

CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS

PLANO DE CURSO

CARGA HORÁRIA: 1.200 HORAS

2023

SUMÁRIO

1. Justificativa e objetivos	3
1.1. Justificativa	3
1.2. Objetivos	5
2. Requisitos de acesso	5
3. Perfil profissional de conclusão	6
3.1. Perfil profissional da qualificação	8
4. Organização curricular	9
4.1. Estrutura de organização curricular	11
4.2. Descrição dos componentes curriculares	18
4.3. Descrição sobre como trabalhar os componentes curriculares comuns a todos os cursos ofertados pela Secretaria da Educação do Estado de São Paulo	39
4.3.1. Carreira e Competências para o Mercado de Trabalho	39
4.3.2. Projeto Multidisciplinar	40
5. Critérios de aproveitamento de estudos, conhecimentos e experiências anteriores	42
6. Critérios de avaliação	42
6.1. Avaliações contínuas	43
6.2. Atividades práticas e de laboratório	44
7. Instalações e equipamentos	45
7.1. Instalações	45
7.2. Equipamentos	46
7.3. Tecnologias digitais imprescindíveis para a área	47
8. Bibliografia	47
9. Pessoal docente e técnico	50
10. Certificados e diplomas	54
11. Estágio supervisionado (não obrigatório)	55
ANEXO 1 – MATRIZ CURRICULAR	56
ANEXO 2 – PLANO E ORIENTAÇÕES PARA ESTÁGIO	57

1. Justificativa e objetivos

1.1. Justificativa

O Estado de São Paulo é um dos maiores polos tecnológicos do Brasil, com ampla demanda por profissionais qualificados em desenvolvimento de sistemas. Empresas de diferentes setores e tamanhos, desde pequenas até grandes corporações, buscam constantemente profissionais talentosos nessa área. Ao obter um diploma de técnico em desenvolvimento de sistemas em São Paulo, a pessoa egressa aumenta suas chances de conseguir emprego, aumento de remuneração e melhores perspectivas de carreira.

Também conhecido por seu vibrante ecossistema de inovação e empreendedorismo, o Estado de São Paulo abriga incubadoras, aceleradoras e espaços de trabalho compartilhados que apoiam *startups* e projetos de desenvolvimento de tecnologias. Participar do curso técnico em Desenvolvimento de Sistemas permite que o estudante esteja imerso nesse ambiente profissional, facilitando o acesso não apenas às oportunidades de emprego, mas de empreendedorismo e colaboração com empresas inovadoras.

Os estudantes que realizam o Curso Técnico em Desenvolvimento de Sistemas em São Paulo podem obter os seguintes benefícios:

- Conhecimento e habilidades técnicas na área: o Curso Técnico em Desenvolvimento de Sistemas fornecerá aos estudantes uma base sólida de conhecimentos teóricos e práticos em programação, *design* de *software*, bancos de dados, redes, segurança da informação e outras subáreas relacionadas. Eles desenvolverão habilidades técnicas essenciais para projetar, desenvolver, implementar e manter sistemas de *software*.
- Experiência prática: os cursos técnicos geralmente têm foco prático, o que permite aos estudantes realizar práticas com o conteúdo desde a sala de aula. Eles terão a oportunidade de trabalhar em projetos reais, resolver problemas complexos e adquirir experiência voltada e relevante para o mercado de trabalho.
- Preparação para o mercado de trabalho: o Curso Técnico em Desenvolvimento de Sistemas foi elaborado visando atender as demandas do mercado de trabalho atual. Os estudantes receberão formação específica nas tecnologias e linguagens de programação mais utilizadas, tornando-se mais preparados para enfrentar os desafios e requisitos das empresas de tecnologia.
- Empregabilidade: a área de desenvolvimento de sistemas está em alta demanda e os estudantes que concluem um curso técnico nessa área têm boas perspectivas de emprego e empreendedorismo. O Estado de São

Paulo, em particular, oferece amplo mercado na área, com inúmeras oportunidades em empresas de diversos setores. Os graduados em Desenvolvimento de Sistemas têm a possibilidade de trabalhar como programadores, analistas de sistemas, desenvolvedores de *software*, testadores de qualidade, entre outras funções relacionadas.

- Crescimento profissional: com um diploma técnico em Desenvolvimento de Sistemas, os estudantes podem começar a trabalhar imediatamente ou optar por continuar seus estudos em um curso de nível superior. Eles podem buscar uma graduação em Ciência da Computação, Engenharia de *Software* ou áreas relacionadas, a fim de ampliar suas oportunidades de crescimento profissional e salarial.
- Rede de contatos: durante o curso, os estudantes terão a oportunidade de interagir com colegas de classe, professores e profissionais do setor de tecnologia. Essa rede de contatos pode ser valiosa para futuras oportunidades de emprego, colaborações em projetos e compartilhamento de conhecimentos.
- Flexibilidade e empreendedorismo: com um diploma técnico em desenvolvimento de sistemas, os estudantes também têm a opção de trabalhar como *freelancers* ou empreender seus próprios negócios. Eles podem desenvolver aplicativos, criar *sites*, prestar serviços de consultoria ou iniciar suas próprias *startups*.

Trabalhar na área de Desenvolvimento de Sistemas está em evidência. O *site* de recrutamento de talentos de alto desempenho, Glassdoor (2020), divulgou um *ranking* com as 50 melhores empresas para se trabalhar em 2020. Dessa lista, os melhores locais são empresas de Tecnologia da Informação (TI).

Mesmo assim, a área de Desenvolvimento de Sistemas enfrenta escassez de profissionais qualificados no mercado, assim como ocorre com outros segmentos de TI. Ou seja, a área oferece boas perspectivas profissionais, com alta demanda de candidatos.

A pesquisa “Achados e Recomendações para Formação Educacional e Empregabilidade em TIC”, realizada pela Associação Brasileira das Empresas de Tecnologia da Informação e Comunicação (Brasscom), apontou que a demanda anual por novos profissionais de TI, no período de 2020 a 2025, será de 80 mil.

Isso nos permite concluir que o técnico em desenvolvimento de sistemas é uma das “profissões do futuro”, pois existe demanda crescente de profissionais com habilidades para conectar toda a infraestrutura tecnológica e tornar disponíveis sistemas e aplicativos.

1.2. Objetivos

O Curso Técnico em Desenvolvimento de Sistemas tem como objetivos gerais:

- formar um profissional capaz de atuar na análise de problemas organizacionais e de selecionar, projetar e utilizar, de forma adequada e econômica, soluções tecnológicas, a fim de aperfeiçoar e viabilizar a implantação de diferenciais estratégicos para as organizações;
- formar um profissional capaz de suprir as necessidades das empresas na área de análise e desenvolvimento de sistemas, que enfatize os aspectos práticos, e apto a trabalhar com ferramentas, equipamentos e metodologia de projetos na produção de sistemas;
- formar profissionais e cidadãos conscientes de sua responsabilidade político-social, empreendedores, e que busquem formação contínua.
- fomentar o uso de normas técnicas para a solução de problemas no processo de desenvolvimento de sistemas e *software*;
- formar profissionais aptos a planejar e executar projetos que garantam as entregas dos produtos digitais usando técnicas e métodos que integram projetos;
- formar profissionais competentes para analisar, projetar, testar e manter sistemas em conformidade com os requisitos de *software* levantados, aplicando metodologias e processos de engenharia e serviços adequados, além de recomendações de qualidade e segurança em sistemas de informação;
- formar profissionais hábeis em realizar prospecções, testes e avaliações de ferramentas e produtos de desenvolvimento de sistemas;
- formar profissionais capazes de utilizar modelagem e pensamento computacional.

Esses objetivos e conceitos fundamentais estão consonantes ao perfil profissional do egresso, à estrutura curricular do curso, ao contexto educacional, às características locais e regionais, assim como, às novas práticas emergentes no campo de conhecimento relacionado ao curso.

2. Requisitos de acesso

O acesso ao Curso Técnico em Desenvolvimento de Sistemas é destinado aos estudantes que tenham concluído o 9º ano do Ensino Fundamental e estejam devidamente matriculados no Ensino Médio na escola da rede pública estadual paulista em que o curso técnico será ofertado.

Por razões de ordem administrativa e/ou pedagógica que sejam justificadas, poderão ser utilizados procedimentos diversificados para ingresso, sendo os candidatos deles notificados por ocasião de suas inscrições.

O acesso direto à 3ª série ou ao longo da 2ª série poderá ocorrer mediante avaliação de competências adquiridas por aproveitamento de estudos realizados, experiências profissionais prévias na área do curso ou reclassificação.

3. Perfil profissional de conclusão

Ao concluir a 3ª série do Ensino Médio, incluindo a integralização das 1.200 horas previstas do curso técnico em questão, o estudante terá a certificação de Ensino Médio com Habilitação Profissional de Técnico em Desenvolvimento de Sistemas.

Em consonância com as necessidades do mercado de trabalho e com a missão do governo do Estado de São Paulo, os egressos do Curso Técnico em Desenvolvimento de Sistemas serão profissionais conscientes de sua responsabilidade político-social, reconhecedores dos benefícios da convivência na diversidade e do ambiente acessível a todos, bem como cidadãos preocupados com os Direitos Humanos que assumam de maneira ética e transformadora seu papel tanto no ambiente de trabalho quanto fora dele. Serão, assim, profissionais com o objetivo de sempre questionar, analisar, repensar e recriar suas metodologias de trabalho, na busca incessante da melhoria do saber-fazer, além de estarem preparados para poder empreender e trazer inovações.

Esse profissional trabalha com ferramentas computacionais, equipamentos de informática e metodologias de implementação de projetos para apoio ao planejamento, ao desenvolvimento e à construção de sistemas. Será capaz de trabalhar em equipe, ter raciocínio lógico, empregar linguagens de programação e seguir as métricas de análise e desenvolvimento existentes. Assim, poderá assegurar o desenvolvimento de *softwares* que ofereçam: usabilidade, funcionalidade, confiabilidade, eficiência, manutenibilidade, portabilidade e acessibilidade, sempre finalizando as entregas balizado em testes de *software*.

O perfil do egresso do curso inclui competências, habilidades e atitudes que atendem às exigências estabelecidas no Catálogo Nacional dos Cursos Técnicos¹.

Competências Técnicas:

1. Desenvolver sistemas computacionais utilizando ambiente de desenvolvimento.

¹ <http://cnct.mec.gov.br/cursos/curso?id=81>

2. Levantar e dimensionar requisitos específicos do negócio, adequando-os às funcionalidades do sistema.
3. Desenvolver soluções de *software* (*back-end*, *front-end* e *full-stack*) por meio de técnicas, métodos, ferramentas e linguagens de programação diversas.
4. Executar manutenção de programas.
5. Codificar aplicações e rotinas mediante linguagens de programação, utilizando *frameworks* adequados.
6. Prestar apoio técnico na elaboração da documentação de sistemas.
7. Migrar sistemas, implementando rotinas e estruturas de dados mais eficazes.
8. Conhecer *frameworks* de desenvolvimento ágeis, utilizando tecnologias de CI e CD que trabalham na segurança do ambiente funcional e entregas, divididas em partes que agregam valor ao negócio de forma rápida.
9. Conhecer técnicas de computação e gerenciamento de dados para soluções em nuvem, parametrizando e dimensionando aplicações de acordo com as necessidades do negócio.
10. Realizar testes funcionais baseados em *scripts* de programas de computador e aplicativos.
11. Além desses, e em consonância com o explicitado no CNCT/2020 4ª edição, também se tem como objetivo desenvolver nos estudantes as seguintes habilidades:

“11. Conhecimentos e saberes relacionados aos processos de planejamento e execução de projetos computacionais de forma a garantir a entrega de produtos digitais, análise de *softwares*, testagem de protótipos, de acordo com suas finalidades.
12. Conhecimentos e saberes relacionados às normas técnicas, à liderança de equipes, à solução de problemas técnicos e à assertividade na comunicação de laudos e análises”.

Competências Socioemocionais:

Visando à formação de uma pessoa desenvolvedora de sistemas, as competências socioemocionais trabalhadas no curso são:

1. Criatividade;
2. Resolução de problemas;
3. Comprometimento;
4. Comunicação (Oratória, Escuta Ativa e Comunicação Assertiva);
5. Relacionamento/Trabalho em equipe;
6. Empreendedorismo/Sentimento de Dono/Liderança de Equipe;

7. Colaboração;
8. Curiosidade;
9. Inteligência Emocional;
10. Resiliência.

O perfil do egresso será um profissional completo, preparado para enfrentar os desafios do mercado de trabalho, tomar decisões embasadas em evidências, trabalhar em equipe, liderar projetos e contribuir para o desenvolvimento e o sucesso das organizações.

3.1. Perfil profissional da qualificação

Qualificação Profissional Técnica de Nível Médio de Auxiliar em Desenvolvimento de Sistemas.

Ao concluir a 2ª série do Ensino Médio, incluindo a integralização da carga horária prevista do curso técnico para a série em questão (600 horas), o estudante terá a certificação intermediária de Qualificação Profissional Técnica de Nível Médio de Auxiliar em Desenvolvimento de Sistemas.

Especificação da Qualificação Profissional Técnica de Nível Médio de Auxiliar em Desenvolvimento de Sistemas:

O Auxiliar em Desenvolvimento de Sistemas é o profissional que sabe os conceitos lógicos e as estruturas de Linguagem de Programação, modela e desenvolve banco de dados relacionais, se preocupando com as redes de computadores e segurança da informação. Essas informações são hospedadas de maneira local ou na nuvem, e as etapas do desenvolvimento são controladas por meio de metodologias ágeis. Tudo isso, aliado com a capacidade de atuar com Inteligência Artificial Geracional sempre pensando em como aplicar todo esse conhecimento na empregabilidade, ou até mesmo de forma empreendedora.

Competências Técnicas:

1. Desenvolver sistemas computacionais utilizando ambiente de desenvolvimento.
2. Levantar e dimensionar requisitos específicos do negócio, adequando-os às funcionalidades do sistema.
3. Prestar apoio técnico na elaboração da documentação de sistemas.
4. Conhecer *frameworks* de desenvolvimento ágeis, utilizando tecnologias de CI e CD que trabalham na segurança do ambiente funcional e entregas, divididas em partes que agregam valor ao negócio de forma rápida.

5. Conhecer técnicas de computação e gerenciamento de dados para soluções em nuvem, parametrizando e dimensionando aplicações de acordo com as necessidades do negócio.
6. Realizar testes funcionais baseados em *scripts* de programas de computador e aplicativos.

Competências Socioemocionais:

Visando à formação de uma pessoa auxiliar em desenvolvimento de sistemas, as competências socioemocionais trabalhadas serão:

1. Criatividade;
2. Resolução de problemas;
3. Comprometimento;
4. Comunicação (Oratória, Escuta Ativa e Comunicação Assertiva);
5. Relacionamento/Trabalho em equipe;
6. Curiosidade;
7. Inteligência Emocional;
8. Resiliência.

4. Organização curricular

A organização curricular foi pensada de modo a permitir observar a relação entre os componentes curriculares e a crescente complexidade das competências desenvolvidas e requeridas ao longo do curso, as quais asseguram a convergência entre esses componentes curriculares determinados dentro do fluxo das suas respectivas ofertas, possibilitando conferir sentido ao processo de ensino e aprendizagem orientados por competências.

A organização curricular do Curso Técnico em Desenvolvimento de Sistemas se alinha ao desejado perfil profissional do egresso, além de formar cidadãos empreendedores e conscientes de sua responsabilidade político-social, de maneira ética e transformadora, conscientes do seu papel tanto em ambiente de trabalho quanto fora deste. Busca, ainda, oportunizar uma formação contínua e o aprofundamento do entendimento da dimensão interdisciplinar dos saberes abordados, proporcionando recursos que permitam ao estudante colocar em prática, vivenciar e entender os processos que permeiam a prática profissional, no que tange à aplicação de conhecimentos, ao aperfeiçoamento de habilidades e ao desenvolvimento de atitudes.

Os conteúdos foram previstos com base em pesquisa e discussão com o mercado de trabalho, para que contenham o “estado da arte” em relação a comportamentos, tecnologias, métodos, modelos, ferramentas e aplicativos.

Unidades curriculares:

As unidades curriculares estão organizadas de forma sequencial e progressiva, permitindo que os estudantes adquiram conhecimentos e competências de forma gradual, ao longo dos dois anos de curso.

A carga horária foi distribuída de maneira equilibrada entre unidades curriculares teóricas, práticas e de projetos interdisciplinares.

Metodologias de ensino:

Para atingir os objetivos educacionais, serão utilizadas diversas estratégias de ensino, visando promover uma aprendizagem significativa e contextualizada. Entre as estratégias metodológicas previstas, destacam-se:

1. Aulas expositivas e dialogadas: para promover a compreensão dos conceitos teóricos fundamentais à área, com a participação ativa dos estudantes por meio de discussões e debates.
2. Estudos de caso: para estimular a análise crítica e a resolução de problemas reais ou hipotéticos, aplicando os conhecimentos teóricos na prática.
3. Trabalhos em grupo: para desenvolver habilidades de trabalho em equipe, negociação, comunicação e liderança.
4. Uso de recursos tecnológicos: *softwares* de gestão, ferramentas *on-line* e recursos multimídia, visando enriquecer o processo de aprendizagem e desenvolver habilidades técnicas e o letramento digital. Os estudantes terão acesso ao Ambiente Virtual de Aprendizagem (AVA), onde realizarão atividades interativas, discussões em fóruns e terão acesso a materiais complementares. O AVA contemplará as demandas diferenciadas, quanto à acessibilidade.
5. Visitas técnicas: possibilidade da realização de visitas a empresas e instituições relevantes do setor na região da escola do estudante, proporcionando a ele a oportunidade de conhecer de perto o ambiente de trabalho e as práticas organizacionais.
6. Integração teoria-prática: busca constante de conectar os conteúdos teóricos com situações e desafios reais, por meio de estudos de caso, exercícios práticos e projetos que simulem a realidade empresarial.

Dessa forma, a organização curricular do Curso Técnico em Desenvolvimento de Sistemas busca proporcionar aos estudantes uma formação abrangente, contextualizada e alinhada às demandas do mercado de trabalho.

4.1. Estrutura de organização curricular

Componente curricular		Carga horária (h)	
Lógica e Linguagem de Programação		120	
Unidades curriculares		Teórica (h)	Prática (h)
I	Estrutura Básica	12	12
II	Estruturas de Decisão Simples	6	6
III	Estruturas de Decisão Composta	6	18
IV	Estruturas de Seleção	6	6
V	Estruturas de Repetição	12	12
VI	Vetores e Matrizes	12	12

Componente curricular		Carga horária (h)	
Versionamento de Código e Sistemas de Mensageria		90	
Unidades curriculares		Teórica (h)	Prática (h)
I	Introdução ao Versionamento de Código, Git e GitHub	4,5	9
II	Fluxo de Trabalho	4,5	4,5

III	Integração Contínua e Entrega Contínua	4,5	9
IV	Boas Práticas de Versionamento e Estratégias Avançadas	4,5	4,5
V	Introdução aos Serviços de Mensageria e Protocolos de Comunicação	4,5	9
VI	RabbitMQ, Amazon Simple Queue Service (SQS) e Apache Kafka	4,5	13,5
VII	Arquitetura e Ferramentas de Apoio e Práticas Avançadas	4,5	9

Componente curricular		Carga horária (h)	
Redes de Computadores e Segurança da Informação na Nuvem		90	
Unidades curriculares		Teórica (h)	Prática (h)
I	Topologia de Redes	13,5	0
II	Meios de Transmissão	13,5	0
III	Protocolos e Camadas	9	4,5
IV	Conceitos, Princípios e Políticas de Segurança da Informação	9	0
V	Características e Aplicações da ISO 27001	9	0
VI	Características e Aplicações da LGPD	9	13,5
VII	Aplicações em Nuvem	0	9

Componente curricular		Carga horária (h)	
Processos de Desenvolvimento de Software e Metodologias Ágeis		90	
Unidades curriculares		Teórica (h)	Prática (h)
I	Levantamento de Requisitos	13,5	0
II	Manifesto Ágil	9	4,5
III	Testes Funcionais e Não Funcionais	4,5	9
IV	Tipos de Modelagem	4,5	4,5
V	Diagramas UML	13,5	0
VI	Mindset Ágil vs Mindset Cascata	4,5	4,5
VII	Pilares da Programação Orientada a Objetos	9	9

Componente curricular		Carga horária (h)	
Inteligência Artificial		90	
Unidades curriculares		Teórica (h)	Prática (h)
I	Conceitos Iniciais	9	0

II	Fundamentos da IA Geracional	9	9
III	Aplicações Práticas da IA Geracional	13,5	9
IV	ChatGPT como Exemplo de IA Geracional	9	13,5
V	Personalização de Mídias	9	9

Componente curricular		Carga horária (h)	
Carreira e Competências para o Mercado de Trabalho		90	
Unidades curriculares		Teórica (h)	Prática (h)
I	Conhecendo o Mercado de Trabalho	18	4,5
II	Planejamento e Preparação Profissional (criação de CV e uso de LinkedIn)	18	9
III	Autoconhecimento e Orientação Profissional	9	4,5
IV	Preparação para Processos Seletivos e Busca de Oportunidades	4,5	4,5
V	Finanças Pessoais	4,5	0
VI	Iniciação Empreendedora	9	4,5

Componente curricular		Carga horária (h)	
Programação Mobile		120	
Unidades curriculares		Teórica (h)	Prática (h)
I	Aplicações Móveis	12	18
II	Armazenamento de Dados e Repositórios	6	24
III	Interfaces com o Usuário	6	24
IV	Publicação nas <i>Stores</i>	6	24

Componente curricular		Carga horária (h)	
Banco de Dados NoSQL (Não Relacionais)		120	
Unidades curriculares		Teórica (h)	Prática (h)
I	Banco de Dados Chave-Valor	12	18
II	Banco de Dados Orientado a Colunas	12	18
III	Banco de Dados Orientado a Documentos	12	18
IV	Banco de Dados Orientado a Grafos	12	18

Componente curricular		Carga horária (h)	
Programação <i>Back-End</i>		90	
Unidades curriculares		Teórica (h)	Prática (h)
I	Introdução à Programação <i>Back-End</i> e Conceitos de Desenvolvimento para <i>Web</i>	9	4,5
II	Linguagens de Programação <i>Back-End</i>	4,5	27
III	Servidores <i>Web</i> , APIs e Segurança das Aplicações	4,5	18
IV	<i>Frameworks Back-End</i> e Integração com Serviços Externos	4,5	18

Componente curricular		Carga horária (h)	
Programação <i>Front-End</i>		90	
Unidades curriculares		Teórica (h)	Prática (h)
I	Introdução à Programação <i>Front-End</i> e Conceitos de Desenvolvimento para <i>Web</i>	9	4,5
II	Linguagens de Programação <i>Front-End</i>	4,5	27
III	Estilização de Páginas, Ferramentas de Desenvolvimento e Performance	4,5	18
IV	Integração com Serviços Externos e Esteiras de Entrega	4,5	18

Componente curricular		Carga horária (h)	
Modelagem e Desenvolvimento de Banco de Dados		120	
Unidades curriculares		Teórica (h)	Prática (h)
I	Levantamento de Dados e Conceitos Fundamentais	6	12
II	Tipos de Relacionamentos	6	12
III	Formas Normais	6	6
IV	Comandos SQL (CRUD)	6	12
V	DDL X DML	6	12
VI	Cláusulas Select	6	12
VII	SGBD Relacional	6	12

Componente curricular		Carga horária (h)	
Projeto Multidisciplinar		90	
Unidades curriculares		Teórica (h)	Prática (h)
I	Introdução ao Projeto Multidisciplinar	4,5	4,5
II	Definição do Projeto e Início das Pesquisas	9	13,5

III	Planejamento e Organização	9	9
IV	Desenvolvimento e Acompanhamento do Projeto (Uso de Ferramentas Office)	18	9
V	Preparação da Apresentação e Discurso do Projeto (<i>Pitch</i>)	4,5	4,5
VI	Apresentação de Projetos	4,5	0

4.2. Descrição dos componentes curriculares

Componente curricular		Carga horária (h)
Lógica e Linguagem de Programação		120
Descrição	Lógica e Linguagem de Programação tem como objetivo abordar: conceito de lógica; lógica aplicada à programação; tipos de algoritmos; conceitos de variáveis e constantes; tipos primitivos de dados; operadores (aritméticos, relacionais e lógicos); estruturas de seleção; estruturas de controle e repetição; vetores e matrizes e ponteiros, funções e passagem de parâmetros (trabalhar com Algoritmos e C# Windows Forms Application).	
Temas	Estrutura Básica; Estruturas de Decisão Simples; Estruturas de Decisão Composta; Estruturas de Seleção; Estruturas de Repetição e Vetores e Matrizes.	
Competências técnicas	Desenvolver sistemas computacionais utilizando ambiente de desenvolvimento; Migrar sistemas, implementando rotinas e estruturas de dados mais eficazes.	
Competências socioemocionais	Agir com curiosidade e criatividade na busca das melhores soluções com lógica de programação; Identificar e analisar problemas, desenvolver alternativas e implementar soluções eficazes durante	

	<p>a execução de um projeto; Ter comprometimento com as demandas; Reconhecer e gerenciar emoções próprias e de outras pessoas, mantendo o equilíbrio emocional em situações desafiadoras; Desenvolver resiliência diante de situações desafiadoras.</p>
<p>Bibliografia básica:</p> <p>MANZANO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo de. <i>Algoritmos: lógica para desenvolvimento de programação de computadores</i>. 29. ed. São Paulo: Érica, 2019.</p> <p>PIVA JUNIOR, Dilermando <i>et al.</i> <i>Algoritmos e programação de computadores</i>. 2. ed. São Paulo: LTC, 2019.</p>	
<p>Bibliografia complementar:</p> <p>DASGUPTA, Sanjoy; CHRISTOS, Papadimitriou; VAZIRANI, Umesh. <i>Algoritmos</i>. São Paulo: AMGH, 2009.</p> <p>MANZANO, José Augusto Navarro Garcia. <i>Estudo dirigido de linguagem C</i>. São Paulo: Érica, 2013.</p> <p>MIZRAHI, Victorine Viviane. <i>Treinamento em linguagem C</i>. 2. ed. São Paulo: Pearson Universidades, 2008.</p> <p>SCHILDT, Hebert. <i>C# 4.0: the Complete Reference</i>. New York: McGraw-Hill, 2010.</p>	

Componente curricular	Carga horária (h)
Versionamento de Código e Sistemas de Mensageria	90
Descrição	<p>Versionamento de Código e Sistemas de Mensageria tem como objetivo abordar: a importância do controle de versões e introdução; Git (conceitos básicos, instalação e configuração); operações básicas (criação, clonagem, adição, <i>commit</i> e <i>push</i>); <i>branches</i> e <i>merges</i>; GitHub (conceitos básicos, colaboração, <i>pull requests</i>, <i>code reviews</i>, <i>issues</i> e projetos); fluxo de trabalho (GitFlow e GitHub Flow); CI/CD (integração contínua e entrega contínua, conceitos básicos e ferramentas); boas práticas (organização do repositório, mensagens de <i>commit</i>, trabalho em</p>

	<p>equipe e documentação do código); ferramentas de apoio (GitKraken, Sourcetree e Visual Studio Code); práticas avançadas (submódulos, <i>rebase</i>, <i>bisect</i> e estratégias de versionamento); serviços de mensageria (introdução e importância); protocolos de comunicação: HTTP, REST, WebSocket, MQTT, AMQP; RabbitMQ (conceitos básicos, instalação e configuração, <i>exchanges</i>, <i>queues</i>, <i>bindings</i>, mensagens persistentes e não persistentes e Workers); Amazon SQS (conceitos básicos, criação e configuração de filas, envio e recebimento de mensagens e integração com serviços da AWS); Apache Kafka (conceitos básicos, instalação e configuração, tópicos, partições, <i>offsets</i>, produtores, consumidores e grupos de consumidores); boas práticas (escalabilidade, disponibilidade, garantia de entrega de mensagens, monitoramento e gerenciamento de erros); ferramentas de apoio (Spring Boot, Apache Camel, AWS SDK) e tendências e novidades (<i>serverless</i>, Kubernetes e mensageria).</p>
Temas	<p>Introdução ao Versionamento de Código, Git e GitHub; Fluxo de Trabalho; Integração Contínua e Entrega Contínua; Boas Práticas de Versionamento e Estratégias Avançadas; Introdução aos Serviços de Mensageria e Protocolos de Comunicação; RabbitMQ, Amazon Simple Queue Service (SQS) e Apache Kafka; Arquitetura e Ferramentas de Apoio e Práticas Avançadas.</p>
Competências técnicas	<p>Desenvolver soluções de <i>software</i> (<i>back-end</i>, <i>front-end</i> e <i>full-stack</i>), utilizando técnicas, métodos, ferramentas e linguagens de programação diversas; Prestar apoio técnico na elaboração da documentação de sistemas; Utilizar <i>frameworks</i> de desenvolvimento ágeis, utilizando tecnologias de CI e CD que trabalham na segurança do ambiente funcional e entregas divididas em partes que agregam valor ao negócio de forma rápida.</p>
Competências socioemocionais	<p>Identificar e analisar problemas, desenvolver alternativas e implementar soluções eficazes durante a execução de um projeto; Ter comprometimento com as demandas; Saber ouvir ativamente, expressar-se de forma clara e assertiva, e construir relacionamentos sólidos; Agir com curiosidade e</p>

	<p>criatividade na busca das melhores soluções em Versionamento de Códigos e Serviços de Mensageria; Reconhecer e gerenciar emoções próprias e de outras pessoas, mantendo o equilíbrio emocional em situações desafiadoras.</p>
<p>Bibliografia básica:</p> <p>BEER, Brent; BELL, Peter. <i>Introdução ao GitHub: um guia que não é técnico</i>. São Paulo: Novatec Editora, 2014. AQUILES, Alexandre; FERREIRA, Rodrigo. <i>Controlando versões com Git e GitHub</i>. São Paulo: Casa do Código, 2014.</p> <p>SAUDATE, Alexandre. <i>SOA aplicado: integrando com web services e além</i>. São Paulo: Casa do Código, 2015.</p>	
<p>Bibliografia complementar:</p> <p>ALMEIDA, Flávio. <i>REST: Construa APIs inteligentes de maneira simples</i>. São Paulo: Casa do Código, 2014.</p> <p>BROWN, Patrick. <i>Continuous Integration: Improving Software Quality and Reducing Risk</i>. Boston: Addison-Wesley Professional, 2006.</p> <p>TOSHEV, Martin. <i>Learning RabbitMQ</i>. Birmingham: Packt Publishing Ltd, 2015. PREUSSER, Brent. <i>GitHub Essentials</i>. Birmingham: Packt Publishing, 2018.</p> <p>RICHARDSON, Chris. <i>Microservices Patterns: With Examples in Java</i>. Sebastopol: O'Reilly Media, 2018.</p>	

Componente curricular	Carga horária (h)
<p>Redes de Computadores e Segurança da Informação na Nuvem</p>	<p>90</p>
<p>Descrição</p>	<p>Redes de Computadores e Segurança da Informação na Nuvem tem como objetivo abordar: conceitos, princípios e políticas de segurança da informação; confidencialidade, integridade e disponibilidade; auditoria em tecnologia da informação e sistemas da informação; classificação e controle dos ativos de informação; responsabilidades do auditor em políticas, conformidades e resposta a incidentes; segurança física e lógica; análise de logs e trilhas de auditoria; resposta a incidentes e controle de vulnerabilidades físicas e lógicas; auditoria interna e externa em</p>

	<p>tecnologia da informação e/ou sistemas da informação; governança em tecnologia da informação, riscos e controles; metodologia de auditoria de tecnologia da informação e sistemas de informação; plano e escopo da auditoria de sistemas da informação; Cobit e COSO como guias de boas práticas de controle; ferramentas de auditoria de tecnologia da informação e sistemas de informação; normas, padrões e exigências regulatórias (ISO 27001, PCI, Sarbanes-Oxley – SOX); técnicas de auditoria de tecnologia da informação e sistemas de informação; guias práticos de segurança (ITIL, NIST, CIS benchmarks, FIPS); condução de auditoria; gerenciamento de risco relacionado ao uso de tecnologia da informação; ISO 27001/2, 31000, 38500; NIST – SP800-30, SP800-53, SP800-60; segurança no desenvolvimento de <i>software</i>, código seguro; ética profissional com foco na LGPD.</p>
Temas	<p>Topologia de Redes; Meios de Transmissão; Protocolos e Camadas; Conceitos, Princípios e Políticas de Segurança da Informação; Definição e Características da ISO 27001; Definição e Características da LGPD e Aplicações em Nuvem.</p>
Competências técnicas	<p>Utilizar técnicas de computação e gerenciamento de dados para soluções em nuvem, parametrizando aplicações e dimensionando de acordo com as necessidades do negócio.</p>
Competências socioemocionais	<p>Identificar e analisar problemas, desenvolver alternativas e implementar soluções eficazes durante a execução de um projeto; Agir com curiosidade e criatividade na busca das melhores soluções para redes de computadores e segurança da informação.</p>
<p>Bibliografia básica:</p> <p>KUROSE, James F.; ROSS, Keith W. <i>Redes de computadores e a internet: Uma Abordagem Top-down</i>. 6. ed. São Paulo: Pearson, 2013.</p> <p>STALLINGS, William. <i>Criptografia e segurança de redes: princípios e práticas</i>. 6. ed. São Paulo: Pearson, 2014.</p> <p>TANENBAUM, Andrew S.; WETHERALL, David J. <i>Redes de computadores</i>. 5. ed. São Paulo: Pearson, 2012.</p>	

Bibliografia complementar:

CLOUD SECURITY ALLIANCE. *Security Guidance for Critical Areas of Focus in Cloud Computing*. Versão 4.0, 2017. Disponível em: <https://cloudsecurityalliance.org/guidance/>. Acesso em: 2 nov. 2023.

VERAS, Manoel. *Computação em nuvem*. Rio de Janeiro: Brasport, 2015.
HENNESSY, John; PATTERSON, David. *Arquitetura de computadores: uma abordagem quantitativa*. 6. ed. São Paulo: GEN LTC, 2019.

MARTINEZ, Luciano. *Curso de direito do trabalho*. São Paulo: SaraivaJur, 2021.
MORAES, Alexandre Fernandes de. *Redes de computadores: fundamentos*. 8. ed. São Paulo: Érica, 2020.

Componente curricular	Carga horária (h)
Processos de Desenvolvimento de <i>Software</i> e Metodologias Ágeis	90
Descrição	<p>Processos de Desenvolvimento de <i>Software</i> e Metodologias Ágeis tem como objetivo abordar: Introdução à Engenharia de <i>Software</i> (histórico, áreas de atuação, sistemas legados); definição de <i>software</i>, sistema, produto de <i>software</i>, processo de <i>software</i> e modelo de processo; atividades no desenvolvimento de <i>software</i> (especificação, desenvolvimento, validação e evolução); modelos de desenvolvimento (cascata, evolucionário); abordagens (análise estruturada, orientada a objetos, orientada a aspectos); engenharia de <i>software</i> baseada em componentes, orientada a serviços; melhores práticas: desenvolvimento iterativo, gerenciamento de requisitos, modelagem visual (UML), arquitetura baseada em componentes, verificação contínua de qualidade, gerenciamento de configuração e mudanças; processo unificado (UP): conceitos, melhores práticas, papéis, atividades, artefatos e fluxos; DDD (Domain Driven Design) na arquitetura de <i>software</i>; Clean Code; Metodologias Ágeis (Extreme Programming, Scrum); DevOps, CI, CD; métricas (conceitos, tipos e uso); modelagem de processos (elementos, ferramentas CASE); desenvolvimento de sistemas críticos; modelagem no desenvolvimento de sistemas (metodologias, notação UML, princípios de</p>

	orientação a objetos, linguagens orientadas a objetos, heurísticas da orientação a objetos e implementação).
Temas	Levantamento de Requisitos; Manifesto Ágil; Testes Funcionais e Não Funcionais; Tipos de Modelagem; Diagramas UML; Mindset Ágil vs Mindset Cascata e Pilares da Programação Orientada a Objetos.
Competências técnicas	Desenvolver sistemas computacionais, utilizando ambiente de desenvolvimento; Desenvolver soluções de <i>software</i> (<i>back-end</i> , <i>front-end</i> e <i>full-stack</i>), utilizando técnicas, métodos, ferramentas e linguagens de programação diversas; Executar manutenção de programas; Prestar apoio técnico na elaboração da documentação de sistemas; Migrar sistemas, implementando rotinas e estruturas de dados mais eficazes; Utilizar <i>frameworks</i> de desenvolvimento ágeis, utilizando tecnologias de CI e CD que trabalham na segurança do ambiente funcional e entregas divididas em partes que agregam valor ao negócio de forma rápida; Realizar testes funcionais baseados em <i>scripts</i> de programas de computador e aplicativos.
Competências socioemocionais	Agir com curiosidade e criatividade na busca das melhores soluções em desenvolvimento de <i>softwares</i> e metodologias ágeis; Identificar e analisar problemas, desenvolver alternativas e implementar soluções eficazes durante a execução de um projeto; Desenvolver resiliência diante de situações desafiadoras.
Bibliografia básica:	
FOWLER, Martin. <i>Agile Software Development: Principles, Patterns, and Practices</i> . Boston: Addison-Wesley Professional, 2002.	
PRESSMAN, Roger S. <i>Engenharia de software: uma abordagem profissional</i> . 7. ed. Porto Alegre: AMGH, 2016.	
SCHWABER, Ken; SUTHERLAND, Jeff. <i>Guia do Scrum - um guia definitivo para o Scrum: as regras do jogo</i> . [atualizado em 2020]. Disponível em: < https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Portuguese-European.pdf >. Acesso em: 15 nov. 2023.	
Bibliografia complementar:	

BECK, Kent. *Extreme Programming Explained: Embrace Change*. 2. ed. Boston: Addison-Wesley, 2004.

COHN, Mike. *User Stories Applied: For Agile Software Development*. Boston: Addison-Wesley Professional, 2004.

FOWLER, Martin. *Refactoring: Improving the Design of Existing Code*. Boston: Addison-Wesley Professional, 2018.

RUMBAUGH, James; JACOBSON, Ivar; BOOCH, Grady. *Unified Modeling Language Reference Manual*. 2. ed. Boston: Addison-Wesley, 2004.

SOMMERVILLE, Ian. *Engenharia de Software*. 9. ed. São Paulo: Pearson Addison Wesley, 2011.

Componente curricular		Carga horária (h)
Inteligência Artificial		90
Descrição	Inteligência Artificial tem como objetivo abordar: Introdução à IA geracional (definição, importância e histórico); exemplos de aplicações; fundamentos da IA geracional (aprendizado de máquina, redes neurais artificiais e <i>deep learning</i>); processamento de linguagem natural (NLP): definição e exemplos de aplicações; aplicações práticas da IA geracional (personalização de conteúdo, <i>chatbots</i> e assistentes virtuais, previsão de demanda e planejamento de produção); o ChatGPT como exemplo de IA geracional (apresentação, arquitetura, treinamento e exemplos de uso); limitações e desafios na aplicação do ChatGPT (viés de dados, ética na IA); o futuro da IA geracional (possíveis avanços tecnológicos, impacto na sociedade e ética na IA).	
Temas	Conceitos Iniciais; Fundamentos da IA Geracional; Aplicações práticas da IA Geracional; ChatGPT como um exemplo de IA Geracional; Personalização de Mídias.	
Competências técnicas	Levantar e dimensionar requisitos específicos do negócio, adequando-os às funcionalidades do sistema; desenvolver soluções de <i>software</i> (<i>back-end</i> , <i>front-end</i> e <i>full-stack</i>), utilizando técnicas, métodos,	

	ferramentas e linguagens de programação diversas.
Competências socioemocionais	Agir com curiosidade e criatividade na busca das melhores soluções em Inteligência Artificial; Saber ouvir ativamente, expressar-se de forma clara e assertiva, e construir relacionamentos sólidos; Reconhecer e gerenciar emoções próprias e de outras pessoas, mantendo o equilíbrio emocional em situações desafiadoras.
<p>Bibliografia básica:</p> <p>COPPIN, Ben. <i>Inteligência artificial</i>. Rio de Janeiro: LTC, 2013.</p> <p>FACELI, Katti et al. <i>Inteligência artificial: uma abordagem de aprendizado de máquina</i>. 2. ed. Rio de Janeiro: LTC, 2021.</p> <p>GABRIEL, Martha. <i>Inteligência Artificial - Do Zero ao Metaverso</i>. São Paulo: Editora Atlas, 2022.</p>	
<p>Bibliografia complementar:</p> <p>HAYKIN, Simon. <i>Redes neurais: princípios e prática</i>. 2. ed. Porto Alegre: Bookman, 2007. E-book. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788577800865/. Acesso em: 2 nov. 2023.</p> <p>LUGER, George F. <i>Inteligência artificial</i>. 6. ed. São Paulo: Pearson Education do Brasil, 2013. E-book. Disponível em: https://plataforma.bvirtual.com.br/Leitor/Publicacao/180430/epub/0. Acesso em: 2 nov. 2023.</p> <p>MEDEIROS, Luciano Frontino de. <i>Inteligência artificial aplicada: uma abordagem introdutória</i>. Curitiba: InterSaberes, 2018. E-book. Disponível em: https://plataforma.bvirtual.com.br/Leitor/Publicacao/161682/epub/0. Acesso em: 2 nov. 2023.</p> <p>RUSSEAL, Stuart; NORVIG, Peter. <i>Inteligência artificial</i>. Rio de Janeiro: LTC, 2021. E-book. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788595156104/. Acesso em: 2 nov. 2023.</p> <p>VALDATI, Line de Brittos. <i>Inteligência artificial – IA</i>. Curitiba: Contentus, 2021. E-book. Disponível em: https://plataforma.bvirtual.com.br/Leitor/Publicacao/191624/pdf/0?code=. Acesso em: 2 nov. 2023.</p>	

Componente curricular	Carga horária (h)
Carreira e Competências para o Mercado de Trabalho	90
Descrição	<p>Carreira e Competências para o Mercado de Trabalho tem como objetivo desenvolver na prática as habilidades fundamentais para o mundo do trabalho. Os estudantes também conhecerão os caminhos para o empreendedorismo em seu campo de atuação. Nesse componente curricular, terão a possibilidade de praticar a comunicação de sua história de vida, bem como suas habilidades e competências, conhecendo também os possíveis caminhos profissionais. Por meio da simulação de situações que os ajudem a trabalhar em equipe, em especial nas diversas situações profissionais. Também estudarão os direitos e deveres do mundo de trabalho e da vida em sociedade, além de fundamentos de planejamento de carreira e educação financeira.</p>
Temas	<p>Características e Tendências do Mundo do Trabalho Atual; Setores e Segmentos de Atuação Profissional na Área de Desenvolvimento de Sistemas; Planejamento de Carreira e Preparação Profissional; Autoconhecimento e Orientação Profissional: Identificação de Habilidades, Interesses e Valores Pessoais; Preparação para Oportunidades de Trabalho; Finanças Pessoais: Uso Consciente e Eficiente dos Recursos Financeiros; Iniciação Empreendedora: Etapas Iniciais do Processo Empreendedor e Elaboração de um Plano de Negócios Simplificado.</p>
Competências técnicas	<p>Criar um plano estruturado para a progressão profissional, considerando objetivos de longo prazo, metas intermediárias e estratégias para alcançá-las; Realizar pesquisa e análise do mercado de trabalho, identificando tendências, demandas, oportunidades e áreas de crescimento; Elaborar um currículo bem organizado, destacando habilidades, experiências e conquistas relevantes para as oportunidades de carreira desejadas; Aplicar técnicas de narrativa pessoal para entrevistas de emprego e para a construção de rede de contatos profissionais; Usar plataformas <i>on-line</i> de busca de emprego, redes sociais profissionais, <i>sites</i> de recrutamento e outras ferramentas relevantes para</p>

	<p>encontrar oportunidades de trabalho; Usar ferramentas digitais para gerenciamento do tempo e melhoria da produtividade; Administrar as finanças pessoais de forma eficaz, incluindo orçamento, planejamento de gastos, poupança e investimentos, a fim de alcançar estabilidade financeira e tomar decisões financeiras informadas.</p>
<p>Competências socioemocionais</p>	<p>Reconhecer e gerenciar emoções próprias e de outras pessoas, mantendo o equilíbrio emocional em situações desafiadoras; Trabalhar em equipe, compartilhando conhecimentos, contribuindo com ideias e colaborando para alcançar objetivos comuns; Avaliar diferentes pontos de vista, questionando pressupostos e tomando decisões fundamentadas; Identificar e analisar problemas, desenvolver alternativas e implementar soluções eficazes durante a execução de um projeto; Saber ouvir ativamente, expressar-se de forma clara e assertiva, e construir relacionamentos sólidos; Agir com ética e integridade pessoal no ambiente de trabalho, demonstrando responsabilidade e honestidade em todas as interações.</p>
<p>Bibliografia básica:</p> <p>DUTRA, Joel Souza. <i>Gestão de carreiras: a pessoa, a organização e as oportunidades</i>. 2. ed. São Paulo: Atlas, 2017.</p> <p>GOLD, Miriam. <i>Gestão de carreira: como ser o protagonista de sua própria história</i>. São Paulo: Saraiva, 2019.</p> <p>OLIVEIRA, Djalma de Pinho Rebouças de. <i>Empreendedorismo: vocação, capacitação e atuação direcionadas para o plano de negócios</i>. São Paulo: Atlas, 2014.</p> <p>PAWLEWSKI, Sarah. <i>Carreiras: guia ilustrado para escolher a profissão certa</i>. São Paulo: Senac São Paulo, 2017.</p> <p>VIEIRA, Paulo; SILVA, Deibson. <i>Decifre seu talento: guia prático para acertar na sua escolha profissional</i>. São Paulo: Gente, 2020.</p>	

Bibliografia complementar:

CARNEGIE, Dale. *Como fazer sua (próxima) carreira decolar*. São Paulo: BestSeller, 2021.

DOLABELA, Fernando. *O segredo de Luísa: uma ideia e uma paixão: como nasce o empreendedor e se cria uma empresa*. São Paulo: Editora Sextante, 2023.

SINEK, Simon; MEAD, David; DOCKER, Peter. *Encontre seu porquê: um guia prático para descobrir o seu propósito e o de sua equipe*. São Paulo: Sextante, 2018.

TERRA, Eberson. *Carreiras exponenciais: torne-se o protagonista da sua própria jornada profissional e multiplique suas oportunidades na era digital*. São Paulo: Alta Books, 2021. 1 v.

Componente curricular	Carga horária (h)
Programação Mobile	120
Descrição	<p>Programação Mobile tem como objetivo abordar: mercado de aplicações móveis, celulares, <i>smartphones</i> e <i>tablets</i>; sistemas operacionais Android, iOS, Symbian, RIM, Windows Mobile e Handset Alliance; exercício de aplicação com interface, menu, diálogo e eventos em Android; plataformas de desenvolvimento para aplicações móveis: J2ME, Android, XCode e outros; recursos de imagens e <i>strings</i> em Android; localização da aplicação; tópicos de Java para aplicações móveis; conhecimentos na linguagem de programação Kotlin; API do Android; mensagens e seus filtros em Android; protocolos sem fio; dispositivos, componentes e acessórios de equipamentos móveis; armazenamento de dados e repositórios de conteúdo em Android; arquitetura Android, <i>framework</i>, bibliotecas; Android Market; exercício de aplicação com armazenamento de dados com Android; ambiente de programação: IDE Eclipse para Android; ferramentas, emulação, testes, Monkey; IDE Android Studio; exemplos de aplicação; mídias em Android (gráficos, áudio, vídeo e mapas); componentes Android: atividades, serviços,</p>

	receptores, repositório de conteúdo, mensagens, arquivo de manifesto; compatibilidade Android com dispositivos e telas; componentes Android: filtros de mensagem, processos e <i>threads</i> ; ciclo de vida; aplicações para <i>web</i> e Android; interface com usuário em Android (leiautes, <i>widgets</i> , eventos de interface, menus, assinatura e publicação de aplicações Android) e interface com usuário em Android (notificação, estilos, temas e visualizações).
Temas	Aplicações Móveis; Armazenamento de Dados e Repositórios; Interfaces com o Usuário e Publicação nas <i>Stores</i> .
Competências técnicas	Levantar e dimensionar requisitos específicos do negócio, adequando às funcionalidades do sistema; Desenvolver soluções de <i>software</i> (<i>back-end</i> , <i>front-end</i> e <i>full-stack</i>), utilizando técnicas, métodos, ferramentas e linguagens de programação diversas; Prestar apoio técnico na elaboração da documentação de sistemas.
Competências socioemocionais	Agir com curiosidade e criatividade na busca das melhores soluções em Programação Mobile; Identificar e analisar problemas, desenvolver alternativas e implementar soluções eficazes durante a execução de um projeto; Ter comprometimento com as demandas; Trabalhar em equipe, compartilhando conhecimentos, contribuindo com ideias e colaborando para alcançar objetivos comuns; Ter virtudes empreendedoras visando contribuir para os resultados e melhorias; Reconhecer e gerenciar emoções próprias e de outras pessoas, mantendo o equilíbrio emocional em situações desafiadoras; Ter resiliência diante de situações adversas.
<p>Bibliografia básica:</p> <p>DEITEL, P.; DEITEL, H. <i>Android for Programmers: an App-Driven Approach</i>. Pearson, 2018.</p> <p>HORTON, John. <i>Android Programming for Beginners</i>. Birmingham: Packt Publishing, 2021.</p>	

Bibliografia complementar:

MEDEIROS, R.; COSTA, F. *Desenvolvimento de aplicações Android com Kotlin*. Novatec Editora, 2019.

HARBOUR, Jonathan S. *Programação de games com Java*. São Paulo: Cengage Learning, 2010.

DOBRYCHTOP, Erik leger et al. *Aplicativos para Android*. São Paulo: Editora Viena, 2022. CIONDER, Shane; DARCEY, Lauren. *Desenvolvimento de aplicativos Wireless para Android, vol. 1: Fundamentos do Android*. 3ª ed. Rio de Janeiro: Ciência Moderna, 2012.

VASILIC, Milos. *Mastering Android Development with Kotlin: Deep Dive into the World of Android to Create Robust Applications with Kotlin*. Birmingham: Packt Publishing, 2017.

Componente curricular	Carga horária (h)
Banco de Dados NoSQL (Não Relacionais)	120
Descrição	<p>Banco de Dados NoSQL (não relacionais) tem como objetivo abordar: bancos de dados chave-valor: fundamentos de bancos de dados chave-valor; Redis (instalação e configuração); Redis (manipulando objetos) e Redis (criando e administrando instâncias).</p> <p>Bancos de dados orientados a colunas: fundamentos de bancos de dados orientados a colunas; Cassandra (instalação e configuração); Cassandra (criando <i>keyspaces</i> e <i>column families</i>) e Cassandra (manipulando registros).</p> <p>Bancos de dados orientados a documentos: fundamentos de bancos de dados orientados a documentos; MongoDB (instalação e configuração); MongoDB (criando, alterando e removendo documentos) e MongoDB (trabalhando com documentos).</p> <p>Bancos de dados orientados a grafos: fundamentos de bancos de dados orientados a grafos; Neo4j (instalação e configuração); Neo4j (criando, editando e apagando relacionamentos) e Neo4j (usando o Cypher para realizar buscas).</p>

Temas	Banco de Dados Chave-Valor; Banco de Dados Orientado a Colunas; Banco de Dados Orientado a Documentos; Banco de Dados Orientado a Grafos.
Competências técnicas	Desenvolver soluções de <i>software</i> (<i>back-end</i> , <i>front-end</i> e <i>full-stack</i>) utilizando técnicas, métodos, ferramentas e linguagens de programação diversas.
Competências socioemocionais	Agir com curiosidade e criatividade na busca das melhores soluções em banco de dados; Saber ouvir ativamente, expressar-se de forma clara e assertiva, e construir relacionamentos sólidos.
Bibliografia básica:	
HECHT, R. D. <i>MongoDB in Action</i> . 2 nd ed. Shelter Island: Manning Publications, 2019.	
CARLSON, Josiah. <i>Redis in Action</i> . 1st ed. Shelter Island: Manning Publications, 2016.	
PANIZ, David. <i>NoSQL: como armazenar os dados de uma aplicação moderna</i> . São Paulo: Casa do Código, 2016.	
Bibliografia complementar:	
CATTELL, R. G. <i>Scalable SQL and NoSQL Data Stores</i> . ACM SIGMOD Record, v. 39, n. 4, p. 12-27, 2011.	
KLEPPMANN, M. <i>Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems</i> . 1st ed. Sebastopol: O'Reilly Media, 2017.	
LAKSHMAN, A.; MALIK, P. <i>Cassandra: a Decentralized Structured Storage System</i> . ACM SIGOPS Operating Systems Review, v. 44, n. 2, p. 35-40, 2010.	
FOWLER, A.. <i>NoSQL for Dummies</i> . Hoboken: Wiley Publishing, 2015.	
HOWS, David; PLUGGE, Eelco; MEMBREY, Peter. <i>Introdução ao MongoDB</i> . São Paulo: Novatec Editora, 2022.	

Componente curricular	Carga horária (h)
Programação <i>Back-End</i>	90
Descrição	Programação <i>Back-End</i> tem como objetivo abordar: definição e linguagens principais; papel do <i>Back-End</i>

	no desenvolvimento web; arquitetura cliente-servidor; protocolo HTTP; linguagens <i>Back-End</i> (C#, Node.js, Ruby, Python); servidores web (conceitos básicos, Apache, Nginx); APIs (conceitos básicos, RESTful, JSON e XML); segurança <i>Back-End</i> [SQL Injection, Cross-site scripting (XSS)]; Frameworks <i>Back-End</i> : (NET, Spring, Node, Ruby on Rails, Django); integração com serviços externos (API de pagamento, API de geolocalização) e DevOps e Deploy (Git, Docker, AWS e Azure).
Temas	Introdução à Programação <i>Back-End</i> e Conceitos de Desenvolvimento para <i>Web</i> ; Linguagens de Programação <i>Back-End</i> ; Servidores <i>Web</i> , APIs e Segurança das Aplicações; Frameworks <i>Back-End</i> e Integração com Serviços Externos.
Competências técnicas	Codificar aplicações e rotinas mediante linguagens de programação Python, C#, Javascript e Java, sempre utilizando <i>frameworks</i> atuais; Migrar sistemas, implementando rotinas e estruturas de dados mais eficazes; Conhecer <i>frameworks</i> de desenvolvimento ágeis, utilizando tecnologias de CI e CD que trabalham na segurança do ambiente funcional e entregas divididas em partes que agregam valor ao negócio de forma rápida.
Competências socioemocionais	Identificar e analisar problemas, desenvolver alternativas e implementar soluções eficazes durante a execução de um projeto; Ter virtudes empreendedoras visando contribuir para os resultados e melhorias; Trabalhar em equipe, compartilhando conhecimentos, contribuindo com ideias e colaborando para alcançar objetivos comuns; Ter resiliência diante de situações adversas.
<p>Bibliografia básica:</p> <p>OLIVEIRA, C. R. <i>Princípios de arquitetura de software: Back-End</i>. Editora MNO, 2023.</p> <p>POWERS, Shelley. <i>Aprendendo Node: usando JavaScript no servidor</i>. São Paulo: Novatec Editora, 2022. PEREIRA, Caio Ribeiro. <i>Construindo APIs REST com Node.js</i>. São Paulo: Casa do Código, 2016.</p>	

Bibliografia complementar:

MACIEL, Francisco Marcelo de Barros. *Python e Django: desenvolvimento WEB moderno e ágil*. Rio de Janeiro: Alta Books, 2021. DOUGLAS, Michael; MARABESI, Matheus. *Aprendendo Laravel: o framework PHP dos artesãos da web*. São Paulo: Novatec Editora, 2017.

SILVA, Delcione Lopes da. *Framework PHP Laravel 8 & AJAX*. São Paulo: [Editora não informada], 2022. PEREIRA, F. S. *Introdução ao framework Spring Boot*. Editora PQR, 2022.

LAWSON, Jeff. *Pergunte ao desenvolvedor*. São Paulo: Benvirá, 2022.

Componente curricular		Carga horária (h)
Programação Front-End		90
Descrição	<p>Programação <i>Front-End</i> tem como objetivo abordar: definição e linguagens principais; papel do <i>Front-End</i> no desenvolvimento <i>web</i>; conceitos básicos de desenvolvimento <i>web</i> (arquitetura cliente-servidor, protocolo HTTP); linguagens de marcação (HTML) e estilização (CSS); linguagem de programação <i>Front-End</i> (JavaScript); <i>frameworks Front-End</i> (Angular, React e Vue.js); estilização de páginas <i>web</i> (conceitos básicos, Bootstrap); testes automatizados; desenvolvimento responsivo; ferramentas de desenvolvimento <i>Front-End</i> (NPM, Gulp e Webpack); performance e acessibilidade (otimização de imagens e ARIA); integração com serviços externos (API de mapas e API de autenticação); DevOps e <i>Deploy</i> [Git, Continuous Integration e Continuous Deployment (CI/CD)].</p>	
Temas	<p>Introdução à Programação <i>Front-End</i> e Conceitos de Desenvolvimento para <i>Web</i>; Linguagens de Programação <i>Front-End</i>; Estilização de Páginas, Ferramentas de Desenvolvimento e Performance; Integração com Serviços Externos e Esteiras de Entrega.</p>	

<p>Competências técnicas</p>	<p>Codificar aplicações e rotinas mediante linguagens de programação Python, C#, Javascript e Java, sempre utilizando <i>frameworks</i> atuais e migrar sistemas, implementando rotinas e estruturas de dados mais eficazes.</p>
<p>Competências socioemocionais</p>	<p>Agir com curiosidade e criatividade na busca das melhores soluções em Programação <i>Front-End</i>; Identificar e analisar problemas, desenvolver alternativas e implementar soluções eficazes durante a execução de um projeto; Ter virtudes empreendedoras visando contribuir para os resultados e melhorias; Trabalhar em equipe, compartilhando conhecimentos, contribuindo com ideias e colaborando para alcançar objetivos comuns; Ter resiliência diante de situações adversas.</p>
<p>Bibliografia básica:</p> <p>DUCKETT, Jon. <i>HTML and CSS: Design and Build Websites</i>. 2nd ed. Indianapolis: John Wiley & Sons, 2014.</p> <p>FLANAGAN, David. <i>JavaScript: The Definitive Guide: Activate Your Web Pages</i>. 7th ed. Sebastopol: O'Reilly Media, 2020.</p> <p>POWERS, Shelley. <i>Learning Node.js: a Hands-On Guide to Building Web Applications in JavaScript</i>. 2nd ed. Sebastopol: O'Reilly Media, 2018.</p>	
<p>Bibliografia complementar:</p> <p>ULLMAN, Larry. <i>Modern JavaScript: Develop and Design</i>. New York: Peachpit Press, 2012.</p> <p>PASSAGLIA, Andrea. <i>Vue.js 2 Cookbook: Build Modern, Interactive Web Applications with Vue.js</i>. Birmingham: Packt Publishing, 2017.</p> <p>SESHADRI, Shyam. <i>AngularJS: Up and Running: Enhanced Productivity with Structured Web Apps</i>. Sebastopol: O'Reilly Media, 2014.</p> <p>MEAD, Andrew. <i>The Complete JavaScript Course 2021: From Zero to Expert!</i>. Udemy, 2021.</p> <p>WIRFS-BROCK, Rebecca; MCKEAN, Alan. <i>Object Design: Roles, Responsibilities, and Collaborations</i>. Reading: Addison-Wesley Professional, 2002.</p>	

Componente curricular	Carga horária (h)
Modelagem e Desenvolvimento de Banco de Dados	120
Descrição	<p>Modelagem e Desenvolvimento de Banco de Dados tem como objetivo abordar: Introdução a Banco de Dados (conceito de dados, informações, componentes de um sistema de banco de dados, tipos de banco de dados); modelagem de dados, levantamento de dados, técnicas para identificação das necessidades do usuário, definições de entidades e atributos, definições de relacionamentos; cardinalidade mínima e máxima; grau de relacionamento, generalização e especialização; entidade associativa; banco de dados relacional; chaves primária e estrangeira; normalização (1ª, 2ª e 3ª forma normal), dicionário de dados; álgebra relacional (operações de união, intersecção, diferença, produto cartesiano, projeção, seleção e junção); comando SELECT do SQL; linguagem SQL (DDL, DML); SGBD relacional; SGBD Oracle (interface SQL Plus); criação e alteração de tabelas, tipos de dados, <i>constraints</i>, DROP TABLE; integridade referencial (<i>constraint</i> de <i>foreign key</i>), ALTER TABLE, INSERT, UPDATE, DELETE, COMMIT, ROLLBACK, SELECT (cláusulas DISTINCT, ORDER BY, WHERE com operadores lógicos, relacionais e aritméticos); funções práticas em laboratório (agrupamento, numéricas, caracteres, datas); cláusulas GROUP BY, HAVING, SUB-QUERIES, JOIN (INNER, OUTER, SELF); operações da álgebra relacional em laboratório (UNION, INTERSECT, MINUS), VIEWS.</p>
Temas	<p>Levantamento de Dados e Conceitos Fundamentais; Tipos de Relacionamentos; Formas Normais; Comandos SQL (CRUD); DDL X DML; Cláusulas Select; SGBD Relacional.</p>
Competências técnicas	<p>Levantar e dimensionar requisitos específicos do negócio, adequando-os às funcionalidades do sistema; Desenvolver soluções de <i>software</i> (<i>back-end</i>, <i>front-end</i> e <i>full-stack</i>), utilizando técnicas, métodos, ferramentas e linguagens de programação</p>

	diversas; Executar manutenção de programas.
Competências socioemocionais	Ter comprometimento com as demandas; Trabalhar em equipe, compartilhando conhecimentos, contribuindo com ideias e colaborando para alcançar objetivos comuns; Agir com curiosidade e criatividade na busca das melhores soluções para modelagem e desenvolvimento de banco de dados.
<p>Bibliografia básica:</p> <p>ELMASRI, Ramez; NAVATHE, Shamkant B. <i>Sistemas de banco de dados</i>. 7. ed. São Paulo: Pearson Education do Brasil, 2018.</p> <p>PRICE, Jason. <i>Oracle Database 11g SQL</i>. Porto Alegre: Bookman, 2009.</p> <p>SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. <i>Sistema de banco de dados</i>. 7. ed. Rio de Janeiro: LTC, 2018.</p>	
<p>Bibliografia complementar:</p> <p>CARDOSO, Virginia M; CARDOSO, Gisele C. <i>Linguagem SQL, fundamentos e práticas</i>. São Paulo: Saraiva Tec, 2013.</p> <p>DATE, Christopher J. <i>Introdução a sistemas de bancos de dados</i>. 8 ed. Rio de Janeiro: Elsevier, 2004.</p> <p>HEUSER, Carlos Alberto. <i>Projeto de banco de dados</i>. 6. ed. Porto Alegre: Bookman, 2008.</p> <p>MACHADO, Felipe Nery R. <i>Banco de dados: projeto e implementação</i>. 4. ed. São José dos Campos: Érica, 2020. SILVA, Robson Soares. <i>Oracle 10g express edition: guia de instalação, configuração e administração com implementação PL/SQL relacional e objeto-relacional</i>. São Paulo: Érica, 2009.</p>	

Componente curricular	Carga horária (h)
Projeto Multidisciplinar	90
Descrição	Projeto Multidisciplinar tem como objetivo mobilizar, por meio do desenvolvimento de projetos, os componentes curriculares deste curso técnico, oferecendo a oportunidade de colocar em prática competências técnicas e socioemocionais de forma articulada e contextualizada.

<p>Temas</p>	<p>Introdução ao Projeto Multidisciplinar: Etapas do Processo de um Projeto Multidisciplinar; Definição do Projeto e Início das Pesquisas; Planejamento e Organização: Elaboração do Plano de Trabalho; Desenvolvimento e Acompanhamento do Projeto com Uso de Ferramentas de Gestão de Projetos; Preparação da Apresentação e Discurso do Projeto; Apresentação do Projeto.</p>
<p>Competências técnicas</p>	<p>Planejar, executar e controlar projetos, aplicando os princípios e as práticas do gerenciamento de projetos; Avaliar a viabilidade técnica, econômica e operacional de projetos e inovações; Realizar pesquisa de mercado, coletar e analisar dados relevantes para embasar decisões estratégicas; Gerir os recursos disponíveis de forma eficiente, como orçamento, materiais, equipe e tempo, visando otimizar a execução do projeto; Criar protótipos de produtos, serviços ou processos, e realizar testes para validar sua viabilidade e coletar <i>feedback</i> dos usuários.</p>
<p>Competências socioemocionais</p>	<p>Reconhecer e gerenciar emoções próprias e de outras pessoas, mantendo o equilíbrio emocional em situações desafiadoras; Trabalhar em equipe, compartilhando conhecimentos, contribuindo com ideias e colaborando para alcançar objetivos comuns; Avaliar diferentes pontos de vista, questionando pressupostos e tomando decisões fundamentadas; Identificar e analisar problemas, desenvolver alternativas e implementar soluções eficazes durante a execução de um projeto; Saber ouvir ativamente, expressar-se de forma clara e assertiva, e construir relacionamentos sólidos; Agir com ética e integridade pessoal no ambiente de trabalho, demonstrando responsabilidade e honestidade em todas as interações; Realizar autogerenciamento e gestão do tempo.</p>
<p>Bibliografia básica:</p> <p>FIGUEIREDO, Paulo N. <i>Gestão da inovação: conceitos, métricas e experiências de empresas no Brasil</i>. 2. ed. São Paulo: Atlas, 2015.</p> <p>KERZNER, Harold. <i>Gerenciamento de projetos: uma abordagem sistêmica para planejamento, programação e controle</i>. São Paulo: Blucher, 2021.</p> <p>MAXIMIANO, Antonio Cesar Amaru. <i>Administração de projetos: como</i></p>	

transformar ideias em resultados. 5. ed. São Paulo: Atlas, 2014.

OSTERWALDER, Alexander; PIGNEUR, Yves. *Business Model Generation: inovação em modelos de negócios*. Rio de Janeiro: Alta Books, 2018.

PINVIDIC, Brant. *O pitch de 3 minutos: fale menos e consiga mais em qualquer apresentação*. São José dos Campos: Benvirá, 2020.

Bibliografia complementar:

BESSANT, John; TIDD, Joe. *Inovação e empreendedorismo*. 3. ed. São Paulo: Bookman, 2019.

MALONE, Michael S.; ISMAIL, Salim; VAN GEEST, Yuri. *Organizações exponenciais: por que elas são 10 vezes melhores, mais rápidas e mais baratas que a sua (e o que fazer a respeito)*. São Paulo: Alta Books, 2019.

PMI. *Um guia do conhecimento em gerenciamento de projetos (Guia PMBOK)*. 7. ed. Project Management Institute, 2021.

SCHERER, Felipe Ost; CARLOMAGNO, Maximiliano Selistre. *Gestão da inovação na prática*. 2. ed. São Paulo: Atlas, 2016.

4.3. Descrição sobre como trabalhar os componentes curriculares comuns a todos os cursos ofertados pela Secretaria da Educação do Estado de São Paulo

4.3.1. Carreira e Competências para o Mercado de Trabalho

O componente curricular Carreira e Competências para o Mercado de Trabalho será trabalhado de forma abrangente e prática, combinando teoria, discussões, estudos de caso e atividades práticas. Ele terá uma abordagem participativa, que envolverá os estudantes de forma ativa no processo de aprendizagem. A seguir, estão algumas estratégias de ensino que podem ser adotadas:

- Aulas expositivas: para apresentar conceitos teóricos, fundamentos e melhores práticas relacionadas à carreira e competência para o trabalho.
- Atividades em grupo: para estimular a colaboração e o compartilhamento de ideias entre os participantes. Isso permitirá a troca de experiências, a ampliação da rede de contatos e a criação de parcerias.
- Debates e discussões: para explorar diferentes perspectivas sobre questões relacionadas à carreira e competências para o trabalho. Os participantes serão incentivados a expressar suas opiniões e argumentar com base em fatos e evidências.
- Exercícios práticos: para que os participantes apliquem os conceitos aprendidos durante o módulo e treinem as competências técnicas.

- Utilização de ferramentas específicas para a evolução do estudante, tais como: guias para consulta de carreiras e profissões, remuneração na carreira/profissão escolhida, análise do ambiente público, privado e terceiro setor, além da realização de testes vocacionais, como 16 personalidades (MBTI), testes sabotadores e teste de coeficiente de inteligência positiva, Matriz SWOT pessoal, Ikigai (propósito), Business Model You | Modelo de Negócios Pessoal e Metodologia Star.
- Palestras e *workshops*: profissionais especializados em áreas específicas do empreendedorismo, como finanças, *marketing*, gestão de operações e inovação, poderão ser convidados para ministrar palestras e *workshops*. Essas atividades práticas permitirão que os participantes obtenham conhecimentos aprofundados em áreas específicas e aprendam com a experiência de profissionais do mercado.
- Visitas a empresas: locais ou incubadoras de negócios para que os participantes possam conhecer de perto empreendedores e *startups* em funcionamento. Isso proporcionará uma visão prática do ambiente empresarial, além de promover *networking* e inspiração.
- Avaliações e *feedback*: avaliações periódicas para verificar o progresso dos participantes e sua compreensão dos conceitos e práticas abordadas. O *feedback* constante será fornecido para orientar o desenvolvimento individual e identificar áreas de melhoria.
- Eventos e competições: oportunidade de participar de eventos, como feiras de empreendedorismo, competições de *startups* ou apresentações de *pitch*, onde poderão colocar em prática suas habilidades de comunicação e apresentação, além de receber *feedback* e visibilidade para seus projetos ou ideias.
- Mentoria e orientação individual: *feedback* individualizado dos instrutores para ajudar estudantes a identificar áreas de melhoria, fortalecer suas competências e desenvolver um plano de ação personalizado para suas carreiras.

4.3.2. Projeto Multidisciplinar

O componente curricular Projeto Multidisciplinar terá como ponto de partida situações-problema da realidade profissional, onde os estudantes desenvolverão, ao longo do segundo ano de curso (3ª série do Ensino Médio), todas as fases de um projeto prático: a) Problema/desafio; b) Desenvolvimento; c) Finalização.

Na fase “Problema/desafio”, será definido o tema, tendo em vista uma situação-problema real identificada no campo de atuação-profissional. Na fase “Desenvolvimento”, serão colocadas em prática as competências técnicas de forma articulada e contextualizada para o desenvolvimento do projeto propriamente dito, por meio do uso de ferramentas de gestão de projetos. E na “Finalização”, os estudantes terão a oportunidade de apresentar seus projetos para profissionais das áreas de seu curso e para a comunidade escolar.

As situações-problema e os projetos (fictícios ou reais) desafiarão os estudantes a aplicar os conhecimentos adquiridos no curso e darão a eles a oportunidade de vivenciar as etapas do gerenciamento de projetos e lidar com situações práticas.

O componente será trabalhado de forma teórica e prática, proporcionando aos estudantes uma compreensão aprofundada dos conceitos, princípios e práticas relacionadas à gestão de projetos e à promoção da inovação. O componente será estruturado em aulas, atividades individuais e em grupo, estudos de caso, projetos práticos e discussões em sala de aula.

Durante as aulas teóricas, os estudantes terão a oportunidade de aprender os fundamentos do gerenciamento de projetos, incluindo conceitos-chave, metodologias e ferramentas utilizadas para planejar, executar e controlar projetos. Serão abordados tópicos como ciclo de vida do projeto, estrutura analítica do projeto, cronograma, gerenciamento de riscos, entre outros.

A inovação será um elemento central do curso, sendo abordada de forma transversal em todas as atividades, uma vez que para o desenvolvimento dos projetos, a partir das diferentes situações-problema propostas, os estudantes farão uso das competências e habilidades desenvolvidas em todos os componentes curriculares do curso. Os estudantes serão expostos a diferentes metodologias e abordagens de inovação, como Design Thinking, Lean Startup, prototipagem, entre outras. Eles terão a chance de explorar técnicas de geração de ideias, desenvolvimento de protótipos e validação de conceitos.

Ao longo do componente curricular, também serão discutidos casos de sucesso de projetos e inovações em diversas áreas, proporcionando aos estudantes *insights* e aprendizados práticos. A interação entre os estudantes, por meio de debates e trabalhos em grupo, será incentivada para promover a troca de experiências e o aprendizado colaborativo.

Além disso, aspectos sociais, ambientais e éticos serão considerados ao abordar a inovação e os projetos, levando em conta a sustentabilidade e a responsabilidade social. Os Objetivos de Desenvolvimento Sustentável (ODS) da

ONU também podem ser integrados nas discussões e nos projetos, visando promover soluções alinhadas com os princípios da sustentabilidade.

Atividades de *feedback* e avaliação: os participantes terão a oportunidade de receber *feedback* sobre seu desempenho e progresso durante o módulo. Isso pode incluir avaliação de trabalhos individuais ou em grupo, apresentações, projetos práticos, entre outros. O *feedback* fornecerá *insights* para o aprimoramento contínuo e a consolidação do aprendizado.

Essas ações garantirão uma abordagem holística no ensino do gerenciamento de projetos, combinando teoria, prática, interação e *feedback* para que os participantes possam adquirir as competências necessárias e desenvolver uma base sólida nessa área.

5. Critérios de aproveitamento de estudos, conhecimentos e experiências anteriores

O aproveitamento de estudos, conhecimentos e experiências anteriores está em consonância com o artigo 46 da Resolução do Conselho Nacional de Educação/Conselho Pleno (CNE/CP) nº 1/2021, que dispõe sobre tais aproveitamentos, e está diretamente relacionado com o perfil profissional de conclusão da habilitação profissional.

Para fins de prosseguimento de estudos, o aproveitamento de competências adquiridas anteriormente pelo estudante por meio da educação formal/informal ou do trabalho será feito mediante avaliação realizada por comissão de professores designada pela Direção da Escola e atendendo os referenciais constantes de sua proposta pedagógica.

O aproveitamento de estudos, conhecimento e experiências anteriores, somente será feito para fins de prosseguimento de estudos, nunca para diplomação.

6. Critérios de avaliação

A avaliação de competências objetivando a expedição de diploma de conclusão seguirá as diretrizes definidas e indicadas pelo Ministério da Educação, bem como o contido na Deliberação CEE nº 107/2011.

Com o objetivo de analisar de forma ampla o desenvolvimento de competências em diferentes indivíduos e em variadas situações de aprendizagem, a avaliação se dará em um processo contínuo e permanente com a utilização de vários instrumentos, como produção textual, *quizzes*, projetos,

relatórios, autoavaliação, roteiros, pesquisas, portfólio, provas, entre outros. Essas avaliações permitem dar subsídios para as decisões e as orientações dos Conselhos de Classe e das Comissões de Professores acerca dos seguintes processos previstos: Classificação, Reclassificação, Recuperação contínua e Progressão parcial.

A Classificação para a série seguinte ou a conclusão do curso ocorrerá caso o estudante tenha obtido aproveitamento suficiente para a promoção – nota final maior ou igual a 5,0 – e a frequência mínima de 75% (setenta e cinco por cento) do total das horas efetivamente trabalhadas pela escola.

A Reclassificação permite ao estudante a matrícula em série diversa daquela em que está classificado. Para tanto, deverá haver parecer positivo elaborado por Comissão de Professores, indicada pela Direção da Escola.

A Recuperação contínua é destinada a estudantes cujo desenvolvimento das competências estabelecidas no Plano de Curso não está sendo identificado no decorrer das aulas. Nessa situação, os docentes devem promover orientações de ensino e atividades diversas adaptadas às dificuldades de cada estudante a fim de realizar a recuperação de aprendizagem dentro do próprio ano letivo.

A Progressão parcial permite ao estudante que obteve, ao final da 2ª série, aproveitamento insatisfatório (menor que 5,0) em até três componentes curriculares cursar, concomitantemente, a série seguinte com a realização das atividades apontadas nesta seção.

6.1. Avaliações contínuas

As avaliações contínuas em sala de aula são uma abordagem alternativa ao modelo tradicional de avaliação, que geralmente envolve apenas provas, trabalhos e exercícios de fixação ao final do ensino de todo o conteúdo. Já na avaliação contínua, busca-se analisar o progresso dos estudantes de forma mais abrangente, ao longo do tempo, considerando diferentes aspectos do aprendizado. A seguir, lista-se elementos-chave para a realização das avaliações contínuas:

- Diversidade de instrumentos de avaliação: para além da prova escrita, incorporar uma variedade de instrumentos e situações de avaliação, como projetos, apresentações, trabalhos em grupo, portfólios, debates, resolução de problemas, entre outros. Essa diversidade permite que os estudantes demonstrem seus conhecimentos e suas habilidades de maneira diversificada, abarcando preferências e aptidões individuais.

- *Feedback* regular: proporcionar aos estudantes retornos frequentes e imediatos sobre seu desempenho. Isso pode ser feito por meio de comentários escritos, discussões em sala de aula ou reuniões individuais com o professor. O *feedback* contínuo ajuda os estudantes a entenderem seus pontos fortes e fracos, permitindo-lhes fazer ajustes e melhorar ao longo do tempo.
- Acompanhamento do progresso individual: acompanhar o progresso de cada estudante ao longo do tempo. Os professores podem observar como os estudantes estão evoluindo em relação aos objetivos de aprendizagem e identificar áreas que precisam de maior atenção. Isso permite que o ensino seja personalizado e adaptado às necessidades individuais dos estudantes.
- Foco no processo de aprendizagem: sem se limitar à avaliação de resultados, as avaliações contínuas valorizam as conquistas dos estudantes ao longo do processo de aprendizagem. Os estudantes são incentivados a refletir sobre seu próprio progresso, a identificar estratégias eficazes de estudo e a aprender com os erros cometidos. Essa abordagem promove uma visão mais abrangente do aprendizado, além de encorajar a autorregulação e a metacognição.
- Participação ativa dos estudantes: envolver os estudantes em seu próprio processo de avaliação, o que pode incluir a definição de metas de aprendizagem individuais, a autorreflexão sobre seu desempenho e o monitoramento de seu progresso ao longo do tempo. Os estudantes são incentivados a tornarem-se mais responsáveis pelo seu próprio aprendizado.

No geral, as avaliações contínuas em sala de aula visam fornecer uma visão mais completa e precisa do desempenho dos estudantes, considerando uma variedade de fatores e habilidades. Essa abordagem, além de promover uma aprendizagem mais significativa e engajada, permite intervenções oportunas para apoiar o progresso dos estudantes.

6.2. Atividades práticas e de laboratório

A criação de projetos que estimulem a integração de elementos práticos na formação do estudante tem papel relevante na efetiva capacitação profissional.

Para garantir essa capacitação, a escola adota as atividades práticas e de laboratório, as quais permitem a interação direta com a realidade do mundo do trabalho, aperfeiçoando habilidades e atitudes inerentes ao processo de educação e formação profissional. Essa estratégia metodológica viabiliza que temáticas ensinadas sejam praticadas e vivenciadas, e que as dificuldades possam aflorar e

ser trabalhadas, no período de apreender a teoria a partir de situações práticas. Também atua como instrumento de incentivo, pois propicia ao estudante a observação e mensuração dos ganhos no processo de aprendizagem.

Para o alcance dos resultados esperados durante a execução das práticas, são necessárias ações de planejamento e organização, seja no local de trabalho, seja em laboratório (virtual ou físico). Por meio de um roteiro de apoio para a realização das atividades e do estabelecimento da correlação entre o objeto do aprendizado e a execução, o estudante deve fazer, registrar e relatar as práticas. Isso permitirá ao docente/tutor avaliar o nível de apreensão e assimilação da teoria e da aplicação de competências necessárias no exercício profissional.

Esse curso técnico prevê a realização de atividades práticas e de laboratório, que receberão uma pontuação para compor a média da disciplina.

7. Instalações e equipamentos

O ambiente das salas de aula, com infraestrutura adequada de iluminação, ventilação, mobiliário e acústica, qualifica a relação proporcionada aos professores e estudantes em tempo real, propiciando práticas pedagógicas que estimulam o estudante a estabelecer uma relação teórico-prática de alto nível.

7.1. Instalações

As salas possuem, além de mobiliários adequados, equipamento de projeção multimídia e acesso *wi-fi*, o que permite aos professores utilizarem diferentes recursos didáticos, de acordo com as competências e habilidades a serem desenvolvidas em cada aula. Essas multimídias possibilitam a realização de dinâmicas interativas entre estudante e professor, também por meio de seus *smartphones*, *tablets* e *notebooks*.

Para melhor desempenho didático-pedagógico, a escola oferece um Ambiente Virtual de Aprendizagem, o *Moodle*, um sistema de gerenciamento de aprendizado *on-line*. É uma plataforma de código aberto que permite criar, gerenciar e entregar cursos *on-line*, facilitando a interação entre professores e estudantes. O termo Moodle é uma sigla e significa Modular Object-Oriented Dynamic Learning Environment (Ambiente de Aprendizagem Dinâmico Modular Orientado a Objetos). O Moodle oferece ampla gama de recursos e ferramentas para facilitar o ensino e a aprendizagem *on-line*. Instrutores podem: criar cursos; adicionar recursos educacionais, como documentos, vídeos e questionários; configurar atividades interativas, como fóruns de discussão e tarefas; e

administrar avaliações *on-line*. Ele também facilita a aplicação de metodologias ativas de aprendizagem e o registro das tarefas exercidas em sala de aula, possibilitando ao professor organizar seu planejamento de aula e gerenciar o processo dos estudantes em ambiente virtual.

7.2. Equipamentos

Para o Curso Técnico em Desenvolvimento de Sistemas, o Laboratório de Informática deve ser utilizado de forma a atender as necessidades específicas de pesquisa e prática na utilização de *softwares* desse campo profissional, viabilizando o aprendizado em informática e a ampliação de conhecimentos no mundo do trabalho. Para utilizar um computador, basta o estudante agendar um horário que coincida com os horários de plantão dos professores de informática. No espaço, existem computadores de última geração, com acesso à internet por banda larga.

Com o objetivo de garantir a aprendizagem, estão previstas atividades pedagógicas em laboratório, seja físico, seja virtual, que podem ser realizadas pelo estudante na escola, sempre com acompanhamento de especialistas na área do curso para suporte didático e análise do aproveitamento.

Os laboratórios possuem rede lógica estruturada exclusiva para interligação entre os computadores e os equipamentos da sala. O laboratório específico de Redes de Computadores tem o seguinte descritivo:

- rede do tipo LAN;
- acesso à internet;
- arquitetura: modelo OSI;
- velocidade da rede: será 10/100/1000 Mbps;
- protocolo de transporte: TCP/IP;
- servidor;
- *rack*;
- *hubs* e *switches*;
- *patch panel*;
- roteadores *wireless*, com suporte 3G/4G;
- microcomputadores;
- estrutura com tomadas RJ45;
- testador de cabo;
- ferramentas de conectorização;
- cabos em geral;
- computadores com IDEs (Eclipse e Visual Studio Community);

- serviço de nuvem para estudantes (AWS ou Azure).

7.3. Tecnologias digitais imprescindíveis para a área

- linguagens de programação (C#, Python, JavaScript e Java);
- *frameworks* (.NET, Spring, Node, Angular e React);
- Inteligência Artificial (ChatBot, Deep Learning, Machine Learning, Lógica Fuzzy);
- Sistemas Gerenciadores de Banco de Dados (SQL);
- banco de dados não relacionais (MongoDB);
- versionamento de código (SCCM, com Git);
- sistemas de mensageria (Kafka, RabbitMQ);
- redes de computadores/computação em nuvem;
- segurança da informação.

8. Bibliografia

Será adotada, preferencialmente, biblioteca *on-line* e, adicionalmente, biblioteca física. Essas devem ser compostas por um volume de cada item da bibliografia básica para cada turma de 40 estudantes.

AQUILES, Alexandre; FERREIRA, Rodrigo. *Controlando versões com Git e GitHub*. São Paulo: Casa do Código, 2014.

BEER, Brent; BELL, Peter. *Introdução ao GitHub: um guia que não é técnico*. São Paulo: Novatec Editora, 2014.

BRASIL. Ministério da Educação. Conselho Nacional de Educação. *Resolução CNE/CP nº 1*, de 05 de janeiro de 2021. Brasília: MEC, 2021. Disponível em: <https://www.in.gov.br/en/web/dou/-/resolucao-cne/cp-n-1-de-5-de-janeiro-de-2021-297767578>. Acesso em: 10 nov. 2023.

CARLSON, Josiah. *Redis in Action*. 1st ed. Shelter Island: Manning Publications, 2016.

COPPIN, Ben. *Inteligência artificial*. Rio de Janeiro: LTC, 2013.

DEITEL, P.; DEITEL, H. *Android for Programmers: an App-Driven Approach*. Pearson, 2018.

DUCKETT, Jon. *HTML and CSS: Design and Build Websites*. 2nd ed. Indianapolis: John Wiley & Sons, 2014.

DUTRA, Joel Souza. *Gestão de carreiras: a pessoa, a organização e as oportunidades*. 2. ed. São Paulo: Atlas, 2017.

ELMASRI, Ramez; NAVATHE, Shamkant B. *Sistemas de banco de dados*. 7. ed. São Paulo: Pearson Education do Brasil, 2018.

FACELI, Katti *et al.* *Inteligência artificial: uma abordagem de aprendizado de máquina*. 2. ed. Rio de Janeiro: LTC, 2021.

FIGUEIREDO, Paulo N. *Gestão da inovação: conceitos, métricas e experiências de empresas no Brasil*. 2. ed. São Paulo: Atlas, 2015.

FLANAGAN, David. *JavaScript: The Definitive Guide: Activate Your Web Pages*. 7th ed. Sebastopol: O'Reilly Media, 2020.

FOWLER, Martin. *Agile Software Development: Principles, Patterns, and Practices*. Boston: Addison-Wesley Professional, 2002.

GABRIEL, Martha. *Inteligência Artificial - Do Zero ao Metaverso*. São Paulo: Editora Atlas, 2022.

GOLD, Miriam. *Gestão de carreira: como ser o protagonista de sua própria história*. São Paulo: Saraiva, 2019.

HECHT, R. D. *MongoDB in Action*. 2nd ed. Shelter Island: Manning Publications, 2019.

HORTON, John. *Android Programming for Beginners*. Birmingham: Packt Publishing, 2021.

KERZNER, Harold. *Gerenciamento de projetos: uma abordagem sistêmica para planejamento, programação e controle*. São Paulo: Blucher, 2021.

KUROSE, James F.; ROSS, Keith W. *Redes de computadores e a internet: Uma Abordagem Top-down*. 6. ed. São Paulo: Pearson, 2013.

MANZANO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo de. *Algoritmos: lógica para desenvolvimento de programação de computadores*. 29. ed. São Paulo: Érica, 2019.

MAXIMIANO, Antonio Cesar Amaru. *Administração de projetos: como transformar ideias em resultados*. 5. ed. São Paulo: Atlas, 2014.

OLIVEIRA, C. R. *Princípios de arquitetura de software: Back-End*. Editora MNO, 2023.

OLIVEIRA, Djalma de Pinho Rebouças de. *Empreendedorismo: vocação, capacitação e atuação direcionadas para o plano de negócios*. São Paulo: Atlas, 2014.

OSTERWALDER, Alexander; PIGNEUR, Yves. *Business Model Generation: inovação em modelos de negócios*. Rio de Janeiro: Alta Books, 2018.

- PANIZ, David. *NoSQL: como armazenar os dados de uma aplicação moderna*. São Paulo: Casa do Código, 2016.
- PAWLEWSKI, Sarah. *Carreiras: guia ilustrado para escolher a profissão certa*. São Paulo: Senac São Paulo, 2017.
- PINVIDIC, Brant. *O pitch de 3 minutos: fale menos e consiga mais em qualquer apresentação*. São José dos Campos: Benvirá, 2020.
- PIVA JUNIOR, Dilermando et al. *Algoritmos e programação de computadores*. 2. ed. São Paulo: LTC, 2019.
- POWERS, Shelley. *Aprendendo Node: usando JavaScript no servidor*. São Paulo: Novatec Editora, 2022. PEREIRA, Caio Ribeiro. *Construindo APIs REST com Node.js*. São Paulo: Casa do Código, 2016.
- POWERS, Shelley. *Learning Node.js: a Hands-On Guide to Building Web Applications in JavaScript*. 2nd ed. Sebastopol: O'Reilly Media, 2018.
- PRESSMAN, Roger S. *Engenharia de software: uma abordagem profissional*. 7. ed. Porto Alegre: AMGH, 2016.
- PRICE, Jason. *Oracle Database 11g SQL*. Porto Alegre: Bookman, 2009.
- SÃO PAULO. CONSELHO ESTADUAL DE EDUCAÇÃO. *Deliberação CEE/SP 107/2011, que dispõe sobre credenciamento de Instituições para avaliação de competências e expedição do diploma na educação profissional de nível médio, no Sistema de Ensino do Estado de São Paulo e dá outras providências*, 2011.
- SAUDATE, Alexandre. *SOA aplicado: integrando com web services e além*. São Paulo: Casa do Código, 2015.
- SCHWABER, Ken; SUTHERLAND, Jeff. *Guia do Scrum - um guia definitivo para o Scrum: as regras do jogo*. [atualizado em 2020]. Disponível em: <<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Portuguese-European.pdf>>. Acesso em: 15 nov. 2023.
- SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. *Sistema de banco de dados*. 7. ed. Rio de Janeiro: LTC, 2018.
- STALLINGS, William. *Criptografia e segurança de redes: princípios e práticas*. 6. ed. São Paulo: Pearson, 2014.
- TANENBAUM, Andrew S.; WETHERALL, David J. *Redes de computadores*. 5. ed. São Paulo: Pearson, 2012.
- VIEIRA, Paulo; SILVA, Deibson. *Decifre seu talento: guia prático para acertar na sua escolha profissional*. São Paulo: Gente, 2020.

9. Pessoal docente e técnico

As escolas estaduais, considerando seu perfil e sua missão, têm como premissa a contratação de docentes com experiência no mercado de trabalho, aderente às áreas de formação dos seus cursos superiores, além de sólida formação acadêmica.

É fundamental ter em sua equipe docente profissionais alinhados com as características da instituição, para que sejam alcançados os resultados esperados ao longo do curso. A seguir, um descritivo das especificações, conforme os componentes curriculares.

Antes disso, é importante mencionar que poderão ser admitidos, em caráter excepcional, profissionais com a seguinte ordem preferencial:

- Na falta de licenciados, os graduados na correspondente área profissional ou de estudos.
- Na falta de profissionais graduados em nível superior nas áreas específicas, profissionais graduados em outras áreas e que tenham comprovada experiência profissional na área do curso.
- Na falta de profissionais graduados, técnicos de nível médio na área do curso, com comprovada experiência profissional.
- Na falta de profissionais de nível técnico com comprovada experiência, outros reconhecidos por sua notória competência e, no mínimo, com Ensino Médio completo. Aos não licenciados, é propiciada formação docente em serviço.

Componente curricular	Especificações de formação/titulação/certificação
Lógica e Linguagem de Programação	<p>Graduação (bacharelado, licenciatura, tecnologia) em Análise e Desenvolvimento de Sistemas, Ciência da Computação, Computação, Engenharia da Computação, Engenharia de Sistemas, Engenharia de <i>Software</i>, Processamento de Dados ou áreas afins.</p> <p>Graduação (bacharelado, licenciatura, tecnologia) nas demais áreas com pós-</p>

	<p>graduação na área de Desenvolvimento de Sistemas ou Análise e Desenvolvimento de Sistemas.</p>
<p>Modelagem e Desenvolvimento de Banco de Dados</p>	<p>Graduação (bacharelado, licenciatura, tecnologia) em Banco de Dados, Ciência da Computação, Análise e Desenvolvimento de Sistemas, Computação, Engenharia da Computação, Engenharia de Sistemas, Engenharia de <i>Software</i>, Processamento de Dados ou áreas afins.</p> <p>Graduação (bacharelado, licenciatura, tecnologia) nas demais áreas com pós-graduação em Desenvolvimento de Sistemas ou Análise e Desenvolvimento de Sistemas.</p>
<p>Redes de Computadores e Segurança da Informação na Nuvem</p>	<p>Graduação (bacharelado, licenciatura, tecnologia) em Redes de Computadores, Tecnologia em Segurança da Informação, Ciência da Computação, Computação, Engenharia da Computação, Engenharia de Sistemas, Engenharia de <i>Software</i>, Processamento de Dados ou áreas afins.</p> <p>Graduação (bacharelado, licenciatura, tecnologia) nas demais áreas com pós-graduação em Desenvolvimento de Sistemas ou Análise e Desenvolvimento de Sistemas.</p>
<p>Processos de Desenvolvimento de <i>Software</i> e Metodologias Ágeis</p>	<p>Graduação (bacharelado, licenciatura, tecnologia) em Análise e Desenvolvimento de Sistemas, Ciência da Computação, Computação, Engenharia da Computação, Engenharia de Sistemas, Engenharia de <i>Software</i>, Processamento de Dados</p>

	<p>ou áreas afins.</p> <p>Graduação (bacharelado, licenciatura, tecnologia) nas demais áreas com pós-graduação em Desenvolvimento de Sistemas ou Análise e Desenvolvimento de Sistemas.</p>
Inteligência Artificial	<p>Graduação (bacharelado, licenciatura, tecnologia) em Análise e Desenvolvimento de Sistemas, Ciência da Computação, Computação, Engenharia da Computação, Engenharia de Sistemas, Engenharia de <i>Software</i>, Processamento de Dados, Estatística ou áreas afins.</p> <p>Graduação (bacharelado, licenciatura, tecnologia) em Desenvolvimento de Sistemas ou Análise e Desenvolvimento de Sistemas.</p>
Carreira e Competências para o Mercado de Trabalho	<p>Graduação (bacharelado, licenciatura, tecnologia) em Administração, Gestão, Negócios, Psicologia, Recursos Humanos ou áreas afins.</p> <p>Graduação (bacharelado, licenciatura, tecnologia) nas demais áreas com pós-graduação em Desenvolvimento de Sistemas, Psicologia ou Recursos Humanos.</p>
Programação Mobile	<p>Graduação (bacharelado, licenciatura, tecnologia) em Análise e Desenvolvimento de Sistemas, Ciência da Computação, Computação, Engenharia da Computação, Engenharia de Sistemas, Engenharia de <i>Software</i>, Processamento de Dados ou áreas afins.</p> <p>Graduação (bacharelado, licenciatura,</p>

	tecnologia) em Desenvolvimento de Sistemas ou Análise e Desenvolvimento de Sistemas.
Banco de Dados NoSQL (Não Relacionais)	<p>Graduação (bacharelado, licenciatura, tecnologia) em Banco de Dados, Ciência da Computação, Análise e Desenvolvimento de Sistemas; Computação, Engenharia da Computação, Engenharia de Sistemas, Engenharia de <i>Software</i>, Processamento de Dados ou áreas afins.</p> <p>Graduação (bacharelado, licenciatura, tecnologia) em Desenvolvimento de Sistemas ou Análise e Desenvolvimento de Sistemas.</p>
Programação <i>Back-End</i>	<p>Graduação (bacharelado, licenciatura, tecnologia) em Análise e Desenvolvimento de Sistemas, Ciência da Computação, Computação, Engenharia da Computação, Engenharia de Sistemas, Engenharia de <i>Software</i>, Processamento de Dados ou áreas afins.</p> <p>Graduação (bacharelado, licenciatura, tecnologia) em Desenvolvimento de Sistemas ou Análise e Desenvolvimento de Sistemas.</p>
Programação <i>Front-End</i>	<p>Graduação (bacharelado, licenciatura, tecnologia) em Análise e Desenvolvimento de Sistemas, Ciência da Computação, Computação, Engenharia da Computação, Engenharia de Sistemas, Engenharia de <i>Software</i>, Processamento de Dados ou áreas afins.</p> <p>Graduação (bacharelado, licenciatura,</p>

	tecnologia) em Desenvolvimento de Sistemas ou Análise e Desenvolvimento de Sistemas.
Versionamento de Código e Sistemas de Mensageria	<p>Graduação (bacharelado, licenciatura, tecnologia) em Análise e Desenvolvimento de Sistemas, Ciência da Computação, Computação, Engenharia da Computação, Engenharia de Sistemas, Engenharia de <i>Software</i>, Processamento de Dados ou áreas afins.</p> <p>Graduação (bacharelado, licenciatura, tecnologia) em Desenvolvimento de Sistemas ou Análise e Desenvolvimento de Sistemas.</p>
Projeto Multidisciplinar	<p>Graduação (bacharelado, licenciatura, tecnologia) em Administração, Gestão, Negócios ou áreas afins.</p> <p>Graduação (bacharelado, licenciatura, tecnologia) nas demais áreas com pós-graduação em Desenvolvimento de Sistemas ou Análise e Desenvolvimento de Sistemas.</p>

10. Certificados e diplomas

TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS

Ao estudante concluinte do curso, será conferido e expedido o diploma de Técnico em Desenvolvimento de Sistemas, satisfeitas as exigências relativas:

- ao cumprimento do currículo previsto para habilitação;
- à apresentação do certificado de conclusão do Ensino Médio ou equivalente.

Ao término do primeiro ano de curso, o estudante fará jus ao Certificado de Qualificação Profissional Técnica de Nível Médio de Auxiliar em Desenvolvimento de Sistemas.

Ao completar os dois anos de curso, com aproveitamento em todos os componentes curriculares, o estudante receberá o diploma de Técnico em Desenvolvimento de Sistemas, pertinente ao eixo tecnológico de “Informação e Comunicação”.

Por se tratar de curso na forma integrada ao Ensino Médio, o estudante receberá também certificado e histórico escolar referentes ao Ensino Médio.

O diploma e o certificado terão validade nacional, obedecendo a legislação vigente, a Lei Federal nº 12.605/2012, que determina às instituições de ensino públicas e privadas a empregarem a flexão de gênero para nomear profissão ou grau nos diplomas expedidos.

Quando se tratar de operação no modelo de parcerias para a execução deste plano, o diploma será expedido pela instituição de ensino parceira e o certificado de conclusão do Ensino Médio será expedido pela secretaria escolar.

11. Estágio supervisionado (não obrigatório)

O Curso Técnico em Desenvolvimento de Sistemas não exige o cumprimento de estágio supervisionado em sua organização curricular. Procedimentos didáticos voltados à prática profissional na área, como simulações, experiências e outras técnicas de ensino, permitirão a vivência dos estudantes em situações próximas à realidade do setor produtivo.

Fica a critério do estudante realizar estágio supervisionado, não sendo esta condição para a conclusão do curso. Quando realizado, as horas efetivamente cumpridas deverão constar do Histórico Escolar do Estudante. A escola acompanhará as atividades de estágio por meio de um Plano de Estágio, que deverá conter:

- sistemática de acompanhamento, controle e avaliação;
- justificativa;
- objetivos;
- metodologias;
- identificação do responsável pela orientação do estágio;
- definição do campo/área para a realização do estágio.

O estágio somente poderá ser realizado de maneira concomitante com o curso, ou seja, apenas enquanto o estudante estiver regularmente matriculado. Após a conclusão do curso, será vedada a realização de estágio supervisionado.

ANEXO 1 – MATRIZ CURRICULAR

CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS			
Ano	Componente curricular	Carga horária do componente (h)	Carga horária anual (h)
Ano 1 (2ª EM)	Lógica e Linguagem de Programação	120	570
	Versionamento de Código e Sistemas de Mensageria	90	
	Redes de Computadores e Segurança da Informação na Nuvem	90	
	Processos de Desenvolvimento de <i>Software</i> e Metodologias Ágeis	90	
	Inteligência Artificial	90	
	Carreira e Competências para o Mercado de Trabalho	90	
Ano 2 (3ª EM)	Programação Mobile	120	630
	Banco de Dados NoSQL (Não Relacionais)	120	
	Programação <i>Back-End</i>	90	
	Programação <i>Front-End</i>	90	
	Modelagem e Desenvolvimento de Banco de Dados	120	
	Projeto Multidisciplinar	90	

ANEXO 2 – PLANO E ORIENTAÇÕES PARA ESTÁGIO

O estágio supervisionado não é obrigatório para os estudantes do Curso Técnico em Desenvolvimento de Sistemas, no entanto é um ótimo complemento, pois proporciona a oportunidade de aplicar os conhecimentos teóricos em um ambiente de trabalho real.

Para elaborar um plano e orientações abrangentes para o estágio, é importante considerar os seguintes fatores:

1. Legislação e regulamentação:

Verificar as leis, diretrizes e regulamentos pertinentes ao estágio supervisionado, garantindo que o plano esteja em conformidade com as exigências legais e normativas. Isso inclui aspectos relacionados a carga horária, remuneração, seguro de acidentes pessoais, entre outros.

2. Parcerias institucionais:

Identificar possíveis parcerias com empresas, organizações ou instituições do setor empresarial para proporcionar oportunidades de estágio aos estudantes. Estabelecer critérios para seleção e acompanhamento das empresas parceiras, garantindo a qualidade das experiências de estágio.

3. Plano de atividades:

Definir um plano de atividades que englobe as diferentes áreas e funções de desenvolvimento de sistemas, permitindo que os estudantes vivenciem ampla gama de experiências durante o estágio. Isso pode incluir rotação entre as diferentes áreas de uma empresa.

4. Integração com o currículo:

Assegurar que o estágio esteja alinhado com os conteúdos e as competências previstos na matriz curricular do Curso Técnico em Desenvolvimento de Sistemas. Integrar o estágio com componentes curriculares específicos, projetos multidisciplinares ou outras atividades acadêmicas, promovendo uma conexão direta entre a teoria e a prática.

5. Orientação e suporte aos estagiários:

Estabelecer mecanismos eficazes de orientação e suporte aos estagiários, garantindo que tenham um supervisor designado na empresa, bem como um orientador acadêmico responsável por acompanhar seu progresso. Traçar diretrizes claras sobre as responsabilidades e expectativas de ambas as partes.

6. Avaliação do estágio:

Definir critérios e instrumentos de avaliação para medir o desempenho dos estagiários durante o período de estágio. Isso pode incluir relatórios de atividades, avaliações de desempenho, *feedback* dos supervisores da empresa e avaliação do orientador acadêmico.

É importante ressaltar que o plano e as orientações de estágio devem ser revisados periodicamente, considerando o *feedback* dos estudantes, supervisores e orientadores, visando aprimorar a qualidade das experiências de estágio e a garantir sua efetividade na formação dos futuros profissionais de Desenvolvimento de Sistemas.

ANEXO 3 – GLOSSÁRIO TERMOS/SIGLAS DE TI

Modelagem e Desenvolvimento de Banco de Dados

- DDL (*Data Definition Language*): Linguagem de Definição de Dados.
- DML (*Data Manipulation Language*): Linguagem de Manipulação de Dados.
- PK (*Primary Key*): Chave Primária.
- FK (*Foreign Key*): Chave Estrangeira.
- UK (*Unique Key*): Chave Única.
- CK (*Check Constraint*): Restrição de Verificação.
- NN (*Not Null*): Não Nulo.
- SGBD (Sistema Gerenciador de Banco de Dados): System Database Management System.
- SQL (*Structured Query Language*): Linguagem de Consulta Estruturada.
- ANSI (American National Standards Institute): Instituto Nacional de Padrões Americanos.
- Oracle®: Sistema de gerenciamento de banco de dados (SGBD) desenvolvido pela Oracle Corporation.
- SQL Plus®: Interface de linha de comando para interagir com o banco de dados Oracle.
- CREATE TABLE: Comando para criar uma tabela.
- DROP TABLE: Comando para excluir uma tabela.
- ALTER TABLE: Comando para alterar a estrutura de uma tabela existente.
- INSERT: Comando para inserir dados em uma tabela.
- UPDATE: Comando para atualizar dados em uma tabela.
- DELETE: Comando para excluir dados de uma tabela.
- COMMIT: Comando para confirmar as alterações feitas em um banco de dados.
- ROLLBACK: Comando para desfazer as alterações feitas em um banco de dados.
- SELECT: Comando para recuperar dados de um banco de dados.
- DISTINCT: Cláusula utilizada para retornar valores únicos em uma consulta.
- ORDER BY: Cláusula utilizada para ordenar os resultados de uma consulta.
- WHERE: Cláusula utilizada para filtrar os resultados de uma consulta.
- GROUP BY: Cláusula utilizada para agrupar os resultados de uma consulta.
- HAVING: Cláusula utilizada para filtrar grupos em uma consulta.
- OUTER JOIN: Tipo de junção que retorna registros correspondentes em ambas as tabelas, além dos registros que não possuem correspondência.

- SELF JOIN: Tipo de junção em que uma tabela é combinada com ela mesma.
- UNION: Operação que retorna a combinação de resultados de duas consultas.
- INTERSECT: Operação que retorna os resultados comuns a duas consultas.
- MINUS: Operação que retorna os resultados de uma consulta que não estão presentes em outra consulta.
- VIEWS: Visões ou visualizações, que são consultas salvas como objetos de banco de dados.

Redes de Computadores e Segurança da Informação

- OSI (*Open Systems Interconnection*): Interconexão de Sistemas Abertos.
- TCP/IP (*Transmission Control Protocol/Internet Protocol*): Protocolo de Controle de Transmissão/Protocolo de Internet.
- LAN (*Local Area Network*): Rede Local.
- IPv4 (*Internet Protocol Version 4*): Protocolo de Internet Versão 4.
- CIDR (*Classless InterDomain Routing*): Roteamento Interdomínio sem Classes.
- IPv6 (*Internet Protocol Version 6*): Protocolo de Internet Versão 6.
- NAT (*Network Address Translation*): Tradução de Endereço de Rede.
- RIP (*Routing Information Protocol*): Protocolo de Informação de Roteamento.
- OSPF (*Open Shortest Path First*): Open Shortest Path First.
- TCP (*Transmission Control Protocol*): Protocolo de Controle de Transmissão.
- UDP (*User Datagram Protocol*): Protocolo de Datagrama de Usuário.
- DNS (*Domain Name System*): Sistema de Nomes de Domínio.
- ITIL (*Information Technology Infrastructure Library*): Biblioteca de Infraestrutura de Tecnologia da Informação.
- NIST (*National Institute of Standards and Technology*): Instituto Nacional de Padrões e Tecnologia.
- CIS (*Center for Internet Security*): Centro de Segurança da Internet.
- FIPS (*Federal Information Processing Standards*): Padrões Federais de Processamento de Informações.
- ISO (*International Organization for Standardization*): Organização Internacional de Padronização.
- PCI (*Payment Card Industry*): Indústria de Cartões de Pagamento.
- SOX (*Sarbanes-Oxley Act*): Lei Sarbanes-Oxley.

- LGPD (Lei Geral de Proteção de Dados): Legislação brasileira que regula a proteção de dados pessoais.

Processos de Desenvolvimento de *Software* e Metodologias Ágeis

- UML (*Unified Modeling Language*): Linguagem de Modelagem Unificada.
- DDD (*Domain Driven Design*): *Design* Orientado a Domínio.
- CI (*Continuous Integration*): Integração Contínua.
- CD (*Continuous Delivery*): Entrega Contínua.
- Case (*Computer-Aided Software Engineering*): Engenharia de *Software* Assistida por Computador.
- XP (*Extreme Programming*): Programação Extrema.
- UP (*Unified Process*): Processo Unificado.
- Scrum: Metodologia ágil de gerenciamento de projetos.
- DevOps: Combinação de desenvolvimento (Dev) e operações (Ops) de *software* para promover a colaboração e integração contínua.
- OO (*Oriented Object*): Orientação a Objetos.

Inteligência Artificial

- IA (Inteligência Artificial): Campo de estudo que se dedica a desenvolver sistemas capazes de realizar tarefas que exigem inteligência humana.
- NLP (*Natural Language Processing*): Processamento de Linguagem Natural.
- CNNs (*Convolutional Neural Networks*): Redes Neurais Convolucionais.
- RNNs (*Recurrent Neural Networks*): Redes Neurais Recorrentes.
- *Deep Learning*: Aprendizado Profundo, uma subárea da IA que se concentra em modelos de redes neurais profundas.
- *Chatbots*: Programas de computador projetados para simular conversas humanas por meio de mensagens de texto ou de voz.
- IA Geracional: Refere-se à geração atual de IA e suas aplicações específicas.
- Filtragem Colaborativa: Método de recomendação que identifica padrões de comportamento e preferências com base na colaboração entre usuários semelhantes.
- Séries Temporais: Conjunto de dados organizados em ordem cronológica para análise de padrões e tendências ao longo do tempo.
- Regressão: Técnica estatística para modelar e analisar a relação entre variáveis dependentes e independentes.
- ChatGPT: Sistema de *chat* baseado no modelo de linguagem GPT (*Generative Pre-trained Transformer*).

- Viés de Dados: Tendência sistemática dos dados usados para treinar modelos de IA, resultando em preconceitos ou distorções nas respostas geradas.
- Ética na IA: Conjunto de princípios e diretrizes que visam garantir o uso responsável, justo e ético da Inteligência Artificial.

Programação Mobile

- Android: Sistema Operacional para dispositivos móveis desenvolvido pelo Google.
- iOS: Sistema Operacional para dispositivos móveis desenvolvido pela Apple.
- Symbian: Sistema Operacional para dispositivos móveis, anteriormente usado pela Nokia.
- RIM: Research In Motion, empresa conhecida por desenvolver o sistema operacional BlackBerry.
- Windows Mobile: Sistema Operacional para dispositivos móveis desenvolvido pela Microsoft.
- J2ME: Java 2 Platform, Micro Edition, uma plataforma para desenvolvimento de aplicativos móveis usando a linguagem Java.
- XCode: Ambiente de desenvolvimento integrado (IDE) usado para criar aplicativos para iOS e macOS.
- API: *Application Programming Interface*, uma interface de programação de aplicativos que define as interações entre diferentes *softwares*.
- IDE: Integrated Development Environment, um ambiente de desenvolvimento integrado que fornece ferramentas para escrever, depurar e testar *softwares*.
- Gráficos, áudio, vídeo e mapas: Recursos de mídia utilizados em aplicativos móveis para exibir imagens, reproduzir áudio e vídeo, e mostrar informações geográficas.
- Atividades, serviços, receptores, repositório de conteúdo, mensagens, arquivo de manifesto: Componentes do *framework* Android que ajudam no desenvolvimento de aplicativos.
- Leiautes, *widgets*, eventos de interface, menus: Elementos de interface do usuário no Android, como a disposição da tela, elementos interativos, interações do usuário e menus.
- Emulação: Processo de execução de um sistema ou *software* em um ambiente que simula o ambiente original.
- Android Market: Antigo nome da loja de aplicativos do Android, atualmente chamada de Google Play Store.
- Android Studio: IDE oficial do Android para desenvolvimento de aplicativos.

- Notificação, estilos, temas, visualizações: Recursos relacionados à interface do usuário do Android para exibir notificações, personalizar o estilo e tema dos aplicativos, e criar diferentes visualizações.

Banco de Dados NoSQL (Não Relacional)

- Bancos de dados chave-valor: Sistemas de gerenciamento de banco de dados que armazenam dados em pares de chave-valor, onde cada valor é associado a uma chave única.
- Redis: Banco de dados chave-valor de código aberto e em memória, usado para armazenar e recuperar dados rapidamente.
- Bancos de dados orientados a colunas: Sistemas de gerenciamento de banco de dados que armazenam dados em colunas em vez de linhas, otimizados para consultas que envolvem grandes quantidades de dados.
- Cassandra: Banco de dados orientado a colunas distribuído, de código aberto e projetado para lidar com grandes volumes de dados e alta escalabilidade.
- Bancos de dados orientados a documentos: Sistemas de gerenciamento de banco de dados que armazenam dados em formato de documentos, geralmente usando o formato JSON ou BSON.
- MongoDB: Banco de dados orientado a documentos de código aberto, projetado para ser escalável e flexível, adequado para aplicativos modernos baseados em documentos.
- Bancos de dados orientados a grafos: Sistemas de gerenciamento de banco de dados que armazenam dados na forma de nós e arestas, permitindo a representação e análise de relacionamentos complexos.
- Neo4j: Banco de dados orientado a grafos de código aberto, projetado para armazenar, consultar e manipular dados em formato de grafo de forma eficiente.

Programação Back-End

- *Back-End*: Desenvolvimento de *software* responsável pela lógica de funcionamento e processamento dos dados de um aplicativo ou *website*.
- C#: Linguagem de programação da Microsoft, utilizada principalmente para desenvolvimento de *software* na plataforma .NET.
- Node.js: Plataforma de desenvolvimento de aplicações em JavaScript baseada no motor de execução V8 do Google Chrome.
- Ruby: Linguagem de programação dinâmica, de código aberto e orientada a objetos.

- Python: Linguagem de programação interpretada, de alto nível e multiparadigma, conhecida por sua simplicidade e legibilidade de código.
- Apache: Servidor *web* de código aberto amplamente utilizado.
- Nginx: Servidor *web* de alto desempenho e escalabilidade, conhecido por sua eficiência no processamento de requisições HTTP.
- APIs: Interfaces de programação de aplicativos que definem métodos e formatos de comunicação entre diferentes componentes de *software*.
- RESTful: Estilo arquitetural para desenvolvimento de APIs *web* que utiliza os princípios do protocolo HTTP.
- JSON: JavaScript Object Notation, formato de dados leve e legível por humanos.
- XML: Extensible Markup Language, formato de dados que permite estruturar informações de forma hierárquica.
- SQL *Injection*: Vulnerabilidade que permite a inserção de comandos SQL maliciosos em uma aplicação, explorando falhas de segurança.
- *Cross-site scripting* (XSS): Vulnerabilidade que permite a execução de *scripts* maliciosos em um *website*, afetando os usuários que acessam o *site*.
- *Frameworks Back-End*: Conjunto de ferramentas e bibliotecas que facilitam o desenvolvimento de *software*, fornecendo estruturas e padrões predefinidos.
- .NET: Plataforma de desenvolvimento da Microsoft que oferece ampla gama de ferramentas e bibliotecas para o desenvolvimento de *software*.
- *Spring Framework* Java que facilita o desenvolvimento de aplicativos corporativos, promovendo a modularidade e a produtividade.
- *Express Framework* Node.js que simplifica o desenvolvimento de aplicações *web* e APIs.
- *Ruby on Rails Framework* Ruby que segue o paradigma de programação *Convention over Configuration* (Convenção sobre Configuração), otimizando a produtividade.
- Django: *Framework* Python que permite o desenvolvimento rápido e seguro de aplicações *web* complexas.
- API de pagamento: Interface de programação que permite a integração de um sistema com serviços de pagamento *on-line*.
- API de geolocalização: Interface de programação que fornece recursos de localização geográfica, como obter coordenadas de um endereço.
- DevOps: Práticas que visam à integração e à colaboração entre equipes de desenvolvimento e às operações para melhorar a eficiência e a qualidade de entrega de *software*.

- Git: Sistema de controle de versão distribuído, amplamente utilizado para o gerenciamento de código-fonte.
- Docker: Plataforma de virtualização de contêineres que facilita o empacotamento e a implantação de aplicativos em ambientes isolados.
- AWS: *Amazon Web Services*, conjunto de serviços em nuvem fornecidos pela Amazon.
- Azure: Plataforma em nuvem da Microsoft que oferece ampla variedade de serviços para desenvolvimento, implantação e gerenciamento de aplicativos.
- DevOps: Combinação de práticas e ferramentas que integra desenvolvimento de *software* (Dev) e operações de infraestrutura (Ops) para acelerar o processo de entrega de *software* e melhorar a colaboração entre as equipes.
- Controle de versionamento de código: Prática de rastrear e gerenciar as alterações feitas no código-fonte de um projeto ao longo do tempo, permitindo o controle e a colaboração entre desenvolvedores.
- Virtualização de ambientes: Técnica que permite criar ambientes virtuais isolados para executar aplicações, proporcionando maior flexibilidade e segurança.
- Git: Sistema de controle de versão distribuído, amplamente utilizado para rastrear as alterações no código-fonte de um projeto, facilitando o trabalho em equipe e a colaboração.
- Docker: Plataforma de código aberto que simplifica o empacotamento e a implantação de aplicativos em contêineres isolados, proporcionando maior eficiência e portabilidade.
- AWS Lambda: Serviço de computação sem servidor da *Amazon Web Services*, que permite a execução de código em resposta a eventos, sem a necessidade de gerenciar a infraestrutura subjacente.
- CI/CD: Integração Contínua (*Continuous Integration*) e Entrega Contínua (*Continuous Delivery*) são práticas que visam automatizar o processo de construção, teste e implantação de *software*, reduzindo o tempo e os riscos envolvidos nas atualizações e lançamentos de novas versões.
- AWS *Elastic Beanstalk*: Serviço da AWS que facilita o *deploy* e a escalabilidade de aplicações *web*, gerenciando a infraestrutura subjacente automaticamente.
- Serviços em nuvem: Recursos computacionais, armazenamento, bancos de dados, entre outros, fornecidos como serviço através da internet, eliminando a necessidade de infraestrutura local e permitindo a escalabilidade e o pagamento conforme o uso.

- Infraestrutura como código (IaC): Abordagem em que a infraestrutura de um ambiente é descrita e gerenciada através de código, possibilitando a automação e a replicabilidade da infraestrutura.
- AWS CloudFormation: Serviço da AWS que permite criar e gerenciar recursos da AWS através de arquivos de modelo, facilitando a implementação e a manutenção da infraestrutura como código.
- Escalabilidade: Capacidade de um sistema de lidar com o aumento ou a diminuição na demanda, adicionando ou removendo recursos automaticamente para manter o desempenho.
- Alta disponibilidade: Característica de um sistema que está sempre disponível e acessível, mesmo em caso de falhas ou interrupções em componentes individuais.
- Monitoramento: Prática de coletar informações sobre o desempenho e o estado de um sistema, permitindo o diagnóstico de problemas e a tomada de ações corretivas.
- Tolerância a falhas: Capacidade de um sistema de continuar operando adequadamente, mesmo quando ocorrem falhas ou interrupções em seus componentes individuais.
- Segurança na nuvem: Conjunto de práticas e medidas de proteção adotadas para garantir a confidencialidade, integridade e disponibilidade dos dados e sistemas armazenados e processados na nuvem.
- IAM: Identity and Access Management (Gerenciamento de Identidade e Acesso) é um serviço que permite controlar o acesso aos recursos da AWS, definindo permissões e políticas para usuários, grupos e funções.
- *Firewall*: Mecanismo de segurança que controla o tráfego de rede com base em regras predefinidas, protegendo a infraestrutura contra ameaças externas.
- VPN: Virtual Private Network (Rede Virtual Privada) é uma rede virtual privada que permite estabelecer uma conexão segura entre redes remotas ou usuários individuais por meio de criptografia.
- Criptografia: Processo de codificação de informações para proteger sua confidencialidade, garantindo que apenas destinatários autorizados possam decifrar e acessar os dados.
- *Backup*: Cópia de segurança dos dados para protegê-los contra perdas acidentais, corrupção ou ataques, permitindo a recuperação dos dados em caso de necessidade.
- Monitoramento de segurança: Prática de monitorar continuamente a infraestrutura e os sistemas em busca de atividades suspeitas ou ameaças de segurança, visando identificar e responder rapidamente a incidentes.

- AWS WAF: Web Application Firewall da AWS, serviço que ajuda a proteger aplicativos *web* contra ataques comuns, filtrando o tráfego malicioso e indesejado.
- Certificados SSL/TLS: Certificados de segurança que estabelecem uma conexão criptografada entre o servidor e o cliente, garantindo a integridade e a autenticidade dos dados transmitidos.
- DDoS: *Distributed Denial of Service* (Negação de Serviço Distribuída) é um tipo de ataque em que muitos dispositivos são usados para sobrecarregar um serviço ou infraestrutura, tornando este inacessível.
- Recuperação de desastres: Conjunto de processos, políticas e infraestrutura que permite a recuperação rápida e eficiente de sistemas e dados em caso de desastres ou interrupções graves.
- *Logs*: Registros detalhados de atividades e eventos que ocorrem em um sistema ou aplicativo, utilizados para monitoramento, auditoria e análise de segurança.
- Princípio da menor permissão: Prática de conceder apenas as permissões necessárias para que um usuário ou sistema execute suas funções, reduzindo o risco de acesso não autorizado a recursos sensíveis.
- Conformidade: Adesão a requisitos, regulamentos e padrões de segurança estabelecidos, garantindo que as práticas e medidas de segurança estejam em conformidade com as normas aplicáveis.
- AWS Shield: Serviço de proteção contra DDoS fornecido pela AWS, que ajuda a proteger as aplicações *web* e os recursos da AWS contra ataques de negação de serviço.

Programação *Front-End*

- *Front-End*: Desenvolvimento de *software* responsável pela interface e interação direta com o usuário em um aplicativo ou *website*.
- HTML: *Hypertext Markup Language* (Linguagem de Marcação de Hipertexto) é a linguagem padrão para estruturar o conteúdo de páginas *web*.
- CSS: *Cascading Style Sheets* (Folhas de Estilo em Cascata) é uma linguagem de estilização utilizada para controlar a aparência e o *layout* de elementos em páginas *web*.
- HTTP: *Hypertext Transfer Protocol* (Protocolo de Transferência de Hipertexto) é um protocolo de comunicação utilizado para transferir informações na *web*.
- JavaScript: Linguagem de programação utilizada para adicionar interatividade e comportamento dinâmico a páginas *web*.

- Angular: *Framework* JavaScript mantido pelo Google, utilizado para construir aplicações *web* complexas.
- React: Biblioteca JavaScript mantida pelo Facebook, utilizada para construir interfaces de usuário interativas e reativas.
- Vue.js: *Framework* JavaScript de código aberto utilizado para construir interfaces de usuário reativas e componentes reutilizáveis.
- Bootstrap: *Framework front-end* popular que oferece uma variedade de componentes e estilos predefinidos para facilitar o desenvolvimento de interfaces responsivas.
- NPM: Node Package Manager (Gerenciador de Pacotes do Node) é um gerenciador de pacotes para a plataforma Node.js, que permite a instalação e o gerenciamento de bibliotecas e dependências de um projeto.
- Gulp: Ferramenta de automação de tarefas em JavaScript, utilizada para simplificar e agilizar o processo de desenvolvimento *front-end*.
- Webpack: Ferramenta de empacotamento de módulos JavaScript, utilizada para gerenciar dependências, otimizar o desempenho e preparar o código para implantação.
- ARIA: *Accessible Rich Internet Applications* (Aplicações Web Ricas e Acessíveis) é um conjunto de atributos e técnicas que tornam o conteúdo *web* mais acessível a pessoas com deficiência.
- API de mapas: Interface de programação que permite a integração de mapas e recursos de geolocalização em um aplicativo ou *website*.
- API de autenticação: Interface de programação que permite a integração de recursos de autenticação e autorização em um aplicativo ou *website*.
- Git: Sistema de controle de versão distribuído, amplamente utilizado para rastrear as alterações no código-fonte de um projeto, facilitando o trabalho em equipe e a colaboração.
- *Continuous Integration* e *Continuous Deployment* (CI/CD): Práticas de desenvolvimento que envolvem a integração frequente de código-fonte em um repositório compartilhado, bem como a implantação automatizada de alterações em um ambiente de produção.

Versionamento de Código e Sistemas de Mensageria

- Git: Sistema de controle de versão distribuído, amplamente utilizado para rastrear as alterações no código-fonte de um projeto, facilitando o trabalho em equipe e a colaboração.
- SCCM: Sistema de Controle de Configuração e Mudanças (*Software Configuration and Change Management*) é um conjunto de práticas e

ferramentas que auxiliam no controle e gerenciamento de mudanças em um ambiente de desenvolvimento de *software*.

- GitHub: Plataforma de hospedagem de código-fonte que utiliza o Git como sistema de controle de versão. Permite a colaboração e o compartilhamento de projetos entre desenvolvedores.
- CI/CD: Integração Contínua e Entrega Contínua (*Continuous Integration and Continuous Delivery*) são práticas de desenvolvimento que envolvem a integração frequente de código-fonte em um repositório compartilhado, bem como a implantação automatizada de alterações em um ambiente de produção.
- *GitFlow*: Método de gerenciamento de *branches* e fluxo de trabalho baseado no Git. É um modelo para controle de versão que define regras para a criação, fusão e publicação de *branches*.
- *GitHub Flow*: Fluxo de trabalho simples baseado no GitHub, em que o desenvolvimento ocorre em *branches* separadas, e as alterações são mescladas (*merge*) no *branch* principal (geralmente o **main** ou **master**) por meio de *pull requests*.
- RabbitMQ: Sistema de mensageria de código aberto que implementa o protocolo AMQP (Advanced Message Queuing Protocol). É utilizado para o envio e recebimento de mensagens entre componentes de um sistema distribuído.
- SQS: Amazon Simple Queue Service é um serviço de filas de mensagens oferecido pela *Amazon Web Services* (AWS). Ele fornece um modo escalável e resiliente de enviar, armazenar e receber mensagens em sistemas distribuídos.
- MQTT: Message Queuing Telemetry Transport é um protocolo de mensageria leve projetado para a comunicação em redes com restrições de largura de banda, como dispositivos IoT (Internet das Coisas).
- AMQP: Advanced Message Queuing Protocol é um protocolo aberto e padronizado para mensageria assíncrona entre sistemas. É amplamente utilizado em sistemas distribuídos para troca de mensagens.
- Apache Kafka: Plataforma de *streaming* distribuída que permite a ingestão, o armazenamento e o processamento de grandes volumes de dados em tempo real. É projetada para lidar com fluxos contínuos de eventos e frequentemente usada em arquiteturas orientadas a eventos.
- Spring Boot: *Framework* de desenvolvimento de aplicativos Java que simplifica a configuração e criação de aplicativos Java. É conhecido por sua facilidade de uso e suporte a diversas tecnologias, incluindo integração com serviços de mensageria.

- Apache Camel: *Framework* de integração de código aberto que oferece um modo simplificado de integrar sistemas e aplicativos, usando diversos padrões de integração. Suporta a integração com diferentes protocolos e tecnologias, incluindo serviços de mensageria.
- AWS SDK: *Amazon Web Services Software Development Kit* é um conjunto de bibliotecas e ferramentas fornecidas pela AWS para ajudar os desenvolvedores a interagir e utilizar os serviços da AWS em suas aplicações.
- Serverless: Modelo de computação em nuvem em que o provedor de nuvem é responsável pela execução e dimensionamento das aplicações, eliminando a necessidade de gerenciar infraestrutura. Os desenvolvedores podem se concentrar no código da aplicação sem se preocupar com a infraestrutura subjacente.
- Kubernetes: Plataforma de orquestração de contêineres de código aberto que permite a implantação, o dimensionamento e o gerenciamento de aplicativos em contêineres. É amplamente utilizada para criar e gerenciar arquiteturas de microsserviços.