<!DOCTYPE html>

<html lang=”pt-BR”>

<head>

<meta charset=”UTF-8”>

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

<title>Carrossel Vertical com Menu</title>

<style>

* {

Margin: 0;

Padding: 0;

Box-sizing: border-box;

}

Body {

Display: flex;

Justify-content: center;

Align-items: center;

Height: 100vh;

Position: relative;

Background-color: black;

Color: #95aed7;

}

/\* Botão de menu (Hambúrguer) \*/

.menu-btn {

Position: absolute;

Top: 10px;

Left: 10px;

Cursor: pointer;

Display: flex;

Flex-direction: column;

Justify-content: center;

Align-items: center;

Gap: 5px;

z-index: 300;

}

.menu-btn div {

Width: 25px;

Height: 3px;

Background-color: #95aed7;

Transition: transform 1.3s;

}

/\* Estilo do menu lateral \*/

.menu {

Position: fixed;

Top: 0;

Left: -250px; /\* Escondido inicialmente \*/

Width: 250px;

Height: 100vh;

Background: rgba(34, 34, 34, 0.5); /\* Transparente \*/

Padding: 20px;

Box-shadow: 2px 0 5px rgba(0, 0, 0, 0.5);

Transition: left 0.3s;

z-index: 200;

overflow-y: auto; /\* Permite rolagem dentro do menu \*/

backdrop-filter: blur(10px); /\* Efeito vidro \*/

border-right: 1px solid rgba(255, 255, 255, 0.2); /\* Borda suave \*/

}

/\* Estilizar os links como os labels e destca slide atual \*/

.menu ul li a,

.menu ul li label {

Display: block;

Padding: 10px;

Color: #a1cca9;

Font-size: 18px;

Text-decoration: none;

Cursor: pointer;

Transition: transform 0.2s ease-in-out, background-color 0.3s;

Border-radius: 5px;

}

.menu ul {

List-style: none;

Padding: 0;

}

.menu ul li {

Margin: 20px 0;

}

.menu ul li a:hover,

.menu ul li label:hover {

Border: 2px solid #95aed7; /\* Blue highlight \*/

Border-radius: 5px; /\* Deixa as bordas arredondadas \*/

Transform: scale(1.1);

}

/\* Estilização da header do menu \*/

.menu-header {

Display: flex;

Justify-content: space-between; /\* Home à esquerda e X à direita \*/

Align-items: center; /\* Centraliza verticalmente \*/

Padding: 15px;

Border-bottom: 1px solid grey; /\* Linha separadora \*/

Margin-bottom: 15px; /\* Espaço extra abaixo do cabeçalho \*/

Color: #95aed7;

}

/\* Container para o ícone Home \*/

.menu-icons {

Display: flex;

Align-items: center;

}

/\* Estilização do botão de fechar \*/

.close-menu {

Background: transparent;

Border: none;

Font-size: 20px;

Cursor: pointer;

}

/\* Efeito de scale ao passar o mouse + mudança de cor na moldura \*/

.close-menu:hover {

Transform: scale(1.1); /\* Aumenta 20% o tamanho \*/

}

/\* Classe para abrir o menu \*/

.menu.open {

Left: 0;

}

.menu ul li label.active {

Background-color: #95aed7; /\* Blue highlight \*/

Color: #151819;

Transform: scale(1.1);

}

/\*\*/

/\* Personalizar a barra de rolagem no WebKit (Chrome, Edge, Safari) \*/

.menu::-webkit-scrollbar {

Width: 5px; /\* Largura inicial menor \*/

Background: transparent;

Transition: width 0.3s ease-in-out;

}

.menu::-webkit-scrollbar-thumb {

Background: rgba(161, 204, 169, 0.5); /\* Verde semi-transparente \*/

Border-radius: 10px;

Transition: width 0.3s ease-in-out, background 0.3s ease-in-out;

}

/\* Quando o mouse passa sobre a barra de rolagem (thumb), aumenta a largura \*/

.menu:hover::-webkit-scrollbar {

Width: 8px; /\* Aumenta a largura da barra quando o menu está em uso \*/

}

/\* Quando o mouse passa sobre a própria barra (thumb), aumenta a opacidade \*/

.menu::-webkit-scrollbar-thumb:hover {

Background: rgba(161, 204, 169, 0.9); /\* Mais opaco ao passar o mouse \*/

}

/\* Carrossel \*/

.carousel {

Width: 100vw;

Height: 100vh;

Overflow: hidden;

Position: relative;

Display: flex;

Justify-content: center;

Align-items: center;

}

.slides {

Width: 50vh;

Height: 300vh;

Transition: transform 0.4s ease-in-out;

}

/\* Quando a classe “no-transition” for adicionada, remove a animação \*/

.slides.no-transition {

Transition: none !important;

}

.slide {

Width: 50vh;

Height: 90vh;

Margin: 5vh 0;

Border-radius: 8px;

Text-align: center;

Padding: 5px;

/\* Gradiente sutil do topo mais claro para a base preta \*/

Background: linear-gradient(

To bottom,

#1f1f1f, /\* topo levemente mais claro \*/

#151819 /\* base mais escura \*/

);

}

.slide img {

Width: 100%;

Height: 100%;

Object-fit: cover;

Border-radius: 4px;

}

/\* Classe para o “TEMA”, alinhado à direita \*/

.tema {

Text-align: right;

Font-family: ‘Calibri’, sans-serif; /\* ou ‘Times New Roman’, serif \*/

Font-size: 20pt; /\* Word costuma usar pt, mas pode adaptar \*/

Font-weight: bold; /\* título em negrito \*/

Margin: 0.5em 0; /\* espaço antes e depois \*/

}

/\* Classe para o “MÓDULO”, também alinhado à direita \*/

.modulo {

Text-align: right;

Font-family: ‘Calibri’, sans-serif;

Font-size: 14pt;

Font-weight: bold;

Margin: 0.3em 0;

}

.textoJustificado {

Text-align: justify;

Text-justify: inter-word; /\* ajuda na distribuição das palavras em alguns navegadores \*/

Font-family: ‘Calibri’, sans-serif; /\* ou outra fonte \*/

Font-size: 12pt;

Line-height: 1.5;

/\* recuo de primeira linha \*/

Text-indent: 2em; /\* Ajuste o valor conforme a necessidade \*/

Margin: 1em 0;

}

Input[type=”radio”] {

Position: absolute;

Left: -9999px;

}

/\* Movimentação do carrossel \*/

/\* Padrão de movimentação para 50 cards no carrossel \*/

#card1:checked ~ .slides { transform: translateY(100vh); }

#card2:checked ~ .slides { transform: translateY(5vh); }

#card3:checked ~ .slides { transform: translateY(-90vh); }

#card4:checked ~ .slides { transform: translateY(-185vh); }

#card5:checked ~ .slides { transform: translateY(-280vh); }

#card6:checked ~ .slides { transform: translateY(-375vh); }

#card7:checked ~ .slides { transform: translateY(-470vh); }

#card8:checked ~ .slides { transform: translateY(-565vh); }

#card9:checked ~ .slides { transform: translateY(-660vh); }

#card10:checked ~ .slides { transform: translateY(-755vh); }

#card11:checked ~ .slides { transform: translateY(-850vh); }

#card12:checked ~ .slides { transform: translateY(-945vh); }

#card13:checked ~ .slides { transform: translateY(-1040vh); }

#card14:checked ~ .slides { transform: translateY(-1135vh); }

#card15:checked ~ .slides { transform: translateY(-1230vh); }

#card16:checked ~ .slides { transform: translateY(-1325vh); }

#card17:checked ~ .slides { transform: translateY(-1420vh); }

#card18:checked ~ .slides { transform: translateY(-1515vh); }

#card19:checked ~ .slides { transform: translateY(-1610vh); }

#card20:checked ~ .slides { transform: translateY(-1705vh); }

#card21:checked ~ .slides { transform: translateY(-1800vh); }

#card22:checked ~ .slides { transform: translateY(-1895vh); }

#card23:checked ~ .slides { transform: translateY(-1990vh); }

#card24:checked ~ .slides { transform: translateY(-2085vh); }

#card25:checked ~ .slides { transform: translateY(-2180vh); }

#card26:checked ~ .slides { transform: translateY(-2275vh); }

#card27:checked ~ .slides { transform: translateY(-2370vh); }

#card28:checked ~ .slides { transform: translateY(-2465vh); }

#card29:checked ~ .slides { transform: translateY(-2560vh); }

#card30:checked ~ .slides { transform: translateY(-2655vh); }

#card31:checked ~ .slides { transform: translateY(-2750vh); }

#card32:checked ~ .slides { transform: translateY(-2845vh); }

#card33:checked ~ .slides { transform: translateY(-2940vh); }

#card34:checked ~ .slides { transform: translateY(-3035vh); }

#card35:checked ~ .slides { transform: translateY(-3130vh); }

#card36:checked ~ .slides { transform: translateY(-3225vh); }

#card37:checked ~ .slides { transform: translateY(-3320vh); }

#card38:checked ~ .slides { transform: translateY(-3415vh); }

#card39:checked ~ .slides { transform: translateY(-3510vh); }

#card40:checked ~ .slides { transform: translateY(-3605vh); }

#card41:checked ~ .slides { transform: translateY(-3700vh); }

#card42:checked ~ .slides { transform: translateY(-3795vh); }

#card43:checked ~ .slides { transform: translateY(-3890vh); }

#card44:checked ~ .slides { transform: translateY(-3985vh); }

#card45:checked ~ .slides { transform: translateY(-4080vh); }

#card46:checked ~ .slides { transform: translateY(-4175vh); }

#card47:checked ~ .slides { transform: translateY(-4270vh); }

#card48:checked ~ .slides { transform: translateY(-4365vh); }

#card49:checked ~ .slides { transform: translateY(-4460vh); }

#card50:checked ~ .slides { transform: translateY(-4555vh); }

</style>

</head>

<body>

<!—Menu Lateral 🡪

<nav class=”menu”>

<div class=”menu-header”>

<!—Ícone Home à esquerda 🡪

<div class=”menu-icons”>

<a href=<https://www.facebook.com/?locale=pt_BR>>

<svg aria-label=”Página inicial” fill=”#95aed7” height=”24” role=”img” viewBox=”0 0 24 24” width=”24”>

<path d=”M22 23h-6.001ª1 1 0 0 1-1-1v-5.455ª2.997 2.997 0 1 0-5.993 0V22a1 1 0 0 1-1 1H2a1 1 0 0 1-1-1V11.543ª1.002 1.002 0 0 1 .31-.724l10-9.543ª1.001 1.001 0 0 1 1.38 0l10 9.543ª1.002 1.002 0 0 1 .31.724V22a1 1 0 0 1-1 1Z”></path>

</svg>

</a>

</div>

<!—Botão de fechar à direita 🡪

<button class=”close-menu” onclick=”toggleMenu()”>✖</button>

</div>

<ul>

<li class=”semester”> 1º SEMESTRE – Abr/24</li>

<li><label for=”card1”>☐ Arquitetura De Computadores</label></li>

<li><label for=”card2”>☐ Paradigmas De Linguagens De Programação Em Python</label></li>

<li><label for=”card3”>☐ Introdução À Segurança Da Informação</label></li>

<li><label for=”card4”>☐ Língua Portuguesa</label></li>

<li><label for=”card5”>☐ Pensamento Computacional</label></li>

<li><label for=”card6”>☐ Processos De Desenvolvimento De Software</label></li>

<li class=”semester”> 2º SEMESTRE – Out/24</li>

<li><label for=”card7”>☐ Fundamentos De Redes De Computadores</label></li>

<li><label for=”card8”>☐ Sistemas De Informação E Sociedade</label></li>

<li><label for=”card9”>☐ Desenv. Web Em Html5, Css, Javascript E Php</label></li>

<li><label for=”card10”>☐ Introdução À Programação De Computadores</label></li>

<li><label for=”card11”>☐ Requisitos De Sistemas</label></li>

<li class=”semester”> 3º SEMESTRE – Abr/25</li>

<li><label for=”card12”>☐ Arquitetura De Sistemas</label></li>

<li><label for=”card13”>☐ Engenharia De Usabilidade</label></li>

<li><label for=”card14”>☐ Estrutura De Dados</label></li>

<li><label for=”card15”>☐ Programação De Software Básico Em C</label></li>

<li><label for=”card16”>☐ Programação Orientada A Objetos Em Java</label></li>

<li><label for=”card17”>☐ Matemática E Lógica</label></li>

<li class=”semester”> 4º SEMESTRE – Out/25</li>

<li><label for=”card18”>☐ Comportamento Organizacional</label></li>

<li><label for=”card19”>☐ Gerência De Configuração</label></li>

<li><label for=”card20”>☐ Indicadores De Desempenho De Ti</label></li>

<li><label for=”card21”>☐ Modelagem De Sistemas Em Uml</label></li>

<li><label for=”card22”>☐ Pensamento Computacional</label></li>

<li><label for=”card23”>☐ Computação Em Nuvem</label></li>

<li class=”semester”> 5º SEMESTRE – Abr/26</li>

<li><label for=”card24”>☐ Desenvolvimento Rápido De Aplicações Em Python</label></li>

<li><label for=”card25”>☐ Desenvolvimento Web Full Mvc</label></li>

<li><label for=”card26”>☐ Engenharia De Produto</label></li>

<li><label for=”card27”>☐ Padrões De Projetos De Software Com Java</label></li>

<li><label for=”card28”>☐ Sistemas Operacionais</label></li>

<li><label for=”card29”>☐ Tópicos De Big Data Em Python</label></li>

<li class=”semester”> 6º SEMESTRE – Out/26</li>

<li><label for=”card30”>☐ Aplic. De Cloud, Iot E Indústria 4.0 Em Python</label></li>

<li><label for=”card31”>☐ Métodos Ágeis Com Scrum</label></li>

<li><label for=”card32”>☐ Modelagem De Processos</label></li>

<li><label for=”card33”>☐ Programação Em Python Para Análise De Dados</label></li>

<li><label for=”card34”>☐ Programação Para Dispositivos Móveis Em Android</label></li>

<li><label for=”card35”>☐ Banco De Dados</label></li>

<li class=”semester”> 7º SEMESTRE – Abr/27</li>

<li><label for=”card36”>☐ Algoritmos E Complexidade</label></li>

<li><label for=”card37”>☐ Banco De Dados Nosql</label></li>

<li><label for=”card38”>☐ Desenvolvimento De Back-end</label></li>

<li><label for=”card39”>☐ Direito Cibernético</label></li>

<li><label for=”card40”>☐ Métodos Quantitativos</label></li>

<li><label for=”card41”>☐ Qualidade E Testes De Software</label></li>

<li class=”semester”> 8º SEMESTRE – Out/27</li>

<li><label for=”card42”>☐ Desenvolvimento De Front-end</label></li>

<li><label for=”card43”>☐ Desenvolvimento De Software Seguro</label></li>

<li><label for=”card44”>☐ Desenvolvimento E Operações (Devops)</label></li>

<li><label for=”card45”>☐ Direitos Humanos</label></li>

<li><label for=”card46”>☐ Educação Ambiental</label></li>

<li><label for=”card47”>☐ Gerenciamento De Projetos</label></li>

<li><label for=”card48”>☐ Linhas De Produtos De Software</label></li>

<li><label for=”card49”>☐ História Dos Povos Indígenas E Afrodescendentes</label></li>

<li><label for=”card50”>☐ Tópicos Em Libras: Surdez E Inclusão</label></li>

</ul>

</nav>

<!—Botão de Menu (Hambúrguer) 🡪

<div class=”menu-btn” onclick=”toggleMenu()”>

<div></div>

<div></div>

<div></div>

</div>

<div class=”carousel”>

<!—50 inputs para o carrossel 🡪

<input type=”radio” name=”carousel” id=”card1” checked tabindex=”0”>

<input type=”radio” name=”carousel” id=”card2” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card3” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card4” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card5” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card6” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card7” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card8” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card9” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card10” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card11” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card12” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card13” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card14” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card15” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card16” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card17” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card18” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card19” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card20” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card21” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card22” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card23” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card24” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card25” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card26” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card27” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card28” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card29” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card30” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card31” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card32” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card33” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card34” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card35” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card36” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card37” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card38” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card39” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card40” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card41” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card42” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card43” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card44” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card45” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card46” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card47” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card48” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card49” tabindex=”0”>

<input type=”radio” name=”carousel” id=”card50” tabindex=”0”>

<div class=”slides”>

<!—Tema 1 – Apresentação 🡪

<div class=”slide”>

<div class=”tema”>

<p>Tema 1 – Apresentação</p>

</div>

<div class=”textoJustificado”>

<p>O que realmente o sistema deve fazer; quais são as entradas disponíveis e as saídas esperadas para o sistema, ou seja, que os objetivos estão alinhados com a necessidade?</p>

</div>

</div>

<!—Tema 2 – Introdução a Requisitos de Sistemas / Módulo 1 🡪

<div class=”slide”>

<div class=”tema”>

<p>Tema 2 – Introdução a Requisitos de Sistemas</p>

</div>

<div class=”modulo”>

Módulo 1 – Introdução

</div>

<div class=”textoJustificado”>

<p>Imagine que uma padaria quer um programa que registre automaticamente a entrada e saída de produtos, como pão, leite e ovos. Assim, ao vender ou receber mercadorias, o sistema atualiza tudo sozinho.</p>

</div>

</div>

<!—Tema 2 / Módulo 2 🡪

<div class=”slide”>

<div class=”modulo”>

Módulo 2 – Análise de Dados

</div>

<div class=”textoJustificado”>

<p>Antes de criar o programa, é importante entender as necessidades da padaria. Por exemplo, se não soubermos que o dono quer ser avisado quando o estoque de farinha estiver baixo, o sistema pode acabar sendo inútil.</p>

</div>

</div>

<!—Tema 2 / Módulo 3 🡪

<div class=”slide”>

<div class=”modulo”>

Módulo 3 – Levantamento de Requisitos

</div>

<div class=”textoJustificado”>

<p>O programador conversa com o dono da padaria para entender o que ele precisa. Assim, descobre que o programa deve emitir alertas quando os produtos estiverem acabando e gerar relatórios de vendas diárias.</p>

</div>

</div>

<!—Tema 2 / Módulo 4 🡪

<div class=”slide”>

<div class=”modulo”>

Módulo 4 – Qualidade do Software

</div>

<div class=”textoJustificado”>

<p>Durante a criação do programa, o desenvolvedor testa várias vezes para garantir que ele funciona bem. Por exemplo, verifica se o alerta de estoque baixo aparece na hora certa.</p>

</div>

</div>

<!—Tema 2 / Módulo 5 🡪

<div class=”slide”>

<div class=”modulo”>

Módulo 5 – Controle e Garantia de Qualidade

</div>

<div class=”textoJustificado”>

<p>Antes de entregar o programa, o dono da padaria e o programador fazem testes juntos para confirmar que tudo funciona como esperado. Se algo estiver errado, dá tempo de corrigir.</p>

</div>

</div>

<!—Tema 2 / Módulo 6 🡪

<div class=”slide”>

<div class=”modulo”>

Módulo 6 – Gerenciamento da Qualidade

</div>

<div class=”textoJustificado”>

<p>O programador cria um plano para garantir que o programa será feito de forma organizada, desde o início até a entrega, sempre checando se está dentro do combinado.</p>

</div>

</div>

<!—Tema 3 – Níveis de Requisitos 🡪

<div class=”slide”>

<div class=”tema”>

<p>Tema 3 – Níveis de Requisitos: Requisitos de Usuários e Requisitos de Sistemas</p>

</div>

<div class=”modulo”>

Módulo 1 – Introdução

</div>

<div class=”textoJustificado”>

<p>Notadamente, o sistema a ser desenvolvido substituirá ou aperfeiçoará algum outro existente ou automatizará um processo que atualmente não está sendo realizado por computador.</p>

</div>

</div>

<!—Tema 3 / Módulo 2 🡪

<div class=”slide”>

<div class=”modulo”>

Módulo 2 – Diferença entre Requisitos de Usuário e de Sistema

</div>

<div class=”textoJustificado”>

<p>Os requisitos de usuário descrevem o que o usuário deseja que o sistema faça, enquanto os requisitos de sistema detalham como o sistema atenderá a essas necessidades.</p>

</div>

</div>

<!—Tema 3 / Módulo 3 🡪

<div class=”slide”>

<div class=”modulo”>

Módulo 3 – Requisitos no Nível do Usuário

</div>

<div class=”textoJustificado”>

<p>O que o usuário quer? Exemplo: visualizar o saldo da conta.</p>

</div>

</div>

<!—Tema 3 / Módulo 4 🡪

<div class=”slide”>

<div class=”modulo”>

Módulo 4 – Requisitos de Sistema

</div>

<div class=”textoJustificado”>

<p>Como o sistema realizará isso? Exemplo: acessará a base de dados via HTTPS.</p>

</div>

</div>

<!—Tema 4 – Tipos de Requisitos 🡪

<div class=”slide”>

<div class=”tema”>

<p>Tema 4 – Tipos de Requisitos: Funcionais, Não Funcionais e Regras de Negócios</p>

</div>

<div class=”modulo”>

Módulo 1 – Requisitos Funcionais

</div>

<div class=”textoJustificado”>

<p>São aqueles que estabelecem e descrevem as funcionalidades do sistema.</p>

<p><strong>• Requisito:</strong> O sistema deve possuir um módulo de saque que será disponibilizado em uma tela separada, essa será exibida conforme o interesse do cliente.</p>

</div>

</div>

<!—Tema 4 / Módulo 2 🡪

<div class=”slide”>

<div class=”modulo”>

Módulo 2 – Requisitos Não Funcionais

</div>

<div class=”textoJustificado”>

<p>Os requisitos não funcionais definem atributos do sistema, como desempenho, segurança e usabilidade.</p>

<p><strong>• Exemplo:</strong> Um aplicativo de banco deve ser confiável e exibir a tela principal em no máximo 3 segundos.</p>

</div>

</div>

<!—Tema 4 / Módulo 3 🡪

<div class=”slide”>

<div class=”modulo”>

Módulo 3 – Regras de Negócio

</div>

<div class=”textoJustificado”>

<p>As regras de negócio são diretrizes e restrições que determinam como o sistema deve operar em um contexto específico.</p>

<p><strong>• Exemplo:</strong> Um cliente só pode sacar um valor se houver saldo suficiente na conta.</p>

</div>

</div>

<!—Tema 4 / Módulo 4 (Métricas para requisitos não funcionais – Parte 1) 🡪

<div class=”slide”>

<div class=”modulo”>

Módulo 4 – Métricas para Especificar Requisitos Não Funcionais

</div>

<div class=”textoJustificado”>

<p>Para garantir que os requisitos não funcionais sejam atendidos, métricas específicas são utilizadas.</p><br>

<p><strong>• Velocidade:</strong> tempo de resposta máximo permitido.</p>

<p><strong>• Tamanho:</strong> uso de memória e espaço em disco.</p>

<p><strong>• Facilidade de uso:</strong> tempo médio para um novo usuário aprender o sistema.</p>

<br><p>Outras métricas importantes incluem:</p><br>

<p><strong>• Confiabilidade:</strong> tempo médio entre falhas do sistema.</p>

<p><strong>• Robustez:</strong> capacidade do sistema de lidar com erros inesperados.</p>

<p><strong>• Portabilidade:</strong> número de plataformas suportadas.</p>

</div>

</div>

<!—Tema 5 – Técnicas de Levantamento de Requisitos 🡪

<div class=”slide”>

<div class=”tema”>

<p>Tema 5 – Técnicas de Levantamento de Requisitos</p>

</div>

<div class=”modulo”>

Módulo 1 – Introdução

</div>

<div class=”textoJustificado”>

<p>Para identificar as necessidades dos usuários, utilizamos diferentes técnicas de levantamento de requisitos.</p>

<p><br>• Objetivo:</br> Coletar informações sobre o sistema desejado e evitar ambiguidades na definição dos requisitos.</p>

</div>

</div>

<!—Tema 5 / Módulo 2 🡪

<div class=”slide”>

<div class=”modulo”>

Módulo 2 – Entrevistas

</div>

<div class=”textoJustificado”>

<p>A entrevista é uma das técnicas mais utilizadas para o levantamento de requisitos, permitindo um contato direto com os stakeholders.</p>

<p><br>• Vantagem:</br> Permite obter informações detalhadas e esclarecer dúvidas em tempo real.</p>

<p><br>• Desvantagem:</br> Pode ser demorado e exigir grande disponibilidade dos envolvidos.</p>

</div>

</div>

<!—Tema 5 / Módulo 3 🡪

<div class=”slide”>

<div class=”modulo”>

Módulo 3 – Questionários

</div>

<div class=”textoJustificado”>

<p>Os questionários são formulários estruturados usados para coletar informações de um grande número de usuários.</p>

<p><br>• Quando usar:</br> Quando é necessário coletar opiniões de muitos usuários sem demandar muito tempo.</p>

<p><br>• Limitação:</br> As respostas podem ser superficiais e não permitir aprofundamento nos requisitos.</p>

</div>

</div>

<!—Tema 5 / Módulo 4 🡪

<div class=”slide”>

<div class=”modulo”>

Módulo 4 – Observação Direta

</div>

<div class=”textoJustificado”>

<p>Essa técnica consiste em observar como os usuários realizam suas atividades para entender melhor suas necessidades.</p>

<p><br>• Benefício:</br> Ajuda a identificar requisitos que os usuários podem não ter mencionado.</p>

<p><br>• Risco:</br> Algumas atividades podem variar ao longo do tempo, dificultando a definição precisa dos requisitos.</p>

</div>

</div>

<!—Tema 6 – Modelagem de Requisitos 🡪

<div class=”slide”>

<div class=”tema”>

<p>Tema 6 – Modelagem de Requisitos</p>

</div>

<div class=”modulo”>

Módulo 1 – Introdução à Modelagem de Requisitos

</div>

<div class=”textoJustificado”>

<p>A modelagem de requisitos ajuda a representar graficamente as necessidades do sistema, tornando-as mais compreensíveis.</p>

<p><br>• Benefício:</br> Facilita a comunicação entre desenvolvedores e usuários.</p>

</div>

</div>

<!—Tema 6 / Módulo 2 🡪

<div class=”slide”>

<div class=”modulo”>

Módulo 2 – Diagramas de Casos de Uso

</div>

<div class=”textoJustificado”>

<p>Os diagramas de casos de uso descrevem as interações entre os usuários e o sistema.</p>

<p><br>• Elementos principais:</br> Atores, casos de uso e relacionamentos.</p>

</div>

</div>

<!—Tema 6 / Módulo 3 🡪

<div class=”slide”>

<div class=”modulo”>

Módulo 3 – Modelagem com UML

</div>

<div class=”textoJustificado”>

<p>A UML (Unified Modeling Language) é amplamente utilizada na modelagem de sistemas, incluindo requisitos.</p>

<p><br>• Vantagem:</br> Permite criar diferentes tipos de diagramas para representar requisitos de forma visual.</p>

</div>

</div>

<!—Tema 7 – Validação e Verificação de Requisitos 🡪

<div class=”slide”>

<div class=”tema”>

<p>Tema 7 – Validação e Verificação de Requisitos</p>

</div>

<div class=”modulo”>

Módulo 1 – Conceito de Validação e Verificação

</div>

<div class=”textoJustificado”>

<p>Validação e verificação são processos essenciais para garantir que os requisitos estão corretos e atendem às necessidades do usuário.</p>

<p><br>• Validação:</br> Verifica se o sistema atende às expectativas dos usuários.</p>

<p><br>• Verificação:</br> Garante que os requisitos estão bem definidos e sem ambiguidades.</p>

</div>

</div>

<!—Tema 7 / Módulo 2 🡪

<div class=”slide”>

<div class=”modulo”>

Módulo 2 – Técnicas de Verificação

</div>

<div class=”textoJustificado”>

<p>As técnicas de verificação incluem inspeção de requisitos, revisões e testes formais.</p>

<p><br>• Inspeção:</br> Revisão detalhada dos requisitos por especialistas.</p>

<p><br>• Testes formais:</br> Garantem que o sistema está de acordo com os requisitos especificados.</p>

</div>

</div>

<!—Tema 7 / Módulo 3 🡪

<div class=”slide”>

<div class=”modulo”>

Módulo 3 – Técnicas de Validação

</div>

<div class=”textoJustificado”>

<p>As técnicas de validação buscam assegurar que os requisitos realmente representam as necessidades do usuário.</p>

<p><br>• Prototipagem:</br> Permite testar requisitos antes da implementação final.</p>

<p><br>• Revisões com usuários:</br> Validam se os requisitos refletem corretamente as necessidades.</p>

</div>

</div>

<!—Tema 8 – Gerenciamento de Requisitos 🡪

<div class=”slide”>

<div class=”tema”>

<p>Tema 8 – Gerenciamento de Requisitos</p>

</div>

<div class=”modulo”>

Módulo 1 – Introdução ao Gerenciamento de Requisitos

</div>

<div class=”textoJustificado”>

<p>O gerenciamento de requisitos envolve o controle e a rastreabilidade dos requisitos ao longo do ciclo de vida do projeto.</p>

<p><br>• Objetivo:</br> Garantir que os requisitos sejam atendidos e que mudanças sejam bem gerenciadas.</p>

</div>

</div>

<!—Tema 8 / Módulo 2 🡪

<div class=”slide”>

<div class=”modulo”>

Módulo 2 – Rastreabilidade de Requisitos

</div>

<div class=”textoJustificado”>

<p>A rastreabilidade permite acompanhar a origem e o impacto de cada requisito dentro do projeto.</p>

<p><br>• Benefício:</br> Ajuda a identificar dependências e reduzir riscos em mudanças.</p>

</div>

</div>

<!—Tema 8 / Módulo 3 🡪

<div class=”slide”>

<div class=”modulo”>

Módulo 3 – Mudanças nos Requisitos

</div>

<div class=”textoJustificado”>

<p>Durante o desenvolvimento, os requisitos podem precisar de ajustes devido a novas necessidades ou restrições.</p>

<p><br>• Estratégia:</br> Definir um processo de controle de mudanças para minimizar impactos negativos.</p>

</div>

</div>

<!—Tema 9 – Engenharia de Requisitos e o Processo de Software 🡪

<div class=”slide”>

<div class=”tema”>

<p>Tema 9 – Engenharia de Requisitos e o Processo de Software</p>

</div>

<div class=”modulo”>

Módulo 1 – Relação entre Engenharia de Requisitos e Desenvolvimento de Software

</div>

<div class=”textoJustificado”>

<p>A engenharia de requisitos é uma etapa essencial no desenvolvimento de software, garantindo que o sistema atenda às necessidades dos usuários.</p>

<p><br>• Impacto:</br> Requisitos bem definidos resultam em software mais eficiente e com menos retrabalho.</p>

</div>

</div>

<!—Tema 9 / Módulo 2 🡪

<div class=”slide”>

<div class=”modulo”>

Módulo 2 – Ciclo de Vida dos Requisitos

</div>

<div class=”textoJustificado”>

<p>Os requisitos passam por diversas fases desde a sua elicitação até a sua implementação e manutenção.</p>

<p><br>• Etapas principais:</br> Levantamento, análise, especificação, validação e gerenciamento.</p>

</div>

</div>

<!—Tema 9 / Módulo 3 🡪

<div class=”slide”>

<div class=”modulo”>

Módulo 3 – Engenharia de Requisitos em Diferentes Metodologias

</div>

<div class=”textoJustificado”>

<p>As abordagens para engenharia de requisitos variam conforme a metodologia de desenvolvimento adotada.</p>

<p><br>• Métodos ágeis:</br> Requisitos são definidos de forma incremental, com entregas contínuas.</p>

<p><br>• Métodos tradicionais:</br> Os requisitos são definidos completamente antes do desenvolvimento iniciar.</p>

</div>

</div>

<!—Tema 10 – Ferramentas para Engenharia de Requisitos 🡪

<div class=”slide”>

<div class=”tema”>

<p>Tema 10 – Ferramentas para Engenharia de Requisitos</p>

</div>

<div class=”modulo”>

Módulo 1 – Importância das Ferramentas de Apoio

</div>

<div class=”textoJustificado”>

<p>Ferramentas especializadas auxiliam na gestão e documentação dos requisitos de um sistema.</p>

<p><br>• Benefício:</br> Melhor organização e rastreabilidade dos requisitos ao longo do projeto.</p>

</div>

</div>

<!—Tema 10 / Módulo 2 🡪

<div class=”slide”>

<div class=”modulo”>

Módulo 2 – Exemplos de Ferramentas para Engenharia de Requisitos

</div>

<div class=”textoJustificado”>

<p>Existem diversas ferramentas que facilitam o trabalho com requisitos.</p>

<p><br>• IBM DOORS:</br> Amplamente usada para rastreamento de requisitos.</p>

<p><br>• Jira:</br> Popular em metodologias ágeis para acompanhar requisitos e tarefas.</p>

<p><br>• RequisitePro:</br> Permite gerenciamento detalhado de requisitos em grandes projetos.</p>

</div>

</div>

<!—Tema 10 / Módulo 3 🡪

<div class=”slide”>

<div class=”modulo”>

Módulo 3 – Critérios para Escolha de uma Ferramenta

</div>

<div class=”textoJustificado”>

<p>A escolha da ferramenta depende das necessidades do projeto e da equipe.</p>

<p><br>• Critérios:</br> Facilidade de uso, integração com outras ferramentas e custo-benefício.</p>

</div>

</div>

</div>

<script>

Let startY;

Let isScrolling = false;

// Function to update active menu item

Function updateActiveMenu() {

Let slides = document.querySelectorAll(‘input[name=”carousel”]’);

Let labels = document.querySelectorAll(“.menu ul li label”);

Slides.forEach(slide => {

If (slide.checked) {

Labels.forEach(label => label.classList.remove(“active”)); // Remove active from all labels

Let activeLabel = document.querySelector(`.menu ul li label[for=”${slide.id}”]`);

If (activeLabel) {

activeLabel.classList.add(“active”);

}

}

});

}

// Toggle menu visibility

Function toggleMenu() {

Let menu = document.querySelector(“.menu”);

Let menuBtn = document.querySelector(“.menu-btn”);

Menu.classList.toggle(“open”);

If (menu.classList.contains(“open”)) {

menuBtn.style.display = “none”; // Esconder botão do menu quando o menu estiver aberto

disableBodyScroll();

} else {

menuBtn.style.display = “flex”; // Mostrar botão do menu quando o menu estiver fechado

enableBodyScroll();

}

}

// Prevent background scrolling when the menu is open

Function disableBodyScroll() {

Document.body.style.overflow = “hidden”;

}

// Allow background scrolling when the menu is closed

Function enableBodyScroll() {

Document.body.style.overflow = “auto”;

}

// Prevent menu from propagating scroll events

Function stopScrollPropagation(event) {

Event.stopPropagation();

}

// Função para navegar para um slide sem animação

Function scrollToCard(index) {

Let slides = document.querySelector(“.slides”);

Let slideInput = document.getElementById(“card” + (index + 1));

If (slideInput) {

Slides.classList.add(“no-transition”); // Remove a animação

// Fecha o menu ANTES de mudar o slide

toggleMenu();

slideInput.checked = true; // Seleciona o slide correspondente

setTimeout(() => {

slides.classList.remove(“no-transition”); // Restaura a animação após 50ms

updateActiveMenu(); // Atualiza o destaque do menu

}, 50);

}

}

// Aplica a função nos itens do menu

Document.querySelectorAll(“.menu ul li label”).forEach((label, index) => {

Label.addEventListener(“click”, () => scrollToCard(index));

});

// Attach event listeners to each slide input (radio buttons)

Document.querySelectorAll(‘input[name=”carousel”]’).forEach(input => {

Input.addEventListener(“change”, updateActiveMenu);

});

// Handle scrolling (mouse wheel) for slide change

Document.addEventListener(“wheel”, function(event) {

Let menu = document.querySelector(“.menu”);

If (menu.classList.contains(“open”)) return; // Don’t scroll carousel when menu is open

If (isScrolling) return;

isScrolling = true;

event.preventDefault();

let slides = document.querySelectorAll(‘input[name=”carousel”]’);

let checkedIndex = [...slides].findIndex(slide => slide.checked);

if (event.deltaY > 0 && checkedIndex < slides.length – 1) {

slides[checkedIndex + 1].checked = true;

} else if (event.deltaY < 0 && checkedIndex > 0) {

Slides[checkedIndex – 1].checked = true;

}

updateActiveMenu();

setTimeout(() => isScrolling = false, 400);

}, { passive: false });

// Handle touch swipe for slide change

Document.addEventListener(“touchstart”, function(event) {

Let menu = document.querySelector(“.menu”);

If (menu.classList.contains(“open”)) return; // Ignore swipe when menu is open

startY = event.touches[0].clientY;

}, { passive: true });

Document.addEventListener(“touchend”, function(event) {

Let menu = document.querySelector(“.menu”);

If (menu.classList.contains(“open”)) return; // Ignore swipe when menu is open

Let endY = event.changedTouches[0].clientY;

Let slides = document.querySelectorAll(‘input[name=”carousel”]’);

Let checkedIndex = [...slides].findIndex(slide => slide.checked);

Let swipeThreshold = 50;

If (startY – endY > swipeThreshold && checkedIndex < slides.length – 1) {

Slides[checkedIndex + 1].checked = true;

} else if (endY – startY > swipeThreshold && checkedIndex > 0) {

Slides[checkedIndex – 1].checked = true;

}

updateActiveMenu();

}, { passive: true });

// Initialize menu highlight on page load

Document.addEventListener(“DOMContentLoaded”, updateActiveMenu);

// Força o foco na página para capturar eventos de tecla

Document.addEventListener(“DOMContentLoaded”, function() {

Document.body.setAttribute(“tabindex”, “0”);

Document.body.focus();

});

// Permitir navegação pelos slides com as setas do teclado

Document.addEventListener(“keydown”, function(event) {

Let slides = document.querySelectorAll(‘input[name=”carousel”]’);

Let checkedIndex = [...slides].findIndex(slide => slide.checked);

If (event.key === “ArrowDown” && checkedIndex < slides.length – 1) {

Slides[checkedIndex + 1].checked = true;

} else if (event.key === “ArrowUp” && checkedIndex > 0) {

Slides[checkedIndex – 1].checked = true;

}

updateActiveMenu(); // Atualizar o menu quando trocar de slide

});

Document.addEventListener(“DOMContentLoaded”, function () {

Const menuBtn = document.querySelector(“.menu-btn”);

Const menu = document.querySelector(“.menu”);

Const cards = document.querySelectorAll(“.slide”); // Seleciona os cards

Let menuVisible = false;

Let menuBtnVisible = false;

// Alterna a visibilidade do menu

Function toggleMenu() {

If (menuVisible) {

Menu.classList.remove(“open”);

} else {

Menu.classList.add(“open”);

}

menuVisible = !menuVisible;

}

// Exibir/esconder o botão de menu

Function toggleMenuButton(show) {

If (show) {

menuBtn.style.display = “flex”;

menuBtnVisible = true;

} else {

menuBtn.style.display = “none”;

menuBtnVisible = false;

}

}

// Evento de clique no botão de menu

menuBtn.addEventListener(“click”, function (event) {

event.stopPropagation(); // Evita que o clique no botão de menu também dispare o evento no card

toggleMenu();

});

// Evento de clique nos cards

Cards.forEach(card => {

Card.addEventListener(“click”, function () {

If (menuVisible) {

// Se o menu estiver aberto, fecha o menu e exibe o botão imediatamente

toggleMenu();

toggleMenuButton(true); // Exibir o botão do menu imediatamente

} else {

If (menuBtnVisible) {

toggleMenuButton(false);

} else {

toggleMenuButton(true);

}

}

});

});

// Ocultar o botão do menu inicialmente

toggleMenuButton(false);

});

</script>

</body>

</html>