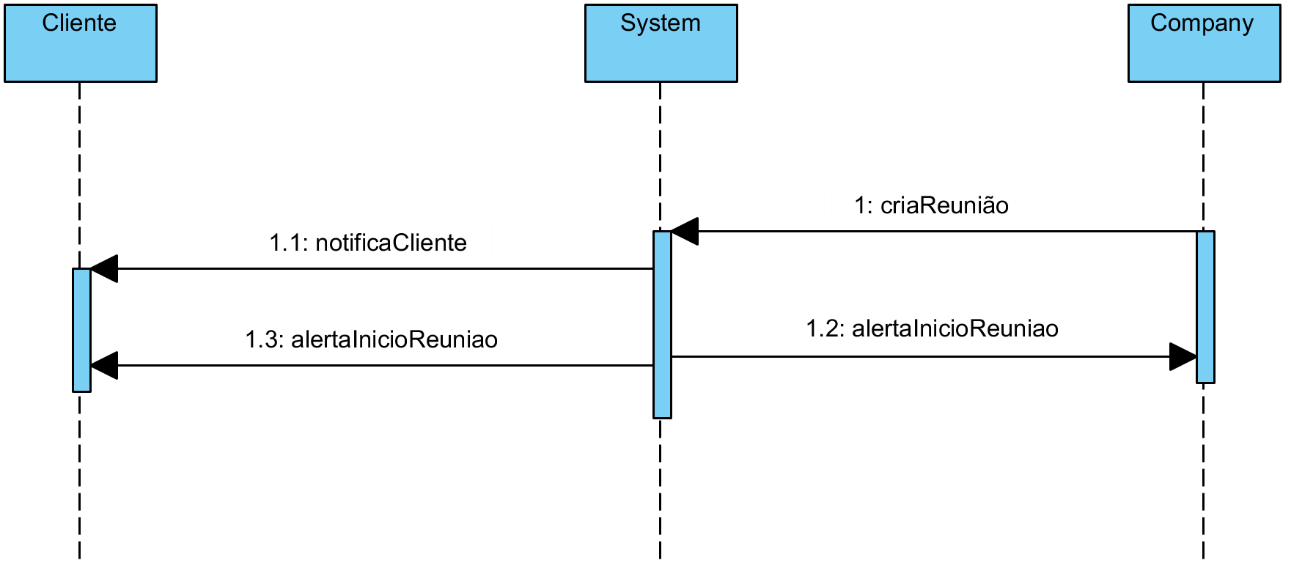


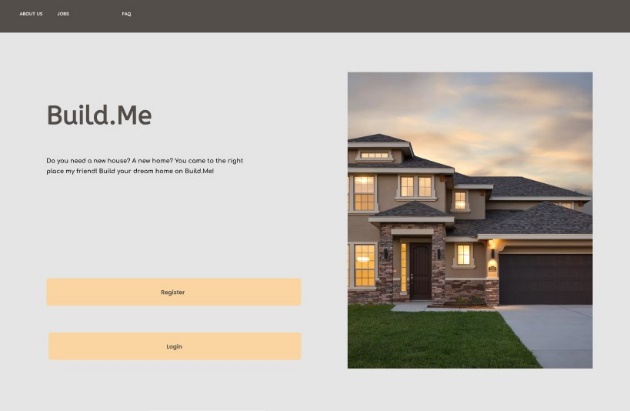
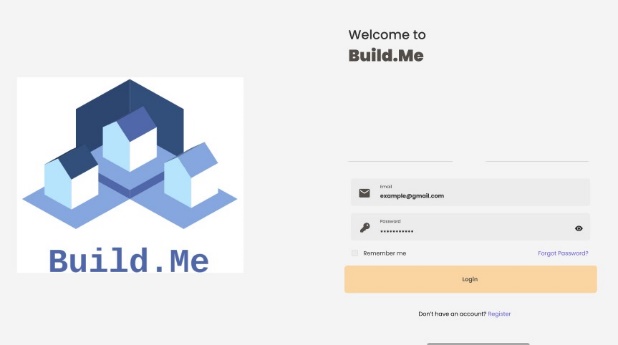
Figura - Diagrama de Sequência para Caso de Uso 3

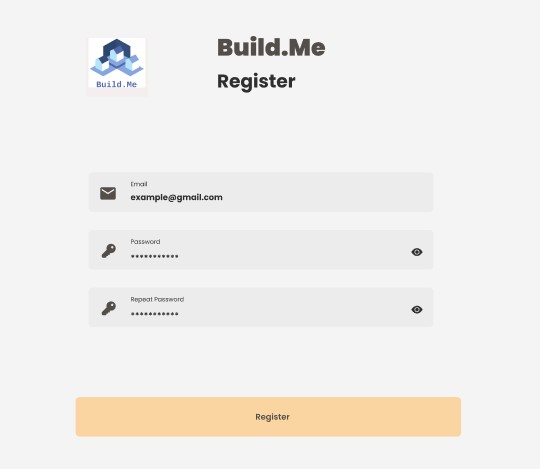
# Mockups

Neste capítulo vamos demonstrar o *front-end* desejado para o final do projeto. Para isso, foram construídas *Mockups* simples, utilizando a ferramenta online *Figma*. Pequenos detalhes podem ser diferentes no produto final, contudo, estes exemplos de *mockups* servem de base e de inspiração para o desenrolar do projeto.

É possível aceder as *mockups* completas ao site:

[*https://www.figma.com/file/usnHWCnkXwYSbvhF2zOWdV/PDS\_22\_23-team-library?node-id=0%3A1&t=8vI7Fu8KOekL1EmV-0*](https://www.figma.com/file/usnHWCnkXwYSbvhF2zOWdV/PDS_22_23-team-library?node-id=0%3A1&t=8vI7Fu8KOekL1EmV-0)



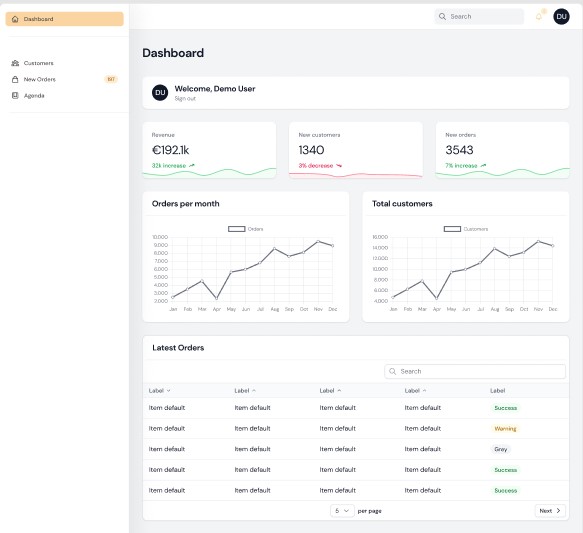


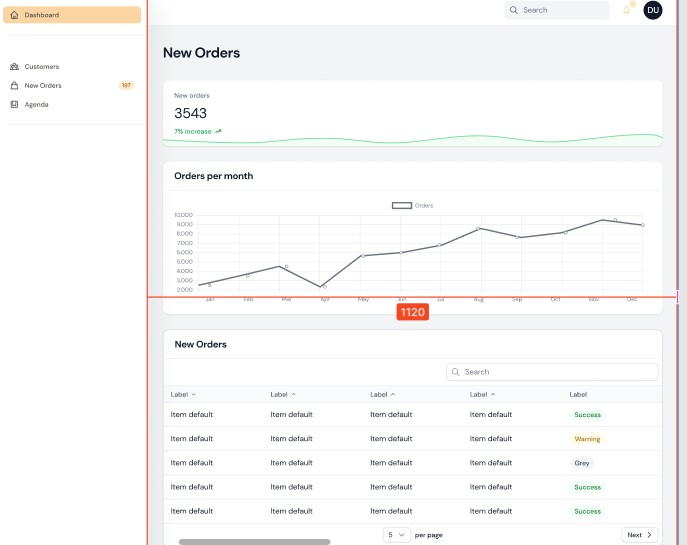


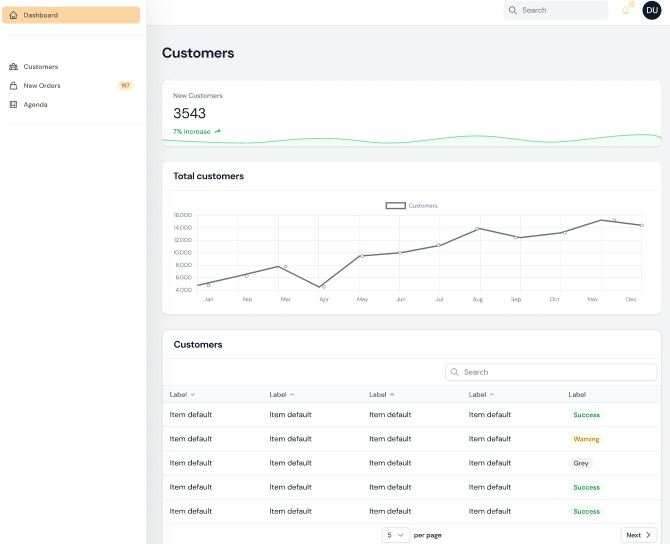
Este é o fluxo que o *User* pode fazer a partir da *Home Page*. Registar-se e fazer o login, ou só fazer o login. Depois, dependendo do tipo do *User* (Empresa ou Normal *User*) é direcionado para um *BackOfifice* diferente, e com funções diferentes.

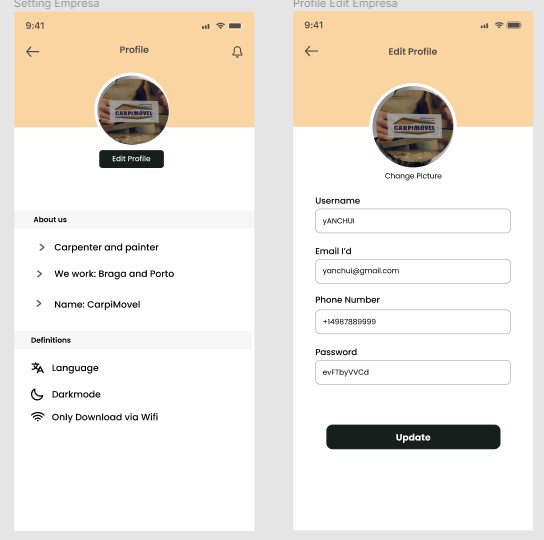


No caso de ser uma empresa:

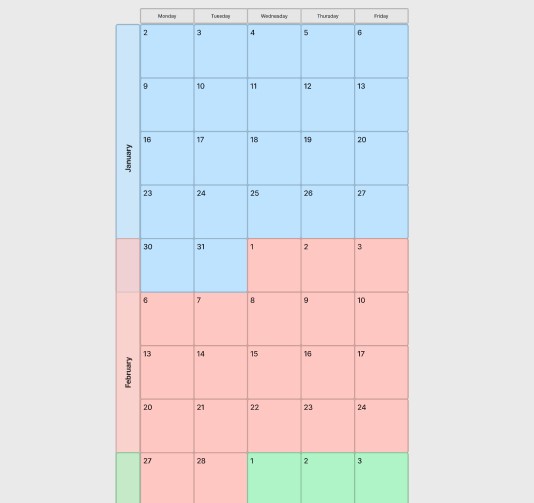






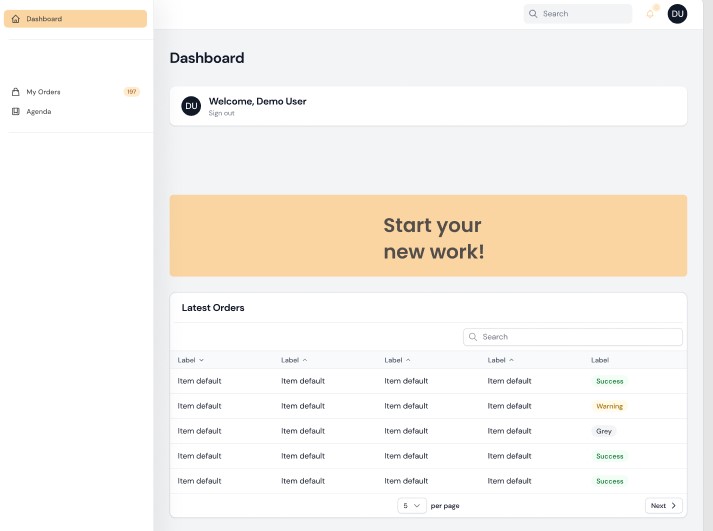


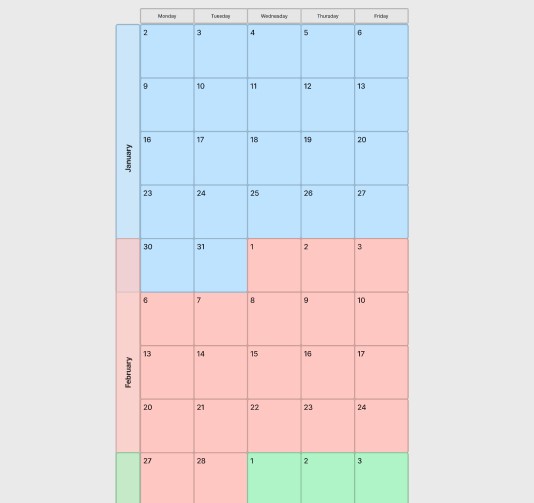


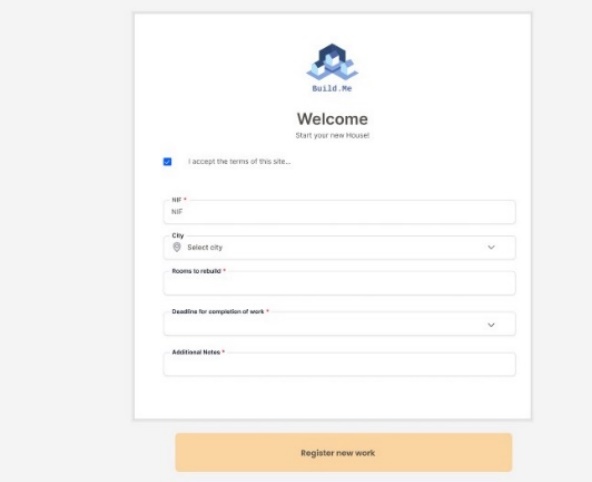


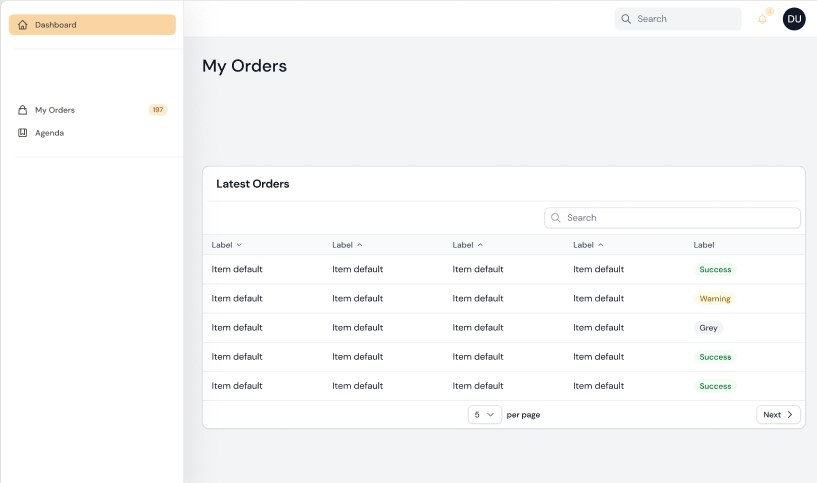
No caso de o User ser do tipo Empresa, ele tem acesso a:

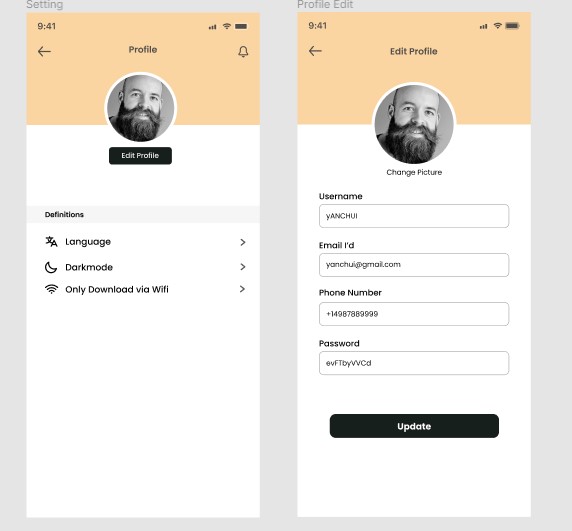
* Uma agenda, onde pode ver a marcação de certos serviços;
* Uma lista de Clientes que o contrataram;
* Ao seu perfil;
* E aos novos trabalhos que foram pedidos e o valor deles;

No caso de ser um User normal:









O User Normal tem acesso a:

* Uma agenda, para ver a data em que os orçamentos podem ser aceites, ou a obra está feita, etc;
* A começar um pedido de obra nova;
* A ver os pedidos que fez até ao momento, e o dinheiro gasto nos projetos;
* A editar e ver o seu perfil;