

Programa Avançado Engenharia de Software



Programa Avançado Engenharia de Software

- » Modalidade: online
- » Duração: 6 meses
- » Certificado: TECH Universidade Tecnológica
- » Dedicção: 16h/semana
- » Horário: no seu próprio ritmo
- » Provas: online

Acesso ao site: www.techtitute.com/br/informatica/programa-avancado/programa-avancado-engenharia-software

Índice

01

Apresentação

pág. 4

02

Objetivos

pág. 8

03

Estrutura e conteúdo

pág. 12

04

Metodologia

pág. 22

05

Certificado

pág. 30

01

Apresentação

Este Programa Avançado foi desenvolvido com o objetivo de atualizar os conhecimentos do aluno em Engenharia de Software através de renomados especialistas com ampla experiência na área. Nesta capacitação o aluno aprenderá as últimas atualizações e avanços da matéria. Trata-se de uma oportunidade única para alcançar sucesso em sua carreira profissional.



“

Este Programa Avançado atualizará os seus conhecimentos em Engenharia de Software de forma prática, 100% online, sem abrir mão do máximo rigor acadêmico”

Este programa destina-se aos profissionais interessados em alcançar um nível mais elevado de conhecimento em Engenharia de Software. Seu principal objetivo é capacitar o aluno para aplicar no mundo real os conhecimentos adquiridos ao longo do programa, além de proporcionar um ambiente de estudo baseado nas condições que podem ser encontradas em seu futuro, de forma rigorosa e realista.

Este Programa Avançado irá preparar o aluno para a prática profissional da Ciências da Computação, através de uma experiência acadêmica transversal e versátil, adaptada às novas tecnologias e inovações nesta área. Além disso, proporcionará um amplo conhecimento de Engenharia de Software em um programa ministrado por profissionais da área.

Oferecendo ao aluno a oportunidade de estudar 100% online, sem abrir mão de suas obrigações e facilitando o seu retorno à universidade. Além de atualizar seus conhecimentos e ajudá-lo a conquistar um certificado, contribuindo para seu crescimento pessoal e profissional.



Aprenda as últimas técnicas e estratégias com este programa e alcance o sucesso como engenheiro de software”

Este **Programa Avançado de Engenharia de Software** conta com o conteúdo mais completo e atualizado do mercado. Suas principais características são:

- ◆ O Desenvolvimento de 100 cenários simulados apresentados por especialistas em Engenharia de Software
- ◆ Seu conteúdo gráfico, esquemático e extremamente prático que fornece informações científicas e práticas sobre Engenharia de Software
- ◆ Atualizações sobre os últimos avanços de Engenharia de Software
- ◆ Exercícios práticos onde o processo de autoavaliação é realizado para melhorar a aprendizagem
- ◆ Sistema interativo de aprendizagem baseado no método do estudo de caso e sua aplicação na prática real
- ◆ Aulas teóricas, perguntas aos especialistas, fóruns de discussão sobre temas controversos e trabalhos de reflexão individual
- ◆ Acesso a todo o conteúdo desde qualquer dispositivo fixo ou portátil com conexão à Internet

“

Este programa de estudos melhorará suas habilidades e atualizará seus conhecimentos em Engenharia de Software”

Com uma equipe de professores constituída por profissionais da área de Ciências da Computação, cuja experiência profissional é somada nesta capacitação, além de reconhecidos especialistas de conceituadas instituições e universidades de prestígio.

Seu conteúdo multimídia, desenvolvido com a mais recente tecnologia educacional, oferece ao profissional uma aprendizagem situada e contextual, ou seja, um ambiente simulado que proporcionará uma aprendizagem imersiva e programada para se capacitar em situações reais.

Este programa de estudos tem como fundamento a Aprendizagem Baseada em Problemas, que permite ao profissional abordar a solução de problemas reais que surgem ao longo do curso e de sua prática profissional. Para isso, o profissional contará com a ajuda de um sistema inovador de vídeo interativo idealizado por conceituados especialistas em Engenharia de Software e com ampla experiência de ensino.

Aproveite a mais recente tecnologia educacional para se atualizar em Engenharia de Software sem sair de casa.

Conheça os últimos avanços em Engenharia de Software com o suporte de especialistas da área.



02

Objetivos

O objetivo desta capacitação é oferecer aos profissionais de TI os conhecimentos e habilidades necessárias para o desempenho da sua atividade utilizando os protocolos e técnicas mais avançadas da atualidade. Através de uma abordagem de trabalho totalmente adaptável ao aluno, este Programa Avançado irá levá-lo progressivamente a adquirir as competências que lhe impulsionam para um nível profissional superior.



“

Alcance o nível de conhecimento que você tanto deseja e domine a área da Engenharia de Software através desta capacitação de alto nível”



Objetivos gerais

- ♦ Capacitar científica e tecnologicamente, assim como preparar para a prática profissional das Ciências da Computação, tudo isso com uma capacitação transversal e versátil adaptada às novas tecnologias e inovações nesta área
- ♦ Proporcionar amplo conhecimento na área de computação, estrutura de computadores e engenharia de software, incluindo os fundamentos matemáticos, estatísticos e físicos essenciais em uma Engenharia

“

Alcance o sucesso profissional como Engenheiro da Computação com este programa intensivo, desenvolvido por profissionais com ampla experiência na área”





Objetivos específicos

- ◆ Estabelecer as bases da Engenharia de Software e da modelagem, aprendendo os principais processos e conceitos
- ◆ Compreender o processo de software e os diferentes modelos para o seu desenvolvimento, incluindo as tecnologias ágeis
- ◆ Conhecer a engenharia de requisitos, seu desenvolvimento, elaboração, negociação e validação
- ◆ Aprender a modelagem de requisitos e os diferentes elementos como cenários, informações, aulas de análise, fluxo, comportamento e padrões
- ◆ Compreender os conceitos e processos de design de software, incluindo também o design de arquitetura e o design em nível de componentes, baseados em padrões
- ◆ Conhecer as principais normas relacionadas à qualidade de software e à gestão de projetos
- ◆ Conhecer detalhadamente as diferentes metodologias ágeis utilizadas na engenharia de software
- ◆ Aprender a desenvolver através das técnicas de *Scrum*, programação extrema (XP) e desenvolvimento de software baseado em reutilização
- ◆ Compreender os diferentes padrões de arquiteturas de sistemas e de design de software, assim como a arquitetura de aplicações na nuvem.
- ◆ Aprender a realizar testes de software com metodologias como *desenvolvimento orientado a testes (Test Driven Development - TDD)*, *desenvolvimento orientado a testes de aceitação (Acceptance Test Driven Development - ATDD)*, *desenvolvimento orientado a comportamento (Behavior Driven Development - BDD)* e *Cucumber*
- ◆ Profundizar na melhoria do processo de desenvolvimento de software e na qualidade do software, usando os padrões ISO/IEC
- ◆ Introduzir o conceito de DevOps e suas principais práticas
- ◆ Compreender a importância da engenharia de requisitos no processo de desenvolvimento de software
- ◆ Ampliar a compreensão das fontes de requisitos e das técnicas de eliciação de requisitos, visto que são partes essenciais do processo
- ◆ Compreender e aplicar a criação de protótipos como parte essencial do processo de desenvolvimento
- ◆ Aprender a realizar análise de requisitos e como documentá-las adequadamente
- ◆ Compreender os processos de validação e negociação de requisitos, bem como a modelagem e gestão de requisitos
- ◆ Adquirir os conhecimentos necessários para o manuseio de sistemas críticos e a especificação formal de requisitos
- ◆ Conhecer o quadro de referência da Engenharia de Software e a norma ISO/IEC 12207
- ◆ Conhecer as características do processo unificado de desenvolvimento de software e o planejamento no contexto do desenvolvimento de software ágil
- ◆ Conhecer os diferentes estilos de design de software distribuído e arquiteturas de software orientadas a serviços
- ◆ Aprender os conceitos essenciais no design de interfaces gráficas de usuário
- ◆ Compreender as bases do desenvolvimento de aplicações web
- ◆ Aprofundar-se em estratégias e técnicas de teste de software, fatores de qualidade de software e diferentes métricas utilizadas

03

Estrutura e conteúdo

O conteúdo deste programa foi elaborado por uma equipe de profissionais de Ciências da Computação, conscientes da relevância da capacitação atual para aprofundar-se nessa área de conhecimento com o objetivo de enriquecer humanisticamente o aluno, elevando o nível de conhecimento em Engenharia de Software através das mais recentes tecnologias educacionais disponíveis.



“

Este Programa Avançado de Engenharia de Software conta com o conteúdo mais completo e atualizado do mercado”

Módulo 1. Engenharia de Software

- 1.1. Introdução à engenharia de software e a modelagem
 - 1.1.1. A Natureza do software
 - 1.1.2. A natureza única das *Webapps*
 - 1.1.3. Engenharia de Software
 - 1.1.4. Processo do software
 - 1.1.5. A prática da engenharia de software
 - 1.1.6. Mitos do software
 - 1.1.7. Como tudo começa
 - 1.1.8. Conceitos orientado a objetos
 - 1.1.9. Introdução à UML
- 1.2. Processo do software
 - 1.2.1. Um modelo geral de processo
 - 1.2.2. Modelo de processo prescritivo
 - 1.2.3. Modelo de processo especializado
 - 1.2.4. Processo unificado
 - 1.2.5. Modelos do processo pessoal e em equipe
 - 1.2.6. O que é agilidade?
 - 1.2.7. O que é um processo ágil?
 - 1.2.8. *Scrum*
 - 1.2.9. Conjunto de ferramentas de um processo ágil
- 1.3. Princípios que orientam a prática da Engenharia de Software
 - 1.3.1. Princípios que orientam o processo
 - 1.3.2. Princípios que orientam a prática
 - 1.3.3. Princípios de comunicação
 - 1.3.4. Princípios de planejamento
 - 1.3.5. Princípios de modelagem
 - 1.3.6. Princípios de construção
 - 1.3.7. Princípios de implantação
- 1.4. Compreensão dos requisitos
 - 1.4.1. Engenharia de requisitos
 - 1.4.2. Estabelecer as bases
 - 1.4.3. Indagação sobre os requisitos
 - 1.4.4. Desenvolvimento de casos de uso
 - 1.4.5. Elaboração do modelo de requisitos
 - 1.4.6. Negociação de requisitos
 - 1.4.7. Validação de requisitos
- 1.5. Modelagem de requisitos: cenários, informações e classes de análise
 - 1.5.1. Análise de requisitos
 - 1.5.2. Modelagem baseada em cenários
 - 1.5.3. Modelos UML que proporcionam o caso de uso
 - 1.5.4. Conceitos de modelagem de dados
 - 1.5.5. Modelagem baseada em aulas
 - 1.5.6. Diagramas de classes
- 1.6. Modelagem de requisitos: fluxo, comportamento e padrões
 - 1.6.1. Requisitos que modelam as estratégias
 - 1.6.2. Modelagem orientada ao fluxo
 - 1.6.3. Diagrama de transição de estados
 - 1.6.4. Criação de um modelo de comportamento
 - 1.6.5. Diagramas de sequência
 - 1.6.6. Diagramas de comunicação
 - 1.6.7. Padrões para modelagem de requisitos
- 1.7. Conceitos de design
 - 1.7.1. Design no contexto da Engenharia de Software
 - 1.7.2. Processo de design
 - 1.7.3. Conceitos de design
 - 1.7.4. Conceitos de design orientado a objetos
 - 1.7.5. O modelo de design

- 1.8. Design de arquitetura
 - 1.8.1. Arquitetura de software
 - 1.8.2. Gêneros arquitetônicos
 - 1.8.3. Estilos arquitetônicos
 - 1.8.4. Design arquitetônico
 - 1.8.5. Evolução dos designs alternativos para a arquitetura
 - 1.8.6. Mapeamento da arquitetura com o uso de fluxos de dados
- 1.9. Design em nível de componentes e baseado em padrões
 - 1.9.1. O que é um componente?
 - 1.9.2. Design de componentes baseados em classe
 - 1.9.3. Realização do design no nível de componentes
 - 1.9.4. Design de componentes tradicionais
 - 1.9.5. Desenvolvimento baseado em componentes
 - 1.9.6. Padrões de design
 - 1.9.7. Design de software baseado em padrões
 - 1.9.8. Padrões arquitetônicos
 - 1.9.9. Padrões de design no nível de componentes
 - 1.9.10. Padrões de design da interface do usuário
- 1.10. Qualidade do software e administração de projetos
 - 1.10.1. Qualidade
 - 1.10.2. Qualidade do software
 - 1.10.3. A importância da qualidade do software
 - 1.10.4. Obter a qualidade do software
 - 1.10.5. Garantia de qualidade do software
 - 1.10.6. O espectro administrativo
 - 1.10.7. Os funcionários
 - 1.10.8. O produto
 - 1.10.9. O processo
 - 1.10.10. O projeto
 - 1.10.11. Princípios e práticas

Módulo 2. Engenharia de Software Avançada

- 2.1. Introdução às metodologias ágeis
 - 2.1.1. Modelos de processo e metodologias
 - 2.1.2. Agilidade e processos ágeis
 - 2.1.3. Manifesto Ágil
 - 2.1.4. Algumas metodologias ágeis
 - 2.1.5. Ágil x Tradicional
- 2.2. *Scrum*
 - 2.2.1. Origens e filosofia do *Scrum*
 - 2.2.2. Valores do *Scrum*
 - 2.2.3. Fluxo do processo *Scrum*
 - 2.2.4. Funções do *Scrum*
 - 2.2.5. Os artefatos do *Scrum*
 - 2.2.6. Os eventos do *Scrum*
 - 2.2.7. Histórias de usuários
 - 2.2.8. Extensões do *Scrum*
 - 2.2.9. Estimativas ágeis
 - 2.2.10. Escalonamento do *Scrum*
- 2.3. Programação extrema
 - 2.3.1. Razões e visão geral de XP
 - 2.3.2. O ciclo de vida no XP
 - 2.3.3. Os cinco valores básicos
 - 2.3.4. As doze práticas básicas do XP
 - 2.3.5. Funções dos participantes
 - 2.3.6. XP Industrial
 - 2.3.7. Avaliação crítica de XP

- 2.4. Desenvolvimento de software baseado em reutilização
 - 2.4.1. A reutilização do software
 - 2.4.2. Níveis de reutilização de código
 - 2.4.3. Técnicas concretas de reutilização
 - 2.4.4. Desenvolvimento baseado em componentes
 - 2.4.5. Vantagens e problemas da reutilização
 - 2.4.6. Planejamento da reutilização
- 2.5. Padrões de arquitetura de sistema e projeto de software
 - 2.5.1. Design arquitetônico
 - 2.5.2. Padrões arquitetônicos gerais
 - 2.5.3. Arquiteturas tolerantes a falhas
 - 2.5.4. Arquiteturas de sistemas distribuídos
 - 2.5.5. Os padrões de projeto
 - 2.5.6. Padrões de Gamma
 - 2.5.7. Padrões de design de interações
- 2.6. Arquitetura de aplicações na nuvem
 - 2.6.1. Fundamentos de *Cloud Computing*
 - 2.6.2. Qualidade de aplicações na nuvem
 - 2.6.3. Estilos de arquitetura
 - 2.6.4. Padrões de design
- 2.7. Testes de software: TDD, ATDD e BDD
 - 2.7.1. Verificação e validação de software
 - 2.7.2. As provas de software
 - 2.7.3. *Test Driven Development* (TDD)
 - 2.7.4. *Acceptance Test Driven Development* (ATDD)
 - 2.7.5. *Behavior Driven Development* (BDD)
 - 2.7.6. BDD e *Cucumber*
- 2.8. A melhoria do processo de software
 - 2.8.1. A melhoria do processo de software
 - 2.8.2. O processo de melhoria de processos
 - 2.8.3. Modelos de maturidade
 - 2.8.4. O modelo CMMI
 - 2.8.5. CMMI/V2.0. 2.8.6. CMMI e Ágil

```
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
echo "Photo gallery"
elseif ($_COOKIE['lang'] == 'rus')
echo "Фотопалерея";
}
else
echo "Foto galerija";
?></h3>-->
<div class="<?if($_GET[type]==1)"
<a href="foto-galerija.php?
<div id="left_sidebar">
<div id="left_ico">
<p <?if($_COOKIE['lang'] == 'rus')
<?
if($_COOKIE['lang'] == 'eng'){
echo "Wood-frame houses";
}elseif($_COOKIE['lang'] == 'rus')
echo "Деревянные каркасные
}else{
echo "Koka karkasa mājas";
```

- 2.9. A qualidade do produto software: SQuaRE
 - 2.9.1. A qualidade do software
 - 2.9.2. Modelo de qualidade do produto software
 - 2.9.3. Família ISO/IEC 25000
 - 2.9.4. ISO/IEC 25010: modelo e características de qualidade
 - 2.9.5. ISO/IEC 25012: a qualidade dos dados
 - 2.9.6. ISO/IEC 25020: medição da qualidade do software
 - 2.9.7. ISO/IEC 25022, 25023 y 25024: métricas de qualidade do software e dos dados
 - 2.9.8. ISO/IEC 25040: avaliação do software
 - 2.9.9. O processo de certificação
- 2.10. Introdução ao DevOps
 - 2.10.1. Conceito de DevOps
 - 2.10.2. Principais práticas

Módulo 3. Engenharia de requisitos

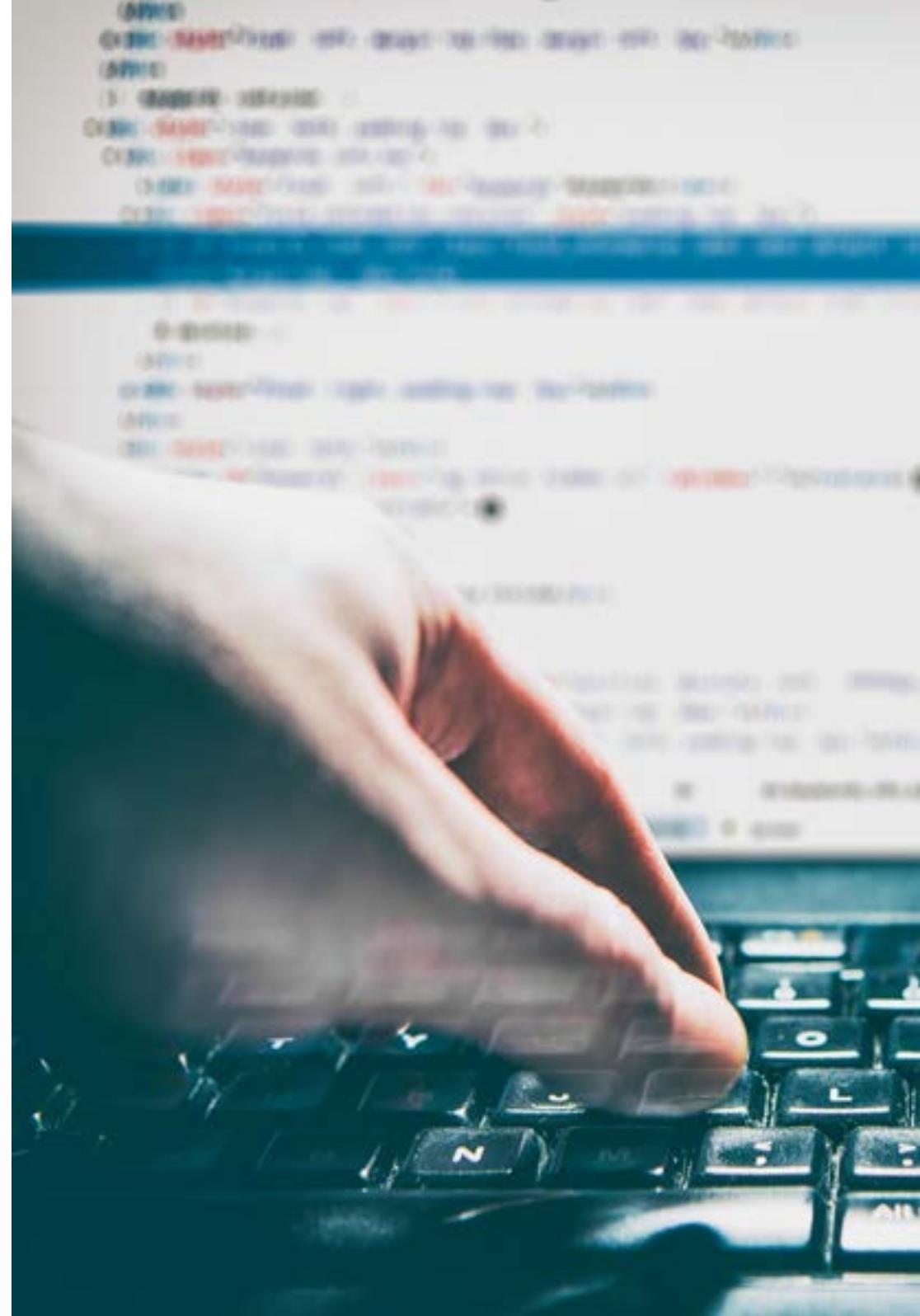
- 3.1. Introdução à Engenharia de Requisitos
 - 3.1.1. A importância dos requisitos
 - 3.1.2. Conceito de requisito
 - 3.1.3. Dimensões de requisitos
 - 3.1.4. Níveis e tipos de requisitos
 - 3.1.5. Características de requisitos
 - 3.1.6. Engenharia de requisitos
 - 3.1.7. O processo de engenharia de Requisitos
 - 3.1.8. *Frameworks* para engenharia de requisitos
 - 3.1.9. Boas práticas em engenharia de requisitos
 - 3.1.10. O analista de negócios
- 3.2. As fontes dos requisitos
 - 3.2.1. A rede de requisitos
 - 3.2.2. Os *Stakeholders*
 - 3.2.3. Os requisitos de negócio
 - 3.2.4. Documento de visão e alcance

- 3.3. Técnicas de elicitação de requisitos
 - 3.3.1. A elicitação de requisitos
 - 3.3.2. Problemas de elicitação de requisitos
 - 3.3.3. Contextos da descoberta
 - 3.3.4. Entrevistas
 - 3.3.5. Observação e aprendizagem
 - 3.3.6. Etnografia
 - 3.3.7. *Workshops*
 - 3.3.8. *Focus Groups*
 - 3.3.9. Questionários
 - 3.3.10. *Brainstorming* e técnicas criativas
 - 3.3.11. Meios grupais
 - 3.3.12. Análise das interfaces do sistema
 - 3.3.13. Análise de documentos e «arqueologia»
 - 3.3.14. Casos de uso e cenários
 - 3.3.15. Os protótipos
 - 3.3.16. A engenharia reversa
 - 3.3.17. Reutilização dos requisitos
 - 3.3.18. Boas práticas de elicitação
- 3.4. Requisitos dos usuários
 - 3.4.1. Pessoas
 - 3.4.2. Casos de uso e histórias de usuários
 - 3.4.3. Cenários
 - 3.4.4. Tipos de cenários
 - 3.4.5. Como descobrir os cenários
- 3.5. Técnicas de Prototipagem
 - 3.5.1. Prototipagem
 - 3.5.2. Protótipos de acordo com seu alcance
 - 3.5.3. Protótipos de acordo com sua temporalidade
 - 3.5.4. A fidelidade de um protótipo
 - 3.5.5. Protótipos de interface do usuário
 - 3.5.6. Avaliação de protótipos
- 3.6. Análise de requisitos
 - 3.6.1. A análise de requisitos
 - 3.6.2. Boas práticas da análise de requisitos
 - 3.6.3. O dicionário de dados
 - 3.6.4. Priorização de requisitos
- 3.7. Documentação dos requisitos
 - 3.7.1. O documento de especificação de requisitos
 - 3.7.2. Estrutura e conteúdo de um SRS (Software Requirements Specification)
 - 3.7.3. Documentação em linguagem natural
 - 3.7.4. EARS: *Easy Approach to Requirements Syntax*
 - 3.7.5. Os requisitos não funcionais
 - 3.7.6. Atributos e templates em forma de tabela
 - 3.7.7. Boas práticas de especificação
- 3.8. Validação e negociação de requisitos
 - 3.8.1. Validação de requisitos
 - 3.8.2. Técnicas de validação de requisitos
 - 3.8.3. Negociação de requisitos
- 3.9. Modelagem e gestão de requisitos
 - 3.9.1. A modelagem de requisitos
 - 3.9.2. A perspectiva do usuário
 - 3.9.3. A perspectiva dos dados
 - 3.9.4. A perspectiva funcional ou orientada ao fluxo
 - 3.9.5. A perspectiva comportamental
 - 3.9.6. A volatilidade dos requisitos
 - 3.9.7. Processo de gestão de requisitos
 - 3.9.8. Ferramentas para a gestão de requisitos
 - 3.9.9. Boas práticas em gestão de requisitos
- 3.10. Sistemas críticos e especificação formal
 - 3.10.1. Os sistemas críticos
 - 3.10.2. Especificação orientada a riscos
 - 3.10.3. Especificação formal

Módulo 4. Processos de Engenharia de Software

- 4.1. Estrutura de Engenharia de Software
 - 4.1.1. Características do software
 - 4.1.2. Os principais processos na engenharia de software
 - 4.1.3. Modelos de processo de desenvolvimento de software
 - 4.1.4. Estrutura de referência padrão para o processo de desenvolvimento de software: a norma ISO/IEC 12207
- 4.2. Processo unificado de desenvolvimento de software
 - 4.2.1. Processo unificado
 - 4.2.2. Dimensões do processo unificado
 - 4.2.3. Processo de desenvolvimento orientado por casos de uso
 - 4.2.4. Fluxos de trabalho fundamentais de processos unificados
- 4.3. Planejamento no contexto do desenvolvimento de software ágil
 - 4.3.1. Características do desenvolvimento de software ágil
 - 4.3.2. Diferentes horizontes temporais de planejamento em um desenvolvimento ágil
 - 4.3.3. Estrutura de desenvolvimento ágil *Scrum* e horizontes temporais de planejamento
 - 4.3.4. Histórias de usuários como uma unidade de planejamento e estimativa
 - 4.3.5. Técnicas comuns para obter uma estimativa
 - 4.3.6. Escalas para a interpretação de estimativas
 - 4.3.7. *Planning Poker*
 - 4.3.8. Tipos de planejamento comuns: planejamento de entregas e planejamento de iteração
- 4.4. Estilos de design de software distribuído e arquiteturas de software orientadas a serviços
 - 4.4.1. Modelos de comunicação em sistemas de software distribuídos
 - 4.4.2. Camada intermediária ou *Middleware*
 - 4.4.3. Padrões de arquitetura para sistemas distribuídos
 - 4.4.4. Processo geral de design de serviços de software
 - 4.4.5. Aspectos de design de serviços de software
 - 4.4.6. Composição dos serviços
 - 4.4.7. Arquitetura de serviços web
 - 4.4.8. Componentes de Infraestrutura e SOA

- 4.5. Introdução ao desenvolvimento de software orientado por modelos
 - 4.5.1. O conceito de modelo
 - 4.5.2. Desenvolvimento de software orientado por modelos
 - 4.5.3. Estrutura de referência de desenvolvimento orientada por modelos MDA
 - 4.5.4. Elementos de um modelo de transformação
- 4.6. Design de interfaces gráficas do usuário
 - 4.6.1. Princípios de design de interfaces do usuário
 - 4.6.2. Padrões de design arquitetônico para sistemas interativos: Modelo-Visão-Controlador (MVC)
 - 4.6.3. Experiência do usuário (UX *User Experience*)
 - 4.6.4. Design centrado no usuário
 - 4.6.5. Processo de análise e design da interface gráfica do usuário
 - 4.6.6. Usabilidade de interfaces do usuário
 - 4.6.7. Acessibilidade em interfaces do usuário
- 4.7. Design de aplicações web
 - 4.7.1. Características das aplicações web
 - 4.7.2. Interface de usuário de uma aplicação web
 - 4.7.3. Design de navegação
 - 4.7.4. Protocolo de interação base para aplicações web
 - 4.7.5. Estilos de arquitetura para aplicações web
- 4.8. Estratégias e técnicas de testes de software e fatores de qualidade de software
 - 4.8.1. Estratégias de testes
 - 4.8.2. Design de casos de testes
 - 4.8.3. Relação custo-qualidade
 - 4.8.4. Modelos de qualidade
 - 4.8.5. Família de normas ISO/IEC 25000 (SQuaRE)
 - 4.8.6. Modelo de qualidade de produto (ISO 2501n)
 - 4.8.7. Modelo de qualidade de dados (ISO 2501n)
 - 4.8.8. Gestão de qualidade do software



- 4.9. Introdução às métricas em Engenharia de Software
 - 4.9.1. Conceitos básicos: medidas, métricas e indicadores
 - 4.9.2. Tipos de métricas em Engenharia de Software
 - 4.9.3. O processo de medição
 - 4.9.4. ISO 25024 Métricas externas e de qualidade em uso
 - 4.9.5. Métrica orientada a objetos
- 4.10. Manutenção e reengenharia de software
 - 4.10.1. Processo de manutenção
 - 4.10.2. Estrutura padrão do processo de manutenção ISO/IEC 14764
 - 4.10.3. Modelo de processo de reengenharia de software
 - 4.10.4. Engenharia inversa

“

*Uma experiência de capacitação única,
fundamental e decisiva para impulsionar
seu crescimento profissional”*

04 Metodologia

Este curso oferece uma maneira diferente de aprender. Nossa metodologia é desenvolvida através de um modo de aprendizagem cíclico: **o Relearning**. Este sistema de ensino é utilizado, por exemplo, nas faculdades de medicina mais prestigiadas do mundo e foi considerado um dos mais eficazes pelas principais publicações científicas, como o ***New England Journal of Medicine***.



“

Descubra o Relearning, um sistema que abandona a aprendizagem linear convencional para realizá-la através de sistemas de ensino cíclicos: uma forma de aprendizagem que se mostrou extremamente eficaz, especialmente em disciplinas que requerem memorização"

Estudo de caso para contextualizar todo o conteúdo

Nosso programa oferece um método revolucionário para desenvolver as habilidades e o conhecimento. Nosso objetivo é fortalecer as competências em um contexto de mudança, competitivo e altamente exigente.

“

Com a TECH você irá experimentar uma forma de aprender que está revolucionando as bases das universidades tradicionais em todo o mundo”



Você terá acesso a um sistema de aprendizagem baseado na repetição, por meio de um ensino natural e progressivo ao longo de todo o programa.



Um método de aprendizagem inovador e diferente

Este curso da TECH é um programa de ensino intensivo, criado do zero, que propõe os desafios e decisões mais exigentes nesta área, em âmbito nacional ou internacional. Através desta metodologia, o crescimento pessoal e profissional é impulsionado em direção ao sucesso. O método do caso, técnica que constitui a base deste conteúdo, garante que a realidade econômica, social e profissional mais atual seja adotada.

“

Nosso programa prepara você para enfrentar novos desafios em ambientes incertos e alcançar o sucesso na sua carreira”

Através de atividades de colaboração e casos reais, o aluno aprenderá a resolver situações complexas em ambientes reais de negócios.

O método do caso é o sistema de aprendizagem mais utilizado nas principais escolas de Informática do mundo, desde que elas existem. Desenvolvido em 1912 para que os estudantes de Direito não aprendessem a lei apenas com base no conteúdo teórico, o método do caso consistia em apresentar-lhes situações realmente complexas para que tomassem decisões conscientes e julgassem a melhor forma de resolvê-las. Em 1924 foi estabelecido como o método de ensino padrão em Harvard.

Em uma determinada situação, o que um profissional deveria fazer? Esta é a pergunta que abordamos no método do caso, um método de aprendizagem orientado para a ação. Ao longo do curso, os alunos vão se deparar com múltiplos casos reais. Terão que integrar todo o conhecimento, pesquisar, argumentar e defender suas ideias e decisões.

Metodologia Relearning

A TECH utiliza de maneira eficaz a metodologia do estudo de caso com um sistema de aprendizagem 100% online, baseado na repetição, combinando elementos didáticos diferentes em cada aula.

Potencializamos o Estudo de Caso com o melhor método de ensino 100% online: o Relearning.

Em 2019 alcançamos os melhores resultados de aprendizagem entre todas as universidades online do mundo.

Na TECH você aprenderá através de uma metodologia de vanguarda, desenvolvida para capacitar os profissionais do futuro. Este método, na vanguarda da pedagogia mundial, se chama Relearning.

Nossa universidade é uma das únicas que possui a licença para usar este método de sucesso. Em 2019 conseguimos melhorar os níveis de satisfação geral dos nossos alunos (qualidade de ensino, qualidade dos materiais, estrutura dos curso, objetivos, entre outros) com relação aos indicadores da melhor universidade online.



No nosso programa, a aprendizagem não é um processo linear, ela acontece em espiral (aprender, desaprender, esquecer e reaprender). Portanto, combinamos cada um desses elementos de forma concêntrica. Esta metodologia já capacitou mais de 650 mil universitários com um sucesso sem precedentes em campos tão diversos como a bioquímica, a genética, a cirurgia, o direito internacional, habilidades administrativas, ciência do esporte, filosofia, direito, engenharia, jornalismo, história, mercados e instrumentos financeiros. Tudo isso em um ambiente altamente exigente, com um corpo discente com um perfil socioeconômico médio-alto e uma média de idade de 43,5 anos.

O Relearning permitirá uma aprendizagem com menos esforço e mais desempenho, fazendo com que você se envolva mais em sua especialização, desenvolvendo o espírito crítico e sua capacidade de defender argumentos e contrastar opiniões: uma equação de sucesso.

A partir das últimas evidências científicas no campo da neurociência, sabemos como organizar informações, ideias, imagens, memórias, mas sabemos também que o lugar e o contexto onde aprendemos algo é fundamental para nossa capacidade de lembrá-lo e armazená-lo no hipocampo, para mantê-lo em nossa memória a longo prazo.

Desta forma, no que se denomina Neurocognitive context-dependent e-learning, os diferentes elementos do nosso programa estão ligados ao contexto onde o aluno desenvolve sua prática profissional.



Neste programa, oferecemos o melhor material educacional, preparado especialmente para os profissionais:



Material de estudo

Todo o conteúdo foi criado especialmente para o curso pelos especialistas que irão ministrá-lo, o que faz com que o desenvolvimento didático seja realmente específico e concreto.

Posteriormente, esse conteúdo é adaptado ao formato audiovisual, para criar o método de trabalho online da TECH. Tudo isso, com as técnicas mais inovadoras que proporcionam alta qualidade em todo o material que é colocado à disposição do aluno.



Masterclasses

Há evidências científicas sobre a utilidade da observação de terceiros especialistas.

O "Learning from an expert" fortalece o conhecimento e a memória, além de gerar segurança para a tomada de decisões difíceis no futuro.



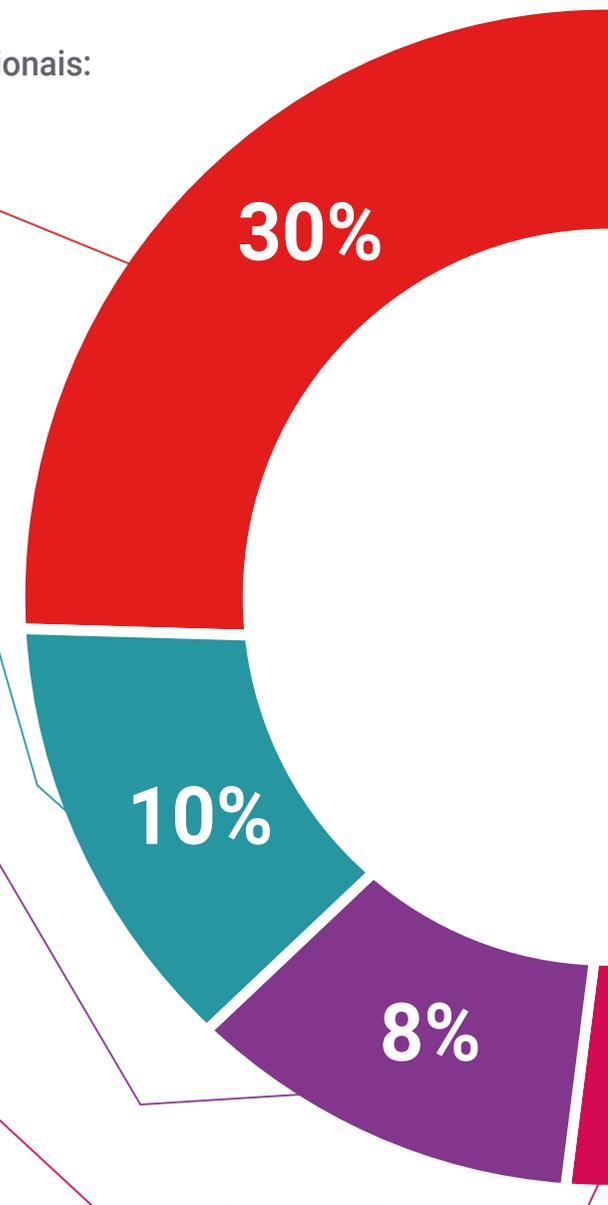
Práticas de habilidades e competências

Serão realizadas atividades para desenvolver competências e habilidades específicas em cada área temática. Práticas e dinâmicas para adquirir e ampliar as competências e habilidades que um especialista precisa desenvolver no contexto globalizado em que vivemos.



Leituras complementares

Artigos recentes, documentos de consenso e diretrizes internacionais, entre outros. Na biblioteca virtual da TECH o aluno terá acesso a tudo o que for necessário para complementar a sua capacitação.





Estudos de caso

Os alunos irão completar uma seleção dos melhores estudos de caso escolhidos especialmente para esta capacitação. Casos apresentados, analisados e orientados pelos melhores especialistas do cenário internacional.



Resumos interativos

A equipe da TECH apresenta o conteúdo de forma atraente e dinâmica através de pílulas multimídia que incluem áudios, vídeos, imagens, gráficos e mapas conceituais para consolidar o conhecimento.

Este sistema exclusivo de capacitação por meio da apresentação de conteúdo multimídia foi premiado pela Microsoft como "Caso de sucesso na Europa".



Testing & Retesting

Avaliamos e reavaliamos periodicamente o conhecimento do aluno ao longo do programa, através de atividades e exercícios de avaliação e autoavaliação, para que possa comprovar que está alcançando seus objetivos.



05

Certificado

O Programa Avançado de Engenharia de Software garante, além da capacitação mais rigorosa e atualizada, o acesso a um título de Programa Avançado emitido pela TECH Universidade Tecnológica.



“

*Conclua este programa de estudos
com sucesso e receba o seu certificado
sem sair de casa e sem burocracias”*

Este **Programa Avançado de Engenharia de Software** conta com o conteúdo mais completo e atualizado do mercado.

Uma vez aprovadas as avaliações, o aluno receberá por correio o certificado* correspondente ao título de **Programa Avançado** emitido pela **TECH Universidade Tecnológica**.

O certificado emitido pela **TECH Universidade Tecnológica** expressará a qualificação obtida no Programa Avançado, atendendo aos requisitos normalmente exigidos pelas bolsas de empregos, concursos públicos e avaliação de carreira profissional.

Título: **Programa Avançado de Engenharia Software**

N.º de Horas Oficiais: **600h**



*Apostila de Haia: Caso o aluno solicite que seu certificado seja apostilado, a TECH EDUCATION providenciará a obtenção do mesmo a um custo adicional.

futuro
saúde confiança pessoas
informação orientadores
educação certificação ensino
garantia aprendizagem
instituições tecnologia
comunidade compromisso
atenção personalizada
conhecimento inovação
presente qualidade
desenvolvimento sustentabilidade

tech universidade
tecnológica

Programa Avançado Engenharia de Software

- » Modalidade: online
- » Duração: 6 meses
- » Certificado: TECH Universidade Tecnológica
- » Dedicção: 16h/semana
- » Horário: no seu próprio ritmo
- » Provas: online

Programa Avançado Engenharia de Software

