ESCOLA SENAI “A. JACOB LAFER”

**CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS**

CARLOS CERQUEIRA

GABRIEL ORTOLANI

GUSTAVO DA SILVA

ISABELLE PACHECO

MIGUEL MERNICK

STEFANY LINO

**PROJETO SCRUM**

ServHub

SANTO ANDRÉ 2025

ESCOLA SENAI “A. JACOB LAFER”

**CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS**

CARLOS CERQUEIRA

GABRIEL ORTOLANI

GUSTAVO DA SILVA

ISABELLE PACHECO

MIGUEL MERNICK

STEFANY LINO

**PROJETO SCRUM**

ServHub

Trabalho acadêmico apresentado ao curso

Técnico em Desenvolvimento de Sistemas do

SENAI, como contribuição para o projeto SCRUM.

Orientadores: Nadja Luz e Raul Lopes.

SANTO ANDRÉ 2025

**RESUMO**

Com o avanço tecnológico e a transformação das formas de inserção no mercado de trabalho, muitos profissionais têm enfrentado dificuldades em divulgar seus serviços de maneira eficaz, principalmente aqueles que ainda utilizam métodos tradicionais, como a entrega manual de currículos ou a busca informal por oportunidades. Segundo a Catho (2021), 63% desses trabalhadores relatam dificuldades em obter entrevistas, enquanto o LinkedIn (2023) aponta que mais de 85% dos recrutadores utilizam plataformas digitais para seleção de candidatos. Diante desse cenário, desenvolveu-se este projeto com o objetivo de criar um site voltado ao "aluguel de serviços", que funcione como uma vitrine digital para profissionais autônomos e prestadores de serviço ampliarem sua visibilidade e alcançarem um público maior. A proposta busca atender à demanda por ferramentas acessíveis, que facilitem a conexão entre profissionais e contratantes, promovendo inclusão digital e inserção no mercado. O levantamento de dados realizado permitiu identificar as principais necessidades dos usuários e orientar a estruturação do sistema. Conforme o Fórum Econômico Mundial (2023), 44% das habilidades exigidas no mercado devem mudar até 2027, sendo o domínio de tecnologias digitais essencial para a competitividade, o que reforça a relevância da presente iniciativa.

**Palavras-chave:** Sistema, implementação, tecnologia.

**ABSTRACT**

With technological advancement and the transformation of how individuals enter the job market, many professionals have faced difficulties in effectively promoting their services, especially those who still rely on traditional methods such as handing out printed résumés or informally searching for opportunities. According to Catho (2021), 63% of these workers report difficulties in securing interviews, while LinkedIn (2023) states that more than 85% of recruiters use digital platforms for candidate selection. In response to this scenario, this project was developed with the objective of creating a website focused on "service rental"—a digital showcase where self-employed professionals and service providers can increase their visibility and reach a broader audience. The proposal aims to meet the demand for accessible tools that facilitate the connection between professionals and clients, promoting digital inclusion and better integration into the labor market. The data collected allowed the identification of users' main needs and guided the structuring of the system. According to the World Economic Forum (2023), 44% of the skills required in the job market are expected to change by 2027, with digital technology proficiency becoming essential for competitiveness, which highlights the relevance of this initiative.

**Palavras-chave:** System, implementation, technology.

[1. INTRODUÇÃO 6](#_Toc199769625)

[1.1 Justificativa 6](#_Toc199769626)

[1.2 Problema 7](#_Toc199769627)

[1.3 Objetivos 7](#_Toc199769628)

[2. EMBASAMENTO TEÓRICO 8](#_Toc199769629)

[2.1 Viabilidade Técnica 8](#_Toc199769630)

[3. DESENVOLVIMENTO 10](#_Toc199769631)

[3.1 Levantamento de requisitos 10](#_Toc199769632)

[3.2 Manual de marca 11](#_Toc199769633)

[3.3 Cronograma 12](#_Toc199769634)

[3.4 Protótipos 14](#_Toc199769635)

[3.5 SPRINT 1 - Descritivo de funcionamento do projeto. 17](#_Toc199769636)

[3.6 SPRINT 2 - Descritivo de continuidade do projeto. 17](#_Toc199769637)

[3.7 SPRINT 3 – Descritivo de conclusão do projeto. 17](#_Toc199769638)

[4. REFERÊNCIAS 19](#_Toc199769639)

[5. APÊNDICE 21](#_Toc199769640)

# INTRODUÇÃO

Com o crescimento acelerado das demandas do mercado e a necessidade crescente das empresas por mão de obra especializada, surgem novos desafios e oportunidades. Atualmente, muitas organizações buscam profissionais altamente capacitados, porém apenas para funções específicas e por tempo determinado. Paralelamente, um número significativo de trabalhadores encontra-se em busca de novas oportunidades. De acordo com dados recentes, a taxa de desemprego no Brasil é de 6,2% (IBGE, 2024), o que evidencia a necessidade de soluções que promovam a reintegração desses profissionais ao mercado de trabalho. Diante desse cenário, a **ServHub** identifica uma oportunidade inovadora ao propor o desenvolvimento de uma plataforma digital que facilite a conexão entre empresas e profissionais qualificados, otimizando o processo de contratação para funções temporárias. Como destaca Drucker (2001), “a inovação sistemática requer a vontade de considerar a mudança como uma oportunidade” (p. 36), o que reforça a importância de soluções tecnológicas que atendam às novas exigências do mercado de trabalho contemporâneo.

## Justificativa

A justificativa deste projeto parte da crescente demanda das empresas por mão de obra especializada e da necessidade de facilitar o acesso da população a oportunidades de trabalho. O desenvolvimento de uma plataforma digital voltada à intermediação entre empresas e profissionais tem como objetivo proporcionar maior agilidade, segurança e praticidade aos processos de contratação. Além disso, espera-se que a ferramenta otimize os contatos entre empregadores e candidatos adequados ao perfil exigido, beneficiando tanto o setor empresarial quanto os trabalhadores, e, por consequência, contribuindo com o desenvolvimento social. De acordo com Chiavenato (2004), “o capital humano é o recurso mais importante das organizações, e sua adequada gestão é essencial para o alcance de resultados organizacionais e sociais” (p. 45), o que reforça a relevância de soluções tecnológicas que auxiliem na aproximação entre talentos e oportunidades de trabalho.

## Problema

No cenário atual, a sociedade enfrenta um aumento exponencial do desemprego, segundo dados, no Brasil cerca de 6,8 milhões de pessoas encaram essa triste realidade (IBGE, 2024), além desta problemática, há a dificuldade de conseguir serviço para estabelecer uma renda, foi pensando nisso que nós da ServHub desenvolvemos a nossa plataforma, a qual facilita que todos os cidadãos tenham uma grande facilidade para venderem seus serviços.

## Objetivos

O objetivo geral deste projeto é desenvolver uma plataforma de gestão de recursos humanos temporários, especificamente uma tabela interativa que permita às empresas visualizar de forma clara e eficiente os funcionários e os seus serviços. A plataforma visa facilitar a busca por profissionais qualificados para um curto período de tempo, otimizando o processo de recrutamento e contratação temporária.

Os objetivos específicos deste projeto são: criar uma interface intuitiva para cadastro e login, gerenciamento de dados, automatizar processos internos, garantir a segurança dos dados. Essas interfaces serão criadas utilizando tecnologias de codificação para o melhor desempenho, desenvolvida pensando na acessibilidade, conforto e segurança de cada usuário, trazendo mais eficiência a procura de funcionários.

# EMBASAMENTO TEÓRICO

O trabalho temporário tem se destacado como uma solução estratégica para atender à crescente demanda por flexibilidade no mercado de trabalho. Segundo o estudo *Flexibility @ Work*, da Randstad, entre 5% e 25% da força de trabalho em países desenvolvidos atua sob contratos com prazo determinado, o que reforça a importância desse modelo de contratação (Redação Mundo RH, 2021). Conhecido também como “aluguel de mão de obra”, o trabalho temporário permite que as empresas atendam a demandas pontuais, como projetos específicos, picos de produção ou substituições momentâneas, sem a necessidade de vínculos permanentes. Esse cenário tem sido impulsionado pelo avanço de tecnologias voltadas à gestão de recursos humanos, que tornam os processos de contratação mais ágeis e eficientes. Assim, o trabalho temporário se apresenta como uma alternativa viável tanto para as organizações quanto para os profissionais em busca de inserção no mercado.

## Viabilidade Técnica

Para o desenvolvimento deste trabalho, utilizaremos alguns recursos fundamentais. O **HTML(Hypertext Markup Language)** será empregado como linguagem de marcação responsável por estruturar as páginas da aplicação, organizando os dados de forma clara e funcional. De acordo com Beaird (2010), “o HTML é a espinha dorsal da web, fornecendo a estrutura básica necessária para a construção de sites” (p. 25). Complementando essa estrutura, utilizaremos o **CSS** (Cascading Style Sheets), uma linguagem voltada à estilização das páginas web, que permite definir cores, fontes, espaçamento, layout e responsividade, garantindo uma apresentação visual atrativa e adaptável a diferentes dispositivos. Segundo W3C (2018), “o CSS permite a separação do conteúdo e do design, proporcionando uma apresentação mais flexível e eficiente” (p. 32). Além disso, o **Figma** será utilizado para a criação de interfaces, wireframes e protótipos. O Figma possibilita que equipes de design trabalhem de forma colaborativa, simultân0065a e remota no mesmo projeto, otimizando o processo criativo (Tera Blog, s.d.). Por fim, o **PHP** (Hypertext Preprocessor) será adotado como a principal linguagem de programação para o backend da plataforma. O PHP é uma linguagem de código aberto amplamente usada no desenvolvimento de sites dinâmicos e interativos, permitindo a criação de funcionalidades como autenticação de usuários, gerenciamento de dados em banco de dados e processamento de formulários. Como afirmado por Welling e Thomson (2017), “o PHP é uma das linguagens mais utilizadas para desenvolver aplicações web dinâmicas e interativas” (p. 115). A escolha do PHP se justifica pela sua flexibilidade, robustez e integração eficaz com o HTML e o CSS, além de ser uma tecnologia consolidada e amplamente utilizada em soluções web.

A combinação de HTML, CSS e PHP oferece uma estrutura completa para o desenvolvimento de plataformas eficientes, escaláveis e de fácil manutenção. A escolha desses recursos para a criação da plataforma de aluguel de funcionários se baseia na simplicidade e na robustez dessas tecnologias, que garantem uma experiência de usuário agradável e uma gestão eficiente das funcionalidades no backend.

O JSON é um formato leve de intercâmbio de dados, de fácil leitura e escrita para humanos, e simples de interpretar e gerar por máquinas. Baseado em texto, o JSON é estruturado em pares chave-valor e listas ordenadas de valores, sendo amplamente utilizado para a transmissão de dados entre aplicações web e servidores. Sua sintaxe é derivada da linguagem JavaScript, mas é compatível com diversas linguagens de programação. Devido à sua simplicidade, legibilidade e compatibilidade, o JSON tornou-se um dos formatos mais populares na comunicação entre sistemas em ambientes distribuídos, especialmente em aplicações baseadas em APIs REST.

# DESENVOLVIMENTO

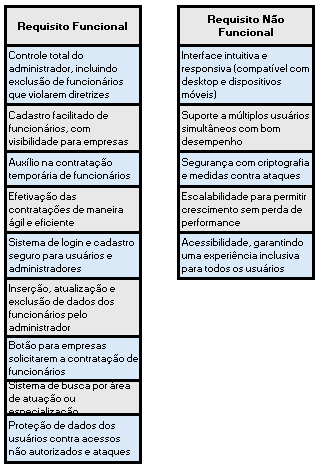
Nossa abordagem começa com a divisão das tarefas principais entre os membros da equipe, garantindo que todos possam se dedicar ao trabalho sem sobrecargas. Após essa divisão, realizamos pesquisas para entender e analisar as melhores opções para o andamento do projeto. Em seguida, elaboramos cronogramas detalhados para assegurar a estruturação do projeto e o cumprimento dos prazos de entrega. Com uma base sólida do projeto montada, discutimos a melhor linguagem de programação para garantir um desenvolvimento eficiente. Optamos por HTML e CSS, focando também na criação de uma interface intuitiva.

Além disso, utilizamos o Figma na fase de prototipagem da plataforma, o que foi fundamental para a criação de wireframes e layouts interativos das páginas. O Figma permitiu que a equipe de design trabalhasse de forma colaborativa e remota, otimizando o processo de criação e garantindo que todas as partes envolvidas pudessem visualizar e modificar o design em tempo real. A prototipagem no Figma também foi essencial para testar a navegação, a usabilidade e a interação entre os elementos antes do desenvolvimento final. Dessa forma, conseguimos validar e ajustar a interface com antecedência, assegurando que a experiência do usuário fosse fluida e eficiente, além de atender às necessidades de acessibilidade e design responsivo.

## Levantamento de requisitos

Requisitos funcionais são elementos essenciais para o bom funcionamento do projeto e devem ser atendidos de forma rigorosa. Segundo Sommerville (2011), "os requisitos funcionais especificam o comportamento ou funções que um sistema deve exibir, sendo fundamentais para garantir que o software atenda às necessidades do usuário e alcance os objetivos do projeto" (p. 82). A observância rigorosa desses requisitos assegura que a aplicação desempenhe corretamente suas funções e atenda às expectativas dos usuários e stakeholders. De acordo com o nosso projeto, a figura de número 1 descreve quais são os requisitos para o funcionamento da interface.

Figura 1 – Tabela de requisitos.



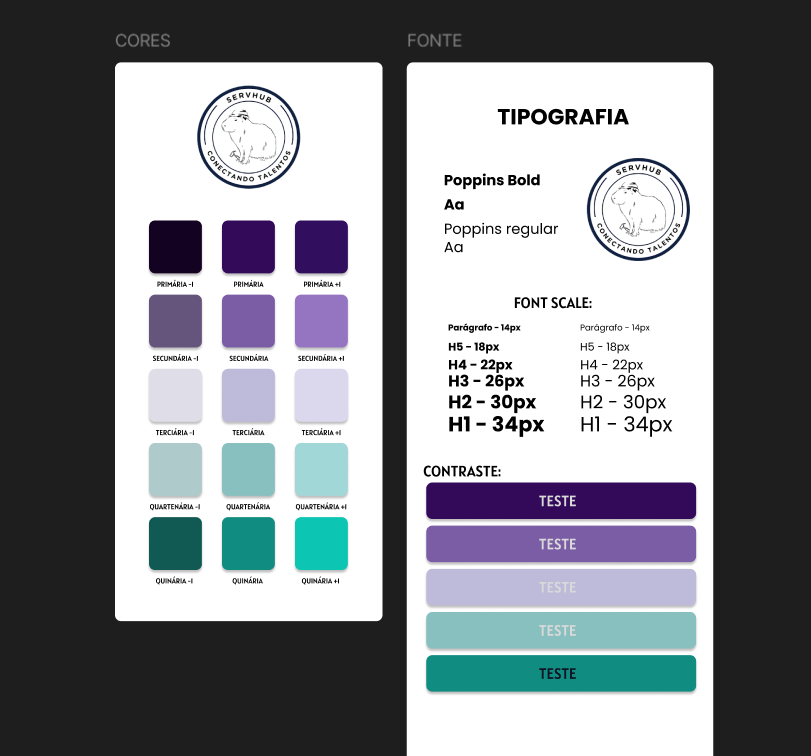
Fonte: Elaborado pelos autores (2025).

## Manual de marca

Manual de marca, se trata de uma etapa crucial no desenvolvimento de um WebSite, pois nele se define inúmeros fatores como, a paleta de cores a ser utilizada(para gerar uma melhor harmonia), tipo de fonte de letras, logomarca da empresa e outros inúmeros fatores (ROCKCONTENT, 2023).

Na figura 2, é possível visualizar a paleta de cores escolhida, fontes e contraste. Tais cores foram escolhidas pensando na harmonia do site, o roxo simboliza criatividade e inspiração, enquanto o azul transmite intelecto e segurança, sentimentos que o nosso site procura passar a quem for acessá-lo.

Figura 2 – Manual de marca (FIGMA)



Fonte – Elaborado pelos autores (2025).

Além disso, a logo da figura 3 foi criada com uma capivara funcionando como mascote, tal animal foi selecionado por serem roedores conhecidos por viverem em grupos com forte organização social (BIOLOGIANET, 2025).

Figura 3 – Logo da empresa

Forma, Calendário

O conteúdo gerado por IA pode estar incorreto.

Fonte – Elaborado pelos autores (2025).

## Cronograma

Para garantir uma organização eficiente e um acompanhamento preciso do progresso do projeto, foram criados dois modelos de cronograma: um Kanban e um Scrum. Ambos serão seguidos rigorosamente ao longo do desenvolvimento, proporcionando uma estrutura clara para a execução das tarefas e o cumprimento dos prazos.

O modelo Kanban foi adotado para gerenciar o fluxo de trabalho de maneira visual, permitindo que todos os membros da equipe acompanhem as etapas do processo de forma dinâmica, identificando rapidamente o status de cada tarefa. Por outro lado, o modelo Scrum será utilizado para estruturar o trabalho em ciclos, conhecidos como sprints, com metas bem definidas a serem alcançadas em períodos específicos. (ATLASSIAN, 2025)

A figura 4 ilustra como esse cronograma é aplicado e acompanhado ao longo de todas as fases do projeto, com cada etapa sendo elencada em grupos diferentes, não iniciado, em progresso e concluído.

Figura 4 – Cronograma Kanban (exemplo).

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Fonte – Elaborado pelos autores (2025).

Por outro lado, o Scrum foca na entrega contínua de funcionalidades incrementais e na melhoria constante do processo de desenvolvimento. (ATLASSIAN, 2025)

A Figura 5 ilustra essa abordagem, mostrando claramente a divisão das tarefas em ciclos de trabalho (sprints) e o status de progresso de cada uma delas.

Figura 5 – Cronograma Scrum (exemplo).

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Fonte – Elaborado pelos autores (2025).

## Protótipos

Iniciamos o projeto criando protótipos de alta fidelidade, esse protótipo serviu como fundamental para a concepção e estruturação do nosso site, permitindo testar e validar ideias antes da implementação final. A figura 6 apresenta a tela de login, na qual o usuário ao entrar no site tem que preencher os campos com seus dados através de um formulário, com campos de usuário e senha. Caso o usuário não tenha um login, através de um botão de “cadastre-se”, se é possível realizar um cadastro para garantir seu acesso no site. Além disso, se há um botão de “esqueci minha senha”, na qual, o usuário acessa um formulário para a alteração da sua senha anterior.

Figura 6 – Área de Login (exemplo).

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

Fonte – Elaborado pelos autores (2025).

A figura 7 demonstra a homepage da página, que servirá como área de fácil acesso para os demais conteúdos da página, através de botões de acesso como “comece já” e “login”, se é possível o acesso as demais telas do site.

Figura 7 – Homepage (exemplo).

Tela de celular com mensagem de texto

O conteúdo gerado por IA pode estar incorreto.

Fonte – Elaborado pelos autores (2025).

A figura 8 demonstra a área de acesso dos formulários de cadastro, com duas opções, cadastro como empresa ou funcionário, que depende das funções que o usuário executar.

Figura 8 – Área de Cadastro (exemplo).

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

Fonte – Elaborado pelos autores (2025).

A figura 9 demonstra a tela de perfil, logada no perfil de usuário, onde se pode visualizar informações de seu perfil pessoal e também outros profissionais que se cadastraram na plataforma, além de uma barra de busca para ver outros profissionais cadastrados na área de sua preferência de pesquisa.

Figura 9 – Área de Perfil Usuário (exemplo).

Interface gráfica do usuário, Site

O conteúdo gerado por IA pode estar incorreto.

Fonte – Elaborado pelos autores (2025).

A figura 10 demonstra a tela de perfil, com acesso aos serviços e configurações do administrador, onde além da visualização de profissionais e a barra de busca, o administrador pode editar informações ou deletá-las.

Figura 10 – Área de Perfil ADM (exemplo).

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

Fonte – Elaborado pelos autores (2025).

## SPRINT 1 - Descritivo de funcionamento do projeto.

Primeiro foi feito o levantamento de tarefas para o projeto e após a divisão das mesmas garantindo uma carga de trabalho igual para a equipe sem sobrecarregar nenhum integrante e melhorar a agilidade dos colaboradores.  
Para organizar as tarefas e prazos foi desenvolvido o kanban e o cronograma, após a criação destes foi feito o levantamento de requisitos,  posteriormente foi iniciado a elaboração da prototipagem e em paralelo o desenvolvimento da documentação.

## SPRINT 2 - Descritivo de continuidade do projeto.

Durante a segunda sprint, demos continuidade ao desenvolvimento dos protótipos de alta fidelidade, aprimorando o design das interfaces com base no planejamento inicial e nos feedbacks coletados até o momento. Essa etapa foi fundamental para garantir que a experiência do usuário fosse fluida, intuitiva e visualmente agradável, além de alinhada com os objetivos do projeto. Simultaneamente, seguimos com a produção e organização da documentação técnica, registrando as decisões tomadas, os fluxos de navegação e os requisitos implementados, o que será essencial para o acompanhamento e manutenção futura da plataforma. Enquanto isso, outros membros da equipe concentraram seus esforços no desenvolvimento do código Front-End, transformando os protótipos em interfaces reais com o uso de HTML, CSS e JavaScript, garantindo que o layout fosse responsivo e funcional em diferentes dispositivos. Essa divisão de tarefas permitiu que o time trabalhasse de forma colaborativa e produtiva, mantendo um bom ritmo de progresso ao longo da sprint.

## SPRINT 3 – Descritivo de conclusão do projeto.

Por fim, durante a Sprint 3, o grupo ficou encarregado de concluir o desenvolvimento completo do projeto. Essa fase foi marcada por diversas atividades técnicas e organizacionais. Inicialmente, trabalhamos na finalização do código já existente, realizando ajustes e correções necessárias no Front-End para garantir melhor usabilidade e coerência visual da interface.

Em seguida, iniciamos a implementação do Back-End, utilizando a linguagem PHP em conjunto com o formato de dados JSON para estruturar a comunicação entre o servidor e o cliente. Desenvolvemos funcionalidades essenciais, como a manipulação de dados, integração com formulários da interface e simulação de armazenamento e recuperação de informações, garantindo que a página fosse totalmente funcional.

Além das atividades técnicas, também dedicamos tempo à organização do projeto: concluímos toda a documentação necessária, incluindo descrições detalhadas das funcionalidades, tecnologias utilizadas, estrutura de diretórios e fluxos de dados. Atualizamos os cronogramas de forma a refletir com precisão o andamento e as entregas de cada sprint, e finalizamos os slides da apresentação, destacando os principais pontos do desenvolvimento, os desafios enfrentados e os resultados obtidos.

# REFERÊNCIAS

BEAIRD, Jason. *The principles of beautiful web design.* 2. ed. SitePoint, 2010.

BIOLOGIA NET. Capivara. Disponível em: https://www.biologianet.com/biodiversidade/capivara.htm. Acesso em: 13 maio 2025.

BRAY, Tim. . IETF, 2017. Disponível em: https://tools.ietf.org/html/rfc8259. Acesso em: 06 maio 2025.

CATHO. Pesquisa de Recrutamento e Seleção. Disponível em: https://www.catho.com.br. Acesso em: 15 abr. 2025.

CHIAVENATO, Idalberto. *Gestão de pessoas: o novo papel dos recursos humanos nas organizações.* 3. ed. Rio de Janeiro: Elsevier, 2004.

DRUCKER, Peter F. *Inovação e espírito empreendedor: prática e princípios.* 3. ed. São Paulo: Cengage Learning, 2001.

ECMA INTERNATIONAL. . 2.ª ed. Geneva, Suíça: Ecma International, 2017. Disponível em: https://www.ecma-international.org/publications-and-standards/standards/ecma-404/. Acesso em: 06 maio 2025.

IBGE. Taxa de desocupação é de 6,2% e taxa de subutilização cai para 17,6% no trimestre encerrado em março de 2024. Instituto Brasileiro de Geografia e Estatística, 2024. Disponível em: https://www.ibge.gov.br. Acesso em: 22 abr. 2025.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE). Desemprego – Explica. Disponível em: https://www.ibge.gov.br/explica/desemprego.php. Acesso em: 15 abr. 2025.

MOZILLA DEVELOPER NETWORK (MDN). . Disponível em: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON. Acesso em: 06 maio 2025.

RGU GUSTAVO. Quadro de tarefas do projeto na plataforma Jira. Disponível em: https://rguustavo61.atlassian.net/jira/software/projects/KAN/boards/1. Acesso em: 15 abr. 2025.

ROCK CONTENT. Manual de marca: como criar e aplicar em sua empresa. Disponível em: https://rockcontent.com/br/blog/manual-de-marca/f. Acesso em: 15 abr. 2025.

SOMMERVILLE, Ian. *Engenharia de software.* 9. ed. São Paulo: Pearson, 2011.

TERA BLOG. Figma: o que é e como usar. 2019. Disponível em: https://www.terablog.com.br/figma-como-usar/. Acesso em: 22 abr. 2025.

W3C. CSS specifications. World Wide Web Consortium, 2018. Disponível em: https://www.w3.org/TR/css/. Acesso em: 22 abr. 2025.

W3SCHOOLS. . Disponível em: https://www.w3schools.com/js/js\_json\_intro.asp. Acesso em: 06 maio 2025.

WELLING, Luke; THOMSON, Laura. *PHP and MySQL Web Development.* 5. ed. Boston: Addison-Wesley, 2017.

WORLD ECONOMIC FORUM. *The Future of Jobs Report 2020.* Disponível em: https://www.weforum.org/reports/the-future-of-jobs-report-2020. Acesso em: 15 abr. 2025.

# APÊNDICE

<?php

    // arquivo de configuração com as constantes do sistema

    define('ARQUIVO\_JSON', \_\_DIR\_\_ . '/../data/funcionarios.json');

    define('ARQUIVO\_USUARIOS', \_\_DIR\_\_ . '/../data/usuarios.json');

    define('SEMANAL\_INICIANTE', 800.00);

    define('SEMANAL\_EXPERIENTE', 1200.00);

    define('SEMANAL\_SENIOR', 1800.00);

?>

[

    {

        "nome": "ford",

        "email": "ford@gmail.com",

        "senha": "$2y$12$EDCLf.4Z.bH1kD.hOiDuUe\/L42ZBzPNuimTn6FOCbeBHM3jvS51Ju",

        "cnpj": "07901368000170",

        "endereco": "rua sim do lado do parque tal",

        "path": "",

        "filename": "",

        "uploaded\_at": ""

    },

    {

        "nome": "honda",

        "email": "honda@gmail.com",

        "senha": "$2y$12$wET21nS08\/f5q2OJ4kV1LeGCMebtG66hoZPws.A265eJyNceuIHce",

        "cnpj": "180230816238190263",

        "endereco": "sim",

        "path": "",

        "filename": "",

        "uploaded\_at": ""

    },

    {

        "nome": "nike",

        "email": "nike@gmail.com",

        "senha": "$2y$12$OhbLOyka4gD0V1b4pzD9e.8t1jzOZoQwnjhLYvi1W7xyr1KHaMFtC",

        "cnpj": "0120310239123",

        "endereco": "não",

        "path": "uploads\/img\_683d9061e04b07.60177230.png",

        "filename": "img\_683d9061e04b07.60177230.png",

        "uploaded\_at": "2025-06-02 11:52:01"

    },

    {

        "nome": "SLINOS",

        "email": "slinos@email.com",

        "senha": "$2y$12$9Kx6MuySQUB4JQ\/KnIOsE.Oe5fWydvUeaFa3LaWFVmCkWo1dREzHO",

        "cnpj": "123456789012",

        "endereco": "Rua Amém da Glória N°7",

        "path": "uploads\/img\_683de8a7b621a6.27589685.png",

        "filename": "img\_683de8a7b621a6.27589685.png",

        "uploaded\_at": "2025-06-02 18:08:39"

    }

]

[

    {

        "nome": "teste",

        "email": "teste@gmail.com",

        "senha": "$2y$12$5dnNJdXndSW\/U5CfCME6YOJgk\/ICzsYtfcqwtqf8bp3GqH0MlNPI2",

        "cpf": "123131231221123",

        "atuacao": "Tecnico",

        "descricao": "traleleiro tralala",

        "experiencia": "iniciante",

        "telefone": "",

        "path": "uploads\/img\_68349953cd9ab3.72437505.png",

        "filename": "img\_68349953cd9ab3.72437505.png",

        "uploaded\_at": "2025-05-26 16:39:47",

        "path\_curriculo": "uploads2\/cv\_68359e62b76a38.74850665.pdf",

        "filename\_curriculo": "cv\_68359e62b76a38.74850665.pdf",

        "curriculo\_uploaded\_at": "2025-05-27 11:13:38",

        "disponivel": true

    },

    {

        "nome": "carlos",

        "email": "carlos@gmail.com",

        "senha": "$2y$12$5YIsR1wNwVYxe5ZMOPtFa.OjBQdtYNHkldk8H5GJZVI7BByjpMlgq",

        "cpf": "123123123132123",

        "atuacao": "sou o malcon",

        "descricao": "baiano",

        "experiencia": "experiente",

        "telefone": "(11) 999999999",

        "path": "uploads\/img\_68349de04fdd59.25745178.jpg",

        "filename": "img\_68349de04fdd59.25745178.jpg",

        "uploaded\_at": "2025-05-26 16:59:12",

        "path\_curriculo": "uploads2\/cv\_683d936d4171c9.81015085.pdf",

        "filename\_curriculo": "cv\_683d936d4171c9.81015085.pdf",

        "curriculo\_uploaded\_at": "2025-06-02 12:05:01",

        "disponivel": true

    },

    {

        "nome": "Stefany Lino Silva",

        "email": "stefany@gmail.com",

        "senha": "$2y$12$spDTvei9GHRbdm1fkSo08u07ACNOpsgPc11rVs8mL9YVoIEiEsSt6",

        "cpf": "12345678910",

        "atuacao": "Desenvolvedor de Sistemas",

        "descricao": "",

        "experiencia": "experiente",

        "telefone": "",

        "path": "",

        "filename": "",

        "uploaded\_at": "",

        "path\_curriculo": "",

        "filename\_curriculo": "",

        "curriculo\_uploaded\_at": "",

        "disponivel": true

    }

]

[

    {

        "tipo": "senior",

        "nome": "teste",

        "experiencia": "senior",

        "disponivel": true

    },

    {

        "tipo": "senior",

        "nome": "carlos",

        "experiencia": "senior",

        "disponivel": false

    },

    {

        "tipo": "experiente",

        "nome": "Stefany Lino Silva",

        "experiencia": "experiente",

        "disponivel": true

    }

]

[

    {

        "filename": "img\_68349953cd9ab3.72437505.png",

        "path": "uploads\/img\_68349953cd9ab3.72437505.png",

        "uploaded\_at": "2025-05-26 16:39:47"

    },

    {

        "filename": "img\_68349a6e30f363.70303430.png",

        "path": "uploads\/img\_68349a6e30f363.70303430.png",

        "uploaded\_at": "2025-05-26 16:44:30"

    },

    {

        "filename": "img\_68349c3208e8a8.30406297.jpg",

        "path": "uploads\/img\_68349c3208e8a8.30406297.jpg",

        "uploaded\_at": "2025-05-26 16:52:02"

    },

    {

        "filename": "img\_68349c3dc34ad8.95154170.png",

        "path": "uploads\/img\_68349c3dc34ad8.95154170.png",

        "uploaded\_at": "2025-05-26 16:52:13"

    },

    {

        "filename": "img\_68349db2c2f686.21149873.png",

        "path": "uploads\/img\_68349db2c2f686.21149873.png",

        "uploaded\_at": "2025-05-26 16:58:26"

    },

    {

        "filename": "img\_68349de04fdd59.25745178.jpg",

        "path": "uploads\/img\_68349de04fdd59.25745178.jpg",

        "uploaded\_at": "2025-05-26 16:59:12"

    },

    {

        "filename": "img\_683d9061e04b07.60177230.png",

        "path": "uploads\/img\_683d9061e04b07.60177230.png",

        "uploaded\_at": "2025-06-02 11:52:01"

    },

    {

        "filename": "img\_683de8a7b621a6.27589685.png",

        "path": "uploads\/img\_683de8a7b621a6.27589685.png",

        "uploaded\_at": "2025-06-02 18:08:39"

    }

]

[

    {

        "username": "grupo4@gmail.com",

        "perfil": "admin",

        "password": "$2y$12$iKGzYCwueiIY2EDMVgr05utvujA5avqPLBHV4Pfupdcuf\/8wBkAjS"

    },

    {

        "username": "honda@gmail.com",

        "perfil": "empresa",

        "password": "$2y$12$wET21nS08\/f5q2OJ4kV1LeGCMebtG66hoZPws.A265eJyNceuIHce"

    },

    {

        "username": "nike@gmail.com",

        "perfil": "empresa",

        "password": "$2y$12$OhbLOyka4gD0V1b4pzD9e.8t1jzOZoQwnjhLYvi1W7xyr1KHaMFtC"

    },

    {

        "username": "teste@gmail.com",

        "perfil": "funcionario",

        "password": "$2y$12$zAg7iwzl2wO5rrSRWrKfruCqWKc4g8Jcjx.zj11kD992kASDIPQkO"

    },

    {

        "username": "teste2@gmail.com",

        "perfil": "funcionario",

        "password": "$2y$12$WlgyUhOg2Yi6DxYP3ItzwO0OONrG0gpR8D\/Li3p7FrQOrHngT3DiW"

    },

    {

        "username": "teste@gmail.com",

        "perfil": "funcionario",

        "password": "$2y$12$5dnNJdXndSW\/U5CfCME6YOJgk\/ICzsYtfcqwtqf8bp3GqH0MlNPI2"

    },

    {

        "username": "carlos@gmail.com",

        "perfil": "funcionario",

        "password": "$2y$12$5YIsR1wNwVYxe5ZMOPtFa.OjBQdtYNHkldk8H5GJZVI7BByjpMlgq"

    },

    {

        "username": "slinos@email.com",

        "perfil": "empresa",

        "password": "$2y$12$9Kx6MuySQUB4JQ\/KnIOsE.Oe5fWydvUeaFa3LaWFVmCkWo1dREzHO"

    },

    {

        "username": "stefany@gmail.com",

        "perfil": "funcionario",

        "password": "$2y$12$spDTvei9GHRbdm1fkSo08u07ACNOpsgPc11rVs8mL9YVoIEiEsSt6"

    }

]

<?php

namespace Interfaces;

// Interface que define os métodos necessários para um veiculo ser locável

interface Locavel {

    public function alugar() : string;

    public function devolver() : string;

    public function isDisponivel() : bool;

}

?>

<?php

namespace Models;

require\_once \_\_DIR\_\_ . '/../config/config.php';

use Interfaces\Locavel;

class Experiente extends Pessoa implements Locavel { // Alterado para herdar de Pessoa

    public function calcularAluguel(int $dias): float {

        return $dias \* SEMANAL\_EXPERIENTE;

    }

    public function alugar(): string{

        if ($this->disponivel){

            $this->disponivel = false;

            return "Funcionario '{$this->nome}' alugado com sucesso!";

        }

        return "Funcionario '{$this->nome}' não está disponivel.";

    }

    public function devolver(): string

    {

        if (!$this->disponivel){

            $this->disponivel = true;

            return "Funcionario '{$this->nome}' devolvido com sucesso!";

        }

        return "Funcionario '{$this->nome}' está disponivel.";

    }

    public function getNivelExperiencia(): string {

        return 'experiente';

    }

}

<?php

namespace Models;

require\_once \_\_DIR\_\_ . '/../config/config.php';

use Interfaces\Locavel;

class Iniciante extends Pessoa implements Locavel { // Alterado para herdar de Pessoa

    public function calcularAluguel(int $dias): float {

        return $dias \* SEMANAL\_INICIANTE;

    }

    public function alugar(): string{

        if ($this->disponivel){

            $this->disponivel = false;

            return "Funcionario '{$this->nome}' alugado com sucesso!";

        }

        return "Funcionario '{$this->nome}' não está disponivel.";

    }

    public function devolver(): string

    {

        if (!$this->disponivel){

            $this->disponivel = true;

            return "Funcionario '{$this->nome}' devolvido com sucesso!";

        }

        return "Funcionario '{$this->nome}' está disponivel.";

    }

    public function getNivelExperiencia(): string {

        return 'iniciante';

    }

}

<?php

namespace Models;

// Classe abstrata para todos os tipos de veículos

abstract class Pessoa {

    protected string $nome;

    protected string $email;

    protected int $experiencia;

    protected bool $disponivel = true;

    public function \_\_construct(string $nome, string $email, int $experiencia) {

        $this->nome = $nome;

        $this->email = $email;

        $this->experiencia = $experiencia;

    }

    public function getNome(): string {

        return $this->nome;

    }

    public function getNivelExperiencia(): string {

        return $this->experiencia;

    }

    public function isDisponivel(): bool {

        return $this->disponivel;

    }

    public function alugar(): string {

        $this->disponivel = false;

        return "Funcionário {$this->nome} alugado.";

    }

    public function devolver(): string {

        $this->disponivel = true;

        return "Funcionário {$this->nome} devolvido.";

    }

    abstract public function calcularAluguel(int $dias): float;

}

<?php

namespace Models;

require\_once \_\_DIR\_\_ . '/../config/config.php';

use Interfaces\Locavel;

class Senior extends Pessoa implements Locavel { // Alterado para herdar de Pessoa

    public function calcularAluguel(int $dias): float {

        return $dias \* SEMANAL\_SENIOR;

    }

    public function alugar(): string{

        if ($this->disponivel){

            $this->disponivel = false;

            return "Funcionario '{$this->nome}' alugado com sucesso!";

        }

        return "Funcionario '{$this->nome}' não está disponivel.";

    }

    public function devolver(): string

    {

        if (!$this->disponivel){

            $this->disponivel = true;

            return "Funcionario '{$this->nome}' devolvido com sucesso!";

        }

        return "Funcionario '{$this->nome}' está disponivel.";

    }

    public function getNivelExperiencia(): string {

        return 'senior';

    }

}

@import url('https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap');

:root{

    --primary-color:#7b5da6;

    --button-hover:#5b3f82;

    --fundo:#efeded;

    --secondary-color:;

}

\*{

    font-family: "Poppins", sans-serif;

    font-weight: bold;

    margin: 0;

    padding: 0;

    border: 0;

    list-style: none;

}

body{

    width: 100%;

    height: 100vh;

}

#fundo {

    height: 100vh;

    background-image: url(../Assets/Login-fundo.png);

    background-size: cover;

    background-position: center;

}

.input-custom {

    margin: 0 0px 20px 0;

}

#login{

    background-color: var(--fundo);

}

#btn-custom {

    background-color: var(--primary-color);

    color: white;

}

#btn-custom:hover {

    background-color: var(--button-hover);

}

#bem-vindo{

    font-weight: lighter;

}

#senha-esqueceu{

    font-weight: lighter;

}

@media screen and (max-width: 426px){

    body{

        background-color: var(--fundo);

    }

    #fundo{

        display: none !important;

        background-image: none !important;

    }

    #login{

        height: 100%;

        padding-top: 50px;

    }

}

#foto-perfil{

    width: 200px;

    height: 200px;

    border-radius: 50%;

}

#adicionar{

    padding: 0!important;

}

/\* navbar homepage \*/

/\* NAVBAR \*/

#navbar {

    background-color: #E9EDEB;

    box-shadow: 0px 0px 5px rgba(0, 0, 0, 0.639);

}

#navbar .navbar-nav{

    flex-direction: row;

    justify-content: flex-end;

}

#navbar.navbar-nav li{

    margin-left: 1em;

}

#navbar button{

    color: #FFF;

}

#navbar li{

    position: relative;

}

.navbar-brand{

    font-weight: 700;

    width: 100%;

    text-align: center;

}

#logout {

    padding: 8px 16px;

    border-radius: 10px;

    background-color: #320A59;

    color: white;

    box-shadow: 3px 3px 2px black;

    margin-right: 20px;

}

.links{

    text-decoration: none;

    color: var(--primary-color);

}

#navbar-items{

    justify-content: space-between;

}

@media(min-width: 991px){

    #navbar-items{

        display: flex;

        width: 20%;

    }

    #formulario{

        width: 50%;

        align-self: flex-start;

    }

    #search-form{

        width: 100%;

        margin: 0;

        border-radius: 10px;

    }

    #logo{

        width: 20%;

    }

    #bot-nav{

        width: 20%;

    }

}

@media screen and (max-width: 768px){

    #formulario{

        display: none;

    }

}

@media (min-width: 426px) and (max-width: 768px){

    #navbar-items{

        width: 50%;

    }

    #navegacao{

        display: flex;

        flex-direction: row;

    }

}

#search-form{

    background-color: #BEBAD9;

    border: 2px solid #626262;

    border-radius: 20px;

    padding: 5px 15px;

}

#search-form input{

    border: none;

    color: #000000;

}

#search-form input:focus{

    box-shadow: none;

}

#search-form input::placeholder{

    color: #000000;

}

/\* BOTTOM NAVBAR \*/

#bottom-navbar-container .container{

    justify-content: flex-end;

}

#bottom-navbar-container a{

    color: #000000;

    font-weight: 600;

}

.navbar-toggler {

    font-size: 1.5em;

}

/\* card principal \*/

#saiba{

    background-color: #320A59;

    border-color: #290849;

}

#responsivo{

    display: none;

}

/\* Celular \*/

@media screen and (max-width:426px){

    #principal{

        display: none;

    }

    #sidebar {

        display: none;

    }

    #responsivo{

        display: block;

    }

    #form-login{

        display: flex;

        align-items: center;

        flex-direction: column;

        justify-content: center;

        margin: 0!important;

    }

    #texto-login{

        text-align: center;

        margin-bottom: 20px;

        margin-top: 40px;

    }

    #texto-cadastro{

        text-align: center;

        align-self: center;

    }

    #form-login form input{

        margin-bottom: 5px;

    }

}

/\* Menu hamburguer \*/

@media (max-width: 991px) {

    #navbar{

        padding-left: 0 !important;

    }

    #navbar-items.show {

      flex-direction: column;

      align-items: flex-start;

      margin-top: 10px;

      padding: 0;

    }

    #navbar-items.show .navbar-nav {

      flex-direction: column;

      width: 100%;

    }

    #navbar-items.show .nav-item {

      width: 100%;

      margin: 0 !important;

      border-bottom: 2px solid rgba(0, 0, 0, 0.15); /\* Linha separadora \*/

    }

    #navbar-items.show .nav-link {

      display: block;

      width: 100%;

      border-radius: 0;

      margin: 0;

      padding: 10px 15px;

    }

    #navbar-items.show .nav-item:last-child {

      border-bottom: none; /\* Remove a linha do último item \*/

    }

    #logout {

      padding: 5px 0;

      background-color: transparent;

      color: black !important;

      box-shadow: none;

      margin-right: 20px;

    }

    #burgue{

        align-self: flex-end;

    }

  }

/\* laptop \*/

@media(min-width: 991px) and (max-width:1024px){

    #formulario{

        width: 40%;

    }

}

/\* Card \*/

#titulo-card{

    margin-top: 5px;

    margin-left: 5px;

    font-size: 3em;

    font-weight: bold;

}

#texto-card{

    width: 60%;

    margin-top: 5px;

    margin-left: 15px;

    font-size: 1.5em;

    font-weight: lighter;

}

#logo-card{

    width: 40%;

    margin-right: 40px;

}

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Servhub</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/css/bootstrap.min.css" rel="stylesheet">

    <link rel="stylesheet" href="CSS/style.css">

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.12.1/font/bootstrap-icons.min.css">

    <link rel="shortcut icon" href="../Assets/ServHub\_branca.png" type="image/x-icon">

</head>

<nav class="navbar navbar-expand-lg primary-bg-color py-4 px-2" id="navbar">

    <div class="container">

        <div class="container d-flex justify-content-between" id="bot-nav">

            <a class="navbar-brand text-black" id="logo" href="../public/exibicao.php">ServHub</a>

            <button id="burgue" class="navbar-toggler text-black" type="button" data-bs-toggle="collapse" data-bs-target="#navbar-items">

                <i class="bi bi-list"></i>

            </button>

        </div>

        <div class="container col-sm-8 mx-2" id="formulario">

            <form class="d-flex" id="search-form">

                <i class="bi bi-search text-black mt-2"></i>

                <input class="form-control me-2 bg-transparent text-white" name="pesquisa" type="search" placeholder="Busque um funcionário...">

                <button class="btn secondary-bg-color text-black" type="submit">Pesquisar</button>

            </form>

        </div>

        <div class="collapse navbar-collapse" id="navbar-items">

            <ul class="navbar-nav mb-2 mb-lg-0 ">

                <li class="nav-item mx-3">

                    <a class="nav-link text-black" href="../public/exibicao.php">HOME</a>

                </li>

                <li class="nav-item mx-3">

                    <a class="nav-link text-black" href="../public/exibicaoaccount.php">CONTA</a>

                </li>

                <li class="nav-item mx-3">

                    <a class="nav-link text-white" href="?logout=1" id="logout">LOGOUT</a>

                </li>

            </ul>

        </div>

    </div>

</nav>

<?php

require\_once \_\_DIR\_\_ . '/../services/Auth.php';

use Services\Auth;

$usuario = Auth::getUsuario();

// Carregando arquivos

$usuariosJson = json\_decode(file\_get\_contents(\_\_DIR\_\_ . '/../data/usuarios.json'), true);

$dataFuncionarioJson = json\_decode(file\_get\_contents(\_\_DIR\_\_ . '/../data/data\_funcionario.json'), true);

$dataEmpresaJson = json\_decode(file\_get\_contents(\_\_DIR\_\_ . '/../data/data\_empresa.json'), true);

$isAdmin = Auth::isAdmin();

$dadosLogado = null;

// Busca dados com base no tipo de perfil

if ($isAdmin) {

    foreach ($usuariosJson as $user) {

        if ($user['username'] === $usuario['username']) {

            $dadosLogado = $user;

            break;

        }

    }

} elseif ($usuario['perfil'] === 'funcionario') {

    foreach ($dataFuncionarioJson as $user) {

        if ($user['email'] === $usuario['username']) {

            $dadosLogado = $user;

            break;

        }

    }

}else{

    foreach ($dataEmpresaJson as $user) {

        if ($user['email'] === $usuario['username']) {

            $dadosLogado = $user;

            break;

        }

    }

}

// Define imagem de perfil

$profileImage = isset($dadosLogado['path']) && !empty($dadosLogado['path'])

    ? $dadosLogado['path']

    : 'Assets/adm.png';

?>

<?php include 'htmls/head.html'; ?>

<body>

<?php include 'htmls/nav.html'; ?>

<div class="py-4 mx-2">

    <div class="container mt-4">

        <div class="row">

            <!-- Sidebar -->

            <div class="col-md-4" id="sidebar">

                <div class="card mb-4">

                    <div class="card-body d-flex flex-column align-items-center">

                        <img id="foto-perfil" src="<?= $profileImage ?>" alt="Foto de perfil">

                        <h3 class="card-title"><?= htmlspecialchars($dadosLogado['nome'] ?? $usuario['username']) ?></h3>

                        <p style="font-size: 0.8rem;"><?= htmlspecialchars($usuario['username']) ?></p>

                    </div>

                </div>

            </div>

            <!-- Conteúdo principal -->

            <div class="col-md-8 d-flex flex-column">

                <div class="card mb-4">

                    <div class="card-header mx-0">

                        <h5>Atualizar dados</h5>

                    </div>

                    <div class="card-body">

                        <!-- Dados -->

                        <form method="POST" action="atualizar\_dados.php">

                            <!-- Campo oculto para identificador fixo -->

                            <?php if ($isAdmin): ?>

                                <input type="hidden" name="user\_identifier" value="<?= htmlspecialchars($dadosLogado['username'] ?? $usuario['username']) ?>">

                            <?php else: ?>

                                <input type="hidden" name="user\_identifier" value="<?= htmlspecialchars($dadosLogado['email'] ?? $usuario['username']) ?>">

                            <?php endif; ?>

                            <div class="row">

                                <div class="col-md-12 mb-3">

                                    <label class="fw-bold">Nome:</label>

                                    <input class="form-control" name="nome" value="<?= htmlspecialchars($dadosLogado['nome'] ?? '') ?>" required>

                                </div>

                                <div class="col-md-12 mb-3">

                                    <label class="fw-bold">Email:</label>

                                    <input class="form-control" type="email" name="email" value="<?= htmlspecialchars($dadosLogado['email'] ?? '') ?>" required>

                                </div>

                                <?php if ($usuario['perfil'] === 'funcionario'): ?>

                                    <div class="col-md-6 mb-3">

                                        <label class="fw-bold">Telefone:</label>

                                        <input class="form-control" name="telefone" value="<?= htmlspecialchars($dadosLogado['telefone'] ?? '') ?>">

                                    </div>

                                    <div class="col-md-6 mb-3">

                                        <label class="fw-bold">CPF:</label>

                                        <input class="form-control" name="cpf" value="<?= htmlspecialchars($dadosLogado['cpf'] ?? '') ?>">

                                    </div>

                                    <div class="col-md-6 mb-3">

                                        <label class="fw-bold">Atuação:</label>

                                        <input class="form-control" name="atuacao" value="<?= htmlspecialchars($dadosLogado['atuacao'] ?? '') ?>">

                                    </div>

                                    <div class="col-md-6 mb-3">

                                        <label class="fw-bold">Nível de Experiência:</label>

                                        <select class="form-control mb-2" name="experiencia" require id="">

                                            <option value="" disabled selected>Selecione seu nível de experiência</option>

                                            <option value="iniciante">Iniciante</option>

                                            <option value="experiente">Experiente</option>

                                            <option value="senior">Sênior</option>

                                        </select>

                                    </div>

                                    <div class="col-md-12 mb-3">

                                        <label class="fw-bold">Descrição:</label>

                                        <textarea class="form-control" name="descricao" rows="4"><?= htmlspecialchars($dadosLogado['descricao'] ?? '') ?></textarea>

                                    </div>

                                <?php elseif ($usuario['perfil'] === 'empresa'): ?>

                                    <div class="col-md-12 mb-3">

                                        <label class="fw-bold">CNPJ:</label>

                                        <input class="form-control" name="cnpj" value="<?= htmlspecialchars($dadosLogado['cnpj'] ?? '') ?>" required>

                                    </div>

                                    <div class="col-md-12 mb-3">

                                        <label class="fw-bold">Endereço:</label>

                                        <input class="form-control" name="endereco" value="<?= htmlspecialchars($dadosLogado['endereco'] ?? '') ?>">

                                    </div>

                                <?php endif; ?>

                                <div class="col-md-12 text-center">

                                    <button type="submit" id="saiba" class="btn btn-success">Salvar Alterações</button>

                                </div>

                            </div>

                        </form>

                    </div>

                </div>

                <div class="row mx-0">

                    <?php if (!$isAdmin): ?>

                        <!-- Upload de imagem -->

                        <div class="card mb-4 <?= $usuario['perfil'] === 'empresa' ? 'col-md-12' : 'col-md-6' ?>" id="adicionar">

                            <div class="card-header">

                                <h5>Alterar foto de perfil</h5>

                            </div>

                            <div class="card-body">

                                <form action="../upload/upload.php" method="post" enctype="multipart/form-data" class="mb-4">

                                    <input type="hidden" name="perfil" value="<?= htmlspecialchars($usuario['perfil']) ?>">

                                    <input type="hidden" name="email" value="<?= htmlspecialchars($dadosLogado['email']) ?>">

                                    <label for="image" class="form-label fw-bold">Foto de Perfil:</label>

                                    <input class="form-control" type="file" name="image" id="image" accept="image/\*">

                                    <button class="btn btn-primary mt-2" id="saiba" type="submit">Enviar Imagem</button>

                                </form>

                            </div>

                        </div>

                        <!-- Upload de currículo -->

                        <?php if ($usuario['perfil'] === 'funcionario'): ?>

                            <div class="card mb-4 col-md-6" id="adicionar">

                                <div class="card-header">

                                    <h5>Enviar currículo</h5>

                                </div>

                                <div class="card-body">

                                    <form action="../upload/upload\_curriculo.php" method="post" enctype="multipart/form-data" class="mt-4">

                                        <input type="hidden" name="email" value="<?= htmlspecialchars($dadosLogado['email']) ?>">

                                        <label class="fw-bold">Currículo:</label>

                                        <input class="form-control" type="file" name="file" accept="application/pdf,image/\*" required>

                                        <button class="btn btn-secondary mt-2" id="saiba" type="submit">Enviar Currículo</button>

                                    </form>

                                </div>

                            </div>

                        <?php endif; ?>

                    <?php endif; ?>

                </div>

            </div>

        </div>

    </div>

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

<?php

require\_once \_\_DIR\_\_ . '/../services/Auth.php';

use Services\Auth;

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

    $usuario = Auth::getUsuario();

    $isAdmin = Auth::isAdmin();

    $usuarioJsonPath = \_\_DIR\_\_ . '/../data/usuarios.json';

    $funcionarioJsonPath = \_\_DIR\_\_ . '/../data/data\_funcionario.json';

    $empresaJsonPath = \_\_DIR\_\_ . '/../data/data\_empresa.json';

    $usuarios = file\_exists($usuarioJsonPath) ? json\_decode(file\_get\_contents($usuarioJsonPath), true) : [];

    $funcionarios = file\_exists($funcionarioJsonPath) ? json\_decode(file\_get\_contents($funcionarioJsonPath), true) : [];

    $userIdentifier = $\_POST['user\_identifier'] ?? null;

    if ($isAdmin) {

        foreach ($usuarios as &$user) {

            if ($user['username'] === $userIdentifier) {

                $user['username'] = $\_POST['email'];

                $user['nome'] = $\_POST['nome'];

                break;

            }

        }

        unset($user);

        file\_put\_contents($usuarioJsonPath, json\_encode($usuarios, JSON\_PRETTY\_PRINT | JSON\_UNESCAPED\_UNICODE));

    } elseif($usuario['perfil'] === 'funcionario') {

        foreach ($funcionarios as &$func) {

            if ($func['email'] === $userIdentifier) {

                $func['nome'] = $\_POST['nome'];

                $func['email'] = $\_POST['email'];

                $func['telefone'] = $\_POST['telefone'] ?? '';

                $func['cpf'] = $\_POST['cpf'] ?? '';

                $func['atuacao'] = $\_POST['atuacao'] ?? '';

                $func['experiencia'] = $\_POST['experiencia'] ?? '';

                $func['descricao'] = $\_POST['descricao'] ?? '';

                break;

            }

        }

        unset($func);

        file\_put\_contents($funcionarioJsonPath, json\_encode($funcionarios, JSON\_PRETTY\_PRINT | JSON\_UNESCAPED\_UNICODE));

    }else{

        $empresas = file\_exists($empresaJsonPath) ? json\_decode(file\_get\_contents($empresaJsonPath), true) : [];

        foreach ($empresas as &$emp) {

            if ($emp['email'] === $userIdentifier) {

                $emp['nome'] = $\_POST['nome'];

                $emp['cnpj'] = $\_POST['cnpj'] ?? '';

                $emp['endereco'] = $\_POST['endereco'] ?? '';

                break;

            }

        }

        unset($emp);

        file\_put\_contents($empresaJsonPath, json\_encode($empresas, JSON\_PRETTY\_PRINT | JSON\_UNESCAPED\_UNICODE));

    }

    header('Location: ../public/exibicaoaccount.php');

    exit;

}

<?php

// Caminho da pasta onde os currículos estão armazenados

$baseDir = \_\_DIR\_\_ . '/uploads2/';

// Verifica se foi passado um nome de arquivo

if (!isset($\_GET['arquivo'])) {

    die('Arquivo não especificado.');

}

// Evita diretórios relativos no nome do arquivo

$arquivo = basename($\_GET['arquivo']);

$caminhoCompleto = $baseDir . $arquivo;

// Verifica se o arquivo existe

if (!file\_exists($caminhoCompleto)) {

    die('Arquivo não encontrado.');

}

// Força o download

header('Content-Description: File Transfer');

header('Content-Type: application/octet-stream');

header('Content-Disposition: attachment; filename="' . $arquivo . '"');

header('Content-Length: ' . filesize($caminhoCompleto));

readfile($caminhoCompleto);

exit;

?>

<?php

    include 'htmls/head.html';

?>

<body id="login">

    <div class="d-flex col-sm-12">

        <div class="col-sm-6" id="fundo"></div>

        <div class="container col-sm-6 d-flex flex-column align-items-center justify-content-center">

            <section class="col-sm-9" id="form-login">

                <div class="align-self-start" id="texto-cadastro">

                    <h1 class="mb-2">Empresa</h1>

                    <p id="bem-vindo">Seja bem vindo(a)! Realize seu cadastro!</p>

                </div>

                <form class="w-100" method="post" action="salvar-empresa.php">

                    <div>

                        <label for="nome" class="fw-normal">Nome:</label>

                        <input type="text" placeholder="Nome da empresa" id="nome\_empresa" name="nome\_empresa" required class="form-control shadow-sm">

                    </div>

                    <div>

                        <label for="email" class="fw-normal">Email:</label>

                        <input type="email" placeholder="Digite o Email comercial" id="email\_empresa" name="email\_empresa" required class="form-control shadow-sm">

                    </div>

                    <div>

                        <label for="senha" class="fw-normal">Senha:</label>

                        <input type="password" min="8" placeholder="Digite seu senha" id="senha\_empresa" name="senha\_empresa" required class="form-control shadow-sm">

                    </div>

                    <div>

                        <label for="cnpj" class="fw-normal">CNPJ:</label>

                        <input type="number" min="14" placeholder="CNPJ da empresa" id="cnpj" name="cnpj" required class="form-control shadow-sm">

                    </div>

                    <div>

                        <label for="endereco" class="fw-normal">Endereço:</label>

                        <input type="text" placeholder="Endereço da empresa" id="endereco" name="endereco" required class="form-control shadow-sm">

                    </div>

                    <button type="submit" class="btn btn-submit mt-3 w-100  mb-2" id="btn-custom">Cadastre-se:</button>

                    <div id="texto-cadastro"><a href="cadastro.php" class="align-self-start links">Voltar <i class="bi bi-caret-right-fill"></i></a></div>

                </form>

            </section>

        </div>

    </div>

    <script src="script.js"></script>

</body>

</html>

<?php

    include 'htmls/head.html';

?>

<body id="login">

    <div class="d-flex col-sm-12">

        <div class="col-sm-6" id="fundo"></div>

        <div class="container col-sm-6 d-flex flex-column justify-content-center align-items-center">

            <section class="col-sm-9" id="form-login">

                <div class="align-self-start" id="texto-cadastro">

                    <h1 class="mb-2">Funcionário</h1>

                    <p id="bem-vindo">Seja bem vindo(a)! Realize seu cadastro!</p>

                </div>

                <form action="salvar-funcionario.php" method="POST">

                    <div>

                        <label for="nome" class="fw-normal">Nome completo:</label>

                        <input type="text" name="nome\_funcionario" id="nome\_funcionario" required class="form-control shadow-sm" placeholder="Digite seu nome">

                    </div>

                    <div>

                        <label for="email" class="fw-normal">Email:</label>

                        <input type="email" name="email\_funcionario" id="email\_funcionario" required class="form-control shadow-sm" placeholder="Digite seu email">

                    </div>

                    <div>

                        <label for="senha" class="fw-normal">Senha:</label>

                        <input type="password" min="8" name="senha\_funcionario" id="senha\_funcionario" required class="form-control shadow-sm" placeholder="Digite sua senha">

                    </div>

                    <div>

                        <label for="cpf" class="fw-normal">CPF:</label>

                        <input type="number" min="11" name="cpf" id="cpf" required class="form-control shadow-sm" placeholder="Digite seu CPF">

                    </div>

                    <div>

                        <label for="atuacao" class="fw-normal">Área de Atuação</label>

                        <input type="text" name="atuacao" id="atuacao" required class="form-control shadow-sm" placeholder="Digite sua area de atuação(Ex. Engenheiro, Medico e etc)">

                    </div>

                    <div>

                        <label for="experiencia" class="fw-normal">Nivel de Experiência</label>

                        <select class="form-control mb-2" name="nivel-experiencia" id="">

                            <option value="" disabled selected>Selecione seu nível de experiência</option>

                            <option value="iniciante">Iniciante</option>

                            <option value="experiente">Experiente</option>

                            <option value="senior">Sênior</option>

                        </select>

                    </div>

                    <button class="btn btn-submit w-100 mb-2" id="btn-custom">Cadastre-se</button>

                    <div id="texto-cadastro"><a href="cadastro.php" class="links">Voltar <i class="bi bi-caret-right-fill"></i></a></div>

                </form>

            </section>

        </div>

    </div>

    <script src="script.js"></script>

    <script>

    document.querySelectorAll('form').forEach(form => {

        form.addEventListener('submit', function(e) {

            const select = form.querySelector('select[name="nivel-experiencia"]');

            if (select && (select.value === "" || select.value === "empty")) {

                e.preventDefault();

                alert("Por favor, selecione um tipo de veículo válido.");

            }

        });

    });

    </script>

</body>

</html>

<?php

    include 'htmls/head.html';

?>

<body>

    <div class="d-flex col-sm-12 " >

        <div class="col-sm-6" id="fundo">

        </div>

        <main class="container col-sm-6 mt-8 d-flex align-items-center justify-content-center flex-column" id="login">

            <section class="col-sm-9 ms-2" id="form-login">

                <div class="align-self-start" id="texto-login">

                    <h1 class="mb-2">Cadastro</h1>

                    <p id="bem-vindo">Seja bem vindo(a)! Escolha uma opção abaixo.</p>

                </div>

                <div class="col-sm-10 d-flex flex-column align-items-center">

                    <a class="btn btn-submit w-100 p-3" id="btn-custom" style="margin-bottom: 30px; border-radius: 10px;" href="cadastro-empresa.php">Cadastre-se como Empresa</a>

                    <a class="btn btn-submit w-100 p-3 mb-2" id="btn-custom" style="border-radius: 10px;" href="cadastro-funcionario.php"> Cadastre-se como Funcionario</a>

                </div>

                <div id="texto-login"><a href="../index.php" class="links align-self-start">Voltar <i class="bi bi-caret-right-fill"></i></a></div>

            </section>

        </main>

    </div>

    <script src="script.js"></script>

</body>

</html>

<?php

    error\_reporting(E\_ALL);

    ini\_set('display\_errors', 1);

    // incluir o autoload

    require\_once \_\_DIR\_\_ . '/../vendor/autoload.php';

    // incluir o arquivo com as variáveis

    require\_once \_\_DIR\_\_ . '/../config/config.php';

    session\_start();

    // importar as classes Locadora e Auth

    use Services\{Locadora, Auth};

    // inportar as classes Carro e Moto

    use Models\{Inicial, Experiente, Senior};

    // Verifica se o usuário está logado

    if(!Auth::verificarLogin()){

        header('Location: login.php');

        exit;

    }

    // Condição para logout

    if(isset($\_GET['logout'])){

        (new Auth())->logout();

        header('Location: login.php');

        exit;

    }

    // Criar uma instância da classe Locadora

    $locadora = new Locadora();

    $mensagem = '';

    $usuario = Auth::getUsuario();

    // veriifica os dados do formulario via POST

    if($\_SERVER['REQUEST\_METHOD'] === 'POST'){

        // verificar se requer permissão de administrador

        if(isset($\_POST['deletar']) || isset($\_POST['alugar']) || isset($\_POST['devolver'])){

            if(!Auth::isAdmin()){

                $mensagem = "Você não tem permissão para realizar essa ação.";

                goto renderizar;

            }

        }

        if(isset($\_POST['alugar'])){

            $nome = $\_POST['nome'] ?? '';

            $dias = isset($\_POST['dias']) ? (int)$\_POST['dias'] : 1;

            if (empty($nome)) {

                $mensagem = "Erro: O nome do funcionário é obrigatório para alocar.";

            } else {

                $mensagem = $locadora->alocarFuncionario($nome, $dias);

            }

        }

        elseif(isset($\_POST['devolver'])){

            $nome = $\_POST['nome'] ?? '';

            if (empty($nome)) {

                $mensagem = "Erro: O nome do funcionário é obrigatório para liberar.";

            } else {

                $mensagem = $locadora->liberarFuncionario($nome);

            }

        }

        elseif(isset($\_POST['deletar'])){

            $nome = $\_POST['nome'] ?? '';

            if (empty($nome)) {

                $mensagem = "Erro: O nome do funcionário é obrigatório para remover.";

            } else {

                $mensagem = $locadora->removerFuncionario($nome);

            }

        }

        elseif(isset($\_POST['calcular'])){

            $dias = (int)$\_POST['dias\_calculo'];

            $tipo = $\_POST['tipo\_calculo'] ?? '';

            $id = $\_POST['id'] ?? null;

            // Verificar se o tipo selecionado é válido

            if (!in\_array($tipo, ['iniciante', 'experiente', 'senior'])) {

                $mensagem = "Erro: Tipo de funcionário inválido.";

            } else {

                $valor = $locadora->calcularPrevisaoAluguel($dias, $tipo);

                $mensagem = "Previsão de trabalho para {$dias} semana(s) ({$tipo}): R$ " . number\_format($valor, 2, ',', '.');

            }

        }

    }

    renderizar:

    require\_once \_\_DIR\_\_ . '/../public/homepage.php';

<?php

    error\_reporting(E\_ALL);

    ini\_set('display\_errors', 1);

    // incluir o autoload

    require\_once \_\_DIR\_\_ . '/../vendor/autoload.php';

    // incluir o arquivo com as variáveis

    require\_once \_\_DIR\_\_ . '/../config/config.php';

    session\_start();

    // importar as classes Locadora e Auth

    use Services\{Locadora, Auth};

    // inportar as classes Carro e Moto

    use Models\{Inicial, Experiente, Senior};

    // Verifica se o usuário está logado

    if(!Auth::verificarLogin()){

        header('Location: login.php');

        exit;

    }

    // Condição para logout

    if(isset($\_GET['logout'])){

        (new Auth())->logout();

        header('Location: login.php');

        exit;

    }

    // Criar uma instância da classe Locadora

    $locadora = new Locadora();

    $mensagem = '';

    $usuario = Auth::getUsuario();

    // veriifica os dados do formulario via POST

    if($\_SERVER['REQUEST\_METHOD'] === 'POST'){

        // verificar se requer permissão de administrador

        if(isset($\_POST['deletar']) || isset($\_POST['alugar']) || isset($\_POST['devolver'])){

            if(!Auth::isAdmin()){

                $mensagem = "Você não tem permissão para realizar essa ação.";

                goto renderizar;

            }

        }

        if(isset($\_POST['adicionar'])){

            $nome = $\_POST['nome'] ?? '';

            $email = $\_POST['email'] ?? '';

            $tipo = $\_POST['tipo'] ?? '';

            if (empty($nome) || empty($email) || empty($tipo)) {

                $mensagem = "Erro: Todos os campos são obrigatórios para adicionar um funcionário.";

            }

        }

        elseif(isset($\_POST['alugar'])){

            $nome = $\_POST['nome'] ?? '';

            $dias = isset($\_POST['dias']) ? (int)$\_POST['dias'] : 1;

            if (empty($nome)) {

                $mensagem = "Erro: O nome do funcionário é obrigatório para alocar.";

            } else {

                $mensagem = $locadora->alocarFuncionario($nome, $dias);

            }

        }

        elseif(isset($\_POST['devolver'])){

            $nome = $\_POST['nome'] ?? '';

            if (empty($nome)) {

                $mensagem = "Erro: O nome do funcionário é obrigatório para liberar.";

            } else {

                $mensagem = $locadora->liberarFuncionario($nome);

            }

        }

        elseif(isset($\_POST['deletar'])){

            $nome = $\_POST['nome'] ?? '';

            if (empty($nome)) {

                $mensagem = "Erro: O nome do funcionário é obrigatório para remover.";

            } else {

                $mensagem = $locadora->removerFuncionario($nome);

            }

        }

        elseif(isset($\_POST['calcular'])){

            $dias = (int)$\_POST['dias\_calculo'];

            $tipo = $\_POST['tipo\_calculo'] ?? '';

            // Verificar se o tipo selecionado é válido

            if (!in\_array($tipo, ['iniciante', 'experiente', 'senior'])) {

                $mensagem = "Erro: Tipo de funcionário inválido.";

            } else {

                $valor = $locadora->calcularPrevisaoAluguel($dias, $tipo);

                $mensagem = "Previsão de trabalho para {$dias} dias ({$tipo}): R$ " . number\_format($valor, 2, ',', '.');

            }

        }

    }

    renderizar:

    require\_once \_\_DIR\_\_ . '/../public/accountpage.php';

<?php

require\_once \_\_DIR\_\_ .'/../services/Auth.php';

require\_once \_\_DIR\_\_ . '/../services/locadora.php';

use Services\Auth;

$mensagem = '';

$usuario = Auth::getUsuario();

$funcionarios = json\_decode(file\_get\_contents(\_\_DIR\_\_ . '/../data/funcionarios.json'), true);

$dado\_funcionarios = json\_decode(file\_get\_contents(\_\_DIR\_\_ . '/../data/data\_funcionario.json'), true);

$dado\_empresa = json\_decode(file\_get\_contents(\_\_DIR\_\_ . '/../data/data\_empresa.json'), true);

// Caminho do arquivo de usuários

$usuarioJsonPath = \_\_DIR\_\_ . '/../data/usuarios.json';

if (file\_exists($usuarioJsonPath)) {

    $usuarios = json\_decode(file\_get\_contents($usuarioJsonPath), true);

} else {

    $usuarios = [];

}

// Deletar funcionário

if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['deletar'])) {

    $id = $\_POST['id'];

    // Verificar se o funcionário existe

    if (isset($dado\_funcionarios[$id])) {

        $emailParaRemover = $dado\_funcionarios[$id]['email'];

        // Remover do arquivo funcionarios.json

        unset($funcionarios[$id]);

        // Remover do arquivo data\_funcionario.json

        unset($dado\_funcionarios[$id]);

        // Reindexar arrays para evitar lacunas

        $funcionarios = array\_values($funcionarios);

        $dado\_funcionarios = array\_values($dado\_funcionarios);

        // Remover do arquivo usuarios.json

        $usuarios = array\_filter($usuarios, function($usuario) use ($emailParaRemover) {

            return $usuario['username'] !== $emailParaRemover;

        });

        // Reindexar usuários

        $usuarios = array\_values($usuarios);

        // Salvar os arquivos modificados

        file\_put\_contents(\_\_DIR\_\_ . '/../data/funcionarios.json', json\_encode($funcionarios, JSON\_PRETTY\_PRINT | JSON\_UNESCAPED\_UNICODE));

        file\_put\_contents(\_\_DIR\_\_ . '/../data/data\_funcionario.json', json\_encode($dado\_funcionarios, JSON\_PRETTY\_PRINT | JSON\_UNESCAPED\_UNICODE));

        file\_put\_contents($usuarioJsonPath, json\_encode($usuarios, JSON\_PRETTY\_PRINT | JSON\_UNESCAPED\_UNICODE));

        header("Location: " . $\_SERVER['PHP\_SELF']);

        exit;

    }

}

// Editar funcionário

if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['editar'])) {

    $id = $\_POST['id'];

    $dado\_funcionarios[$id]['nome'] = $\_POST['nome'];

    $dado\_funcionarios[$id]['email'] = $\_POST['email'];

    $dado\_funcionarios[$id]['experiencia'] = $\_POST['nivel'];

    $dado\_funcionarios[$id]['descricao'] = $\_POST['descricao'];

    file\_put\_contents(\_\_DIR\_\_ . '/../data/data\_funcionario.json', json\_encode($dado\_funcionarios, JSON\_PRETTY\_PRINT | JSON\_UNESCAPED\_UNICODE));

    header("Location: " . $\_SERVER['PHP\_SELF']);

    exit;

}

// --- CÁLCULO DE ALUGUEL NO MODAL ---

if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['calcular'])) {

    $dias = isset($\_POST['dias\_calculo']) ? (int)$\_POST['dias\_calculo'] : 1;

    $tipo = $\_POST['tipo\_calculo'] ?? '';

    $nome = $\_POST['nome\_calculo'] ?? '';

    require\_once \_\_DIR\_\_ . '/../config/config.php';

    $valor = 0;

    if ($tipo === 'iniciante' && defined('SEMANAL\_INICIANTE')) {

        $valor = $dias \* SEMANAL\_INICIANTE;

    } elseif ($tipo === 'experiente' && defined('SEMANAL\_EXPERIENTE')) {

        $valor = $dias \* SEMANAL\_EXPERIENTE;

    } elseif ($tipo === 'senior' && defined('SEMANAL\_SENIOR')) {

        $valor = $dias \* SEMANAL\_SENIOR;

    }

    $mensagem = "Previsão de contratação do(a) $nome para $dias semana(s) ($tipo): R$ " . number\_format($valor, 2, ',', '.');

}

// --- ALUGAR FUNCIONÁRIO ---

if (isset($\_POST['alugar'])) {

    $nome = $\_POST['nome\_aluguel'] ?? '';

    $tipo = strtolower($\_POST['tipo\_aluguel'] ?? '');

    $dias = isset($\_POST['dias\_aluguel']) ? (int)$\_POST['dias\_aluguel'] : 1;

    if (empty($nome)) {

        $mensagem = "Erro: O nome do funcionário não encontrado tente novamente.";

    } else {

        $mensagem = $locadora->alugarFuncionario($nome, $tipo, $dias);

    }

}

// --- DEVOLVER FUNCIONÁRIO ---

if (isset($\_POST['devolver'])) {

    $nome = $\_POST['nome\_devolver'] ?? '';

    if (empty($nome)) {

        $mensagem = "Erro: O nome do funcionário não encontrado tente novamente.";

    } else {

        $mensagem = $locadora->devolverFuncionario($nome);

    }

}

?>

<?php include 'htmls/head.html'; ?>

<body>

<?php include 'htmls/nav.html'; ?>

<div class="py-4 mx-2">

    <div class="container mt-4">

        <div class="row">

            <div class="col-md-8">

                <div class="card mb-4" id="principal">

                    <div class="card-body d-flex flex-row justify-content-between">

                        <div>

                            <h1 class="card-title" id="titulo-card">SERVHUB</h1>

                            <p class="card-text" id="texto-card">Conectando talentos e impulsionando oportunidades!</p>

                        </div>

                        <img id="logo-card" src="Assets/logo.png" alt="" class="img-fluid">

                    </div>

                </div>

                <div class="card mb-4" id="responsivo">

                    <div class="card-body d-flex flex-row justify-content-around">

                        <img style="width: 50px;" src="Assets/adm.png" alt="" class="img-fluid">

                        <a href="exibicaoaccount.php" class="btn btn-submit w-100 mb-2" id="btn-custom">Editar Perfil</a>

                    </div>

                </div>

            </div>

            <div class="col-md-4" id="sidebar">

                <div class="card mb-4">

                    <div class="card-body d-flex flex-column align-items-center">

                        <?php if (Auth::isAdmin()): ?>

                            <img style="width: 200px;" src="Assets/adm.png" alt="Administrador">

                            <h3 class="card-title"><?= htmlspecialchars($usuario['username']) ?></h3>

                            <p style="font-size: 0.8rem; font-weight: lighter;" id="descricao-adm" class="mt-0 mx-3 fw-normal">Você é o administrador da empresa, altere o que for necessário e deslogue sua conta para ter mais segurança.</p>

                        <?php else: ?>

                            <?php

                                $funcionarioLogado = null;

                                foreach ($dado\_funcionarios as $funcionario) {

                                    if (isset($funcionario['email']) &&

                                        $funcionario['email'] === $usuario['username'] &&

                                        isset($funcionario['descricao']) &&

                                        isset($funcionario['path'])) {

                                        $funcionarioLogado = $funcionario;

                                        break;

                                    }

                                }

                                $empresa = null;

                                foreach ($dado\_empresa as $emp){

                                    if (isset($emp['email']) && $emp['email'] === $usuario['username']) {

                                        $empresa = $emp;

                                        break;

                                    }

                                }

                            ?>

                            <?php if ($usuario['perfil'] != 'empresa') : ?>

                                <img id="foto-perfil" src="<?= htmlspecialchars($funcionarioLogado['path'] ?? 'Assets/default.png') ?>" alt="Funcionário">

                                <h3 style="font-size: 1.2rem; font-weight: bold;" class="card-title"><?= htmlspecialchars($funcionarioLogado['nome'] ?? 'Sem nome') ?></h3>

                                <p style="font-size: 1rem;" class="mt-0"><?= htmlspecialchars($funcionarioLogado['email'] ?? 'Sem email') ?></p>

                                <?php if ($funcionarioLogado['descricao'] === ""): ?>

                                    <p style="font-size: 0.8rem; font-weight: lighter;" class="mt-0 mx-3">Sem descrição.</p>

                                <?php else: ?>

                                    <p style="font-size: 0.8rem; font-weight: lighter;" class="mt-0 mx-3"><?= htmlspecialchars($funcionarioLogado['descricao']) ?></p>

                                <?php endif; ?>

                            <?php else: ?>

                                <img id="foto-perfil" src="<?= htmlspecialchars($empresa['path'] ?? 'Assets/adm.png') ?>" alt="Funcionário">

                                <h3 style="font-size: 1.2rem; font-weight: bold;" class="card-title"><?= htmlspecialchars($empresa['nome'] ?? 'sem nome') ?></h3>

                                <p style="font-size: 1rem;" class="mt-0"><?= htmlspecialchars($empresa['email'] ?? 'sem nome') ?></p>

                            <?php endif; ?>

                        <?php endif; ?>

                        <a href="exibicaoaccount.php" class="btn btn-submit w-100 mb-2" id="btn-custom">Editar Perfil</a>

                    </div>

                </div>

            </div>

        </div>

        <?php if ($mensagem): ?>

            <div class="alert alert-info alert-dismissible fade show" role="alert">

                <?= htmlspecialchars($mensagem) ?>

                <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>

            </div>

        <?php endif; ?>

        <?php

            $busca = $\_GET['pesquisa'] ?? '';

            $dado\_funcionarios = json\_decode(file\_get\_contents('../data/data\_funcionario.json'), true);

            // Filtrar funcionários, se houver termo de busca

            if ($busca) {

                $busca = strtolower($busca);

                $dado\_funcionarios = array\_filter($dado\_funcionarios, function ($func) use ($busca) {

                    return strpos(strtolower($func['nome']), $busca) !== false ||

                        strpos(strtolower($func['atuacao']), $busca) !== false;

                });

            }

        ?>

        <div class="row mt-3">

        <?php foreach ($dado\_funcionarios as $index => $funcionario): ?>

        <?php

            $modalId = "editarFuncionario" . $index;

            $saibaMaisModalId = "saibaMaisFuncionario" . $index;

        ?>

        <div class="col-sm-6 col-md-4 col-lg-4 mb-4">

            <div class="card h-100">

                <div style="height: 200px;" class="d-flex align-items-center justify-content-center">

                    <img src="<?= htmlspecialchars($funcionario['path'] ?? 'Assets/default.png') ?>" alt="profissional" class="card-img-top w-100 h-100">

                </div>

                <div class="card-body">

                    <h5 class="card-title"><?= htmlspecialchars($funcionario['nome'] ?? 'Sem nome') ?></h5>

                    <p class="card-text"><?= htmlspecialchars($funcionario['atuacao'] ?? 'Sem Area de atuação') ?></p>

                    <div class="d-flex gap-2">

                        <button id="saiba" class="btn btn-success" data-bs-toggle="modal" data-bs-target="#<?= $saibaMaisModalId ?>">Saiba mais...</button>

                        <?php if (Auth::isAdmin()): ?>

                            <button class="btn btn-warning" data-bs-toggle="modal" data-bs-target="#<?= $modalId ?>"><i class="bi bi-pen-fill text-black"></i></button>

                            <form method="POST" onsubmit="return confirm('Tem certeza que deseja excluir este funcionário?')">

                                <input type="hidden" name="id" value="<?= $index ?>">

                                <button type="submit" name="deletar" class="btn btn-danger">

                                    <i class="bi bi-trash-fill text-black"></i>

                                </button>

                            </form>

                        <?php endif; ?>

                    </div>

                </div>

            </div>

        </div>

        <!-- Modal de Edição -->

        <div class="modal fade" id="<?= $modalId ?>" tabindex="-1" aria-labelledby="<?= $modalId ?>Label" aria-hidden="true">

            <div class="modal-dialog">

                <div class="modal-content">

                    <form method="POST">

                        <div class="modal-header">

                            <h5 class="modal-title" id="<?= $modalId ?>Label">Editar Funcionário</h5>

                            <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Fechar"></button>

                        </div>

                        <div class="modal-body">

                            <input type="hidden" name="id" value="<?= $index ?>">

                            <div class="mb-3">

                                <label class="form-label">Nome</label>

                                <input type="text" name="nome" class="form-control" value="<?= htmlspecialchars($funcionario['nome'] ?? '') ?>" required>

                            </div>

                            <div class="mb-3">

                                <label class="form-label">Email</label>

                                <input type="email" name="email" class="form-control" value="<?= htmlspecialchars($funcionario['email'] ?? '') ?>" required>

                            </div>

                            <div class="mb-3">

                                <label class="form-label">Nível de Experiência</label>

                                <select class="form-control mb-2" name="nivel" required>

                                    <option value="" disabled <?= empty($funcionario['experiencia']) ? 'selected' : '' ?>>Selecione seu nível de experiência</option>

                                    <option value="iniciante" <?= ($funcionario['experiencia'] === 'iniciante') ? 'selected' : '' ?>>Iniciante</option>

                                    <option value="experiente" <?= ($funcionario['experiencia'] === 'experiente') ? 'selected' : '' ?>>Experiente</option>

                                    <option value="senior" <?= ($funcionario['experiencia'] === 'senior') ? 'selected' : '' ?>>Sênior</option>

                                </select>

                            </div>

                            <div class="mb-3">

                                <label class="form-label">Descrição</label>

                                <textarea name="descricao" class="form-control" rows="3" required><?= htmlspecialchars($funcionario['descricao'] ?? '') ?></textarea>

                            </div>

                        </div>

                        <div class="modal-footer">

                            <button type="submit" name="editar" class="btn btn-primary">Salvar</button>

                            <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cancelar</button>

                        </div>

                    </form>

                </div>

            </div>

        </div>

        <!-- Modal Saiba Mais -->

        <div class="modal fade" id="<?= $saibaMaisModalId ?>" tabindex="-1" aria-labelledby="<?= $saibaMaisModalId ?>Label" aria-hidden="true">

            <div class="modal-dialog modal-lg">

                <div class="modal-content">

                    <div class="modal-header">

                        <h5 class="modal-title" id="<?= $saibaMaisModalId ?>Label">Detalhes do Funcionário</h5>

                        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Fechar"></button>

                    </div>

                    <div class="modal-body d-flex flex-column flex-md-row align-items-center gap-3 flex-wrap">

                        <div class="col-sm-5">

                            <img src="<?= htmlspecialchars($funcionario['path'] ?? 'Assets/default.png') ?>" alt="Foto do Funcionário" class="w-100" style="max-height: 300px; border-radius: 20px" >

                        </div>

                        <div class="col-sm-6">

                            <div class="col-sm-12">

                                <h4 class="fw-bold mb-0"><?= htmlspecialchars($funcionario['nome'] ?? 'Sem nome') ?></h4>

                                <p class="fw-normal mt-0 mb-1"><?= htmlspecialchars($funcionario['atuacao'] ?? 'Não especificada') ?></p>

                                <p class="fw-normal mt-2"><?= nl2br(htmlspecialchars($funcionario['descricao'] ?? 'Sem descrição')) ?></p>

                            </div>

                            <div class="col-sm-12 d-flex flex-row">

                                <div class="col-sm-6 me-5">

                                    <p><strong>Experiência:</strong><br> <?= htmlspecialchars($funcionario['experiencia'] ?? 'Não informado') ?></p>

                                </div>

                                <div class="col-sm-6 me-5">

                                    <p><strong>Salário:</strong><br>

                                        <?php if ($funcionario['experiencia'] === 'iniciante'): ?>

                                            R$ <?= number\_format(defined('SEMANAL\_INICIANTE') ? SEMANAL\_INICIANTE : 800, 2, ',', '.') ?> por semana

                                        <?php elseif ($funcionario['experiencia'] === 'experiente'): ?>

                                            R$ <?= number\_format(defined('SEMANAL\_EXPERIENTE') ? SEMANAL\_EXPERIENTE : 1200, 2, ',', '.') ?> por semana

                                        <?php elseif ($funcionario['experiencia'] === 'senior'): ?>

                                            R$ <?= number\_format(defined('SEMANAL\_SENIOR') ? SEMANAL\_SENIOR : 1800, 2, ',', '.') ?> por semana

                                        <?php else: ?>

                                            Não informado

                                        <?php endif; ?>

                                    </p>

                                </div>

                            </div>

                            <?php if ($usuario['perfil'] === 'empresa'): ?>

                                <div class="col-sm-12 d-flex flex-row">

                                    <div class="col-sm-6 me-5">

                                        <p><strong>Email:</strong></p>

                                        <p><?= htmlspecialchars($funcionario['email'] ?? 'Sem email') ?></p>

                                    </div>

                                    <div class="col-sm-6">

                                        <p><strong>Telefone:</strong></p>

                                        <p><?= htmlspecialchars($funcionario['telefone'] ?? 'Sem telefone') ?></p>

                                    </div>

                                </div>

                            <?php endif; ?>

                        </div>

                        <!-- FORMULÁRIO DE CÁLCULO -->

                        <div class="col-sm-12">

                            <form action="exibicao.php" class="needs-validation" novalidate method="post" id="formCalculo-<?= $index ?>">

                                <div class="mb-3">

                                    <input type="text" hidden name="tipo\_calculo" value="<?= htmlspecialchars($funcionario['experiencia'] ?? '') ?>">

                                    <input type="text" hidden name="nome\_calculo" value="<?= htmlspecialchars($funcionario['nome'] ?? '') ?>">

                                    <input type="hidden" name="id" value="<?= $index ?>">

                                    <label for="dias" class="form-label">

                                        Calcule a previsão de aluguel desse funcionário<br>

                                        Quantidade de semanas:

                                    </label>

                                    <input type="number" name="dias\_calculo" class="form-control" value="1" required>

                                </div>

                                <button class="btn btn-info w-100" id="calcular" type="submit" name="calcular">Calcular</button>

                            </form>

                        </div>

                        <?php if (!empty($funcionario['disponivel']) && $funcionario['disponivel'] === true): ?>

                            <?php if (Auth::isAdmin() || $usuario['perfil'] === 'empresa'): ?>

                                <div class="col-sm-12">

                                    <form method="post" class="needs-validation" novalidate>

                                        <div class="mb-3">

                                            <input type="hidden" name="tipo\_aluguel" value="<?= htmlspecialchars($funcionario['experiencia'] ?? '') ?>">

                                            <input type="hidden" name="nome\_aluguel" value="<?= htmlspecialchars($funcionario['nome'] ?? '') ?>">

                                            <label for="dias" class="form-label">Faça a contratação desse funcionário</label>

                                            <input type="number" name="dias\_aluguel" class="form-control" value="1" required>

                                        </div>

                                        <button type="submit" name="alugar" class="btn btn-success w-100">Alugar</button>

                                    </form>

                                </div>

                            <?php endif; ?>

                        <?php elseif ($usuario['perfil'] === 'empresa'): ?>

                            <p>Funcionário já contratado. Espere ele ser devolvido para poder contratá-lo.</p>

                        <?php elseif (Auth::isAdmin()): ?>

                            <div class="d-flex flex-column gap-0">

                                <p>Funcionário não disponível para contratação.</p><br>

                                <form action="" class="needs-validation" novalidate method="post">

                                    <input type="hidden" name="nome\_devolver" value="<?= htmlspecialchars($funcionario['nome'] ?? '') ?>">

                                    <button name="devolver" class="btn btn-success" type="submit">Devolver</button>

                                </form>

                            </div>

                        <?php endif; ?>

                    </div>

                    <?php if (Auth::isAdmin()): ?>

                        <div class="modal-footer">

                            <form method="POST" onsubmit="return confirm('Tem certeza que deseja excluir este funcionário?')">

                                <input type="hidden" name="id" value="<?= $index ?>">

                                <button type="submit" name="deletar" class="btn btn-danger">

                                    <i class="bi bi-trash-fill"></i> Excluir

                                </button>

                            </form>

                            <button class="btn btn-warning" data-bs-dismiss="modal" data-bs-toggle="modal" data-bs-target="#<?= $modalId ?>">

                                <i class="bi bi-pen-fill"></i> Editar

                            </button>

                        </div>

                    <?php endif; ?>

                    <a href="baixar\_curriculo.php?arquivo=<?= urlencode($funcionario['filename\_curriculo']) ?>" class="btn btn-primary">

                        Baixar Currículo

                    </a>

                </div>

            </div>

        </div>

         <!-- Modal add funcionário por adm -->

        <div class="modal fade" id="funcionario" tabindex="-1"  aria-hidden="true">

            <div class="modal-dialog modal-lg">

                <div class="modal-content">

                    <div class="modal-header">

                        <h5 class="modal-title" >Adicionar funcionário</h5>

                        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Fechar"></button>

                    </div>

                    <div class="modal-body d-flex flex-column flex-md-row align-items-center gap-3 flex-wrap">

                        <form class="w-100" action="salvar-funcionario.php" method="POST">

                            <div>

                                <label for="nome" class="fw-normal">Nome completo:</label>

                                <input type="text" name="nome\_funcionario" id="nome\_funcionario" required class="form-control shadow-sm" placeholder="Digite seu nome">

                            </div>

                            <div>

                                <label for="email" class="fw-normal">Email:</label>

                                <input type="email" name="email\_funcionario" id="email\_funcionario" required class="form-control shadow-sm" placeholder="Digite seu email">

                            </div>

                            <div>

                                <label for="senha" class="fw-normal">Senha:</label>

                                <input type="password" min="8" name="senha\_funcionario" id="senha\_funcionario" required class="form-control shadow-sm" placeholder="Digite sua senha">

                            </div>

                            <div>

                                <label for="cpf" class="fw-normal">CPF:</label>

                                <input type="number" min="11" name="cpf" id="cpf" required class="form-control shadow-sm" placeholder="Digite seu CPF">

                            </div>

                            <div>

                                <label for="atuacao" class="fw-normal">Área de Atuação</label>

                                <input type="text" name="atuacao" id="atuacao" required class="form-control shadow-sm" placeholder="Digite sua area de atuação(Ex. Engenheiro, Medico e etc)">

                            </div>

                            <div>

                                <label for="experiencia" class="fw-normal">Nivel de Experiência</label>

                                <select class="form-control mb-2" name="nivel-experiencia" id="">

                                    <option value="" disabled selected>Selecione seu nível de experiência</option>

                                    <option value="iniciante">Iniciante</option>

                                    <option value="experiente">Experiente</option>

                                    <option value="senior">Sênior</option>

                                </select>

                            </div>

                            <button class="btn btn-submit w-100 mb-2" id="btn-custom">Cadastre-se</button>

                        </form>

                    </div>

                </div>

            </div>

        </div>

        <!-- Modal add empresa por adm -->

        <div class="modal fade" id="empresa" tabindex="-1"  aria-hidden="true">

            <div class="modal-dialog modal-lg">

                <div class="modal-content">

                    <div class="modal-header">

                        <h5 class="modal-title" >Adicionar Empresa</h5>

                        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Fechar"></button>

                    </div>

                    <div class="modal-body d-flex flex-column flex-md-row align-items-center gap-3 flex-wrap">

                        <form class="w-100" method="post" action="salvar-empresa.php">

                            <div>

                                <label for="nome" class="fw-normal">Nome:</label>

                                <input type="text" placeholder="Nome da empresa" id="nome\_empresa" name="nome\_empresa" required class="form-control shadow-sm">

                            </div>

                            <div>

                                <label for="email" class="fw-normal">Email:</label>

                                <input type="email" placeholder="Digite o Email comercial" id="email\_empresa" name="email\_empresa" required class="form-control shadow-sm">

                            </div>

                            <div>

                                <label for="senha" class="fw-normal">Senha:</label>

                                <input type="password" min="8" placeholder="Digite seu senha" id="senha\_empresa" name="senha\_empresa" required class="form-control shadow-sm">

                            </div>

                            <div>

                                <label for="cnpj" class="fw-normal">CNPJ:</label>

                                <input type="number" min="14" placeholder="CNPJ da empresa" id="cnpj" name="cnpj" required class="form-control shadow-sm">

                            </div>

                            <div>

                                <label for="endereco" class="fw-normal">Endereço:</label>

                                <input type="text" placeholder="Endereço da empresa" id="endereco" name="endereco" required class="form-control shadow-sm">

                            </div>

                            <button type="submit" class="btn btn-submit mt-3 w-100  mb-2" id="btn-custom">Cadastre-se:</button>

                        </form>

                    </div>

                </div>

            </div>

        </div>

    <?php endforeach; ?>

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

<?php

ini\_set('display\_errors', 1);

ini\_set('display\_startup\_errors', 1);

error\_reporting(E\_ALL);

// incluir o auto load do composer para carregar automaticamente as classes utilizadas

require\_once \_\_DIR\_\_ . '/../vendor/autoload.php';

// incluir o arquivo com as variáveis

require\_once \_\_DIR\_\_ . '/../config/config.php';

session\_start();

// inserir a classe de autenticação

use Services\Auth;

// Inicializa a variável para mensagens de erro

$mensagem = '';

// instanciar a classe de autenticação

$auth = new Auth();

// verifica se já foi autenticado

if ($auth->verificarLogin()) { // Alterado para usar a instância $auth

    echo "Usuário já autenticado. Redirecionando...";

    header('Location: exibicao.php');

    exit;

}

// verifica se o formulário foi enviado

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

    $username = $\_POST['email'] ?? '';

    $password = $\_POST['senha'] ?? '';

    if ($auth->login($username, $password)) {

        header('Location: exibicao.php');

        exit;

    } else {

        $mensagem = 'Usuário ou senha incorretos!';

    }

}

?>

<?php

    include 'htmls/head.html';

?>

<body>

    <div class="d-flex col-sm-12 " >

        <div class="col-sm-6" id="fundo">

        </div>

        <main class="container col-sm-6 d-flex align-items-center justify-content-center flex-column">

            <section class="col-sm-9" id="form-login">

                <div id="texto-login">

                    <h1 class="mb-2">Login</h1>

                    <p id="bem-vindo">Seja bem vindo(a)! Faça seu login abaixo.</p>

                </div>

                <form action="#" method="POST" class="form w-100">

                    <div>

                        <label for="email" class="fw-normal">Email:</label>

                        <input type="email" id="email" name="email" class="form-control input-custom shadow-sm" required />

                    </div>

                    <div>

                        <label for="senha" class="fw-normal">Senha:</label>

                        <input type="password" id="senha" name="senha" class="form-control input-custom shadow-sm"  required />

                    </div>

                    <button type="submit" class="btn btn-submit mt-3 mb-3 w-100 " id="btn-custom">Entrar</button>

                </form>

                <div class="d-flex justify-content-start flex-column align-items-start">

                    <p id="senha-esqueceu" class="mb-2">Não tem uma conta? <a href="cadastro.php" class="links">Cadastre-se</a></p>

                    <div id="texto-cadastro" class="aling-self-center"><a href="../index.php" class="links" >Voltar <i class="bi bi-caret-right-fill"></i></a></div>

                </div>

            </section>

        </main>

    </div>

    <script src="script.js"></script>

</body>

</html>

<?php

// Corrigir o caminho do arquivo JSON para usar \_\_DIR\_\_

$dados = json\_decode(file\_get\_contents(\_\_DIR\_\_ . '/../data/data\_empresa.json'), true);

$usuario = json\_decode(file\_get\_contents(\_\_DIR\_\_ . '/../data/usuarios.json'), true);

// verifica se deu certo a comverção

if (!is\_array($dados)){

    $dados = [];

}

if (!is\_array($usuario)){

    $usuario = [];

}

// Codificar a senha antes de salvar

$senhaCodificada = password\_hash($\_POST["senha\_empresa"], PASSWORD\_DEFAULT);

// adiciona as informações do form as respectivas categorias

$novoDado = [

    "nome" => $\_POST["nome\_empresa"],

    "email" => $\_POST["email\_empresa"],

    "senha" => $senhaCodificada, // Senha codificada

    "cnpj" => $\_POST["cnpj"],

    "endereco" => $\_POST["endereco"],

    "path" => "",

    "filename" => "",

    "uploaded\_at" => "",

];

// adiciona as informações do form as respectivas categorias

$novoUsuario = [

    "username" => $\_POST["email\_empresa"],

    "perfil" => "empresa",

    "password" => $senhaCodificada // Senha codificada

];

// adiciona os novos dados aos já existentes

if (strlen($\_POST["senha\_empresa"])) {

    $dados[] = $novoDado;

    // transforma os dados novamente em JSON e salva no arquivo data\_empresa.json

    file\_put\_contents(\_\_DIR\_\_ . '/../data/data\_empresa.json', json\_encode($dados, JSON\_PRETTY\_PRINT));

    // Adicionar o novo usuário ao array de usuários

    $usuario[] = $novoUsuario;

    // transforma os dados novamente em JSON e salva no arquivo usuarios.json

    file\_put\_contents(\_\_DIR\_\_ . '/../data/usuarios.json', json\_encode($usuario, JSON\_PRETTY\_PRINT));

    echo "dados salvos com sucesso ! <a href='cadastro-empresa.php'>voltar </a>";

} else {

    echo "Erro: os dados não foram salvos! <a href='cadastro-empresa.php'>voltar </a>";

}

?>

 <script>

    setTimeout(() => {

      window.location.href = "Login.php";

    }, 0);

</script>

<?php

// Corrigir o caminho do arquivo JSON para usar \_\_DIR\_\_

$dados = json\_decode(file\_get\_contents(\_\_DIR\_\_ . '/../data/data\_funcionario.json'), true);

$usuario = json\_decode(file\_get\_contents(\_\_DIR\_\_ . '/../data/usuarios.json'), true);

$funcionarios = json\_decode(file\_get\_contents(\_\_DIR\_\_ . '/../data/funcionarios.json'), true);

// verifica se deu certo a comverção

if (!is\_array($dados)){

    $dados = [];

}

if (!is\_array($funcionarios)) {

    $funcionarios = [];

}

// Codificar a senha antes de salvar

$senhaCodificada = password\_hash($\_POST["senha\_funcionario"], PASSWORD\_DEFAULT);

// adiciona as informações do form as respectivas categorias

$novoDado = [

    "nome" => $\_POST["nome\_funcionario"],

    "email" => $\_POST["email\_funcionario"],

    "senha" => $senhaCodificada, // Senha codificada

    "cpf" => $\_POST["cpf"],

    "atuacao" => $\_POST["atuacao"],

    "descricao" => "",

    "experiencia" => $\_POST["nivel-experiencia"],

    "telefone" => "",

    "path" => "",

    "filename"=> "",

    "uploaded\_at"=> "",

    "path\_curriculo"=> "",

    "filename\_curriculo"=> "",

    "curriculo\_uploaded\_at"=> "",

    "disponivel" => true,

];

// adiciona as informações do form as respectivas categorias

$novoUsuario = [

    "username" => $\_POST["email\_funcionario"],

    "perfil" => "funcionario",

    "password" => $senhaCodificada, // Senha codificada

];

$novoFuncionario = [

    "tipo" => $\_POST["nivel-experiencia"],

    "nome" => $\_POST["nome\_funcionario"],

    "experiencia" => $\_POST["nivel-experiencia"],

    "disponivel" => true,

];

// adiciona os novos dados aos já existentes

if (strlen($\_POST["nome\_funcionario"])) {

    $dados[] = $novoDado;

    // transforma os dados novamente em JSON e salva no arquivo data\_empresa.json

    file\_put\_contents(\_\_DIR\_\_ . '/../data/data\_funcionario.json', json\_encode($dados, JSON\_PRETTY\_PRINT));

    // Adicionar o novo usuário ao array de usuários

    $usuario[] = $novoUsuario;

    // transforma os dados novamente em JSON e salva no arquivo usuarios.json

    file\_put\_contents(\_\_DIR\_\_ . '/../data/usuarios.json', json\_encode($usuario, JSON\_PRETTY\_PRINT));

    // Adicionar o novo funcionário ao array de funcionários

    $funcionarios[] = $novoFuncionario;

    // Salvar o array atualizado em funcionarios.json

    file\_put\_contents(\_\_DIR\_\_ . '/../data/funcionarios.json', json\_encode($funcionarios, JSON\_PRETTY\_PRINT));

    echo "dados salvos com sucesso ! <a href='cadastro-empresa.php'>voltar </a>";

} else {

    echo "Erro: os dados não foram salvos! <a href='cadastro-empresa.php'>voltar </a>";

}

?>

 <script>

    setTimeout(() => {

      window.location.href = "Login.php";

    }, 0);

</script>

<?php

// define espaços para organização do código

namespace Services;

class Auth{

    private array $usuario = [];

    // Método cronstuctor

    public function \_\_construct()

    {

        $this ->carregarUsuarios();

    }

    // Método para carregar os usuários do arquivo JSON

    private function carregarUsuarios(): void

    {

        // Verifica se o arquivo existe e carrega os usuários

        if (file\_exists(ARQUIVO\_USUARIOS)) {

            // Lê o conteúdo do arquivo JSON e decodifica para um array

            $conteudo = json\_decode(file\_get\_contents(ARQUIVO\_USUARIOS), true);

            $this->usuario = is\_array($conteudo) ? $conteudo : [];

        } else {

            // Se o arquivo não existir, cria usuarios padrão

            $this->usuario = [

                [

                'username' => 'admin',

                'password' => password\_hash('admin123,', PASSWORD\_DEFAULT),

                'perfil' => 'admin',

                ],

                [

                'username' => 'usuario',

                'password' => password\_hash('user123', PASSWORD\_DEFAULT),

                'perfil' => 'usuario',

                ],

                [

                'username' => 'Gabriel',

                'password' => password\_hash('user123', PASSWORD\_DEFAULT),

                'perfil' => 'admin',

                ]

            ];

            $this->salvarUsuarios();

        }

    }

    // Método para salvar os usuários no arquivo JSON

    private function salvarUsuarios(): void

    {

        $dir = dirname(ARQUIVO\_USUARIOS);

        // Verifica se o diretório existe, caso contrário cria

        if (!is\_dir($dir)) {

            mkdir($dir, 0777, true);

        }

        // Salva os usuários no arquivo JSON

        file\_put\_contents(ARQUIVO\_USUARIOS, json\_encode($this->usuario, JSON\_PRETTY\_PRINT));

    }

    // Método para login

    public function login(string $username, string $password): bool

    {

        // Verifica se o usuário existe e a senha está correta

        foreach ($this->usuario as $usuario) {

            if ($usuario['username'] === $username && password\_verify($password, $usuario['password'])) {

                $\_SESSION['auth'] = [

                    'logado' => true,

                    'username' => $username,

                    'perfil' => $usuario['perfil'],

                ];

                return true; // Login bem-sucedido

            }

        }

        return false;// Login falhou

    }

    public function logout(): void

    {

        session\_destroy();

    }

    // verifica se o usuário está logado

    public static function verificarLogin(): bool {

        return isset($\_SESSION['auth']) && $\_SESSION['auth']['logado'] === true;

    }

    public static function isPerfil(string $perfil): bool

    {

        return isset($\_SESSION['auth']) && $\_SESSION['auth']['perfil'] === $perfil;

    }

    public static function isAdmin(): bool

    {

        return self::isPerfil('admin');

    }

    public static function getUsuario(): ?array

    {

        // retorna os dados da sessao ou nulo se nao existir

        return $\_SESSION['auth'] ?? null;

    }

}

?>

<?php

namespace Services;

use Models\{Pessoa, Iniciante, Experiente, Senior};

// Classe para gerenciar a locação

class Locadora {

    private array $funcionarios = [];

    public function \_\_construct() {

        $this->carregarFuncionarios();

    }

    // Adicionar logs para depuração ao carregar e salvar funcionários.

    private function carregarFuncionarios(): void {

        $dados = json\_decode(file\_get\_contents(\_\_DIR\_\_ . '/../data/funcionarios.json'), true);

        foreach ($dados as $funcionario) {

            $experiencia = strtolower($funcionario['experiencia']);

            $nome = $funcionario['nome'];

            $disponivel = $funcionario['disponivel'];

            if ($experiencia === 'iniciante') {

                $this->funcionarios[] = new Iniciante($nome, '', 1, $disponivel);

            } elseif ($experiencia === 'experiente') {

                $this->funcionarios[] = new Experiente($nome, '', 3, $disponivel);

            } elseif ($experiencia === 'senior') {

                $this->funcionarios[] = new Senior($nome, '', 5, $disponivel);

            }

        }

    }

    private function salvarFuncionarios(): void {

        $dados = [];

        foreach ($this->funcionarios as $funcionario) {

            $dados[] = [

                'tipo' => ($funcionario instanceof Iniciante) ? 'iniciante' : (($funcionario instanceof Experiente) ? 'experiente' : 'senior'),

                'nome' => $funcionario->getNome(),

                'experiencia' => strtolower($funcionario->getNivelExperiencia()),

                'disponivel' => $funcionario->isDisponivel()

            ];

        }

        error\_log("Salvando dados no JSON: " . print\_r($dados, true));

        $arquivoJson = \_\_DIR\_\_ . '/../data/funcionarios.json';

        file\_put\_contents($arquivoJson, json\_encode($dados, JSON\_PRETTY\_PRINT));

    }

    private function salvarFuncionariosData(): void {

        $arquivoJson = \_\_DIR\_\_ . '/../data/data\_funcionario.json';

        // Carrega os dados atuais

        $dadosAtuais = [];

        if (file\_exists($arquivoJson)) {

            $dadosAtuais = json\_decode(file\_get\_contents($arquivoJson), true);

            if (!is\_array($dadosAtuais)) {

                $dadosAtuais = [];

            }

        }

        // Reorganiza para acesso fácil por nome

        $mapaDados = [];

        foreach ($dadosAtuais as $item) {

            if (isset($item['nome'])) {

                $mapaDados[$item['nome']] = $item;

            }

        }

        // Atualiza ou adiciona os dados

        foreach ($this->funcionarios as $funcionario) {

            $nome = $funcionario->getNome();

            // Mantém dados antigos ou cria novo

            $mapaDados[$nome]['nome'] = $nome;

            $mapaDados[$nome]['disponivel'] = $funcionario->isDisponivel();

        }

        // Salva de volta como array

        $dadosParaSalvar = array\_values($mapaDados);

        error\_log("Salvando dados no JSON: " . print\_r($dadosParaSalvar, true));

        file\_put\_contents($arquivoJson, json\_encode($dadosParaSalvar, JSON\_PRETTY\_PRINT));

    }

    // Remover funcionário

    public function removerFuncionario(string $nome): string {

        $removido = false;

        // Remove da lista de funcionários

        foreach ($this->funcionarios as $key => $funcionario) {

            if ($funcionario->getNome() === $nome) {

                unset($this->funcionarios[$key]);

                $this->funcionarios = array\_values($this->funcionarios);

                $this->salvarFuncionarios();

                $removido = true;

                break;

            }

        }

        if (!$removido) {

            return "Erro: Funcionário '{$nome}' não encontrado.";

        }

        // Arquivos onde também será removido

        $arquivos = [

            \_\_DIR\_\_ . '/../data/usuario.json',

            \_\_DIR\_\_ . '/../data/data\_funcionario.json'

        ];

        foreach ($arquivos as $arquivo) {

            $this->removerDeArquivo($nome, $arquivo);

        }

        return "Funcionário '{$nome}' removido com sucesso de todos os arquivos!";

    }

    private function removerDeArquivo(string $nome, string $caminhoArquivo): void {

        if (!file\_exists($caminhoArquivo)) {

            return; // Arquivo não existe, então não precisa fazer nada

        }

        $dados = json\_decode(file\_get\_contents($caminhoArquivo), true);

        if (!is\_array($dados)) {

            return; // Dados inválidos, evita erro

        }

        // Remove todos os registros que tenham o mesmo nome

        $dados = array\_filter($dados, function($item) use ($nome) {

            return !(isset($item['nome']) && $item['nome'] === $nome);

        });

        // Reorganiza os índices

        $dados = array\_values($dados);

        // Salva novamente

        file\_put\_contents($caminhoArquivo, json\_encode($dados, JSON\_PRETTY\_PRINT));

    }

    // Contratar funcionario por n dias

    public function alugarFuncionario(string $nome, string $tipo, int $dias = 1): string {

        foreach ($this->funcionarios as $key => $funcionario) {

            if ($funcionario->getNome() === $nome && $funcionario->isDisponivel()) {

                if($tipo === 'iniciante') {

                    $valorAluguel = (new Iniciante('', '', 1))->calcularAluguel($dias);

                } elseif ($tipo === 'experiente') {

                    $valorAluguel = (new Experiente('', '', 3))->calcularAluguel($dias);

                } elseif ($tipo === 'senior') {

                    $valorAluguel = (new Senior('', '', 5))->calcularAluguel($dias);

                } else {

                    return "Tipo de funcionário inválido.";

                }

                $mensagem = $funcionario->alugar();

                // Atualiza o funcionário na lista para manter a alteração

                $this->funcionarios[$key] = $funcionario;

                $this->salvarFuncionariosData($tipo);

                return $mensagem . " Valor do aluguel: R$ " . number\_format($valorAluguel, 2, ',', '.');

            }

        }

        return "Funcionário não disponível ou não encontrado.";

    }

    // Devolver Funcionario

    public function devolverFuncionario(string $nome): string {

        foreach ($this->funcionarios as $funcionario) {

            if ($funcionario->getNome() === $nome) {

                $mensagem = $funcionario->devolver();

                $this->salvarFuncionariosData();

                return $mensagem;

            }

        }

        return "Funcionario já disponível ou não encontrado.";

    }

    // Retorna a lista de veículos

    public function listarFuncionarios(): array {

        return $this->funcionarios;

    }

    // Calcular previsão do valor de aluguel

    public function calcularPrevisaoAluguel(int $dias, string $tipo): float {

        $tipo = strtolower($tipo); // Normalizar o tipo para minúsculas

        if ($tipo === 'iniciante') {

            return (new Iniciante('', '', 1))->calcularAluguel($dias);

        } elseif ($tipo === 'experiente') {

            return (new Experiente('', '', 3))->calcularAluguel($dias);

        } elseif ($tipo === 'senior') {

            return (new Senior('', '', 5))->calcularAluguel($dias);

        }

        throw new \InvalidArgumentException("Tipo de funcionário inválido.");

    }

    public function liberarFuncionario(string $nome): string {

        foreach ($this->funcionarios as $funcionario) {

            if ($funcionario->getNome() === $nome && !$funcionario->isDisponivel()) {

                $mensagem = $funcionario->devolver();

                $this->salvarFuncionarios();

                return $mensagem;

            }

        }

        return "Erro: Funcionário '{$nome}' já está disponível ou não foi encontrado.";

    }

}

[

    {

        "filename": "perfilTeste.jpg",

        "path": "uploads\/perfilTeste.jpg",

        "uploaded\_at": "2025-05-26 12:22:05"

    },

    {

        "filename": "perfilTeste.jpg",

        "path": "uploads\/perfilTeste.jpg",

        "uploaded\_at": "2025-05-26 12:26:20"

    },

    {

        "filename": "img-2.png",

        "path": "uploads\/img-2.png",

        "uploaded\_at": "2025-05-26 12:27:15"

    }

]

<?php

function atualizarFuncionarioJson($fileName, $filePath, $emailUsuario) {

    $dataFile = \_\_DIR\_\_ . '/../data/data\_funcionario.json';

    if (!file\_exists($dataFile)) return;

    $funcionarios = json\_decode(file\_get\_contents($dataFile), true);

    foreach ($funcionarios as &$funcionario) {

        if ($funcionario['email'] === $emailUsuario) {

            $funcionario['filename'] = $fileName;

            $funcionario['path'] = $filePath;

            $funcionario['uploaded\_at'] = date('Y-m-d H:i:s');

            break;

        }

    }

    file\_put\_contents($dataFile, json\_encode($funcionarios, JSON\_PRETTY\_PRINT | JSON\_UNESCAPED\_UNICODE));

}

// Verifica se um e-mail foi enviado

$emailUsuario = $\_POST['email'] ?? null;

if (!$emailUsuario) {

    echo "E-mail do usuário não fornecido.";

    exit;

}

// Upload de currículo

if (isset($\_FILES['file']) && $\_FILES['file']['error'] === UPLOAD\_ERR\_OK) {

    $ext = pathinfo($\_FILES['file']['name'], PATHINFO\_EXTENSION);

    $allowedExts = ['pdf', 'png', 'jpg', 'jpeg']; // você pode adaptar se quiser só PDF

    if (!in\_array(strtolower($ext), $allowedExts)) {

        echo "Formato de arquivo inválido. Envie um PDF ou imagem.";

        exit;

    }

    $curriculoNome = uniqid('cv\_', true) . '.' . $ext;

    $uploadDir = \_\_DIR\_\_ . '/../public/uploads2/';

    $webPath = 'uploads2/' . $curriculoNome;

    if (!is\_dir($uploadDir)) {

        mkdir($uploadDir, 0777, true);

    }

    $fileTmp = $\_FILES['file']['tmp\_name'];

    $filePath = $uploadDir . $curriculoNome;

    if (move\_uploaded\_file($fileTmp, $filePath)) {

        // Atualiza no data\_funcionario.json

        $dataFile = \_\_DIR\_\_ . '/../data/data\_funcionario.json';

        if (file\_exists($dataFile)) {

            $funcionarios = json\_decode(file\_get\_contents($dataFile), true);

            foreach ($funcionarios as &$funcionario) {

                if ($funcionario['email'] === $emailUsuario) {

                    $funcionario['filename\_curriculo'] = $curriculoNome;

                    $funcionario['path\_curriculo'] = $webPath;

                    $funcionario['curriculo\_uploaded\_at'] = date('Y-m-d H:i:s');

                    break;

                }

            }

            file\_put\_contents($dataFile, json\_encode($funcionarios, JSON\_PRETTY\_PRINT | JSON\_UNESCAPED\_UNICODE));

        }

        echo "Currículo enviado com sucesso!";

    } else {

        echo "Erro ao mover o arquivo do currículo.";

    }

} elseif (isset($\_FILES['file'])) {

    echo "Erro ao processar o arquivo do currículo.";

}

?>

<script>

    setTimeout(() => {

        window.location.href = "../public/exibicaoaccount.php";

    }, 0);

</script>

<?php

$perfil = $\_POST['perfil']; // Padrão para 'funcionario'

function atualizarFuncionarioJson($fileName, $filePath, $emailUsuario, $perfil) {

    if($perfil === 'funcionario'){

    $dataFile = \_\_DIR\_\_ . '/../data/data\_funcionario.json';

    } else {

        $dataFile = \_\_DIR\_\_ . '/../data/data\_empresa.json';

    }

    if (!file\_exists($dataFile)) return;

    $funcionarios = json\_decode(file\_get\_contents($dataFile), true);

    foreach ($funcionarios as &$funcionario) {

        if ($funcionario['email'] === $emailUsuario) {

            $funcionario['filename'] = $fileName;

            $funcionario['path'] = $filePath;

            $funcionario['uploaded\_at'] = date('Y-m-d H:i:s');

            break;

        }

    }

    file\_put\_contents($dataFile, json\_encode($funcionarios, JSON\_PRETTY\_PRINT | JSON\_UNESCAPED\_UNICODE));

}

// Verifica se um e-mail foi enviado

$emailUsuario = $\_POST['email'] ?? null;

if (!$emailUsuario) {

    echo "E-mail do usuário não fornecido.";

    exit;

}

// Upload de imagem

if (isset($\_FILES['image']) && $\_FILES['image']['error'] === UPLOAD\_ERR\_OK) {

    $ext = pathinfo($\_FILES['image']['name'], PATHINFO\_EXTENSION);

    $imagemNome = uniqid('img\_', true) . '.' . $ext;

    $uploadDir = \_\_DIR\_\_ . '/../public/uploads/';

    $webPath = 'uploads/' . $imagemNome;

    if (!is\_dir($uploadDir)) {

        mkdir($uploadDir, 0777, true);

    }

    $fileTmp = $\_FILES['image']['tmp\_name'];

    $filePath = $uploadDir . $imagemNome;

    if (move\_uploaded\_file($fileTmp, $filePath)) {

        $imageData = [

            'filename' => $imagemNome,

            'path' => $webPath,

            'uploaded\_at' => date('Y-m-d H:i:s')

        ];

        $jsonFile = \_\_DIR\_\_ . '/../data/imagens.json';

        $existingData = [];

        if (file\_exists($jsonFile)) {

            $jsonContent = file\_get\_contents($jsonFile);

            $existingData = json\_decode($jsonContent, true) ?? [];

        }

        $existingData[] = $imageData;

        file\_put\_contents($jsonFile, json\_encode($existingData, JSON\_PRETTY\_PRINT | JSON\_UNESCAPED\_UNICODE));

        // Atualiza também no data\_funcionario.json

        atualizarFuncionarioJson($imagemNome, $webPath, $emailUsuario, $perfil);

        echo "Imagem enviada com sucesso!";

    } else {

        echo "Erro ao mover o arquivo da imagem.";

    }

} elseif (isset($\_FILES['image'])) {

    echo "Erro ao processar o arquivo da imagem.";

}

?>

<script>

    setTimeout(() => {

        window.location.href = "../public/exibicaoaccount.php";

    }, 0);

</script>

<?php

    $senha = "555";

    $hash\_admin = password\_hash($senha, PASSWORD\_DEFAULT);

    $hash\_usuario = password\_hash($senha, PASSWORD\_DEFAULT);

    echo "Hash para admin:" . $hash\_admin . "<br>";

    echo "Hash para usuario:" . $hash\_usuario . "<br>";

    echo $senha;

?>

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-SgOJa3DmI69IUzQ2PVdRZhwQ+dy64/BUtbMJw1MZ8t5HZApcHrRKUc4W0kG879m7" crossorigin="anonymous">

    <title>ServHub</title>

    <link rel="shortcut icon" href="public/Assets/ServHub\_branca.png" type="image/x-icon">

</head>

<style>

    @import url('https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap');

    @import url('https://fonts.googleapis.com/css2?family=Bebas+Neue&display=swap');

    \*{

        font-family: "Poppins", sans-serif;

        font-weight: bold;

        margin: 0;

        padding: 0;

        border: 0;

        list-style: none;

    }

    body{

        width: 100%;

        height: 90.3vh;

    }

    .fonte-index{

        font-family: "Bebas Neue", sans-serif;

        font-size: 10rem;

        color:rgb(255, 255, 255);

        text-align: center;

        margin-top: 50px;

    }

    #background {

        background: url(public/Assets/Fundo-index.png);

        background-size: fill;

        background-repeat: no-repeat;

        width: 100%;

        height: 100%;

    }

    #Nav {

        background-color: #BEBAD9;

        height: 80px;

    }

    .link-custom {

        text-decoration: none;

        text-align: center;

        font-size: 1.2rem;

    }

    ul{

        width: 150px;

        justify-content: space-around;

        align-items: center;

        margin: 0;

    }

    main{

        justify-content: center;

        align-items: center;

    }

    #login {

        padding: 8px 16px;

        border-radius: 10px;

        background-color: #320A59;

        color: white;

        box-shadow: 3px 3px 2px black;

        margin-right: 20px;

    }

    #principal{

        height: 85vh;

    }

    .botão {

        padding: 15px 18px;

        color: white;

        text-decoration: none;

        font-size: 2rem;

        background-color: #320A59;

        border-radius: 10px;

        box-shadow: 3px 3px 3px black;

    }

    .brand-custom{

        margin-left: 15px;

        font-size: 1.6rem;

    }

    #Serv{

        margin-bottom: 50px;

    }

    #logo{

        width: 60%;

    }

    @media (max-width: 426px) {

        .fonte-index{

            font-size: 4rem;

            margin-top: 0;

        }

        .botão {

            font-size: 1.5rem;

        }

        #Serv{

            padding-left: 0px;

        }

        #principal{

            flex-direction: column-reverse!important;

            height: 70%;

        }

        #logo{

            margin-top: 30px;

            margin-bottom: 15px;

            width: 50%;

        }

    }

    @media (min-width: 426px) and (max-width: 768px) {

        .fonte-index{

            font-size: 5rem;

            margin-top: 0;

        }

        .botão {

            font-size: 1.5rem;

        }

        #Serv{

            padding-left: 0px;

        }

        #logo{

            width: 60%;

        }

    }

</style>

<body id="background">

    <nav class="navbar" id="Nav">

        <div class="container-fluid">

            <a href="#" class="brand-custom navbar-brand">ServHub</a>

            <ul class="d-flex">

                <li ><a href="public/Login.php" class="link-custom" id="login">Login</a></li>

            </ul>

        </div>

    </nav>

    <main class="d-flex flex-row-reverse" id="principal">

        <div class="d-flex flex-column col-md-6 justify-content-center align-items-center" id="Serv">

            <h1 class="fonte-index">SERVHUB</h1>

            <a href="public/cadastro.php" class="botão">Comece já</a>

        </div>

        <div class="logo col-md-6 d-flex justify-content-center align-items-center">

            <img src="public/Assets/ServHub\_branca.png" class="img-fluid" id="logo" alt="">

        </div>

    </main>

</body>

</html>