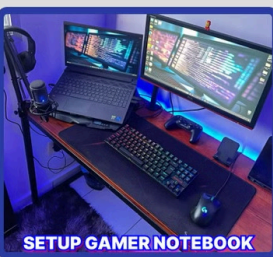
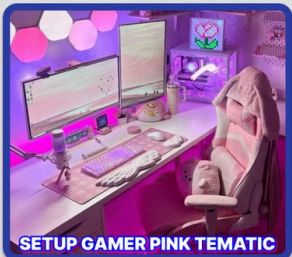
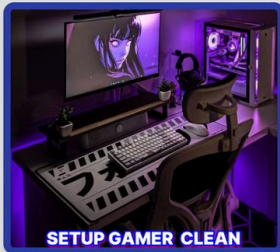




# Introdução à plataforma de venda Setups de Computador

Nossa plataforma de venda de setups de computador foi pensada em quando você, seus amigos ou parentes, que se interessam no mundo game, quiser encontrar tudo o que precisa para montar um setup ideal, sendo ele moderno ou rustico, nós pensamos em uma plataforma onde se pode encontrar todos os tipos de setups completos e você pode adquiri-los com apenas 1 click.



# Apresentação da ferramenta Figma

## Prototipagem interativa

Criar protótipos interativos de alta fidelidade, permitindo que você visualize como sua plataforma funcionará.

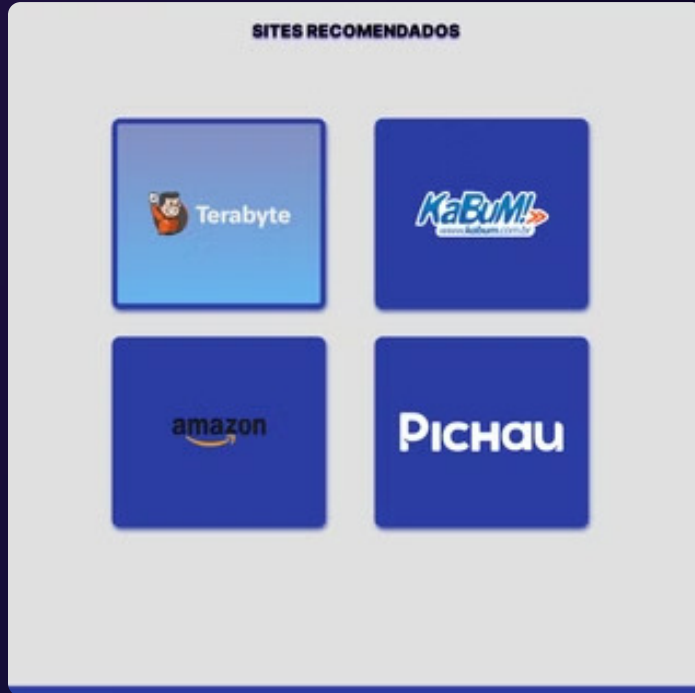
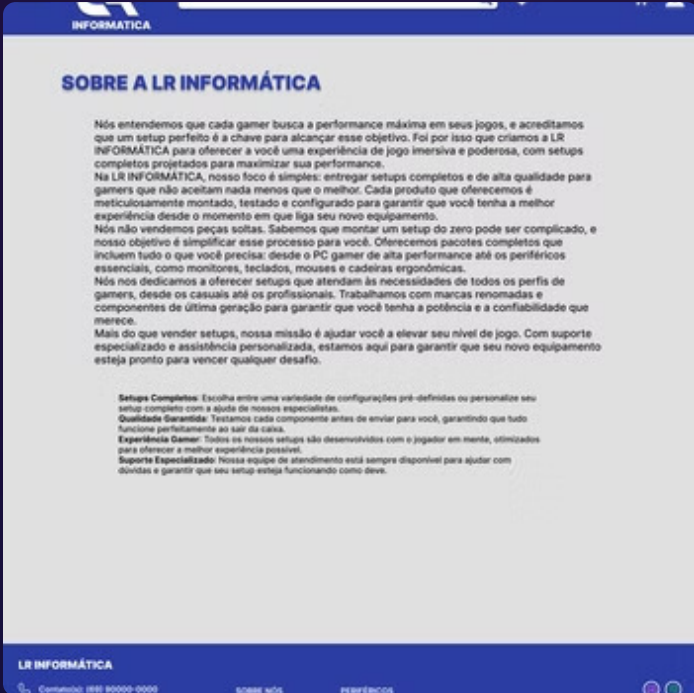
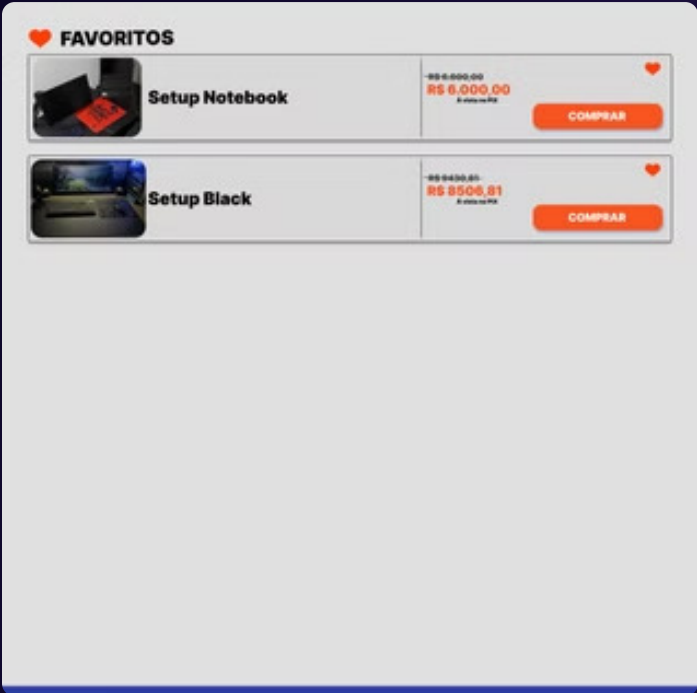
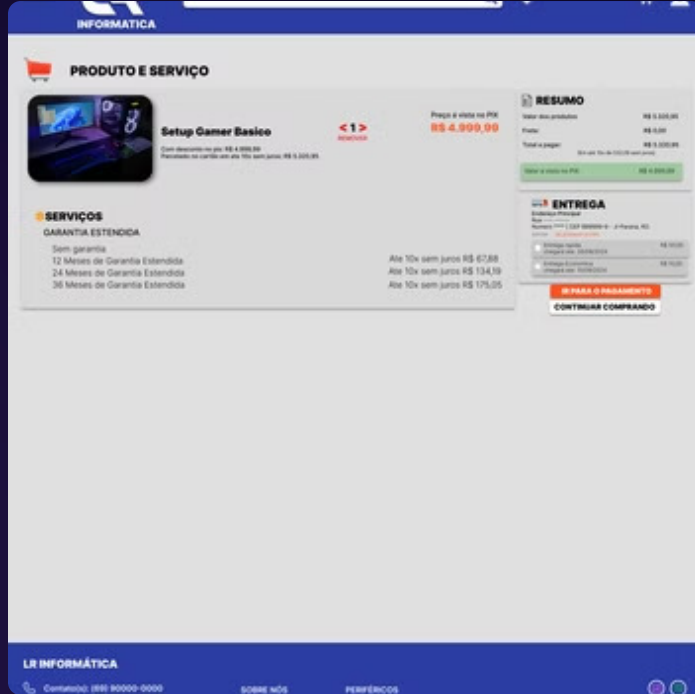
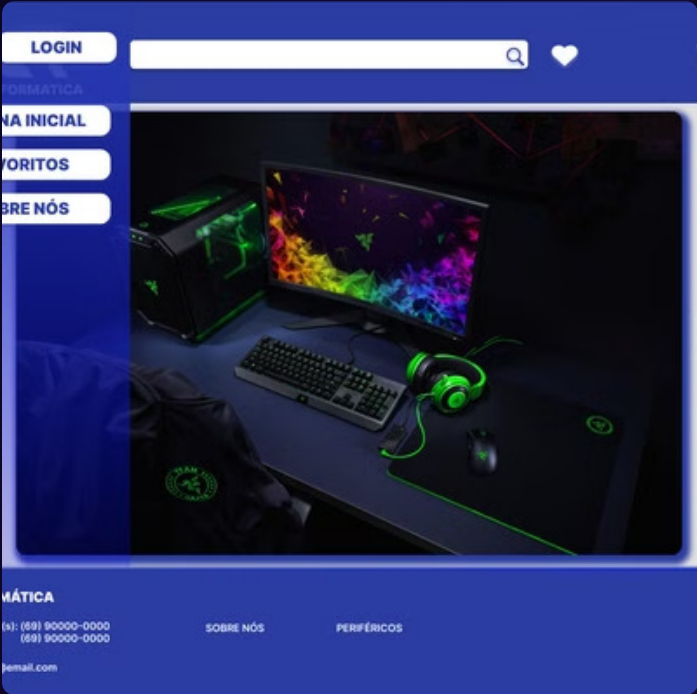
## Design colaborativo

Trabalhamos em tempo real com facilidade nas tomada de decisões.

## Recursos avançados

Utiliza-se componentes, estilos e bibliotecas para manter a consistência e agilizar o processo de design.

# Prototipo





# Tecnologias utilizadas na construção da plataforma

## Front-end

- HTML
- Tailwind CSS
- JavaScript

HTML



JS



CSS



Tailwind CSS



# HTML: estruturação da página

## 1 Estrutura básica

Definindo a estrutura da página com elementos como cabeçalho, conteúdo principal e rodapé.

## 2 Organização do conteúdo

Organizando o conteúdo de forma lógica, utilizando elementos como parágrafos, listas, tabelas e imagens.

## 3 Acessibilidade

Garantindo que a página seja acessível para todos, com o uso de elementos semânticos e atributos de acessibilidade.



# Tailwind: estilização e design responsivo

1

## Estilização rápida

Aplicamos estilos pré-definidos com classes CSS reutilizáveis, reduzindo o tempo de desenvolvimento.

2

## Design responsivo

Criamos layouts que se adaptam, em alguns casos, a diferentes tamanhos de tela, garantindo uma experiência ótima em qualquer dispositivo.

3

## Personalização

Personalizamos os estilos padrão do Tailwind CSS com variáveis, criando um design único.



```
alterarSenha.html login.html item.html JS scripts.js x
ts.js > resetAutoSlide
const openCanvasButton = document.getElementById("openCanvas");
const closeCanvasButton = document.getElementById("closeCanvas");
const offcanvas = document.getElementById("offcanvas");
const overlay = document.getElementById("overlay");

Função para abrir o offcanvas
nine: Edit | Test | Explain | Document | Ask
openCanvasButton.addEventListener("click", () => {
  offcanvas.classList.remove("-translate-x-full");
  overlay.classList.remove("hidden");
});

Função para fechar o offcanvas
nine: Edit | Test | Explain | Document | Ask
closeCanvasButton.addEventListener("click", () => {
  offcanvas.classList.add("-translate-x-full");
  overlay.classList.add("hidden");
});

Fechar o offcanvas ao clicar no overlay
nine: Edit | Test | Explain | Document | Ask
overlay.addEventListener("click", () => {
  offcanvas.classList.add("-translate-x-full");
  overlay.classList.add("hidden");
});

const carousel = document.getElementById("carousel");
const totalSlides = carousel.children.length;
let currentSlide = 0;

Função para mover o carrossel para o slide desejado
nine: Edit | Test | Explain | Document | Ask
function moveToSlide(slideIndex) {
  const offset = -slideIndex * 100; // Move o carrossel pela largura dos sli
  carousel.style.transform = `translateX(${offset}%)`;

  Botão para o slide anterior
  nine: Edit | Test | Explain | Document | Ask
  document.getElementById("prevButton").addEventListener("click", () => {
    currentSlide = (currentSlide - 1 + totalSlides) % totalSlides;
    moveToSlide(currentSlide);
  });
  resetAutoSlide(); // Reinicia o temporizador
};
```

# JavaScript: interatividade e funcionalidades

1

## Interação com o usuário

Criamos elementos interativos, como botões, menus e formulários, para uma melhor experiência do usuário.

2

## Funcionalidades dinâmicas

Implementamos funcionalidades como carrinhos de compras, filtros e animações para tornar a plataforma mais interativa.

3

## Manipulação de dados

Interaja com o servidor para recuperar e enviar dados, como informações de produtos e pedidos.



# CSS: personalização visual

## Auxilio

Utilizamos apenas quando o tailwind não dava o suporte necessario, quando viamos a nessecidade utilizamos o css dentro de `<style></style>` ou no componente `<p style="..."></p>`.

