

## CONTEÚDO PROGRAMÁTICO RESIDÊNCIA EM TIC/SOFTWARE SERRATEC 2022

Disciplina	Conteúdo	CH (horas)
Lógica de Programação (Javascript ou Portugol)	<ol style="list-style-type: none"> <li>1. Compreender a lógica booleana (E, OU, NOT).</li> <li>2. Entender a árvore de decisão.</li> <li>3. Saber utilizar as estruturas de laço.</li> <li>4. Conhecer o conceito de recursividade.</li> <li>5. Entender a estrutura de dados (vetor, matriz, fila, pilha...).</li> <li>6. Compreender o conceito de variável e constante.</li> <li>7. Saber utilizar um sistema de controle de versão.</li> <li>8. Interface de linha de comando (navegar por diretórios, criar diretórios, apagar diretórios, copiar arquivos, apagar arquivos, mover ou renomear arquivos, listar conteúdo de um diretório, limpar tela do terminal)</li> </ol>	60
Banco de Dados (Postgres)	<ol style="list-style-type: none"> <li>1. Entender os conceitos de entidade, atributo e relacionamento.</li> <li>2. Saber mapear e projetar um banco de dados.</li> <li>3. Criar um banco de dados (DDL- Data Definition Language).</li> <li>4. Manter a integridade referencial do banco de dados (chave estrangeira).</li> <li>5. Manipular banco de dados (DML - Data Manipulation Language).</li> <li>6. Consultar banco de dados (DQL - Data Query Language).</li> <li>7. Consultar múltiplas tabelas de um banco de dados (junção de tabelas).</li> <li>8. Utilizar funções de agregação (soma, máximo, mínimo, média, etc).</li> <li>9. Fazer agrupamento.</li> <li>10. Criar índices para pesquisa no banco de dados.</li> <li>11. Entender o conceito de normalização de banco de dados.</li> </ol>	42
Programação Orientada a Objetos (Java)	<ol style="list-style-type: none"> <li>1. Modelagem de Sistemas (UML) e principais diagramas (Classe, Sequência, Atividades, Estados, etc)</li> <li>2. Compreender compilação, execução e entrada de dados (i/o, teclado, console).</li> <li>3. Dominar os identificadores, palavras chave, tipos e operadores (tipos primitivos, conversão, variáveis, hierarquia de operadores e parênteses).</li> <li>4. Saber utilizar as estruturas de programação (if, loop, switch, break, * Enum , métodos de classe, recursividade) e escopo de variáveis.</li> <li>5. Gerir objetos e classes: acesso, getters, setters, construtores, static.</li> <li>6. Gerir objetos e classes: herança, reescrita, polimorfismo, classes abstratas e Interfaces.</li> <li>7. Manipular data e hora.</li> <li>8. Saber utilizar as estruturas de dados (arrays, coleções).</li> <li>9. Realizar o tratamento de erros.</li> <li>10. Acessar um banco de dados a partir do backend.</li> <li>11. Empacotar aplicações, criar e utilizar bibliotecas.</li> </ol>	96

FrontEnd Essencial (HTML, Javascript e CSS)	<ol style="list-style-type: none"> <li>1. Conhecer os pilares de uma aplicação web, identificar a diferença entre HTML, CSS e JavaScript.</li> <li>2. Saber escrever um documento HTML com suas principais tags.</li> <li>3. Fazer um arquivo CSS com suas propriedades básicas: margin, width, height, padding, color, background, border e outros.</li> <li>4. Utilizar seletores do CSS, conhecendo suas especificidades.</li> <li>5. Saber fatiar um layout, criar uma tela entendendo como funcionam os principais displays: block, inline, none, flex e etc.</li> <li>6. Conhecer o gridsystem do bootstrap e as principais classes para utilização em alert, buttons, modais, group e outros.</li> <li>7. Saber utilizar os conceitos básicos de lógica de programação em JavaScript (variáveis, condicionais, loops, escopos, referência, entre outros).</li> <li>8. Conhecer o DOM, capturar elementos, manipular elementos.</li> <li>9. Conhecer Ecma6, arrow function e class, map, filter.</li> <li>10. Escrever arquivos HTML, CSS e JavaScript seguindo todas as boas práticas e convenções de cada linguagem.</li> <li>12. Fazer requisição AJAX.</li> </ol>	54
Desenvolvimento de API RESTFUL (Java)	<ol style="list-style-type: none"> <li>1. Entender os protocolos HTTP e REST.</li> <li>2. Saber utilizar uma ferramenta de automação, compilação, gerenciamento de dependências e empacotamento de projeto.</li> <li>3. Criar um projeto utilizando framework backend.</li> <li>4. Criar uma api REST (controllers, mappings, rotas, etc).</li> <li>5. Entender e saber aplicar os conceitos de inversão de controle e injeção de dependências.</li> <li>6. Tratar erros do backend e códigos de status HTTP.</li> <li>7. Fazer o mapeamento objeto-relacional.</li> <li>8. Utilizar linguagem de consultas objeto relacional.</li> <li>9. Habilitar a documentação com Swagger.</li> <li>10. Saber acessar outras APIs REST a partir do backend.</li> <li>11. Entender o protocolo HTTPS.</li> <li>12. Gerenciar autenticações.</li> <li>13. Fazer upload de arquivos.</li> <li>14. Criar recursos para envio de e-mails.</li> </ol>	102
Desenvolvimento Web (React)	<ol style="list-style-type: none"> <li>1. Conhecer a estrutura e os conceitos de um framework frontend web.</li> <li>2. Utilizar gerenciador de pacotes para projetos frontend web e análise de estrutura.</li> <li>3. Compreender estado e imutabilidade.</li> <li>4. Gerenciar propriedades e estados.</li> <li>5. Utilizar estado, ciclo de vida, entre outras funcionalidades sem a necessidade de escrevermos componentes com classe (hooks).</li> <li>6. Compreender o conceito de componentização.</li> <li>7. Entender a renderização condicional.</li> <li>8. Gerenciar de rotas.</li> <li>9. Saber estilizar componentes.</li> <li>10. Realizar requisições à API backend (get, post, put e delete).</li> </ol>	54

Desenvolvimento de aplicativo Mobile (React Native)	<ol style="list-style-type: none"> <li>1. Entender os conceitos do desenvolvimento mobile.</li> <li>2. Saber utilizar um framework frontend mobile.</li> <li>3. Saber configurar o ambiente de desenvolvimento mobile.</li> <li>4. Criar a estrutura do projeto.</li> <li>5. Utilizar uma ferramenta de gerenciamento, construção, deploy e emulação para aplicações mobile.</li> <li>6. Utilizar IDE e plugins para desenvolvimento de aplicações mobile.</li> <li>7. Criar componentes de classe.</li> <li>8. Gerenciar propriedades.</li> <li>9. Gerenciar estados.</li> <li>10. Criar componentes funcionais.</li> <li>11. Fazer layout e estilização.</li> <li>12. Definir navegação.</li> <li>13. Desenvolver aplicativo utilizando banco de dados embarcado.</li> <li>14. Realizar requisições à API backend (get, post, put e delete).</li> <li>15. Estilizar a interface do usuário com visual nativo do sistema operacional.</li> </ol>	66
Projeto Aplicado	Trabalhar em equipe em projeto proposto por empresas parceiras dentro do conteúdo programático ministrado no curso, com o apoio do professor da disciplina e de mentores das empresas.	86
	<b>Carga horária total</b>	<b>560h</b>