RELATÓRIO – Iteração 3

Protótipo e objetivos de arquitetura

| Projeto: | AutoHub: Serviços automóveis |
| --- | --- |
| Grupo: | 110509: Rui Albuquerque  117450: Ellen Sales  119859: João Leite  120202: Paulo Lacerda |
| Data/versão: | 2024/11/07, v1 |
| Sumário: | O AutoHub é uma plataforma digital para pesquisa, agendamento e gestão de serviços de automóveis. Nesta iteração, foram definidos os principais módulos da arquitetura do sistema, identificados os requisitos com impacto técnico e desenvolvida uma primeira versão do protótipo de navegação, cobrindo funcionalidades como registo, pesquisa de serviços e agendamento. |

**Conteúdo**

[**1**](#_heading=h.c7cqep8jb5yd) **Introdução 1**

[1.1](#_heading=h.rnipu0kxhj68) Sumário executivo 1

[1.2](#_heading=h.qf4okakxp5yq) Controlo de versões 2

[1.3](#_heading=h.mp2yhwn5251z) Referências e recursos suplementares 2

[**2**](#_heading=h.lr7r3tqvgkwr) **Arquitetura do sistema 2**

[2.1](#_heading=h.xdty69a81i1g) Objetivos gerais 2

[2.2](#_heading=h.1wgn8ma5mabn) Requisitos com impacto na arquitetura 3

[2.3](#_heading=h.re7sh342ixdq) Decisões e justificação 3

[2.4](#_heading=h.tq5lgu454xzv) Arquitetura do software 4

[**3**](#_heading=h.l0e33k2r9adp) **Protótipo das interações 5**

# Introdução

## Sumário executivo

Este relatório apresenta a arquitetura proposta para a solução e os resultados da prototipagem, tendo por base o âmbito funcional já caracterizado no relatório de análise de requisitos.

A solução proposta inclui uma área para o cliente, um painel de gestão para os prestadores de serviços, uma interface de administração, e um módulo de pagamentos.

Um aspeto essencial no desenho da plataforma computacional é o princípio "mobile first", garantindo que a experiência do utilizador seja otimizada em dispositivos móveis, com aplicações responsivas que se adaptam a diferentes tamanhos de tela. Este enfoque assegura acessibilidade e usabilidade em qualquer dispositivo.

A prototipagem rápida foca-se nas histórias relativas à criação, acesso e gestão de uma conta, pesquisa e agendamento de serviços e avaliação após o serviço. Estas são as funcionalidades essenciais que acreditamos para validar a experiência base do utilizador.

## Controlo de versões

| **Quando?** | **Responsável** | **Alterações significativas** |
| --- | --- | --- |
| 03 / 04 | Paulo Lacerda | Capítulo 1 |
| 13 / 04 | João Leite | Capítulo 3 |
| 04 / 04 | Ellen Sales | Protótipo Figma |
| 05 / 04 | Rui Albuquerque | Capítulo 2 |

## Referências e recursos suplementares

**Eclipse Process Framework – OpenUP.** *Open Unified Process – A lightweight, agile process framework.* Disponível em: https://epf.eclipse.org/wikis/openup/index.htm. Data da pesquisa: 03/04/2025

# Arquitetura do sistema

Esta seção descreve a arquitetura global do sistema AutoHub, detalhando os objetivos da arquitetura, os requisitos que influenciam as escolhas técnicas, as decisões tomadas e a organização lógica do software. A abordagem adotada visa garantir uma solução escalável, robusta e responsiva, que suporte a integração com sistemas externos (por exemplo, APIs de pagamento e geolocalização), proporcionando uma experiência uniforme em diversos dispositivos (web, smartphones, tablets)

## Objetivos gerais

Os principais objetivos da arquitetura do AutoHub são:

* **Acessibilidade e Responsividade:** Permitir que os clientes acedam à plataforma através de browsers e aplicações móveis sem necessidade de instalar software adicional. A interface deve ser responsiva, garantindo uma experiência consistente em dispositivos desktop e móveis.
* **Escalabilidade e Alta Disponibilidade:** O sistema deve suportar um elevado número de utilizadores simultâneos e ser configurado para funcionar em ambiente cloud com redundância, assegurando uma disponibilidade mínima de 99.9%.
* **Integração com Sistemas Externos:** Integrar serviços essenciais como processamento de pagamentos, geolocalização e notificações push. A arquitetura deve permitir a substituição ou atualização destes serviços sem comprometer a operação global.
* **Segurança e Proteção de Dados:** Garantir que toda a informação relativa aos clientes (dados pessoais, histórico de serviços e transações) seja armazenada de forma cifrada e com acesso controlado.
* **Modularidade e Manutenibilidade:** Promover uma separação clara entre as camadas de apresentação, lógica de negócio e dados, facilitando a implementação de novas funcionalidades e a manutenção do sistema ao longo do tempo.

## Requisitos com impacto na arquitetura

A seguir, apresenta-se uma tabela com os principais requisitos não-funcionais e atributos de qualidade que influenciam as escolhas arquiteturais:

| **Requisitos** | **Descrição** |
| --- | --- |
| RD-4 | O sistema deve suportar a utilização sustentada de 500 sessões em simultâneo, garantindo desempenho consistente mesmo em períodos de elevado tráfego.. |
| RD-5 | Todos os serviços críticos devem ser configurados de forma redundante, utilizando soluções cloud, para assegurar uma disponibilidade de, pelo menos, 99.9%. |
| RS-1 | Os dados pessoais e históricos dos clientes, incluindo transações e avaliações, deverão ser armazenados de forma cifrada, garantindo a segurança e conformidade com as normas de privacidade. |
| RU-3 | O portal deve adaptar-se automaticamente a diferentes tamanhos de ecrã (smartphones, tablets, desktops), mantendo uma experiência de utilizador consistente e intuitiva. |
| RN-1 | O sistema deve permitir a integração com APIs externas (pagamentos, geolocalização e notificações) de forma modular, possibilitando a substituição destes serviços sem impacto direto nas operações. |

## Decisões e justificação

Tendo em conta os objetivos definidos e os requisitos acima identificados, foram tomadas as seguintes decisões arquiteturais:

* **Frontend e Aplicações Móveis:** O desenvolvimento da camada de apresentação será realizado utilizando a biblioteca **React**. Esta escolha justifica-se pela sua robustez, grande comunidade e experiência prévia da eq  
  uipa. Para a construção das aplicações móveis, será utilizada **React Native**, promovendo a reutilização de código e uma transição suave entre as plataformas web e móveis.
* **Backend e Lógica de Negócio:** A camada de aplicação será implementada utilizando **Node.js** (ou outra tecnologia similar), numa abordagem baseada em micro-serviços ou numa arquitetura modular, de forma a permitir a escalabilidade e a integração facilitada com serviços externos. A utilização de frameworks modernos e a adesão a boas práticas de desenvolvimento (p. ex.: RESTful APIs) são essenciais para garantir a robustez do sistema.
* **Base de Dados e Armazenamento:** Optou-se por uma solução baseada em cloud (por exemplo, **Cloud Firestore** ou similar) que permite uma escalabilidade dinâmica e um armazenamento seguro dos dados. O uso de bases de dados NoSQL pode favorecer uma maior flexibilidade na modelação do domínio e facilitar a integração com a camada de aplicação.
* **Integração com Serviços Externos:** A integração com APIs de pagamento, geolocalização e notificações será efetuada através de módulos específicos, isolando as dependências e possibilitando a troca ou atualização destes serviços sem impactos significativos na arquitetura global.
* **Comunicação e Mensageria:** Para a comunicação assíncrona entre os serviços (por exemplo, para enviar notificações ou atualizar o status dos agendamentos), será implementado um sistema de filas/mensageria, o que contribui para a robustez e tolerância a falhas.

Estas escolhas refletem a necessidade de uma plataforma moderna, modular, e preparada para suportar um ambiente de elevada simultaneidade, mantendo a segurança e a qualidade da experiência de utilizador.

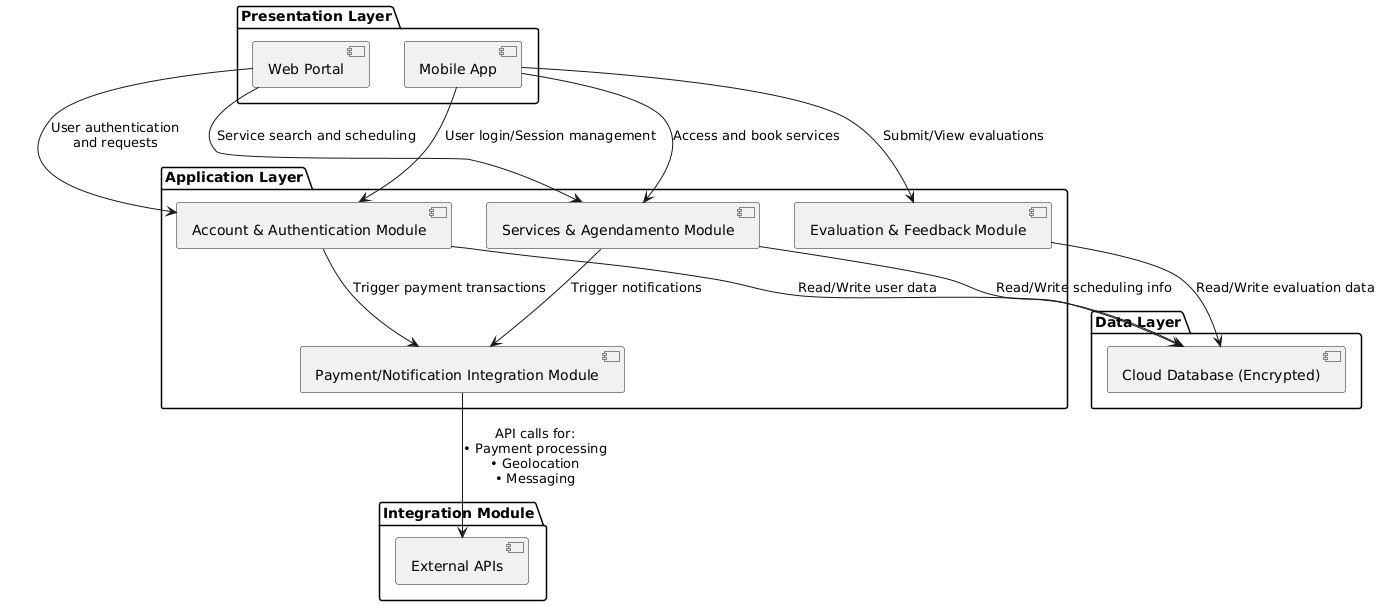
## Arquitetura do software

A proposta da arquitetura do software segue uma abordagem em camadas (layered architecture), representada de forma lógica pelos seguintes módulos:

* **Camada de Apresentação (Frontend):** Responsável por gerir a interação com o utilizador. Inclui o portal web e as aplicações móveis, onde os clientes podem pesquisar serviços, efetuar agendamentos, submeter avaliações e acompanhar o histórico de interações.
* **Camada de Aplicação (Lógica de Negócio):** Responsável pela orquestração dos processos do negócio. Divide-se nos seguintes módulos:  
  + **Gestão de Contas e Autenticação:** Responsável pela criação e manutenção de perfis de utilizador, gestão de sessões e autenticação segura.
  + **Gestão de Serviços e Agendamentos:** Cuida da procura, seleção, agendamento e cancelamento dos serviços oferecidos, interagindo com o módulo de parceiros e prestadores de serviços.
  + **Sistema de Avaliações e Feedback:** Permite a submissão e consulta de avaliações, contribuindo para a confiança e transparência dos serviços.
  + **Integração de Pagamentos e Notificações:** Gestão das transações financeiras e envio de notificações (ex.: via push notifications) para informar os clientes sobre alterações ou novas oportunidades.
* **Camada de Dados:** Responsável pelo armazenamento e gestão dos dados persistentes. As informações relativas aos clientes, agendamentos, serviços e transações são guardadas de forma cifrada em bases de dados cloud, suportando a escalabilidade e a redundância exigidas.
* **Módulo de Integração:** Atua como intermediário entre a camada de aplicação e os serviços externos (APIs de pagamento, geolocalização, etc.), permitindo uma comunicação segura e modular que facilita a manutenção e a eventual substituição destes serviços.

#### **Relação e Articulação dos Módulos**

1. **Fluxo de Utilizador:** O cliente interage com o portal ou aplicação móvel (camada de apresentação), que envia pedidos para a camada de aplicação, onde são processadas as operações de autenticação, agendamento e avaliação.
2. **Processamento e Comunicação:** A camada de aplicação comunica com a camada de dados para operações de leitura e escrita e utiliza o módulo de integração para interagir com serviços externos (ex.: pagamento, geolocalização).
3. **Notificações e Atualizações:** A arquitetura inclui um mecanismo assíncrono (ex.: filas de mensageria) para o envio de notificações e a atualização de estados, garantindo que as operações são processadas de forma robusta, mesmo em cenários de elevada simultaneidade.



A articulação entre os módulos decorre da seguinte forma:

**Presentation Layer:** Inclui o **Web Portal** e o **Mobile App**, que são os pontos de interação com o utilizador.

**Application Layer:** Se encontra constituída por:

* **Account & Authentication Module:** Gerencia o login e a gestão de contas.
* **Services & Agendamento Module:** Responsável pela procura e agendamento dos serviços.
* **Evaluation & Feedback Module:** Permite a submissão e consulta de avaliações.
* **Payment/Notification Integration Module:** Gerencia a integração com sistemas de pagamentos e a emissão de notificações.

**Integration Module:** Centraliza a comunicação com serviços externos (APIs), tais como processamento de pagamentos, geolocalização e sistema de mensageria.

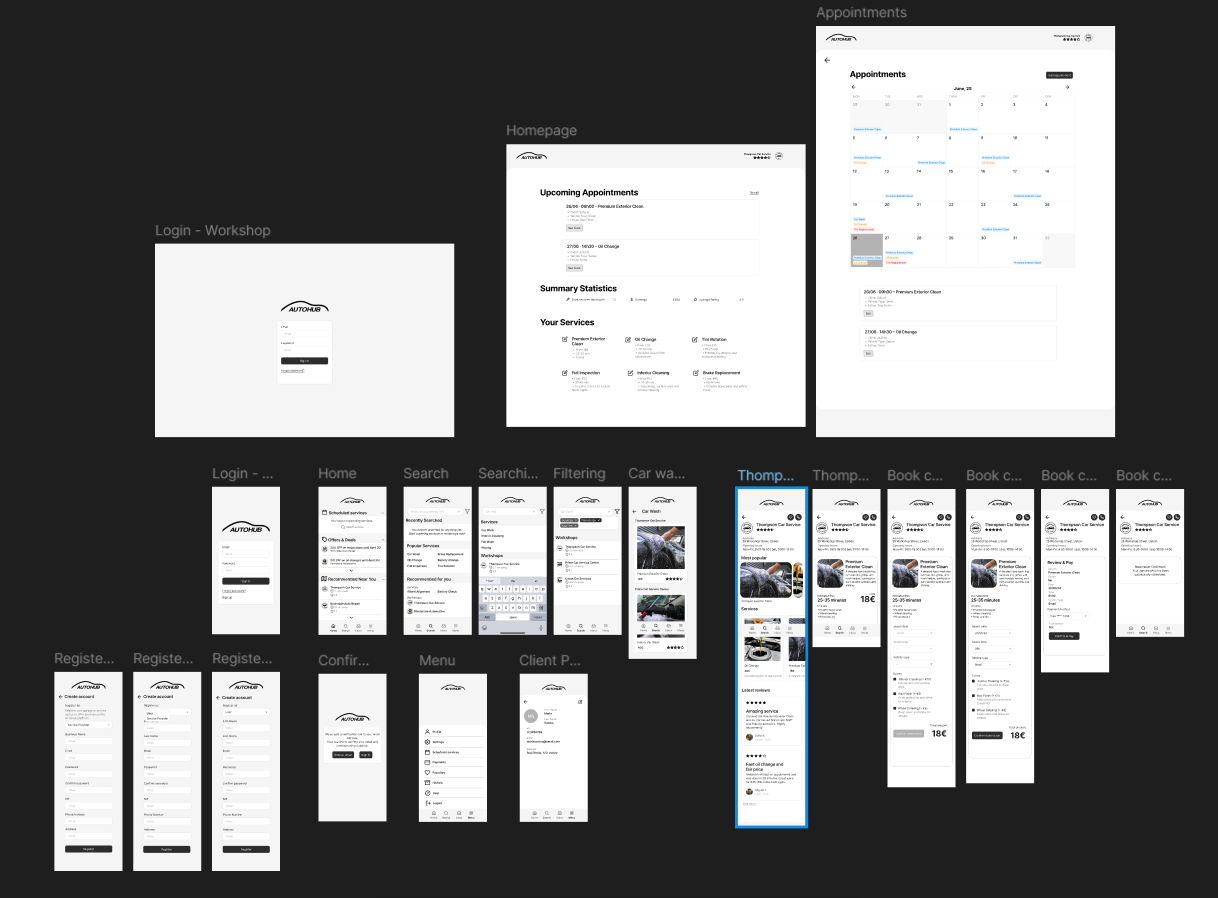
**Data Layer:** Representa o armazenamento dos dados em uma base de dados na cloud, com acesso cifrado.

**Conexões:** As setas indicam o fluxo de informação entre as camadas, desde a interação dos utilizadores até à persistência dos dados e integração com serviços externos.

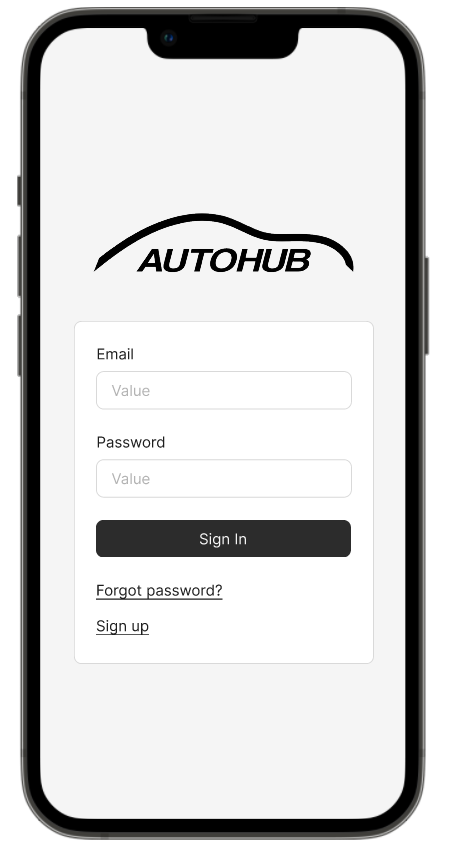
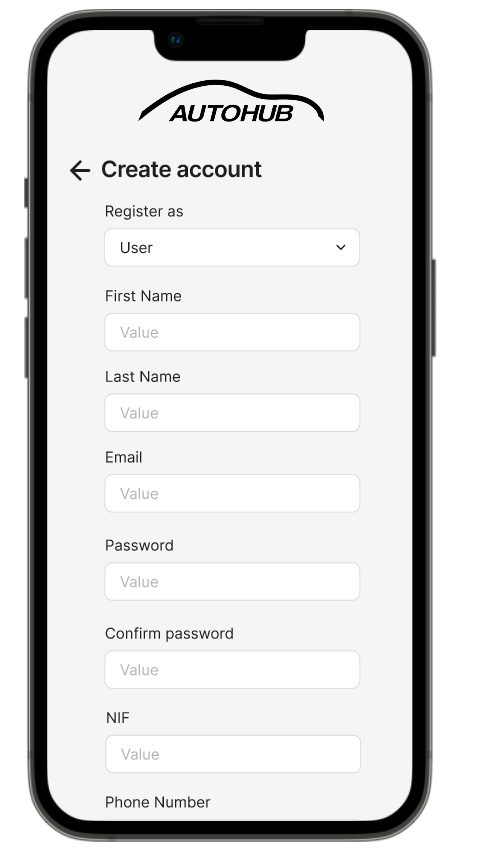
# Protótipo das interações

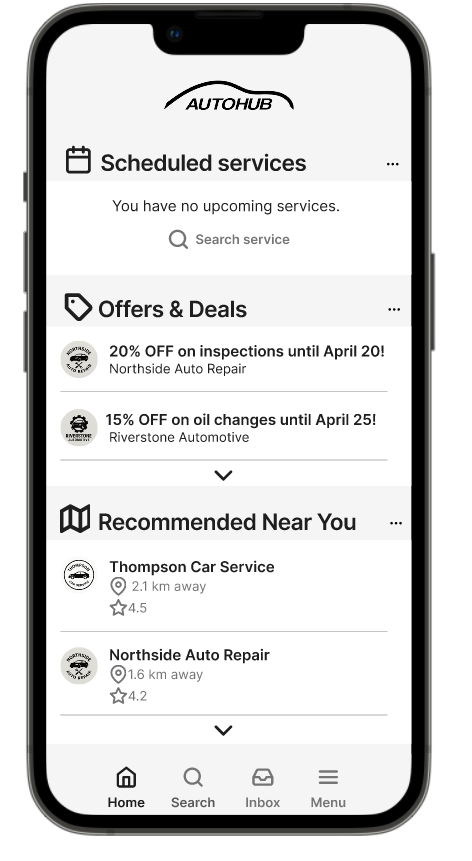
Para a realização deste protótipo utilizamos o Figma. A seguir estarão as páginas principais deste protótipo.

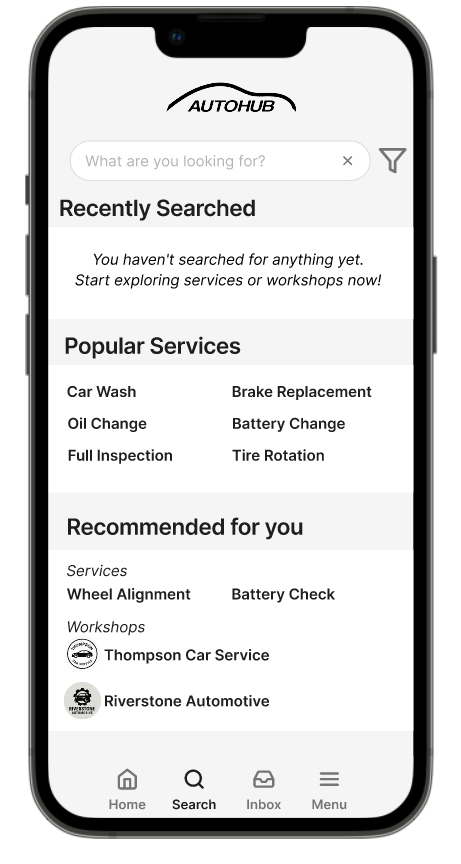
* Esquema de páginas:

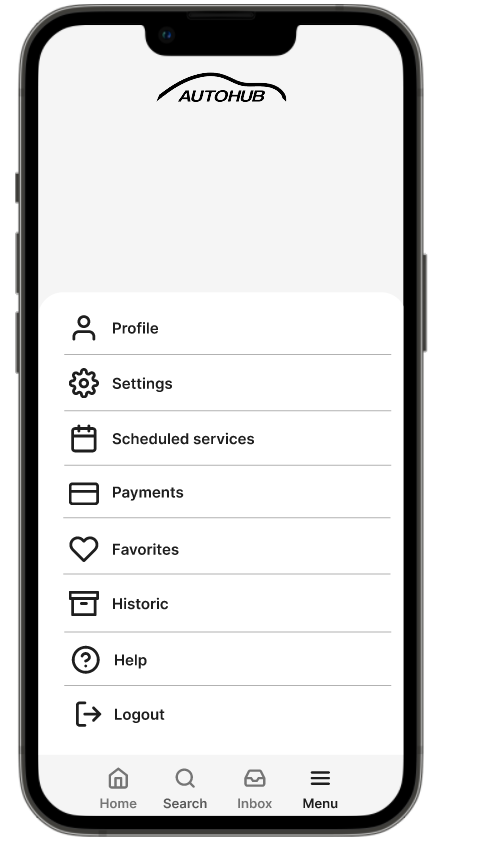


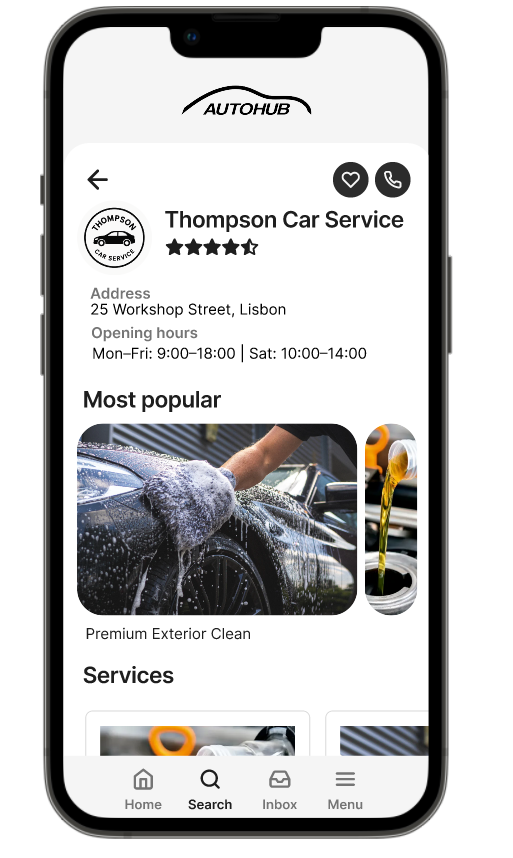
* Login/Registo:

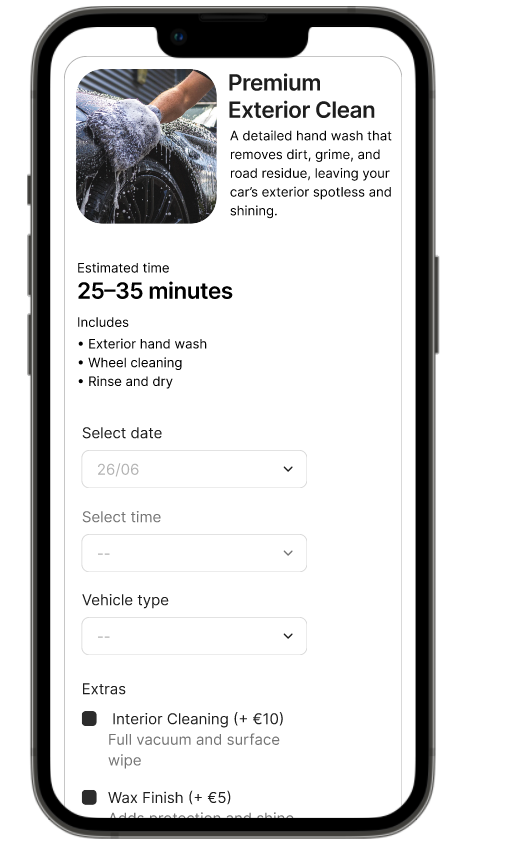
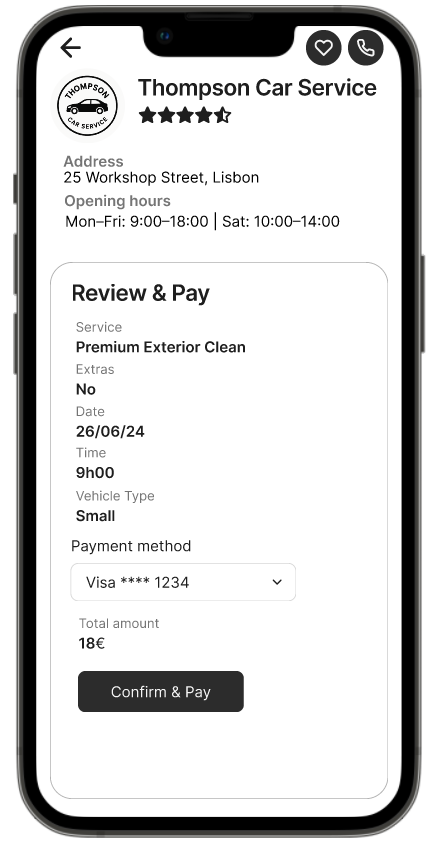


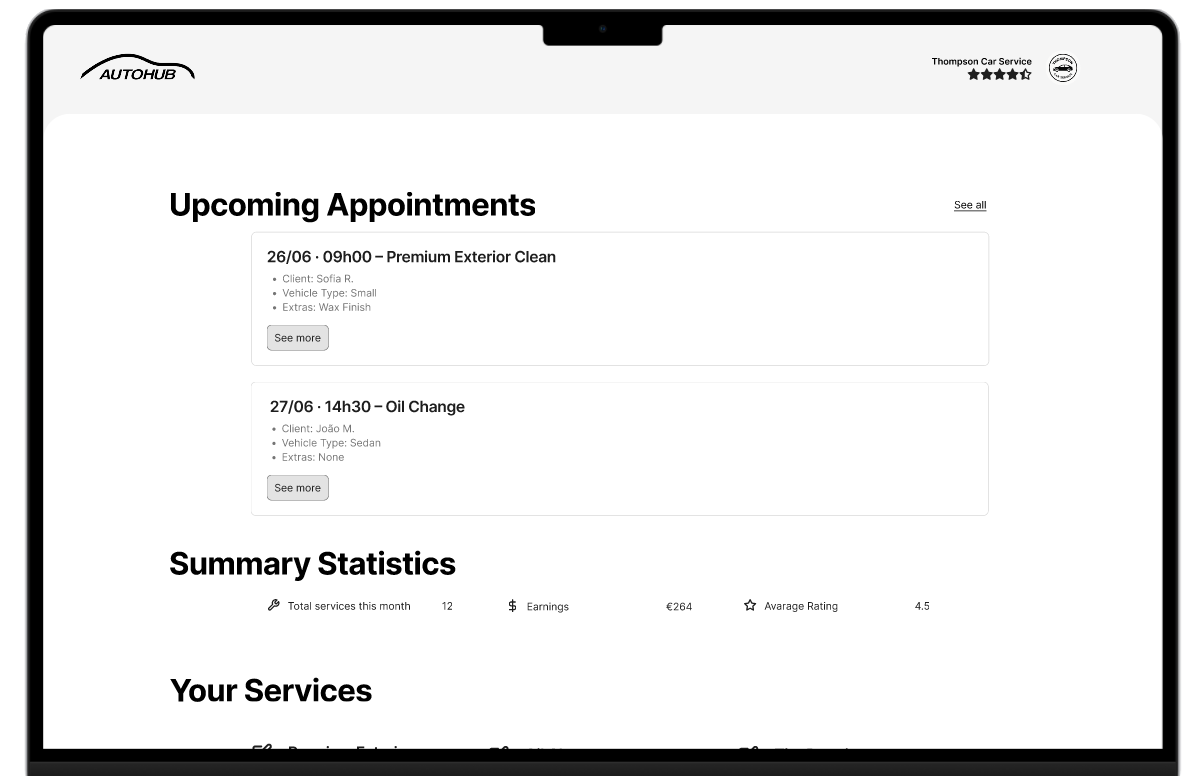
* Página Principal:
* Pesquisa:



* Menu de utilizador:
* Empresa e seus serviços:



* Marcar serviço e pagamento:
* Página principal Empresa:



* Página de serviços marcados: