

Master Privato

Ingegneria del Software Avanzata

```
back the deselected mirror modifier object
```

```
jects.active = modifier_ob  
str(modifier_ob)) # modifier ob is the active ob
```

```
select =  
context.select(projects[0])
```



tech università
tecnologica

Master Privato Ingegneria del Software Avanzata

- » Modalità: online
- » Durata: 12 mesi
- » Titolo: TECH Università
Tecnologica
- » Orario: a scelta
- » Esami: online

Accesso al sito web: www.techitute.com/it/informatica/master/master-ingegneria-software-avanzata

Indice

01

Presentazione

pag. 4

02

Obiettivi

pag. 8

03

Competenze

pag. 14

04

Struttura e contenuti

pag. 18

05

Metodologia

pag. 32

06

Titolo

pag. 40

01

Presentazione

Lo sviluppo di programmi informatici non sarebbe possibile se non esistesse l'ingegneria del software. Grazie ai progressi tecnologici, oggi è possibile trovare sistemi e strutture sempre più complessi e specifici, progettati sulla base di indicazioni di programmazione che soddisfano le esigenze e le richieste del mercato. Si tratta di un processo metodico e ordinato che richiede una conoscenza specialistica dell'informatica e dei suoi strumenti, che lo studente sarà in grado di acquisire nel corso di questo programma esaustivo. Grazie a un'esperienza multidisciplinare, potrai conoscere a fondo i principali processi informatici e l'integrazione dei sistemi, e acquisirai le competenze necessarie per realizzare progetti di alto livello a seconda delle esigenze degli *stakeholders*. Il tutto mediante una preparazione 100% online della durata di soli 12 mesi.





“

Se sei alla ricerca di un programma che ti consenta di creare il tuo progetto software da zero, questo è il Master Privato che fa per te. Cosa aspetti a iscriverti?”

Da oltre sessant'anni, l'ingegneria del software è il fulcro della rivoluzione tecnologica grazie allo sviluppo di software e applicazioni sempre più complessi e specializzati. Si tratta di un settore che è servito da supporto a molti altri per avanzare verso il progresso e la cui applicazione può essere esportata praticamente a tutte le specialità esistenti: medicina, agricoltura, insegnamento, amministrazione, industria, ecc. Da qualsiasi punto di vista lo si guardi, anche il più semplice processo informatico, come l'invio di un'e-mail o l'uso della messaggistica istantanea, che al giorno d'oggi è francamente un fatto quotidiano, richiede una progettazione e una programmazione approfondite per raggiungere il suo scopo: soddisfare le esigenze degli esseri umani.

L'ampia gamma di opportunità derivanti da questa scienza e le sue molteplici applicazioni la rendono una delle più ricercate del mercato del lavoro, non solo per la creazione di nuovi progetti, ma anche per la supervisione, la manutenzione e l'aggiornamento di quelli esistenti. TECH, fedele alla massima di offrire a tutti i suoi studenti la possibilità di specializzarsi in questo campo, ha deciso di lanciare questo programma completo in Ingegneria del Software Avanzata.

Un'esperienza accademica che offre 1.500 ore dei migliori contenuti teorico-pratici e complementari, coprendo l'intero settore, dall'origine alla progettazione, creazione e gestione di sistemi informativi innovativi e moderni. Durante i 12 mesi del programma, l'informatico potrà conoscere a fondo i dettagli di questa specialità: i suoi requisiti tecnici e strutturali, le chiavi per creare architetture sicure, l'integrazione di servizi basati sulle TIC, la gestione degli *stakeholders* e del loro ambito di intervento, lo sviluppo di un progetto dalla base al suo lancio, e molto altro ancora!

Il tutto in un formato 100% online, grazie al quale lo studente potrà avere accesso ai contenuti di questo Master Privato quando e da dove vuole, senza lezioni frontali o orari prefissati. Inoltre, potrà accedere al Campus Virtuale da qualsiasi dispositivo dotato di connessione a internet, sia esso un PC, un tablet o un telefono cellulare. Si tratta quindi di un'opportunità unica per specializzarsi in Ingegneria del Software grazie a un corso in grado di soddisfare le tue esigenze accademiche e i requisiti più esigenti dell'industria informatica di oggi.

Questo **Master Privato in Ingegneria del Software Avanzata** possiede il programma più completo e aggiornato del mercato. Le caratteristiche principali del programma sono:

- ◆ Lo sviluppo di casi di studio presentati da esperti in Ingegneria Informatica
- ◆ I contenuti grafici, schematici ed eminentemente pratici con cui sono stati concepiti forniscono informazioni pratiche riguardo alle discipline essenziali per l'esercizio della professione
- ◆ Esercizi pratici che offrono un processo di autovalutazione per migliorare l'apprendimento
- ◆ La sua speciale enfasi sulle metodologie innovative
- ◆ Lezioni teoriche, domande all'esperto, forum di discussione su questioni controverse e compiti di riflessione individuali
- ◆ Contenuti disponibili da qualsiasi dispositivo fisso o mobile dotato di connessione a internet



*Una qualifica che approfondisce i
fondamenti dell'Ingegneria del Software:
dalle sue origini fino ai processi di
calcolo che vengono eseguiti oggi"*

“

Avrai a disposizione una guida pratica che riguarda i principi dell'ingegneria del software, dal processo iniziale fino alla costruzione e allo sviluppo"

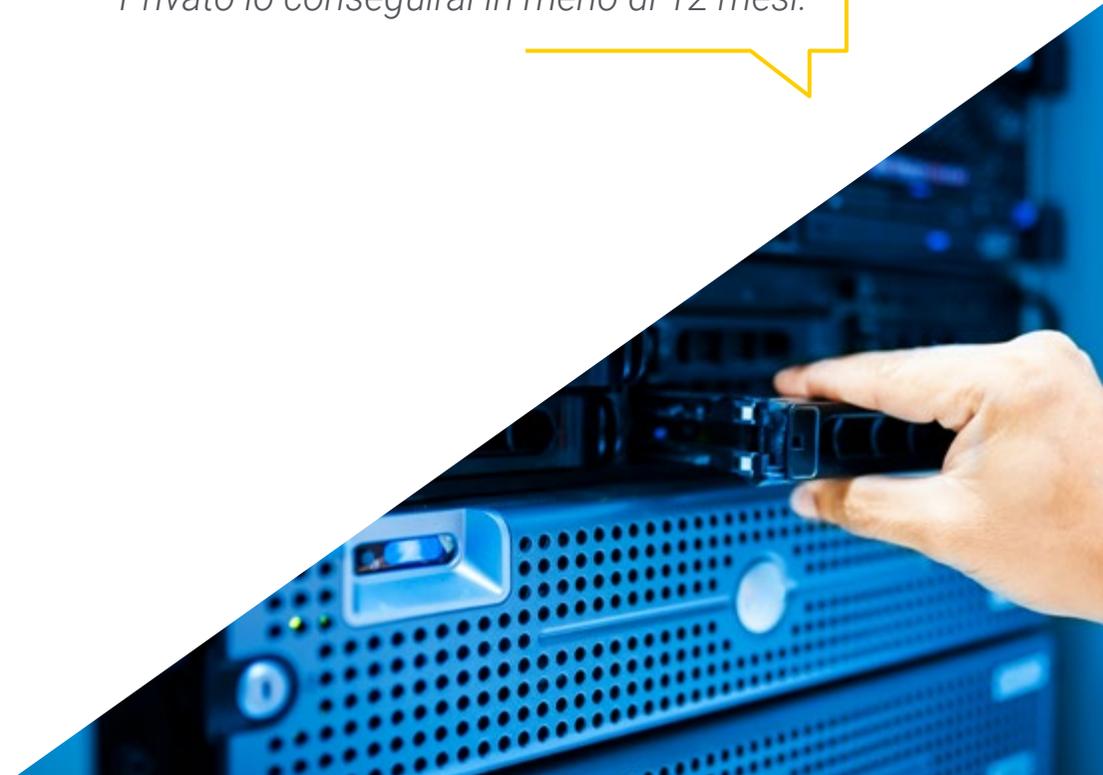
Il personale docente del programma comprende prestigiosi professionisti che apportano la propria esperienza, così come specialisti riconosciuti e appartenenti a società scientifiche di primo piano.

I contenuti multimediali, sviluppati in base alle ultime tecnologie educative, forniranno al professionista un apprendimento coinvolgente e localizzato, ovvero inserito in un contesto reale.

La progettazione di questo programma è incentrata sull'Apprendimento Basato sui Problemi, mediante il quale il professionista deve cercare di risolvere le diverse situazioni di pratica professionale che gli si presentano durante il corso. Sarai supportato da un innovativo sistema video interattivo sviluppato da esperti rinomati.

Un programma progettato per consentirti di applicare fin da subito le strategie più innovative in modellazione dei requisiti.

Ti piacerebbe acquisire le competenze necessarie per progettare architetture complesse e alternative mediante il flusso di dati? Grazie a questo Master Privato lo conseguirai in meno di 12 mesi.



02 Obiettivi

L'Ingegneria del Software è divenuta uno degli strumenti principali dello sviluppo tecnologico odierno. Senza il lavoro di milioni di professionisti dell'informatica, non sarebbe stato possibile raggiungere l'altissimo livello di innovazione digitale che conosciamo oggi. L'obiettivo di questo Master Privato è dunque quello di fornire agli studenti interessati a questo settore, le informazioni necessarie per conoscerlo nel dettaglio e per essere in grado di sviluppare i propri programmi e applicazioni con successo, efficienza e con la massima qualità e rigore.





“

Se uno dei tuoi obiettivi è quello di padroneggiare Scrum e altre tecniche di metodologia agile, questo Master Privato di TECH ti fornirà le garanzie necessarie e tutto ciò che ti serve per raggiungerlo”



Obiettivi generali

- ◆ Preparare scientificamente e tecnologicamente, nonché arricchire la pratica professionale nell'Ingegneria del Software, il tutto attraverso un'esperienza accademica trasversale e versatile adeguata alle nuove tecnologie e alle innovazioni del settore.
- ◆ Acquisirai conoscenze approfondite nel campo dell'Ingegneria del Software, ma anche nel campo dell'informatica e della struttura dei computer, comprese le basi matematiche, statistiche e fisiche essenziali per l'ingegneria

“

Lavorerai intensamente sul testing del software mediante TDD, ATDD e BDD, in modo da essere in grado di costruire strutture software della massima qualità"





Obiettivi specifici

Modulo 1. Ingegneria del Software

- ◆ Gettare le basi dell'ingegneria del software e della modellazione, apprendendo i principali processi e concetti
- ◆ Comprendere il processo del software e i diversi modelli per il suo sviluppo, comprese le tecnologie agili
- ◆ Conoscere i principali standard relativi alla qualità del software e alla gestione dei progetti

Modulo 2. Ingegneria del Software Avanzata

- ◆ Conoscere in maniera approfondita le diverse metodologie agili utilizzate nell'Ingegneria del software
- ◆ Imparare a sviluppare utilizzando Scrum, la programmazione estrema e le tecniche di sviluppo software basate sul riuso
- ◆ Comprendere i concetti e i processi di progettazione del software, imparando anche a conoscere l'architettura di progettazione e la progettazione a livello di componenti e basata su modelli
- ◆ Familiarizzarsi con il concetto di DevOps e le sue pratiche principali
- ◆ Imparare a realizzare prove del software, con metodologie come Test Driven Development, Acceptance Test Driven Development, Behavior Driven Development, BDD e Cucumber
- ◆ Comprendere le diverse architetture di sistema e i modelli di progettazione del software, nonché l'architettura delle applicazioni cloud

Modulo 3. Ingegneria dei requisiti

- ◆ Comprendere l'ingegneria dei requisiti, il loro sviluppo, elaborazione, negoziazione e validazione
- ◆ Imparare la modellazione dei requisiti e i diversi elementi come scenari, informazioni, classi di analisi, flussi, comportamenti e modelli
- ◆ Comprendere l'importanza dell'ingegneria dei requisiti nel processo di sviluppo del software
- ◆ Imparare a eseguire l'analisi dei requisiti e a documentarli correttamente
- ◆ Approfondire la comprensione delle fonti dei requisiti e delle tecniche di elicitazione dei requisiti, in quanto parte essenziale del processo
- ◆ Comprendere i processi di convalida e negoziazione dei requisiti, nonché la loro modellazione e gestione
- ◆ Acquisire le conoscenze necessarie per la gestione dei sistemi critici e la specificazione formale dei requisiti

Modulo 4. Processi di Ingegneria del Software

- ◆ Approfondire le tecniche di miglioramento del processo di sviluppo del software e della qualità del software utilizzando gli standard ISO/IEC
- ◆ Comprendere e applicare la prototipazione come parte essenziale del processo di sviluppo
- ◆ Conoscere il quadro di riferimento dell'Ingegneria del software e lo standard ISO/IEC 12207
- ◆ Imparare le caratteristiche del processo di sviluppo software unificato e della pianificazione nel contesto dello sviluppo software agile
- ◆ Comprendere i diversi stili di progettazione del software distribuito e delle architetture software orientate ai servizi
- ◆ Imparare i concetti essenziali della progettazione di interfacce grafiche per l'utente
- ◆ Comprendere le basi dello sviluppo di applicazioni web

Modulo 5. Qualità e revisioni di sistemi informativi

- ◆ Studiare in modo approfondito le strategie e le tecniche di test del software, i fattori di qualità del software e le diverse metriche utilizzate
- ◆ Acquisire le conoscenze essenziali sui sistemi di gestione della sicurezza informatica
- ◆ Introdurre i concetti di proprietà intellettuale nei sistemi di gestione delle informazioni
- ◆ Preparare gli studenti alla creazione di piani di continuità aziendale e di disaster recovery
- ◆ Imparare a pianificare la gestione della sicurezza e a gestire i principali meccanismi di protezione del patrimonio informativo
- ◆ Comprendere i diversi tipi di revisione e il processo svolto durante l'audit in informatica

Modulo 6. Integrazione dei sistemi

- ◆ Acquisire i concetti essenziali relativi ai sistemi informativi aziendali e identificare le opportunità e le esigenze dei sistemi informativi aziendali
- ◆ Conoscere le basi della *Business Intelligence*, le sue strategie, la sua implementazione, nonché il suo presente e il suo futuro
- ◆ Comprendere il funzionamento dei sistemi integrati di gestione delle risorse aziendali
- ◆ Comprendere la trasformazione digitale dal punto di vista dell'innovazione aziendale, della gestione finanziaria e produttiva, del marketing e della gestione delle risorse umane

Modulo 7. Riuso del software

- ◆ Conoscere il quadro generale della strategia di riuso del software
- ◆ Imparare i diversi modelli relativi al riuso del software in termini di progettazione, creazione, struttura e comportamento
- ◆ Introdurre il concetto di *framework*, nonché i principali tipi, come quelli per la progettazione di interfacce grafiche, lo sviluppo di applicazioni web e la gestione della persistenza degli oggetti nei database
- ◆ Capire come funziona il pattern Model-View-Controller (MVC), attualmente molto diffuso

Modulo 8. Servizi di Tecnologia dell'informazione

- ◆ Preparare al processo decisionale di investimento nelle TIC e alla pianificazione dei sistemi informativi
- ◆ Conoscere gli obiettivi di controllo per le tecnologie dell'informazione e le tecnologie correlate (COBIT)
- ◆ Apprendere il funzionamento dell'Information Technology Infrastructure Library (ITIL), le strategie, la progettazione dei servizi, le transizioni e le operazioni
- ◆ Approfondire le conoscenze del sistema di gestione dei servizi, in particolare dei principi di base della norma UNE-ISO/IEC 20000-1, della struttura della serie di norme ISO/IEC 20000 e dei requisiti del sistema di gestione dei servizi (SMS)
- ◆ Comprendere il funzionamento dei sistemi e delle tecnologie informatiche, i loro componenti, le classificazioni, le architetture e le forme di integrazione dei sistemi
- ◆ Imparare lo standard ISO/IEC 12207, l'analisi, la progettazione, l'implementazione e l'accettazione dei sistemi informativi

Modulo 9 Sicurezza dei sistemi informativi

- ◆ Imparare lo sviluppo della pianificazione per la gestione del tempo, lo sviluppo del budget e la risposta ai rischi
- ◆ Analizzare la natura degli attacchi sul web e i diversi tipi di architetture di sicurezza
- ◆ Comprendere le diverse tecniche di protezione del sistema e di sviluppo sicuro del codice
- ◆ Comprendere i componenti essenziali di *botnets* e spam, nonché del malware e del codice maligno
- ◆ Gettare le basi per l'analisi forense nel mondo del software e dell'audit informatico
- ◆ Acquisire una prospettiva globale sulla sicurezza, la crittografia e la crittoanalisi classica
- ◆ Comprendere i fondamenti della crittografia simmetrica e asimmetrica, nonché i loro principali algoritmi

Modulo 10 Gestione dei progetti

- ◆ Comprendere il funzionamento della gestione della qualità dei progetti, compresi la pianificazione, la garanzia, il controllo, i concetti statistici e gli strumenti disponibili
- ◆ Comprendere il funzionamento dei processi di approvvigionamento, esecuzione, monitoraggio, controllo e chiusura di un progetto
- ◆ Acquisire le conoscenze essenziali relative alla responsabilità professionale nella gestione dei progetti
- ◆ Comprendere i concetti fondamentali del project management e del suo ciclo di vita
- ◆ Comprendere le diverse fasi della gestione del progetto, come l'avvio, la pianificazione, la gestione degli *stakeholders* e lo scoping

03

Competenze

Una delle caratteristiche più significative di tutte le qualifiche offerte da TECH è quella di consentire agli studenti di perfezionare le proprie competenze nel corso dei loro studi. Ciò è possibile grazie all'acquisizione di conoscenze specialistiche della materia, in questo caso l'Ingegneria del Software. Tuttavia, la chiave per avere la garanzia di raggiungere questo obiettivo risiede nella risoluzione di casi d'uso basati su situazioni reali del panorama informatico attuale, grazie ai quali gli studenti possono mettere in pratica le loro conoscenze, applicare le strategie delineate nel programma e ampliare le loro competenze nella progettazione, nell'analisi, nella gestione e nel lancio di applicazioni e programmi.



“

Il miglior programma del mercato accademico attuale, che ti permetterà di implementare le tecniche di prototipazione più innovative ed efficaci dell'ambiente software nel tuo lavoro come informatico"

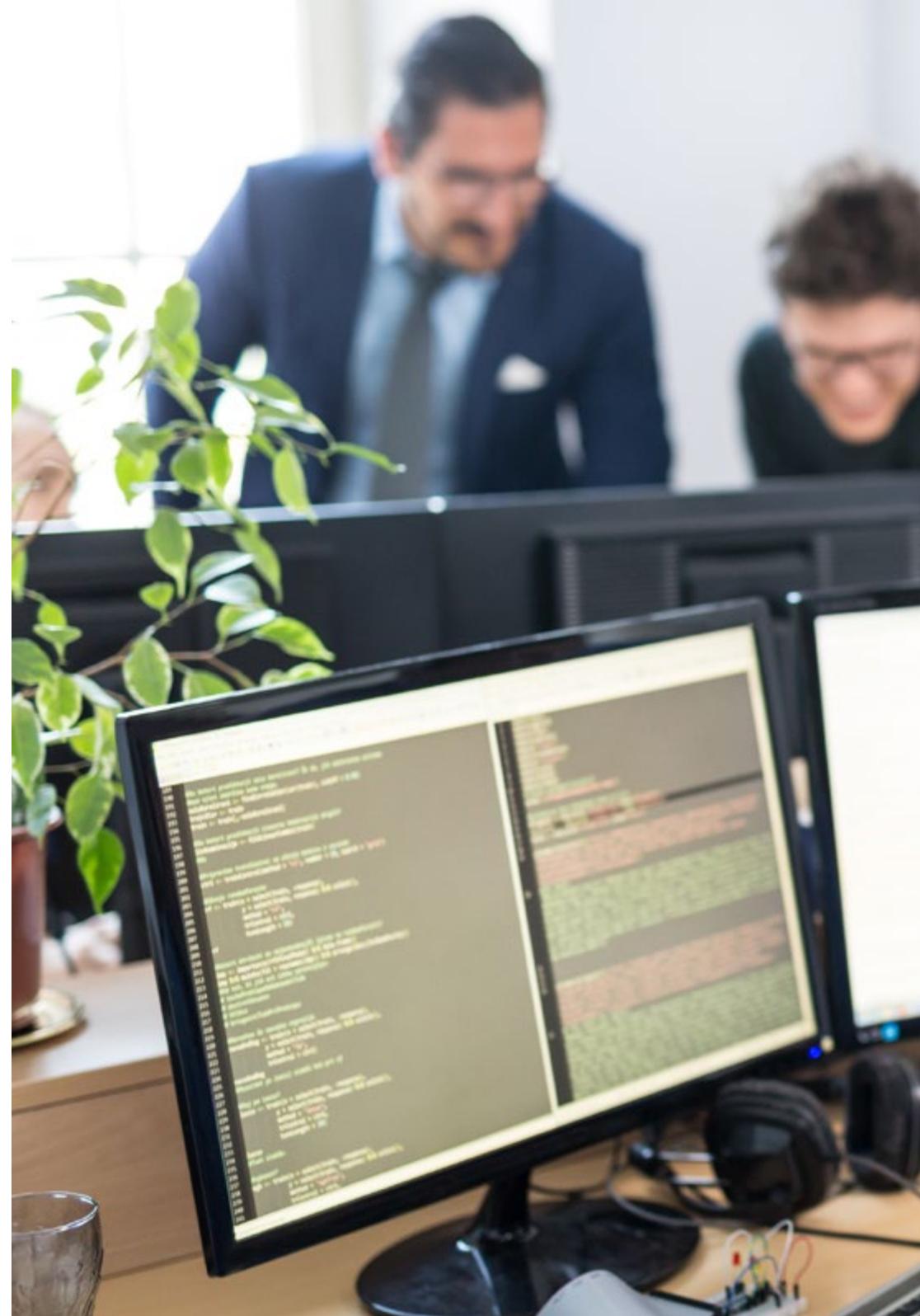


Competenze generali

- ◆ Rispondere alle necessità attuali dell'area dell'Ingegneria del Software Avanzata
- ◆ Padroneggiare i diversi sistemi di lavoro dell'Ingegneria del Software Avanzata
- ◆ Descrivere e sfruttare il software libero e la conoscenza aperta disponibile sul web

“

Grazie alla conoscenza delle strategie e delle tecniche di testing del software, sarai in grado di lavorare al miglioramento di progetti altrui per renderli più intuitivi e specializzati”





Competenze specifiche

- ◆ Acquisire una comprensione approfondita di tutti gli aspetti dell'interazione uomo-computer e di come questi coinvolgano gli sviluppi informatici
- ◆ Essere abili nell'uso delle banche dati
- ◆ Sviluppare diversi tipi di applicazioni in rete
- ◆ Lavorare come ingegnere del software
- ◆ Controllare l'uso di database avanzati
- ◆ Eseguire una programmazione avanzata
- ◆ Conoscere il riuso del software
- ◆ Creazione di interfacce e applicazioni in rete

04

Struttura e contenuti

Il programma di studi è stato progettato sulla base dell'efficacia educativa, selezionando con attenzione i contenuti per offrire un percorso completo, che includa i campi di studio imprescindibili per ottenere una reale conoscenza della materia. Con gli aggiornamenti e gli aspetti più innovativi del settore.



“

Tutte le materie e le aree di conoscenza sono state riunite in un programma di studio completo e aggiornato, al fine di consentire allo studente di raggiungere il massimo livello di conoscenza teorica e pratica"

Modulo 1. Ingegneria del Software

- 1.1. Introduzione all'Ingegneria del Software e alla modellazione
 - 1.1.1. La natura del software
 - 1.1.2. La natura unica delle web app
 - 1.1.3. Ingegneria del Software
 - 1.1.4. Il processo del software
 - 1.1.5. La pratica dell'Ingegneria del Software
 - 1.1.6. Miti del software
 - 1.1.7. Come tutto ebbe inizio
 - 1.1.8. Concetti orientati agli oggetti
 - 1.1.9. Introduzione a UML
- 1.2. Il processo del software
 - 1.2.1. Un modello generale di processo
 - 1.2.2. Modelli di processo prescrittivi
 - 1.2.3. Modelli di processo specializzato
 - 1.2.4. Il processo unificato
 - 1.2.5. Modelli di processo personali e di gruppo
 - 1.2.6. Che cos'è l'agilità?
 - 1.2.7. Che cos'è un processo agile?
 - 1.2.8. Scrum
 - 1.2.9. Kit di strumenti per i processi agili
- 1.3. Principi che guidano la pratica dell'Ingegneria del Software
 - 1.3.1. Principi che guidano il processo
 - 1.3.2. Principi che guidano la pratica
 - 1.3.3. Principi di comunicazione
 - 1.3.4. Principi di pianificazione
 - 1.3.5. Principi di modellazione
 - 1.3.6. Principi di costruzione
 - 1.3.7. Principi di implementazione
- 1.4. Comprendere i requisiti
 - 1.4.1. Ingegneria dei requisiti
 - 1.4.2. Gettare le basi
 - 1.4.3. Indagare i requisiti
 - 1.4.4. Sviluppo di casi d'uso
 - 1.4.5. Elaborazione del modello dei requisiti
 - 1.4.6. Negoziazione dei requisiti
 - 1.4.7. Convalida dei requisiti
- 1.5. Modellazione dei requisiti I: scenari, informazioni e classi di analisi
 - 1.5.1. Analisi dei requisiti
 - 1.5.2. Modellazione basata su scenari
 - 1.5.3. Modelli UML che forniscono il caso d'uso
 - 1.5.4. Concetti di modellazione dei dati
 - 1.5.5. Modellazione basata su classi
 - 1.5.6. Diagrammi di classe
- 1.6. Modellazione dei requisiti II: flusso, comportamento e modelli
 - 1.6.1. Strategie di definizione dei requisiti
 - 1.6.2. Modellazione orientata al flusso
 - 1.6.3. Diagrammi di stato
 - 1.6.4. Creazione di una modellazione del comportamento
 - 1.6.5. Diagrammi di sequenza
 - 1.6.6. Diagrammi di comunicazione
 - 1.6.7. Linee guida per la modellazione dei requisiti
- 1.7. Concetti di progettazione
 - 1.7.1. La progettazione nel contesto dell'Ingegneria del Software
 - 1.7.2. Il processo di progettazione
 - 1.7.3. Concetti di progettazione
 - 1.7.4. Concetti di progettazione orientati agli oggetti
 - 1.7.5. Il modello di progettazione

- 1.8. Progettazione dell'architettura
 - 1.8.1. Architettura del software
 - 1.8.2. Generi architettonici
 - 1.8.3. Stili architettonici
 - 1.8.4. Progettazione architettonica
 - 1.8.5. Evoluzione dei progetti alternativi per l'architettura
 - 1.8.6. Mappatura dell'architettura utilizzando i flussi di dati
- 1.9. Progettazione a livello di componente e basata su modelli
 - 1.9.1. Che cos'è un componente?
 - 1.9.2. Progettazione di componenti basati su classi
 - 1.9.3. Realizzazione del progetto a livello di componenti
 - 1.9.4. Progettazione di componenti tradizionali
 - 1.9.5. Sviluppo basato su componenti
 - 1.9.6. Modelli di disegno
 - 1.9.7. Progettazione di software basato su modelli
 - 1.9.8. Modelli architettonici
 - 1.9.9. Modelli di progettazione a livello di componenti
 - 1.9.10. Modelli di progettazione dell'interfaccia utente
- 1.10. Qualità del software e gestione dei progetti
 - 1.10.1. Qualità
 - 1.10.2. Qualità del software
 - 1.10.3. Il dilemma della qualità del software
 - 1.10.4. Raggiungere la qualità del software
 - 1.10.5. Garanzie della qualità del software
 - 1.10.6. Lo spettro amministrativo
 - 1.10.7. Il personale
 - 1.10.8. Il prodotto
 - 1.10.9. Il processo
 - 1.10.10. Il progetto
 - 1.10.11. Principi e pratiche

Modulo 2. Ingegneria del Software Avanzata

- 2.1. Introduzione alle Metodologie Agili
 - 2.1.1. Modelli di processo e metodologie
 - 2.1.2. Agilità e processi agili
 - 2.1.3. Manifesto agile
 - 2.1.4. Alcune metodologie agili
 - 2.1.5. Agile vs. Tradizionale
- 2.2. Scrum
 - 2.2.1. Origini e filosofia di Scrum
 - 2.2.2. Valori di Scrum
 - 2.2.3. Flusso del processo Scrum
 - 2.2.4. Ruoli di Scrum
 - 2.2.5. Prodotti Scrum
 - 2.2.6. Eventi Scrum
 - 2.2.7. Storie dell'utente
 - 2.2.8. Estensioni Scrum
 - 2.2.9. Stime agili
 - 2.2.10. Scala Scrum
- 2.3. Programmazione estrema
 - 2.3.1. Giustificazione e panoramica di XP
 - 2.3.2. Il ciclo di vita in XP
 - 2.3.3. I cinque valori fondamentali
 - 2.3.4. Le dodici pratiche di base di XP
 - 2.3.5. Ruoli dei partecipanti
 - 2.3.6. XP Industriale
 - 2.3.7. Valutazione critica di XP
- 2.4. Sviluppo software basato sul riuso
 - 2.4.1. Riuso del software
 - 2.4.2. Livelli di riuso del codice
 - 2.4.3. Tecniche di riuso specifiche
 - 2.4.4. Sviluppo basato su componenti
 - 2.4.5. Benefici e problemi del riuso
 - 2.4.6. Pianificazione del riuso

- 2.5. Architettura di sistema e modelli di progettazione del software
 - 2.5.1. Progettazione architettonica
 - 2.5.2. Modelli architettonici generici
 - 2.5.3. Architetture con tolleranza ai guasti
 - 2.5.4. Architetture di sistema distribuito
 - 2.5.5. Modelli di progettazione
 - 2.5.6. Modelli di Gamma
 - 2.5.7. Modelli di progettazione dell'interazione
- 2.6. Architettura di applicazioni in cloud
 - 2.6.1. Fondamenti di Cloud Computing
 - 2.6.2. Qualità di applicazioni in cloud
 - 2.6.3. Stili di architetture
 - 2.6.4. Modelli di progettazione
- 2.7. Test del software: TDD, ATDD e BDD
 - 2.7.1. Verifica e validazione del software
 - 2.7.2. Test del 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. Miglioramenti dei processi di software
 - 2.8.1. Miglioramenti dei processi di software
 - 2.8.2. Il miglioramento dei processi
 - 2.8.3. Modelli di maturità
 - 2.8.4. Il Modello CMMI
 - 2.8.5. CMMI V2.0
 - 2.8.6. CMMI e Agile

- 2.9. Qualità del prodotto software: SQuaRE
 - 2.9.1. La qualità del software
 - 2.9.2. Modelli di qualità del prodotto software
 - 2.9.3. Famiglia ISO/IEC 25000
 - 2.9.4. ISO/IEC 25010: modello e caratteristiche di qualità
 - 2.9.5. ISO/IEC 25012: la qualità dei dati
 - 2.9.6. ISO/IEC 25020: misurazione della qualità del software
 - 2.9.7. ISO/IEC 25022, 25023 y 25024: metriche della qualità del software e dei dati
 - 2.9.8. ISO/IEC 25040: valutazione del software
 - 2.9.9. Processo di certificazione
- 2.10. Introduzione a DevOps
 - 2.10.1. Concetto di DevOps
 - 2.10.2. Pratiche fondamentali

Modulo 3. Ingegneria dei requisiti

- 3.1. Introduzione all'Ingegneria dei Requisiti
 - 3.1.1. L'importanza dei requisiti
 - 3.1.2. Concetto di requisito
 - 3.1.3. Dimensioni dei requisiti
 - 3.1.4. Livelli e tipi di requisiti
 - 3.1.5. Caratteristiche dei requisiti
 - 3.1.6. Ingegneria dei requisiti
 - 3.1.7. Il processo di Ingegneria dei Requisiti
 - 3.1.8. *Frameworks* per Ingegneria dei Requisiti
 - 3.1.9. Buone pratiche in Ingegneria dei Requisiti
 - 3.1.10. Analista di business
- 3.2. Le fonti dei requisiti
 - 3.2.1. Le reti dei requisiti
 - 3.2.2. Gli *stakeholder*
 - 3.2.3. I requisiti del business
 - 3.2.4. Documento di visione e scoping

- 3.3. Tecniche di elicitazione dei requisiti
 - 3.3.1. La elicitazione dei requisiti
 - 3.3.2. Problemi di elicitazione dei requisiti
 - 3.3.3. Contesti di scoperta
 - 3.3.4. Interviste
 - 3.3.5. Osservazione e “apprendimento”
 - 3.3.6. Etnografia
 - 3.3.7. Workshop
 - 3.3.8. *Focus group*
 - 3.3.9. Questionari
 - 3.3.10. *Brainstorming* tecniche di creatività
 - 3.3.11. Mezzi di gruppo
 - 3.3.12. Analisi delle interfacce del sistema
 - 3.3.13. Analisi dei documenti e “archeologia”
 - 3.3.14. Casi d'uso e scenari
 - 3.3.15. I prototipi
 - 3.3.16. Ingegneria inversa
 - 3.3.17. Riutilizzo dei requisiti
 - 3.3.18. Buone pratiche di elicitazione
- 3.4. Requisiti degli utenti
 - 3.4.1. Persone
 - 3.4.2. Casi di uso e storie dell'utente
 - 3.4.3. Scenari
 - 3.4.4. Tipi di scenari
 - 3.4.5. Come scoprire scenari
- 3.5. Tecniche di prototipazione
 - 3.5.1. Prototipazione
 - 3.5.2. Prototipi in base alla loro portata
 - 3.5.3. Prototipi in base alla loro temporalità
 - 3.5.4. La fedeltà di un prototipo
 - 3.5.5. Prototipi di interfaccia utente
 - 3.5.6. Valutazione dei prototipi
- 3.6. Analisi dei requisiti
 - 3.6.1. Analisi dei requisiti
 - 3.6.2. Buone pratiche dell'analisi dei requisiti
 - 3.6.3. Il dizionario dei dati
 - 3.6.4. Priorizzazione dei requisiti
- 3.7. Documentazione dei requisiti
 - 3.7.1. Il documento di specifica dei requisiti
 - 3.7.2. Struttura e contenuti di un SRS
 - 3.7.3. Documentazione in linguaggio naturale
 - 3.7.4. EARS: Easy Approach to Requirements Syntax
 - 3.7.5. Requisiti non funzionali
 - 3.7.6. Attributi e modelli in forma di tabella
 - 3.7.7. Buone prassi di specificazione
- 3.8. Convalida e negoziazione dei requisiti
 - 3.8.1. Convalida dei requisiti
 - 3.8.2. Tecniche di convalida dei requisiti
 - 3.8.3. Negoziazione dei requisiti
- 3.9. Modellazione e gestione di requisiti
 - 3.9.1. Modellazione dei requisiti
 - 3.9.2. La prospettiva dell'utente
 - 3.9.3. La prospettiva dei dati
 - 3.9.4. La prospettiva funzionale o orientata al flusso
 - 3.9.5. La prospettiva del comportamento
 - 3.9.6. La volatilità dei requisiti
 - 3.9.7. Processo di gestione dei requisiti
 - 3.9.8. Strumenti per la gestione dei requisiti
 - 3.9.9. Buone pratiche nella gestione dei requisiti
- 3.10. Sistemi critici e specifiche formali
 - 3.10.1. Sistemi critici
 - 3.10.2. Specifiche basate sul rischio
 - 3.10.3. Specifica formale

Modulo 4. Processi di Ingegneria del Software

- 4.1. Struttura dell'Ingegneria del software
 - 4.1.1. Caratteristiche del software
 - 4.1.2. I principali processi dell'Ingegneria del software
 - 4.1.3. Modellazione del processo di sviluppo software
 - 4.1.4. Quadro di riferimento standard per il processo di sviluppo del software: lo standard ISO/IEC 12207
- 4.2. Processo di sviluppo software unificato
 - 4.2.1. Il processo unificato
 - 4.2.2. Dimensioni del processo unificato
 - 4.2.3. Processo di sviluppo guidato dai casi d'uso
 - 4.2.4. Flussi di lavoro fondamentali del processo unificato
- 4.3. Pianificazione nel contesto dello sviluppo agile del software
 - 4.3.1. Caratteristiche dello sviluppo agile del software
 - 4.3.2. Diversi orizzonti temporali di pianificazione nello sviluppo agile
 - 4.3.3. Quadro di sviluppo agile Scrum e pianificazione degli orizzonti temporali
 - 4.3.4. Le storie utente come unità di pianificazione e stima
 - 4.3.5. Tecniche comuni per ricavare una stima
 - 4.3.6. Scale di interpretazione delle stime
 - 4.3.7. *Planning poker*
 - 4.3.8. Tipi di pianificazione comuni: pianificazione della consegna e pianificazione dell'iterazione
- 4.4. Comprendere stili di progettazione del software distribuito e delle architetture software orientate ai servizi
 - 4.4.1. Modelli di comunicazione nei sistemi software distribuiti
 - 4.4.2. Livello intermedio o middleware
 - 4.4.3. Modelli di architettura per sistemi distribuiti
 - 4.4.4. Processo generale di progettazione dei servizi software
 - 4.4.5. Aspetti della progettazione dei servizi software
 - 4.4.6. Composizione dei servizi
 - 4.4.7. Architettura dei servizi web
 - 4.4.8. Componenti infrastrutturali e SOA
- 4.5. Introduzione allo sviluppo di software guidato da modelli
 - 4.5.1. Il concetto di modello
 - 4.5.2. Sviluppo di software guidato da modelli
 - 4.5.3. Quadro di sviluppo guidato dal modello MDA
 - 4.5.4. Elementi di un modello di trasformazione
- 4.6. Progettazione dell'interfaccia grafica utente
 - 4.6.1. Principi di progettazione interfaccia utente
 - 4.6.2. Modelli di progettazione architettonica per sistemi interattivi: Model view controller (MVC)
 - 4.6.3. Esperienza utente (UX User Experience)
 - 4.6.4. Progettazione incentrata sull'utente
 - 4.6.5. Processo di analisi e progettazione dell'interfaccia grafica utente
 - 4.6.6. Usabilità dell'interfaccia utente
 - 4.6.7. Accessibilità nelle interfacce utente
- 4.7. Disegno delle applicazioni web
 - 4.7.1. Caratteristiche delle applicazioni web
 - 4.7.2. Interfaccia utente di un'applicazione web
 - 4.7.3. Progettazione della navigazione
 - 4.7.4. Protocollo di interazione di base per le applicazioni web
 - 4.7.5. Stili di architettura per applicazioni web
- 4.8. Strategie e tecniche di test del software e fattori di qualità del software
 - 4.8.1. Strategie di test
 - 4.8.2. Progettazione dei casi di test
 - 4.8.3. Rapporto qualità/prezzo
 - 4.8.4. Modelli di qualità
 - 4.8.5. Famiglia di norme ISO/IEC 25000 (SQuaRE)
 - 4.8.6. Modelli di qualità di prodotto (ISO 2501n)
 - 4.8.7. Modelli di qualità di dati (ISO 2501n)
 - 4.8.8. Gestione della qualità del software

- 4.9. Introduzione alle metriche nell'Ingegneria del software
 - 4.9.1. Concetti di base: misure, metriche e indicatori
 - 4.9.2. Tipi di metriche nell'ingegneria del software
 - 4.9.3. Processo di misurazione
 - 4.9.4. ISO 25024. Metriche esterne e di qualità in uso
 - 4.9.5. Metriche orientate agli oggetti
- 4.10. Manutenzione e reingegnerizzazione del software
 - 4.10.1. Processo di manutenzione
 - 4.10.2. Quadro di riferimento per i processi di manutenzione standard. ISO/EIEC 14764
 - 4.10.3. Modellazione del processo di reingegnerizzazione software
 - 4.10.4. Ingegneria inversa

Modulo 5. Qualità e revisioni di sistemi informativi

- 5.1. Introduzione ai Sistemi di Gestione di Sicurezza delle Informazioni (SGSI)
 - 5.1.1. Principi dei SGSI
 - 5.1.2. Regole d'oro dei SGSI
 - 5.1.3. Ruolo dell'audit informatico nei SGSI
- 5.2. Pianificazione della gestione della sicurezza
 - 5.2.1. Concetti legati alla gestione della sicurezza
 - 5.2.2. Classificazione delle informazioni: obiettivi, concetti e ruoli
 - 5.2.3. Implementazione delle politiche di sicurezza: politiche, standard e procedure di sicurezza
 - 5.2.4. Gestione del rischio: principi e analisi del rischio degli asset informativi
- 5.3. Principali meccanismi di protezione del patrimonio informativo I
 - 5.3.1. Panoramica dei principali strumenti crittografici per la protezione della triade CID
 - 5.3.2. Considerazione dei requisiti di privacy, anonimato e corretta gestione della tracciabilità degli utenti
- 5.4. Principali meccanismi di protezione del patrimonio informativo II
 - 5.4.1. Sicurezza delle comunicazioni: protocolli, dispositivi e architetture di sicurezza
 - 5.4.2. Sicurezza dei sistemi operativi
- 5.5. Controlli interni dei SGSI
 - 5.5.1. Tassonomia dei controlli SGSI: controlli amministrativi, logici e fisici
 - 5.5.2. Classificazione dei controlli in base al modo in cui affrontano la minaccia: controlli per la prevenzione, il rilevamento e la correzione della minaccia
 - 5.5.3. Implementazione dei sistemi di controllo interno nei SGSI
- 5.6. Tipi di audit
 - 5.6.1. Differenza tra audit e controllo interno
 - 5.6.2. Audit interno ed esterno
 - 5.6.3. Classificazione dell'audit in base all'obiettivo e al tipo di analisi
- 5.7. Sceneggiatore e sceneggiatura: soggetto e oggetto protetti dal diritto della proprietà intellettuale
 - 5.7.1. Introduzione ai test di penetrazione e all'analisi forense
 - 5.7.2. Definizione e rilevanza dei concetti di *fingerprinting* e *footprinting*
- 5.8. Scansione delle vulnerabilità e monitoraggio del traffico di rete
 - 5.8.1. Strumenti di analisi delle vulnerabilità nei sistemi
 - 5.8.2. Principali vulnerabilità nel contesto delle applicazioni web
 - 5.8.3. Analisi dei protocolli di comunicazione
- 5.9. Il processo di audit informatico
 - 5.9.1. Il concetto di ciclo di vita nello sviluppo di sistemi
 - 5.9.2. Monitoraggio delle attività e dei processi: raccolta ed elaborazione delle prove
 - 5.9.3. Metodologia di un audit informatico
 - 5.9.4. Processo di un audit informatico
 - 5.9.5. Identificazione dei principali reati e illeciti nel contesto delle tecnologie informatiche
 - 5.9.6. Indagine sui crimini informatici: un'introduzione all'analisi forense e al suo rapporto con l'audit informatico
- 5.10. Piani di continuità aziendale e di disaster recovery
 - 5.10.1. Definizione di piano di continuità aziendale e concetto di interruzione dell'attività
 - 5.10.2. Raccomandazione NIST sulla pianificazione della continuità operativa
 - 5.10.3. Piano di disaster recovery
 - 5.10.4. Processo di un piano di disaster recovery

Modulo 6. Integrazione dei sistemi

- 6.1. Introduzione ai sistemi informativi nell'azienda
 - 6.1.1. Il ruolo dei sistemi informativi
 - 6.1.2. Cos'è un sistema informativo?
 - 6.1.3. Dimensioni dei sistemi informativi
 - 6.1.4. Processi aziendali e sistemi informativi
 - 6.1.5. Il dipartimento di SI/TI
- 6.2. Opportunità ed esigenze dei sistemi informativi nell'azienda
 - 6.2.1. Organizzazioni e sistemi informativi
 - 6.2.2. Caratteristiche dell'organizzazione
 - 6.2.3. Impatti delle sistemi informativi nell'azienda
 - 6.2.4. Sistemi informativi per il vantaggio competitivo
 - 6.2.5. Uso dei sistemi nell'amministrazione e nella gestione aziendale
- 6.3. Nozioni di base sui sistemi e sulle tecnologie dell'informazione
 - 6.3.1. Dati, informazioni e conoscenze
 - 6.3.2. Tecnologia e sistemi informativi
 - 6.3.3. Componenti della tecnologia
 - 6.3.4. Classificazione e tipi di sistemi informativi
 - 6.3.5. Architetture basate su servizi e processi aziendali
 - 6.3.6. Forme di integrazione del sistema
- 6.4. Sistemi integrati di gestione delle risorse aziendali
 - 6.4.1. Necessità d'impresa
 - 6.4.2. Un sistema informativo aziendale integrato
 - 6.4.3. Acquisizione vs. Sviluppo
 - 6.4.4. Implementazione dell'ERP
 - 6.4.5. Implicazioni per la gestione
 - 6.4.6. I principali fornitori di ERP
- 6.5. Sistemi informativi per la gestione della catena di approvvigionamento e delle relazioni con i clienti
 - 6.5.1. Definizione della catena di fornitura
 - 6.5.2. Gestione della catena di fornitura
 - 6.5.3. Il ruolo dei sistemi informativi
 - 6.5.4. Benefici della gestione della catena di fornitura
 - 6.5.5. La gestione dei rapporti con i clienti
 - 6.5.6. Il ruolo dei sistemi informativi
 - 6.5.7. Implementazione di un sistema CRM
 - 6.5.8. Fattori critici di successo nell'implementazione del CRM
 - 6.5.9. CRM, e-CRM e altre tendenze
- 6.6. Processo decisionale di investimento nelle TIC e pianificazione dei sistemi informativi
 - 6.6.1. Criteri per le decisioni di investimento nelle TIC
 - 6.6.2. Collegamento del progetto con la gestione e il piano aziendale
 - 6.6.3. Implicazioni della gestione
 - 6.6.4. Riprogettazione di processi aziendali
 - 6.6.5. Decisione del management sulle metodologie di implementazione
 - 6.6.6. Necessità di pianificazione dei sistemi informativi
 - 6.6.7. Obiettivi, partecipanti e momenti
 - 6.6.8. Struttura e sviluppo del piano dei sistemi
 - 6.6.9. Monitoraggio e aggiornamento
- 6.7. Considerazioni di sicurezza nell'uso delle TIC
 - 6.7.1. Analisi dei rischi
 - 6.7.2. Sicurezza dei sistemi informativi
 - 6.7.3. Consigli pratici
- 6.8. Fattibilità dell'implementazione di progetti ICT e aspetti finanziari dei progetti di sistemi informativi
 - 6.8.1. Descrizione e obiettivi
 - 6.8.2. Partecipanti nell'EVS
 - 6.8.3. Tecniche e pratiche
 - 6.8.4. Struttura dei costi
 - 6.8.5. La proiezione finanziaria
 - 6.8.6. Preventivi

- 6.9. *Management*
 - 6.9.1. Cos'è l'intelligenza del business?
 - 6.9.2. Strategia e implementazione della Business Intelligence (BI)
 - 6.9.3. Presente e futuro della BI
- 6.10. ISO/IEC 12207
 - 6.10.1. Cos'è la ISO/IEC 12207?
 - 6.10.2. Analisi dei Sistema Informativi
 - 6.10.3. Progettazione dei Sistema Informativi
 - 6.10.4. Implementazione e accettazione dei Sistema Informativi

Modulo 7. Riutilizzo del software

- 7.1. Quadro generale di riutilizzo del software
 - 7.1.1. Che cos'è il riutilizzo del software?
 - 7.1.2. Vantaggi e svantaggi del riutilizzo del software
 - 7.1.3. Tecniche principali di riutilizzo del software
- 7.2. Introduzione ai modelli di progettazione
 - 7.2.1. Cos'è un modello di progettazione?
 - 7.2.2. Catalogo dei principali modelli di progettazione
 - 7.2.3. Come utilizzare i modelli per risolvere i problemi di progettazione
 - 7.2.4. Come selezionare il miglior modello di progettazione
- 7.3. Modelli creativi I
 - 7.3.1. Modelli creativi
 - 7.3.2. Modello Abstract Factory
 - 7.3.3. Esempio di implementazione del Modello Abstract Factory
 - 7.3.4. Modello Builder
 - 7.3.5. Esempio di implementazione del Builder
 - 7.3.6. Modello Abstract Factory vs. Builder
- 7.4. Modelli creativi II
 - 7.4.1. Modello Factory Method
 - 7.4.2. Factory Method vs. Abstract Factory
 - 7.4.3. Modello Singleton
- 7.5. Modelli strutturali I
 - 7.5.1. Modelli strutturali
 - 7.5.2. Modello Adapter
 - 7.5.3. Modello Bridge
- 7.6. Modelli strutturali II
 - 7.6.1. Modello Composite
 - 7.6.2. Modello Decorator
- 7.7. Modelli strutturali III
 - 7.7.1. Modello Facade
 - 7.7.2. Modello Proxy
- 7.8. Modelli di comportamento I
 - 7.8.1. Concetto di modelli di comportamento
 - 7.8.2. Modello di comportamento: catena di responsabilità
 - 7.8.3. Modelli di comportamento Ordine
- 7.9. Modelli di comportamento II
 - 7.9.1. Modello Interprete o *Interpreter*
 - 7.9.2. Modello Iterativo
 - 7.9.3. Modello Osservatore
 - 7.9.4. Modello Strategia
- 7.10. *Framework*
 - 7.10.1. Concetto di *framework*
 - 7.10.2. Sviluppo mediante *framework*
 - 7.10.3. Model View Controller
 - 7.10.4. *Framework* per la progettazione dell'interfaccia grafica utente
 - 7.10.5. *Framework* per lo sviluppo di applicazioni web
 - 7.10.6. *Frameworks* per la gestione della persistenza degli oggetti nei database

Modulo 8. Servizi di Tecnologia dell'informazione

- 8.1. La trasformazione digitale I
 - 8.1.1. Innovazione aziendale
 - 8.1.2. Gestione della produzione
 - 8.1.3. Gestione finanziaria
- 8.2. La trasformazione digitale II
 - 8.2.1. Il marketing
 - 8.2.2. Gestione delle risorse umane
 - 8.2.3. Un sistema informativo integrato
- 8.3. Caso di studio
 - 8.3.1. Presentazione d'impresa
 - 8.3.2. Metodologie di analisi delle acquisizioni IT
 - 8.3.3. Determinazione di costi, benefici e rischi
 - 8.3.4. Valutazione economica dell'investimento
- 8.4. Governance e gestione delle TIC
 - 8.4.1. Definizione di governance delle tecnologie e dei sistemi informativi
 - 8.4.2. Differenza tra governance e gestione delle TSI
 - 8.4.3. Quadri per la governance e la gestione delle TSI
 - 8.4.4. Le norme e governance e gestione delle TSI
- 8.5. Governance aziendale delle TIC
 - 8.5.1. Che cos'è la buona governance aziendale?
 - 8.5.2. Il contesto della governance delle TIC
 - 8.5.3. Norma ISO/IEC 38500:2008
 - 8.5.4. Attuare una buona governance delle TIC
 - 8.5.5. Governance delle TIC e migliori pratiche
 - 8.5.6. La governance aziendale. Panoramica e tendenze
- 8.6. Obiettivi di controllo per l'informazione e le tecnologie correlate (COBIT)
 - 8.6.1. Quadro di applicazione
 - 8.6.2. Dominio: pianificazione e organizzazione
 - 8.6.3. Dominio: approvvigionamento e implementazione
 - 8.6.4. Dominio: consegna e supporto
 - 8.6.5. Dominio: monitoraggio e valutazione
 - 8.6.6. Applicazione della guida COBIT

- 8.7. La Biblioteca dell'Infrastruttura tecnologica dell'informazione (ITIL)
 - 8.7.1. Introduzione a ITIL
 - 8.7.2. Strategie dei servizi
 - 8.7.3. Progettazione del servizio
 - 8.7.4. Transizione del servizio
 - 8.7.5. Funzionamento del servizio
 - 8.7.6. Miglioramento del servizio
- 8.8. Sistema di gestione dei servizi
 - 8.8.1. Principi di base di UNE-ISO/IEC 20000-1
 - 8.8.2. Struttura della serie di norme ISO/IEC 20000
 - 8.8.3. Requisiti del Sistema di gestione dei servizi (SGS)
 - 8.8.4. Progettazione e transizione di servizi nuovi o modificati
 - 8.8.5. Processi di fornitura del servizio
 - 8.8.6. Gruppi di processo
- 8.9. Sistema di gestione di asset del software
 - 8.9.1. Giustificazione della necessità
 - 8.9.2. Antecedenti
 - 8.9.3. Presentazione della norma 19770
 - 8.9.4. Implementazione della gestione
- 8.10. Gestione della continuità del business
 - 8.10.1. Piano di continuità di un business
 - 8.10.2. Implementazione di un BCM

Modulo 9. Sicurezza dei sistemi informativi

- 9.1. Acquisire una prospettiva globale sulla sicurezza, la crittografia e la crittoanalisi classica
 - 9.1.1. Sicurezza informatica: una prospettiva storica
 - 9.1.2. Ma cosa si intende esattamente per sicurezza?
 - 9.1.3. Storia della crittografia
 - 9.1.4. Criptatori di sostituzione
 - 9.1.5. Caso di studio: la macchina Enigma

- 9.2. Crittografia simmetrica
 - 9.2.1. Introduzione e terminologia di base
 - 9.2.2. Crittografia simmetrica
 - 9.2.3. Modalità di funzionamento
 - 9.2.4. DES
 - 9.2.5. Il nuovo standard AES
 - 9.2.6. Crittografia nel flusso
 - 9.2.7. Crittoanalisi
- 9.3. Crittografia asimmetrica
 - 9.3.1. Origini della crittografia a chiave pubblica
 - 9.3.2. Concetti di base e funzionamento
 - 9.3.3. L'algoritmo RSA
 - 9.3.4. Certificati digitali
 - 9.3.5. Conservazione e gestione delle chiavi
- 9.4. Attacchi di rete
 - 9.4.1. Minacce e attacchi alla rete
 - 9.4.2. Enumerazione
 - 9.4.3. Intercettazione del traffico: *sniffers*
 - 9.4.4. Attacchi di negazione del servizio
 - 9.4.5. Attacchi ARP poisoning
- 9.5. Architettura di sicurezza
 - 9.5.1. Architettura di sicurezza tradizionale
 - 9.5.2. Secure Socket Layer: SSL
 - 9.5.3. Protocollo SSH
 - 9.5.4. Reti private virtuali (VPN)
 - 9.5.5. Meccanismi di protezione dell'unità di archiviazione esterna
 - 9.5.6. Meccanismi di protezione hardware
- 9.6. Tecniche di protezione del sistema e sviluppo di codice sicuro
 - 9.6.1. Sicurezza delle operazioni
 - 9.6.2. Risorse e controlli
 - 9.6.3. Monitoraggio
 - 9.6.4. Sistemi di rilevamento delle intrusioni
 - 9.6.5. IDS di host
 - 9.6.6. IDS di rete
 - 9.6.7. IDS basati sulla firma
 - 9.6.8. Sistemi di esche
 - 9.6.9. Principi di sicurezza di base nello sviluppo del codice
 - 9.6.10. Gestione dei guasti
 - 9.6.11. Nemico pubblico numero 1: i buffer overflow
 - 9.6.12. Inganni crittografici
- 9.7. *Botnets* e spam
 - 9.7.1. Origine del problema
 - 9.7.2. Processo dello spam
 - 9.7.3. Invio di spam
 - 9.7.4. Raffinare le mailing list
 - 9.7.5. Tecniche di protezione
 - 9.7.6. Servizio antispam offerto da terzi
 - 9.7.7. Casi di studio
 - 9.7.8. Spam esotico
- 9.8. Audit e attacchi web
 - 9.8.1. Raccolta informazioni
 - 9.8.2. Tecniche di attacco
 - 9.8.3. Strumenti
- 9.9. Malware e codice maligno
 - 9.9.1. Cos'è il malware?
 - 9.9.2. Tipi di malware
 - 9.9.3. Virus
 - 9.9.4. Cryptovirus
 - 9.9.5. Worm
 - 9.9.6. Adware
 - 9.9.7. Spyware
 - 9.9.8. Hoaxes
 - 9.9.9. Pishing
 - 9.9.10. Troyan
 - 9.9.11. L'economia del malware
 - 9.9.12. Possibili soluzioni

- 9.10. Analisi forense
 - 9.10.1. Raccolta di evidenze
 - 9.10.2. Analisi delle evidenze
 - 9.10.3. Tecniche anti-forensi
 - 9.10.4. Caso di studio pratico

Modulo 10. Gestione dei progetti

- 10.1. Concetti fondamentali della gestione dei progetti e del ciclo di vita della gestione dei progetti
 - 10.1.1. Cos'è un progetto?
 - 10.1.2. Metodologia comune
 - 10.1.3. Che cos'è la direzione/gestione dei progetti?
 - 10.1.4. Cos'è un piano di progetto?
 - 10.1.5. Benefici
 - 10.1.6. Cicli di vita del progetto
 - 10.1.7. Gruppi di processo o ciclo di vita della gestione del progetto
 - 10.1.8. Il rapporto tra gruppi di processi e aree di conoscenza
 - 10.1.9. Relazioni tra ciclo di vita del prodotto e del progetto
- 10.2. Avvio e pianificazione
 - 10.2.1. Dall'idea al progetto
 - 10.2.2. Sviluppare del documento del progetto
 - 10.2.3. Riunione di avvio del progetto
 - 10.2.4. Compiti, conoscenze e competenze nel processo di avvio
 - 10.2.5. Piani del progetto
 - 10.2.6. Sviluppo del piano base Fasi
 - 10.2.7. Compiti, conoscenze e competenze nel processo di pianificazione
- 10.3. Gestione degli *stakeholders* e delle attività di sensibilizzazione
 - 10.3.1. Identificazione delle parti interessate
 - 10.3.2. Sviluppare il piano di gestione degli interessati
 - 10.3.3. Gestire l'impegno degli interessati
 - 10.3.4. Controllo dell'impegno degli interessati
 - 10.3.5. Obiettivi del progetto
 - 10.3.6. La gestione della portata e del suo piano
 - 10.3.7. Raccogliere i requisiti
 - 10.3.8. Definire l'ambito di applicazione
 - 10.3.9. Creare la WBS (EDT)
 - 10.3.10. Verifica e controllare il campo di applicazione
- 10.4. Sviluppo della tabella di marcia
 - 10.4.1. La gestione del tempo e del suo piano
 - 10.4.2. Definire le attività
 - 10.4.3. Stabilire la sequenza delle attività
 - 10.4.4. Stima della risorse delle attività
 - 10.4.5. Stima della durata delle attività
 - 10.4.6. Sviluppo della timeline e calcolo del percorso critico
 - 10.4.7. Controllo della tabella di marcia
- 10.5. Lo sviluppo del budget e la risposta ai rischi
 - 10.5.1. Stimare i costi
 - 10.5.2. Sviluppare il budget e la curva a S
 - 10.5.3. Controllo dei costi e metodo earned value
 - 10.5.4. I concetti di rischio
 - 10.5.5. Come fare una analisi dei rischi
 - 10.5.6. Sviluppo del piano di risposta
- 10.6. Gestione della qualità
 - 10.6.1. Pianificazione della qualità
 - 10.6.2. Garanzia di qualità
 - 10.6.3. Controllo di qualità
 - 10.6.4. Concetti statistici di base
 - 10.6.5. Strumenti della gestione della qualità
- 10.7. Comunicazione e risorse umane
 - 10.7.1. Pianificare la gestione delle comunicazioni
 - 10.7.2. Analisi dei requisiti di comunicazione
 - 10.7.3. Tecnologia delle comunicazioni
 - 10.7.4. Modelli di comunicazione
 - 10.7.5. Metodi di comunicazione
 - 10.7.6. Piano di gestione delle comunicazioni
 - 10.7.7. Gestione della comunicazione



- 10.7.8. Gestione delle risorse umane
- 10.7.9. Attori principali e loro ruoli nei progetti
- 10.7.10. Tipi di organizzazione
- 10.7.11. Organizzazione del progetto
- 10.7.12. Squadre di lavoro
- 10.8. La fornitura
 - 10.8.1. Il processo di acquisizione
 - 10.8.2. Pianificazione
 - 10.8.3. Ricerca di fornitori e richiesta delle offerte
 - 10.8.4. Assegnazione del contratto
 - 10.8.5. Amministrazione del contratto
 - 10.8.6. I contratti
 - 10.8.7. Tipi di contratto
 - 10.8.8. Negoziazione del contratto
- 10.9. Attuazione, monitoraggio e controllo e chiusura
 - 10.9.1. I gruppi dei processi
 - 10.9.2. L'esecuzione del progetto
 - 10.9.3. Monitoraggio e controllo del progetto
 - 10.9.4. Chiusura del progetto
- 10.10. Responsabilità professionali
 - 10.10.1. Responsabilità professionali
 - 10.10.2. Caratteristiche della responsabilità sociale e professionale
 - 10.10.3. Codice etico del leader di progetto
 - 10.10.4. Responsabilità vs. PMP®
 - 10.10.5. Esempi di responsabilità
 - 10.10.6. Benefici della professionalizzazione



*Un'esperienza di studio unica,
cruciale e decisiva per favorire la
tua crescita professionale"*

05 Metodologia

Questo programma ti offre un modo differente di imparare. La nostra metodologia si sviluppa in una modalità di apprendimento ciclico: ***il Relearning***.

Questo sistema di insegnamento viene applicato nelle più prestigiose facoltà di medicina del mondo ed è considerato uno dei più efficaci da importanti pubblicazioni come il ***New England Journal of Medicine***.



“

Scopri il Relearning, un sistema che abbandona l'apprendimento lineare convenzionale, per guidarti attraverso dei sistemi di insegnamento ciclici: una modalità di apprendimento che ha dimostrato la sua enorme efficacia, soprattutto nelle materie che richiedono la memorizzazione”

Caso di Studio per contestualizzare tutti i contenuti

Il nostro programma offre un metodo rivoluzionario per sviluppare le abilità e le conoscenze. Il nostro obiettivo è quello di rafforzare le competenze in un contesto mutevole, competitivo e altamente esigente.

“

Con TECH potrai sperimentare un modo di imparare che sta scuotendo le fondamenta delle università tradizionali in tutto il mondo"



Avrai accesso a un sistema di apprendimento basato sulla ripetizione, con un insegnamento naturale e progressivo durante tutto il programma.



Imparerai, attraverso attività collaborative e casi reali, la risoluzione di situazioni complesse in ambienti aziendali reali.

Un metodo di apprendimento innovativo e differente

Questo programma di TECH consiste in un insegnamento intensivo, creato ex novo, che propone le sfide e le decisioni più impegnative in questo campo, sia a livello nazionale che internazionale. Grazie a questa metodologia, la crescita personale e professionale viene potenziata, effettuando un passo decisivo verso il successo. Il metodo casistico, la tecnica che sta alla base di questi contenuti, garantisce il rispetto della realtà economica, sociale e professionale più attuali.

“

Il nostro programma ti prepara ad affrontare nuove sfide in ambienti incerti e a raggiungere il successo nella tua carriera”

Il Metodo Casistico è stato il sistema di apprendimento più usato nelle migliori Scuole di Informatica del mondo da quando esistono. Sviluppato nel 1912 affinché gli studenti di Diritto non imparassero la legge solo sulla base del contenuto teorico, il metodo casistico consisteva nel presentare loro situazioni reali e complesse per prendere decisioni informate e giudizi di valore su come risolverle. Nel 1924 fu stabilito come metodo di insegnamento standard ad Harvard.

Cosa dovrebbe fare un professionista per affrontare una determinata situazione?

Questa è la domanda con cui ti confrontiamo nel metodo dei casi, un metodo di apprendimento orientato all'azione. Durante il corso, gli studenti si confronteranno con diversi casi di vita reale. Dovranno integrare tutte le loro conoscenze, effettuare ricerche, argomentare e difendere le proprie idee e decisioni.

Metodologia Relearning

TECH coniuga efficacemente la metodologia del Caso di Studio con un sistema di apprendimento 100% online basato sulla ripetizione, che combina diversi elementi didattici in ogni lezione.

Potenziamo il Caso di Studio con il miglior metodo di insegnamento 100% online: il Relearning.

Nel 2019 abbiamo ottenuto i migliori risultati di apprendimento di tutte le università online del mondo.

In TECH imparerai con una metodologia all'avanguardia progettata per formare i manager del futuro. Questo metodo, all'avanguardia della pedagogia mondiale, si chiama Relearning.

La nostra università è l'unica autorizzata a utilizzare questo metodo di successo. Nel 2019, siamo riusciti a migliorare il livello di soddisfazione generale dei nostri studenti (qualità dell'insegnamento, qualità dei materiali, struttura del corso, obiettivi...) rispetto agli indicatori della migliore università online.



Nel nostro programma, l'apprendimento non è un processo lineare, ma avviene in una spirale (impariamo, disimpariamo, dimentichiamo e re-impariamo). Pertanto, combiniamo ciascuno di questi elementi in modo concentrico. Questa metodologia ha formato più di 650.000 laureati con un successo senza precedenti in campi diversi come la biochimica, la genetica, la chirurgia, il diritto internazionale, le competenze manageriali, le scienze sportive, la filosofia, il diritto, l'ingegneria, il giornalismo, la storia, i mercati e gli strumenti finanziari. Tutto questo in un ambiente molto esigente, con un corpo di studenti universitari con un alto profilo socio-economico e un'età media di 43,5 anni.

Il Relearning ti permetterà di apprendere con meno sforzo e più performance, impegnandoti maggiormente nella tua specializzazione, sviluppando uno spirito critico, difendendo gli argomenti e contrastando le opinioni: un'equazione diretta al successo.

Dalle ultime evidenze scientifiche nel campo delle neuroscienze, non solo sappiamo come organizzare le informazioni, le idee, le immagini e i ricordi, ma sappiamo che il luogo e il contesto in cui abbiamo imparato qualcosa è fondamentale per la nostra capacità di ricordarlo e immagazzinarlo nell'ippocampo, per conservarlo nella nostra memoria a lungo termine.

In questo modo, e in quello che si chiama Neurocognitive Context-dependent E-learning, i diversi elementi del nostro programma sono collegati al contesto in cui il partecipante sviluppa la sua pratica professionale.



Questo programma offre i migliori materiali didattici, preparati appositamente per i professionisti:



Materiali di studio

Tutti i contenuti didattici sono creati appositamente per il corso dagli specialisti che lo impartiranno, per fare in modo che lo sviluppo didattico sia davvero specifico e concreto.

Questi contenuti sono poi applicati al formato audiovisivo che supporterà la modalità di lavoro online di TECH. Tutto questo, con le ultime tecniche che offrono componenti di alta qualità in ognuno dei materiali che vengono messi a disposizione dello studente.



Master class

Esistono evidenze scientifiche sull'utilità dell'osservazione di esperti terzi.

Imparare da un esperto rafforza la conoscenza e la memoria, costruisce la fiducia nelle nostre future decisioni difficili.



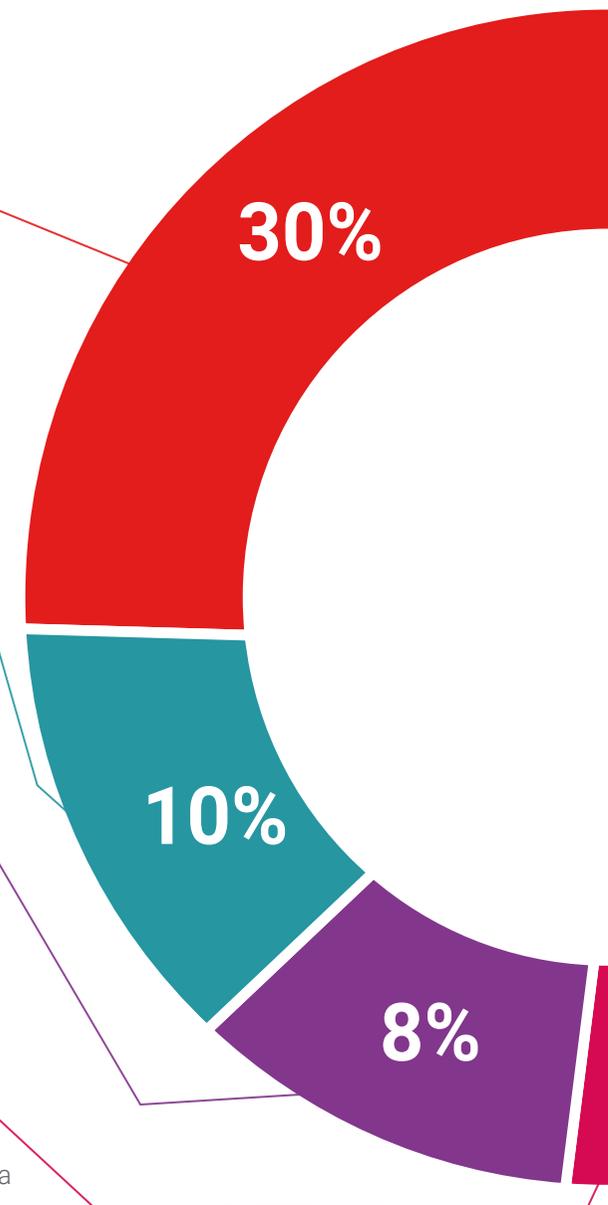
Pratiche di competenze e competenze

Svolgerai attività per sviluppare competenze e capacità specifiche in ogni area tematica. Pratiche e dinamiche per acquisire e sviluppare le competenze e le abilità che uno specialista deve sviluppare nel quadro della globalizzazione in cui viviamo.



Letture complementari

Articoli recenti, documenti di consenso e linee guida internazionali, tra gli altri. Nella biblioteca virtuale di TECH potrai accedere a tutto il materiale necessario per completare la tua specializzazione.





Casi di Studio

Completerai una selezione dei migliori casi di studio scelti appositamente per questo corso. Casi presentati, analizzati e monitorati dai migliori specialisti del panorama internazionale.



Riepiloghi interattivi

Il team di TECH presenta i contenuti in modo accattivante e dinamico in pillole multimediali che includono audio, video, immagini, diagrammi e mappe concettuali per consolidare la conoscenza.

Questo esclusivo sistema di specializzazione per la presentazione di contenuti multimediali è stato premiato da Microsoft come "Caso di successo in Europa".



Testing & Retesting

Valutiamo e rivalutiamo periodicamente le tue conoscenze durante tutto il programma con attività ed esercizi di valutazione e autovalutazione, affinché tu possa verificare come raggiungi progressivamente i tuoi obiettivi.



06 Titolo

Il Master Privato in Ingegneria del Software Avanzata ti garantisce, oltre alla preparazione più rigorosa e aggiornata, l'accesso a una qualifica di Master Privato rilasciata da TECH Università Tecnologica.



“

Porta a termine questo programma e ricevi la tua qualifica universitaria senza spostamenti o fastidiose formalità”

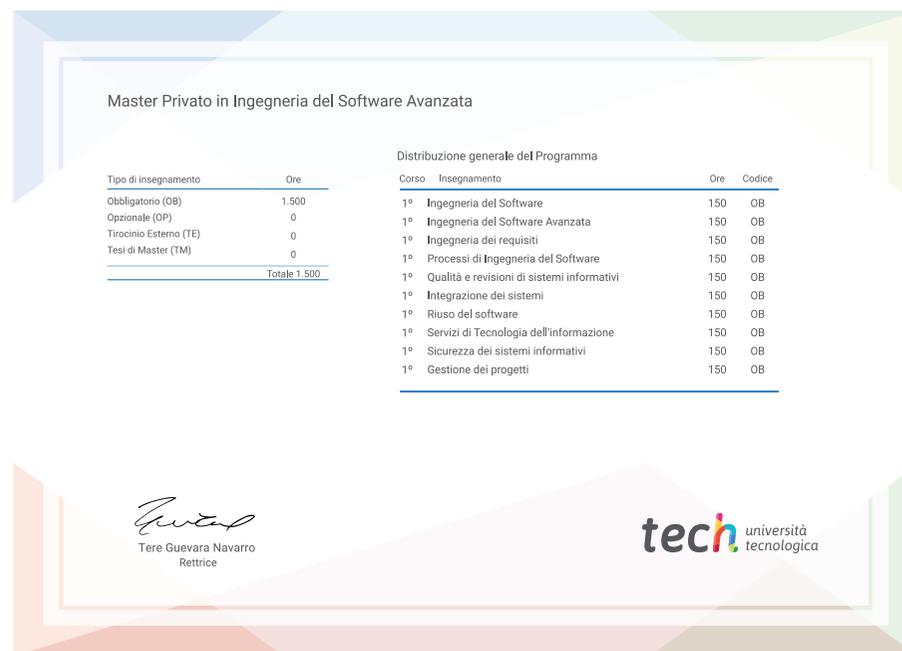
Questo **Master Privato in Ingegneria del Software Avanzata** possiede il programma più completo e aggiornato del mercato.

Dopo aver superato la valutazione, lo studente riceverà mediante lettera certificata* con ricevuta di ritorno, la sua corrispondente qualifica di **Master Privato** rilasciata da **TECH Università Tecnologica**.

Il titolo rilasciato da **TECH Università Tecnologica** esprime la qualifica ottenuta nel Master Privato, e riunisce tutti i requisiti comunemente richiesti da borse di lavoro, concorsi e commissioni di valutazione di carriere professionali.

Titolo: **Master Privato in Ingegneria del Software Avanzata**

N° Ore Ufficiali: **1.500 O.**



*Se lo studente dovesse richiedere che il suo diploma cartaceo sia provvisto di Apostille dell'Aia, TECH EDUCATION effettuerà le gestioni opportune per ottenerla pagando un costo aggiuntivo.

futuro
salute fiducia persone
educazione informazione tutor
garanzia accreditamento insegnamento
istituzioni tecnologia apprendimento
comunità impegno
attenzione personalizzata innovazione
conoscenza presente qualità
formazione online
sviluppo istituzioni
classe virtuale lingue

tech università
tecnologica

Master Privato
Ingegneria del
Software Avanzata

- » Modalità: **online**
- » Durata: **12 mesi**
- » Titolo: **TECH Università
Tecnologica**
- » Orario: **a scelta**
- » Esami: **online**

Master Privato

Ingegneria del Software Avanzata