

Master Semipresenziale

Ingegneria del Software Avanzata



tech università
tecnologica

Master Semipresenziale Ingegneria del Software Avanzata

Modalità: Semipresenziale (Online + Tirocinio)

Durata: 12 mesi

Certificazione: TECH Università Tecnologica

Crediti: 60 + 4 ECTS

Accesso web: www.techtitude.com/it/informatica/master-semipresenziale/master-semipresenziale-ingegneria-software-avanzata

Indice

01

Presentazione

pag. 4

02

Perché iscriversi a questo
Master Semipresenziale?

pag. 8

03

Obiettivi

pag. 12

04

Competenze

pag. 18

05

Struttura e contenuti

pag. 22

06

Tirocinio

pag. 38

07

Dove posso svolgere il
tirocinio?

pag. 44

08

Metodologia

pag. 48

09

Certificazione

pag. 56

01

Presentazione

L'Ingegneria del Software Avanzata è emersa come componente cruciale nello sviluppo di sistemi complessi e robusti, fondamentali per la moderna economia digitale. In un ambiente in cui la domanda di software di alta qualità e scalabilità è in costante aumento, le tecniche avanzate di ingegneria del software consentono agli sviluppatori di creare soluzioni efficienti e sostenibili. È quindi essenziale che i professionisti dell'IT aggiornino regolarmente le loro conoscenze per incorporare le metodologie più innovative nella loro pratica quotidiana, al fine di migliorare la qualità del software e accelerare il ciclo di sviluppo. In questo contesto, TECH lancia un rivoluzionario corso di laurea incentrato sulle tecniche più all'avanguardia dell'Ingegneria del Software Avanzata.



“

*Con questo Master Semipresenziale,
progetterai i sistemi software più scalabili,
affidabili e gestibili”*

La qualità e la sicurezza del software sono diventate elementi cruciali dell'Ingegneria del Software Avanzata. A questo proposito, un rapporto dell'Organizzazione per la Cooperazione e lo Sviluppo Economico rivela che il 58% delle aziende subisce violazioni della sicurezza a causa della mancanza di adeguate pratiche di garanzia della qualità. Ciò può avere gravi conseguenze, che vanno dalla perdita di fatturato o dalla mancanza di fiducia dei clienti a multe significative. Da qui l'importanza per i professionisti di incorporare nella loro pratica le strategie più sofisticate per garantire la qualità e la sicurezza del software.

In questo contesto, TECH presenta un innovativo Master Semipresenziale in Ingegneria del Software Avanzata. Si tratta di un programma che combina 1.920 ore dei migliori contenuti teorici con 3 settimane di formazione pratica in un'azienda leader del settore. Composto da 10 moduli completi, l'itinerario accademico approfondirà argomenti come l'ingegneria dei requisiti, la progettazione di applicazioni web o l'audit dei sistemi informativi. Inoltre, il programma fornirà agli studenti le tecniche più avanzate per la protezione dei sistemi e lo sviluppo di codici altamente sicuri. Gli studenti acquisiranno così competenze avanzate per analizzare problemi software complessi e sviluppare soluzioni efficienti e innovative.

D'altra parte, questo titolo universitario prevede che gli studenti svolgano un tirocinio presso un'istituzione prestigiosa nel campo dell'Ingegneria del Software Avanzata. In questo modo, parteciperanno attivamente ai progetti in fase di sviluppo. Va sottolineato che un tutor specializzato guiderà gli studenti durante l'esperienza accademica, garantendo il completamento di un piano di attività che consentirà loro di migliorare le proprie competenze in modo esponenziale e in base alle esigenze dell'attuale domanda del mercato del lavoro.

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

- ♦ Sviluppo di oltre 100 casi pratici presentati da esperti in Ingegneria Informatica
- ♦ Contenuti grafici, schematici e prettamente pratici che forniscono informazioni tecniche riguardo alle discipline essenziali per l'esercizio della professione
- ♦ Esercizi pratici che offrono un processo di autovalutazione per migliorare l'apprendimento
- ♦ Particolare enfasi sulle metodologie innovative
- ♦ Tutto ciò sarà integrato da lezioni teoriche, domande all'esperto, forum di discussione su questioni controverse e lavoro di riflessione individuale
- ♦ Contenuti disponibili da qualsiasi dispositivo fisso o mobile dotato di connessione a internet
- ♦ Inoltre, potrai fare un tirocinio presso una delle migliori aziende del settore



Guiderai progetti di sviluppo software, dalla concezione all'implementazione e alla valutazione"

“

Effettuerai uno stage intensivo di 3 settimane presso un'istituzione leader nel campo dell'Ingegneria del Software Avanzata”

In questa proposta di Master, di natura professionale e in modalità semipresenziale, il programma è finalizzato all'aggiornamento dei professionisti dell'Informatica. I contenuti sono basati sulle ultime evidenze scientifiche, orientati in modo didattico per integrare le conoscenze teoriche nella pratica Informatica e tali elementi faciliteranno l'aggiornamento delle conoscenze.

Grazie ai loro contenuti multimediali elaborati con le più recenti tecnologie educative, consentiranno al professionista un apprendimento localizzato e contestuale, vale a dire un ambiente simulato che fornisca un apprendimento immersivo programmato per specializzarsi in situazioni reali. La creazione di questo programma è incentrata sull'Apprendimento Basato su Problemi, mediante il quale lo specialista deve cercare di risolvere le diverse situazioni che gli si presentano durante il corso. Lo studente potrà usufruire di un innovativo sistema di video interattivi creati da esperti di rinomata fama.

Aggiorna le tue conoscenze in Ingegneria del Software Avanzata grazie a contenuti multimediali innovativi.

TECH è un'università tecnologica all'avanguardia, che mette tutte le sue risorse a tua disposizione per raggiungere il successo come informatico.



02

Perché iscriversi a questo Master Semipresenziale?

Una delle principali priorità delle aziende è innovare e rimanere competitive in un mercato globale. Per questo motivo, le organizzazioni sono alla ricerca di specialisti in Ingegneria del Software Avanzata, in grado di sviluppare nuove applicazioni che forniscano loro vantaggi competitivi. Per sfruttare queste opportunità, i professionisti devono tenersi aggiornati sugli ultimi sviluppi del settore. Con questo obiettivo, TECH ha creato un programma accademico unico e dirompente nell'attuale panorama educativo, che permetterà agli specialisti di immergersi in un ambiente di lavoro reale e di applicare le più recenti procedure e tecniche di Ingegneria del Software Avanzata.



```
manager.create(this, TOOLTIP_ID_CONTACT)
viewById(R.id.btnContact), TooltipManager.GRAVITY_BOTTOM)
policy(TooltipManager.ClosePolicy.TouchOutside, 3000)
    .activateDelay(300)
    .fitToScreen(true)
    .withStyleId(R.style.TooltipLayoutStyle)
    .text(R.string.label_tips_contact);
mTooltipContact.show();
}

@Override
public void onBackPressed() {
    super.onBackPressed();
}

private String validationRegisterForm() {
    String errorMessage = null;
```

“

Svolgerai il tirocinio in un'azienda di riferimento, dove analizzerai gli ultimi progressi dell'Ingegneria del Software Avanzata”

1. Aggiornarsi a partire dalle più recenti tecnologie disponibili

Le nuove tecnologie hanno trasformato in modo significativo il campo dell'Ingegneria del Software Avanzata, aumentando l'efficienza, la qualità e l'innovatività dello sviluppo del software. Un esempio è rappresentato dai test automatizzati, che consentono ai professionisti di individuare e correggere rapidamente gli errori. Per consentire agli specialisti di familiarizzare con questi strumenti, TECH offre una formazione pratica che permette agli studenti di immergersi in un ambiente di lavoro all'avanguardia, dove avranno accesso alle tecnologie più avanzate in questo campo.

2. Approfondire a partire dall'esperienza dei migliori specialisti

Durante la formazione pratica, un team di esperti in Ingegneria del Software Avanzata guiderà gli studenti, aiutandoli a trarre il massimo da questa esperienza accademica. In questo senso, questi professionisti forniranno agli studenti tutte le conoscenze di cui hanno bisogno per dare una spinta alla loro carriera di informatici.

3. Accedere ad ambienti professionali di prim'ordine

L'obiettivo principale di TECH è offrire programmi universitari di alta qualità e accessibili a tutti. Per questo motivo, TECH seleziona con cura i centri in cui gli studenti svolgeranno i loro stage. Questo garantisce agli informatici l'accesso a istituzioni leader nel campo dell'Ingegneria del Software Avanzata. Gli studenti potranno così sperimentare in prima persona il lavoro quotidiano in un ambiente esigente e rigoroso, applicando sempre le tecniche e le metodologie più avanzate.



4. Combinare la migliore teoria con la pratica più avanzata

Nel panorama accademico odierno, è comune trovare programmi universitari che offrono solo contenuti teorici, trascurando l'importanza della pratica, che consente agli studenti di applicare le loro conoscenze in situazioni lavorative reali. Al contrario, TECH presenta un modello di apprendimento completamente pratico, che consente ai professionisti IT di acquisire esperienza pratica e di affrontare le sfide reali che possono presentarsi nella loro carriera professionale.

5. Ampliare le frontiere della conoscenza

TECH offre agli studenti l'opportunità di conseguire questo Master Semipresenziale presso prestigiose istituzioni internazionali. Ciò consente agli informatici di ampliare i propri orizzonti e di aggiornare le proprie conoscenze con i migliori professionisti che lavorano nelle aziende più importanti. È un'opportunità eccezionale che solo TECH, la più grande università digitale del mondo, può offrire.

“

Avrai l'opportunità di svolgere un tirocinio all'interno di un centro a tua scelta”

03 Obiettivi

Al termine di questo Master Semipresenziale, gli informatici avranno una comprensione completa della progettazione, dell'architettura, dell'implementazione e della manutenzione di sistemi software complessi. In questo senso, gli studenti applicheranno metodologie di sviluppo software sia agili che tradizionali per gestire efficacemente i progetti. Inoltre, i professionisti utilizzeranno le metriche per misurare e migliorare la qualità delle applicazioni.



“

Incorpora nella tua pratica professionale le più innovative tecniche di sviluppo sicuro per proteggere i sistemi software dalle minacce e dalle vulnerabilità”

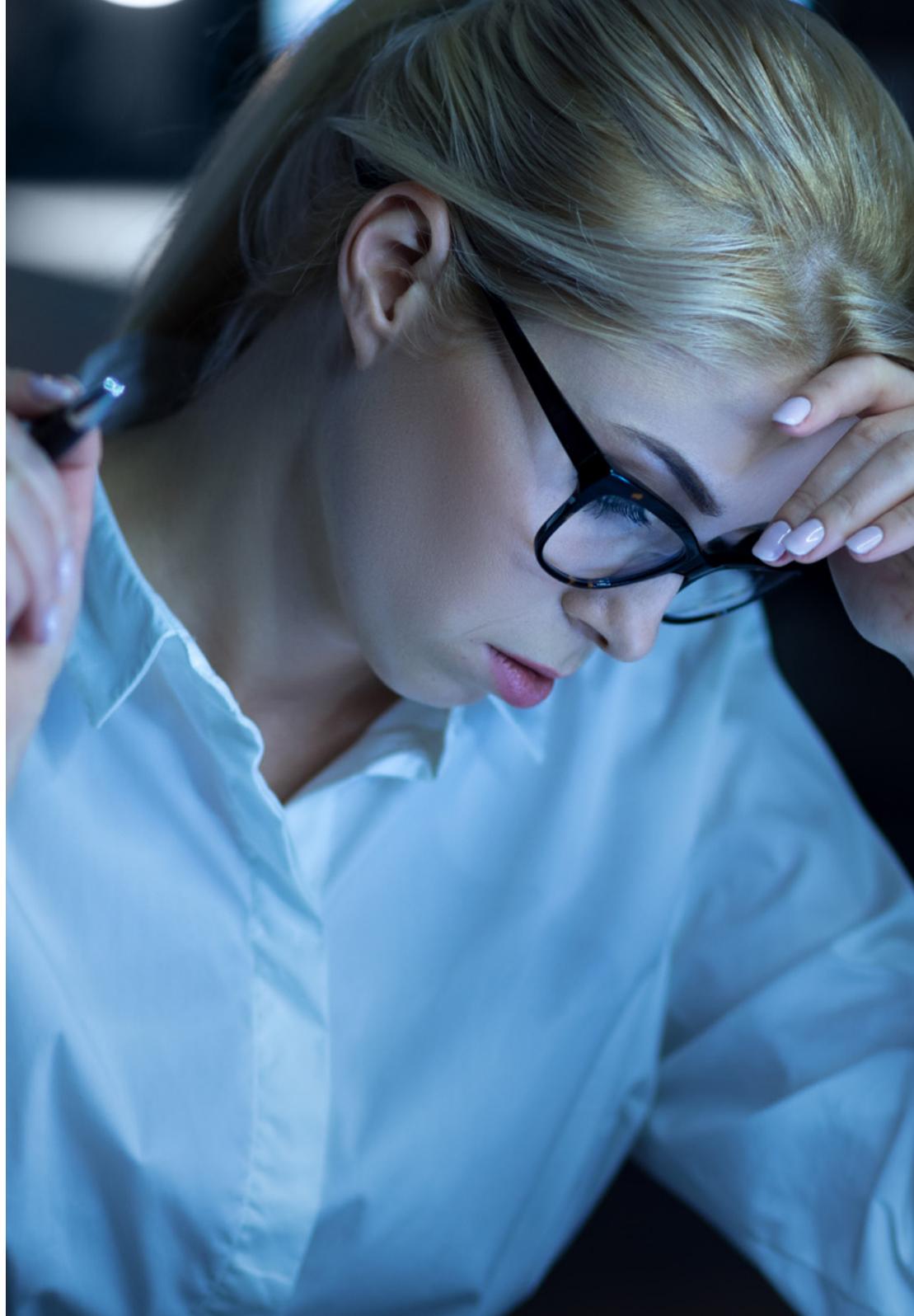


Obiettivi generali

- ♦ Questo Master Semipresenziale in Ingegneria del Software Avanzata fornirà ai professionisti dell'IT competenze avanzate per progettare architetture software scalabili, affidabili e gestibili utilizzando modelli di progettazione e principi di architettura software. Allo stesso tempo, gli studenti applicheranno tecniche di modellazione e simulazione per prevedere e risolvere in modo efficiente potenziali problemi nei sistemi software.

“

*Imparerai, attraverso casi reali, e la
risoluzione di situazioni complesse in
ambienti di apprendimento simulati”*





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 di sviluppo del software, comprese le tecnologie agili

Modulo 2. Ingegneria del Software Avanzata

- ♦ Acquisire una conoscenza approfondita delle 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, apprendendo anche la progettazione dell'architettura, la progettazione a livello di componenti e la progettazione basata su pattern
- ♦ Introdurre il concetto di DevOps e le sue pratiche principali
- ♦ Imparare a realizzare prove al software, con metodologie come *Test Driven Development*, *Acceptance Test Driven Development*, *Behavior Driven Development*, *BDD* e *Cucumber*
- ♦ Comprendere i diversi modelli di architetture di sistema e di progettazione del software, nonché l'architettura delle applicazioni cloud

Modulo 3. Ingegneria dei requisiti

- ♦ Comprendere l'ingegneria dei requisiti, il suo sviluppo, la elaborazione, la negoziazione e la convalida
- ♦ 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 di questi, in quanto parte essenziale del processo
- ♦ Comprendere i processi di convalida e negoziazione dei requisiti, nonché la loro modellazione e la gestione

Modulo 4. Processi di ingegneria del software

- ♦ Approfondire la comprensione del miglioramento del processo di sviluppo del software e della sua qualità utilizzando gli standard ISO/IEC
- ♦ Comprendere e applicare la prototipazione come parte essenziale del processo di sviluppo
- ♦ Comprendere il quadro 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
- ♦ Conoscere i diversi stili di progettazione del software distribuito e architetture software orientate ai servizi
- ♦ Imparare i concetti essenziali della progettazione di interfacce grafiche

Modulo 5. Qualità e controllo dei sistemi informativi

- ♦ Conoscere 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 operativa e di *disaster recovery*
- ♦ Imparare a pianificare la gestione della sicurezza e a gestire i principali meccanismi di protezione del patrimonio informativo
- ♦ Conoscere i diversi tipi di audit e il processo che si svolge durante l'audit informatico

Modulo 6. Integrazione dei sistemi

- ♦ Acquisire i concetti essenziali relativi ai sistemi informativi in ambito aziendale e identificarne le opportunità e le esigenze all'interno dell'azienda stessa
- ♦ Conoscere le basi della *Business Intelligence*, le sue strategie e la sua attuazione, nonché il presente e il futuro della BI
- ♦ Comprendere il funzionamento dei sistemi per la gestione integrata 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. Riutilizzo di software

- ♦ Conoscere il panorama del riutilizzo di software
- ♦ Apprendere i diversi modelli relativi al riutilizzo del software in termini di progettazione, creazione, struttura e comportamento
- ♦ Introdurre il concetto di *framework*, nonché conoscerne le principali tipologie come quelle destinate alla progettazione di interfacce grafiche, allo sviluppo di applicazioni web e alla gestione della persistenza degli oggetti nei database
- ♦ Comprendere il funzionamento del pattern *Model-View-Controller* (MVC), ampiamente utilizzato

Modulo 8. Servizi di tecnologia dell'informazione

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



Modulo 9. Sicurezza nei sistemi informativi

- ♦ Pianificare una tabella di marcia per la gestione del tempo, lo sviluppo del budget e la risposta ai rischi
- ♦ Analizzare la natura degli attacchi di rete e i diversi tipi di architetture di sicurezza
- ♦ Comprendere le tecniche di protezione del sistema e di sviluppo del codice sicuro
- ♦ Conoscere i componenti essenziali di *botnet* e spam, oltre che del malware e del codice dannoso
- ♦ Gettare le basi per l'analisi forense nel mondo del software e del controllo informatico
- ♦ Comprendere i fondamenti della crittografia simmetrica e asimmetrica e i loro principali algoritmi

Modulo 10. Gestione dei progetti

- ♦ Comprendere il funzionamento della gestione della qualità nei 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 ciclo di vita della gestione dei progetti

04

Competenze

Grazie a questa laurea, i professionisti dell'IT acquisiranno competenze avanzate per progettare sistemi software scalabili, gestibili e affidabili utilizzando i principi dell'architettura di progetto. In linea con ciò, gli studenti saranno altamente qualificati per l'applicazione di metodologie agili (come Scrum o Kanban) nel ciclo di vita dello sviluppo del software. Inoltre, gli esperti implementeranno processi di garanzia della qualità per assicurare la qualità delle applicazioni attraverso revisioni del codice, audit e test approfonditi.





“

Utilizzerai i framework per mantenere la coerenza nella progettazione e nell'implementazione del software”



Competenze generali

- ♦ Rispondere alle necessità attuali del settore dell'Ingegneria del Software Avanzata
- ♦ Padroneggiare i diversi sistemi di lavoro dell'ingegneria del Software Avanzata

“

Avrai a disposizione una biblioteca aperta 24 ore su 24 con risorse didattiche e materiali che si distinguono per la loro qualità”





Competenze specifiche

- ♦ Avere una conoscenza approfondita di tutte le sfaccettature dell'interazione uomo-macchina e di come queste coinvolgano gli sviluppi informatici
- ♦ Essere abili nell'uso delle banche dati
- ♦ Sviluppare diversi tipi di applicazioni di rete
- ♦ Lavorare come ingegnere del software
- ♦ Controllare l'uso di database avanzati
- ♦ Eseguire una programmazione avanzata
- ♦ Saper riutilizzare il software
- ♦ Creare interfacce e applicazioni di rete



Combinerai teoria e pratica professionale attraverso un approccio educativo impegnativo e gratificante"

05

Struttura e contenuti

I materiali didattici che compongono questo Master Semipresenziale sono stati creati da un team di professionisti altamente specializzati nell'Ingegneria del Software Avanzata. In questo modo, hanno sviluppato un programma di studi che si distingue per l'eccellente qualità e per essere in linea con le esigenze dell'attuale mercato del lavoro. Composto da 10 moduli specializzati, il programma di studi approfondirà le ultime innovazioni in materie come la modellazione dei requisiti, l'architettura del software o la progettazione di interfacce grafiche. Inoltre, il programma fornirà ai laureati le tecniche più avanzate per la manutenzione del software.

```
manager.js  
main.js  
readme.rst  
sequences.js  
sessions.js  
settings.js  
tasks.js  
templates
```

56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71

```
sel  
ret  
}  
if (e  
e.s  
e.p  
sel  
sel  
ret  
}  
//up/  
if (e  
e.p  
e.s  
if  
/  
i
```

```
f.deactivate(true);  
return; // this is a modified line  
  
.keyCode == 13) {  
  topImmediatePropagation();  
  reventDefault();  
  f.search();  
  f.deactivate();  
  return;  
  
down  
.keyCode == 38 || e.keyCode == 40) {  
  reventDefault();  
  topImmediatePropagation();  
  (e.keyCode == 38) { // up  
    // show previous search query  
    f (hist.currentIndex ==  
      hist.temporaryQuery  
    // skip previous  
    if (hist.t  
      hist  
  }  
}
```

“

Imparerai a conoscere le Metodologie Agili per migliorare l'efficienza, la flessibilità e la reattività dei team di sviluppo al cambiamento”

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 webapp
 - 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 ha 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 specializzati
 - 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 Indagine sui 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 sulle classi
 - 1.5.6 Diagrammi di classe
- 1.6. Modellazione dei requisiti II: flusso, comportamento e modelli
 - 1.6.1 Requisiti che definiscono le strategie
 - 1.6.2 Modellazione orientata al flusso
 - 1.6.3 Diagrammi di stato
 - 1.6.4 Creare un modello comportamentale
 - 1.6.5 Diagrammi di sequenza
 - 1.6.6 Diagrammi di comunicazione
 - 1.6.7 Schemi per la modellazione dei requisiti
- 1.7. Concetti di design
 - 1.7.1 Il design nel contesto dell'ingegneria del software
 - 1.7.2 Processo di progettazione
 - 1.7.3 Concetti di design
 - 1.7.4 Concetti di design orientati agli oggetti
 - 1.7.5 Il modello di design
- 1.8. Design architettonico
 - 1.8.1 Architettura del software
 - 1.8.2 Generi architettonici
 - 1.8.3 Stili architettonici
 - 1.8.4 Design architettonico
 - 1.8.5 Evoluzione dei design alternativi per l'architettura
 - 1.8.6 Mappatura dell'architettura con l'uso di flussi di dati
- 1.9. Design a livello di componente e basato su pattern
 - 1.9.1 Che cos'è un componente?
 - 1.9.2 Design dei componenti basato sulle classi
 - 1.9.3 Realizzazione del progetto a livello di componenti
 - 1.9.4 Design dei componenti tradizionali
 - 1.9.5 Sviluppo basato su componenti
 - 1.9.6 Modelli di progettazione
 - 1.9.7 Il design del software basato su modelli
 - 1.9.8 Modelli architettonici
 - 1.9.9 Modelli di design a livello di componenti
 - 1.9.10. Modelli di design 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 Garanzia di 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 Artefatti di Scrum
 - 2.2.6 Eventi di Scrum
 - 2.2.7 Storie degli utenti
 - 2.2.8 Estensioni Scrum
 - 2.2.9 Stime agili
 - 02.2.10. Scalabilità di Scrum
- 2.3. Programmazione estrema
 - 2.3.1 Motivazione 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 del software basato sul riutilizzo
 - 2.4.1 Riutilizzo del software
 - 2.4.2 Livelli di riutilizzo del codice
 - 2.4.3 Tecniche di riutilizzo specifiche
 - 2.4.4 Sviluppo basato su componenti
 - 2.4.5 Benefici e problemi del riutilizzo
 - 2.4.6 Pianificazione del riutilizzo
- 2.5. Architettura di sistema e modelli di progettazione del software
 - 2.5.1 Design architettonico
 - 2.5.2 Modelli architettonici generali
 - 2.5.3 Architetture con tolleranza ai guasti
 - 2.5.4 Architetture di sistemi distribuiti
 - 2.5.5 Modelli di progettazione
 - 2.5.6 Modelli gamma
 - 2.5.7 Modelli di progettazione dell'interazione
- 2.6. Architettura di applicazioni nel cloud
 - 2.6.1 Fondamenti di Cloud Computing
 - 2.6.2 Qualità delle applicazioni nel cloud
 - 2.6.3 Stili di architettura
 - 2.6.4 Modelli di progettazione
- 2.7. Test del software: TDD, ATDD e BDD
 - 2.7.1 Verifica e convalida 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. Miglioramento dei processi software
 - 2.8.1 Miglioramento dei processi 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. La qualità del prodotto software: SQuaRE
 - 2.9.1 La qualità del software
 - 2.9.2 Modello 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 e 25024: metriche di 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 principali

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 *Framework* per ingegneria dei requisiti
 - 3.1.9 Pratica corretta in ingegneria dei requisiti
 - 3.1.10. Analista di business
- 3.2. Fonti dei requisiti
 - 3.2.1 Rete dei requisiti
 - 3.2.2 Gli *stakeholder*
 - 3.2.3 Requisiti aziendali
 - 3.2.4 Documento di visione e scopo

- 3.3. Tecniche di elicitazione dei requisiti
 - 3.3.1 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 Workshops
 - 3.3.8 *Focus groups*
 - 3.3.9 Questionari
 - 3.3.10. *Brainstorming* e tecniche creative
 - 3.3.11. Media del gruppo
 - 03.3.12. Analisi delle interfacce di sistema
 - 3.3.13. Analisi dei documenti e "archeologia"
 - 03.3.14. Casi d'uso e scenari
 - 03.3.15. Prototipi
 - 03.3.16. Ingegneria inversa
 - 3.3.17. Riutilizzo dei requisiti
 - 3.3.18. Pratiche corrette di elicitazione
- 3.4. Requisiti dell'utente
 - 3.4.1 Persone
 - 3.4.2 Casi di uso e storie dell'utente
 - 3.4.3 Scenari
 - 3.4.5 Tipi di scenari
 - 3.4.6 Come scoprire gli scenari
- 3.5. Tecniche di prototipazione
 - 3.5.1 Prototipazione
 - 3.5.2 Prototipi in base al percorso
 - 3.5.3 Prototipi in base alla temporalità
 - 3.5.4 La fedeltà di un prototipo
 - 3.5.5 Prototipi di interfaccia utente
 - 3.5.6 Valutazione del prototipo
- 3.6. Analisi dei requisiti
 - 3.6.1 Analisi dei requisiti
 - 3.6.2 Pratica corretta di analisi dei requisiti
 - 3.6.3 Il dizionario dei dati
 - 3.6.4 Prioritarizzazione 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 Pratiche corrette 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 dei requisiti
 - 3.9.1 Modellazione dei requisiti
 - 3.9.2 La prospettiva dell'utente
 - 3.9.3 La prospettive 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 di gestione dei requisiti
 - 3.9.9 Pratica corretta per la 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 Principali processi di ingegneria del Software
 - 4.1.3 Modelli di processo di sviluppo software
 - 4.1.4 Quadro di riferimento standard per il processo di sviluppo del software: Standard ISO/IEC 12207
- 4.2. Processo unificato di sviluppo software
 - 4.2.1 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 unificati fondamentali per i processi
- 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 Storie dell'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 dell'iterazione
- 4.4. Stili di progettazione del software distribuito e 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 di 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
 - 4.6.1 Principi di progettazione dell'interfaccia utente
 - 4.6.2 Modelli di progettazione architettonica per sistemi interattivi: Model-View-Controller (MVC)
 - 4.6.3 Esperienza dell'utente (UX User Experience)
 - 4.6.4 Design incentrato sull'utente
 - 4.6.5 Processo di analisi e progettazione dell'interfaccia utente
 - 4.6.6 Usabilità delle interfacce utente
 - 4.6.7 Accessibilità delle 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 Design 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 dello stesso
 - 4.8.1 Strategie di prova
 - 4.8.2 Progettazione dei casi di prova
 - 4.8.3 Relazione costo-qualità
 - 4.8.4 Modelli di qualità
 - 4.8.5 Famiglia di norme ISO/IEC 25000 (SQuaRE)
 - 4.8.6 Modello di qualità del prodotto (ISO 2501n)
 - 4.8.7 Modello di qualità dei 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 Metrica orientata agli oggetti
- 4.10. Manutenzione e reingegnerizzazione del software
 - 4.10.1 Processo di manutenzione
 - 4.10.2 Quadro standard del processo di manutenzione: ISO/EIEC 14764
 - 4.10.3 Modello del processo di reingegnerizzazione del software
 - 4.10.4 Ingegneria inversa

Modulo 5. Qualità e controllo dei sistemi informativi

- 5.1. Introduzione ai Sistemi di Gestione della Sicurezza delle Informazioni
 - 5.1.1 Principi dei SGSI
 - 5.1.2 Regole d'oro dei SGSI
 - 5.1.3 Ruolo del controllo informatico nei SGSI
- 5.2. Pianificazione nella gestione della sicurezza
 - 5.2.1 Concetti relativi 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 Sintesi dei principali strumenti crittografici per la protezione della triade CID
 - 5.3.2 Considerazione della privacy, dell'anonimato e dei requisiti di 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 revisione
 - 5.6.1 Differenza tra revisione e controllo interno
 - 5.6.2 Revisione interna vs esterna
 - 5.6.3 Classificazione della revisione in base all'obiettivo e al tipo di analisi
- 5.7. Sceneggiatore e sceneggiatura: soggetto e oggetto protetti dalla 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 scansione delle vulnerabilità del sistema
 - 5.8.2 Principali vulnerabilità nel contesto delle applicazioni web
 - 5.8.3 Analisi dei protocolli di comunicazione
- 5.9. Il processo di controllo informatico
 - 5.9.1 Concetto di ciclo di vita dello sviluppo dei sistemi
 - 5.9.2 Monitoraggio delle attività e dei processi: raccolta ed elaborazione delle prove
 - 5.9.3 Metodologia di controllo informatico
 - 5.9.4 Processo di controllo informatico
 - 5.9.5 Individuazione dei principali reati e illeciti nel contesto informatico
 - 5.9.6 Indagine sui reati informatici: introduzione all'analisi forense e al suo rapporto con il controllo informatico
- 5.10. Piani di continuità operativa e di disaster recovery
 - 5.10.1 Definizione di piano di continuità aziendale e concetto di interruzione dell'attività
 - 5.10.2 Raccomandazione NIST sui piani di continuità aziendale
 - 5.10.3 Piano di disaster recovery
 - 5.10.4 Processo del piano di disaster recovery

Modulo 6. Integrazione dei sistemi

- 6.1. Introduzione ai sistemi informativi aziendali
 - 6.1.1 Il ruolo dei sistemi informativi
 - 6.1.2 Che 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 dei sistemi informativi

- 6.2. Opportunità e necessità dei sistemi informativi nell'impresa
 - 6.2.1 Organizzazioni e sistemi informativi
 - 6.2.2 Caratteristiche delle organizzazioni
 - 6.2.3 Impatto dei sistemi informativi sull'impresa
 - 6.2.4 Sistemi informativi per il vantaggio competitivo
 - 6.2.5 Uso dei sistemi nell'amministrazione e nella gestione aziendale
- 6.3. Concetti di base dei sistemi e delle tecnologie dell'informazione
 - 6.3.1 Dati, informazioni e conoscenza
 - 6.3.2 Tecnologia e sistemi informativi
 - 6.3.3 Componenti tecnologici
 - 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 dei sistemi
- 6.4. Sistemi per la gestione integrata delle risorse aziendali
 - 6.4.1 Requisiti aziendali
 - 6.4.2 Un sistema informativo aziendale integrato
 - 6.4.3 Acquisizione vs Sviluppo
 - 6.4.4 Implementazione dell'ERP
 - 6.4.5 Implicazioni nella gestione
 - 6.4.6 Principali fornitori di ERP
- 6.5. Sistemi informativi per la gestione della *supply chain* e delle relazioni con i clienti
 - 6.5.1 Definizione di catena di approvvigionamento
 - 6.5.2 Gestione della catena di approvvigionamento
 - 6.5.3 Il ruolo dei sistemi informativi
 - 6.5.4 Soluzioni per la gestione della catena di approvvigionamento
 - 6.5.5 Gestione delle relazioni 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 sugli investimenti nelle TIC e nella pianificazione dei sistemi informativi
 - 6.6.1 Criteri per le decisioni di investimento nelle TIC
 - 6.6.2 Collegare il progetto al piano di gestione e di business
 - 6.6.3 Implicazioni nella gestione
 - 6.6.4 Riprogettazione dei processi aziendali
 - 6.6.5 Decisione della direzione sulle metodologie di implementazione
 - 6.6.6 Necessità di pianificazione dei sistemi informativi
 - 6.6.7 Obiettivi, partecipanti e tempistiche
 - 6.6.8 Struttura e sviluppo del piano di sistema
 - 6.6.9 Monitoraggio e aggiornamento
- 6.7. Considerazioni sulla sicurezza nell'uso delle TIC
 - 6.7.1 Analisi dei rischi
 - 6.7.2 Sicurezza nei sistemi informativi
 - 6.7.3 Consigli pratici
- 6.8. Fattibilità dell'implementazione di progetti TIC e aspetti finanziari dei progetti di sistemi informativi
 - 6.8.1 Descrizione e obiettivi
 - 6.8.2 Partecipanti all'EVS
 - 6.8.3 Tecniche e pratiche
 - 6.8.4 Struttura dei costi
 - 6.8.5 Proiezione finanziaria
 - 6.8.6 Bilanci
- 6.9. *Business Intelligence*
 - 6.9.1 Cos'è la Business Intelligence?
 - 6.9.2 Strategia e implementazione della BI
 - 6.9.3 Presente e futuro in BI
- 6.10. ISO/IEC 12207
 - 6.10.1 Cos'è «ISO/IEC 12207»?
 - 6.10.2 Analisi dei sistemi informativi
 - 6.10.3 Progettazione di un sistema informativo
 - 6.10.4 Implementazione e accettazione del sistema informativo

Modulo 7. Riutilizzo di software

- 7.1. Panoramica del riutilizzo di software
 - 7.1.1 In che consiste il riutilizzo di software
 - 7.1.2 Vantaggi e svantaggi del riutilizzo di software
 - 7.1.3 Principali tecniche di riutilizzo di software
- 7.2. Introduzione ai modelli di progettazione
 - 7.2.1 Che 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 di creazione I
 - 7.3.1 Modelli di creazione
 - 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 metodo Builder
 - 7.3.6 Modello Abstract Factory vs Builder
- 7.4. Modelli di creazione II
 - 7.4.1 Factory Method
 - 7.4.2 Factory Method vs. Abstract Factory
 - 7.4.3 Pattern 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 Pattern 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 Modelli di comportamento: catena di responsabilità
 - 7.8.3 Modello di comportamento Ordine
- 7.9. Modelli di comportamento II
 - 7.9.1 Modello Interprete o *Interpreter*
 - 7.9.2 Modello di iterazione
 - 7.9.3 Modello di osservazione
 - 7.9.4 Modello di strategia
- 7.10. *Framework*
 - 7.10.1 Concetto di *framework*
 - 7.10.2 Sviluppo mediante *framework*
 - 7.10.3 Modello Model View Controller
 - 7.10.4 *Framework* per la progettazione di interfacce grafiche
 - 7.10.5 *Framework* per lo sviluppo di applicazioni web
 - 7.10.6 *Framework* per la gestione della persistenza degli oggetti nei database

Modulo 8. Servizi di tecnologia dell'informazione

- 8.1. Trasformazione digitale I
 - 8.1.1 Innovazione aziendale
 - 8.1.2 Gestione della produzione
 - 8.1.3 Gestione finanziaria
- 8.2. Trasformazione digitale II
 - 8.2.1 Il marketing
 - 8.2.2 La gestione delle risorse umane
 - 8.2.3 Un sistema di informazione integrato
- 8.3. Caso di studio
 - 8.3.1 Presentazione dell'azienda
 - 8.3.2 Metodologie di analisi degli acquisti 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 di riferimento per la **governance** e la gestione delle TSI
 - 8.4.4 **Governance** e gestione delle TSI e standard
- 8.5. La **governance** aziendale delle TIC
 - 8.5.1 Che cos'è la buona **governance** aziendale?
 - 8.5.2 Background sulla **governance** delle TIC
 - 8.5.3 La norma ISO/IEC 38500:2008
 - 8.5.4 Attuare una buona **governance** delle TIC
 - 8.5.5 **Governance** delle TIC e **best practice**
 - 8.5.6 La **governance** aziendale: Panoramica e tendenze
- 8.6. Obiettivi di Controllo per l'Informazione e la Tecnologia Correlata (COBIT)
 - 8.6.1 Quadro di attuazione
 - 8.6.2 Dominio: pianificazione e organizzazione
 - 8.6.3 Dominio: acquisizione e implementazione
 - 8.6.4 Dominio: consegna e supporto
 - 8.6.5 Dominio: monitoraggio e valutazione
 - 8.6.6 Implementazione della guida COBIT
- 8.7. La Biblioteca dell'Infrastruttura Tecnologica dell'Informazione (ITIL)
 - 8.7.1 Introduzione alla ITIL
 - 8.7.2 Strategie di servizio
 - 8.7.3 Design del servizio
 - 8.7.4 Transizione del servizio
 - 8.7.5 Funzionamento del servizio
 - 8.7.6 Miglioramento del servizio
- 8.8. Il sistema di gestione dei servizi
 - 8.8.1 Principi fondamentali di UNE-ISO/IEC 20000-1
 - 8.8.2 La 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. Il sistema di gestione dei beni dei software
 - 8.9.1 Giustificazione della necessità
 - 8.9.2 Contesto
 - 8.9.3 Presentazione della norma 19770
 - 8.9.4 Attuazione della gestione
- 8.10. Gestione della continuità aziendale
 - 8.10.1 Piani di continuità aziendale
 - 8.10.2 Implementazione di un BCM

Modulo 9. Sicurezza nei sistemi informativi

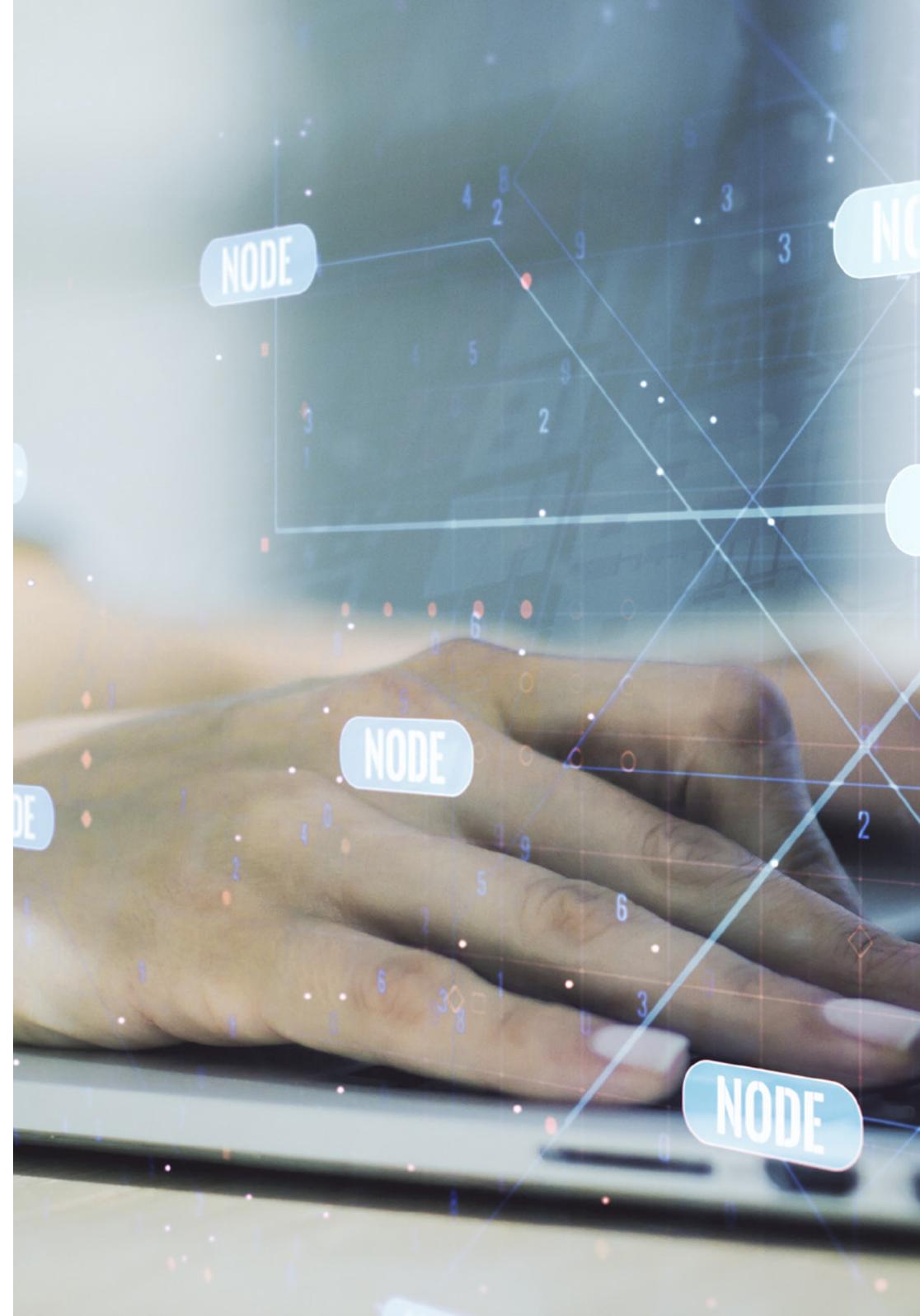
- 9.1. Panoramica sulla sicurezza, la crittografia e le crittoanalisi classiche
 - 9.1.1 Sicurezza informatica: prospettiva storica
 - 9.1.2 Ma cos'è esattamente la sicurezza?
 - 9.1.3 Storia della crittografia
 - 9.1.4 Cifrari sostitutivi
 - 9.1.5 Caso di studio: la macchina Enigma
- 9.2. Crittografia simmetrica
 - 9.2.1 Introduzione e terminologia 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 del 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 in 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. Architetture di sicurezza
 - 9.5.1 Architetture di sicurezza tradizionali
 - 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 sicuro del codice
 - 9.6.1 Sicurezza operativa
 - 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: buffer overflow
 - 9.6.12. Botch crittografici
- 9.7. *Botnet* e Spam
 - 9.7.1 Origine del problema
 - 9.7.2 Processo di spam
 - 9.7.3 Invio di spam
 - 9.7.4 Affinamento della 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. Controllo e attacchi web
 - 9.8.1 Raccolta di informazioni
 - 9.8.2 Tecniche di attacco
 - 9.8.3 Strumenti
- 9.9. Malware e codice maligno
 - 9.9.1 Che cos'è il malware?
 - 9.9.2 Tipi di malware
 - 9.9.3 Virus
 - 9.9.4 Criptovirus
 - 9.9.5 Worm
 - 9.9.6 Adware
 - 9.9.7 Spyware
 - 9.9.8 Hoaxes
 - 9.9.9 Pishing
 - 09.9.10. Trojan
 - 09.9.11. L'economia del malware
 - 09.9.12. 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 del project management 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 gestione dei progetti?
 - 10.1.4 Che cos'è un piano di progetto?
 - 10.1.5 Benefici
 - 10.1.6 Ciclo 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 processo 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 Sviluppo della carta del progetto
 - 10.2.3 Riunione di avvio del progetto
 - 10.2.4 Compiti, nozioni e competenze nel processo di start-up
 - 10.2.5 Il piano di progetto
 - 10.2.6 Sviluppo del piano base: Passaggi
 - 10.2.7 Compiti, nozioni e competenze nel processo di pianificazione
- 10.3. Gestione degli *stakeholder* e del campo di applicazione
 - 10.3.1 Identificare le parti interessate
 - 10.3.2 Sviluppare un piano per la gestione delle parti interessate
 - 10.3.3 Gestire il coinvolgimento delle parti interessate
 - 10.3.4 Controllare il coinvolgimento delle parti interessate
 - 10.3.5 L'obiettivo del progetto
 - 10.3.6 La gestione e il piano relativo al campo di applicazione
 - 10.3.7 Riunire i requisiti
 - 10.3.8 Definire l'ambito di applicazione
 - 10.3.9 Creare la WBS
 - 10.3.10. Verificare e controllare il campo di applicazione
- 10.4. L'ideazione della tabella di marcia
 - 10.4.1 La gestione del tempo e il relativo piano
 - 10.4.2 Definire le attività
 - 10.4.3 Sequenza delle attività
 - 10.4.4 Stima delle risorse delle attività
 - 10.4.5 Stimare la durata delle attività
 - 10.4.6 Sviluppo della tabella di marcia e calcolo del percorso critico
 - 10.4.7 Controllare il cronoprogramma
- 10.5. Sviluppo del budget e risposta ai rischi
 - 10.5.1 Fare una stima dei costi
 - 10.5.2 Definire 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 un'analisi dei rischi
 - 10.5.6 Lo sviluppo del piano di risposta
- 10.6. Gestione qualità
 - 10.6.1 Pianificazione della qualità
 - 10.6.2 Garanzia della qualità
 - 10.6.3 Controllo di qualità
 - 10.6.4 Concetti statistici di base
 - 10.6.5 Strumenti per la 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 Gestire le comunicazioni
- 10.7.8 La gestione delle risorse umane
- 10.7.9 Soggetti principali e il loro ruolo nei progetti
- 10.7.10. Tipi di organizzazione
- 10.7.11. Organizzazione del progetto
- 10.7.12. Squadre di lavoro
- 10.8. Approvvigionamento
 - 10.8.1 Il processo di acquisto
 - 10.8.2 Pianificazione
 - 10.8.3 Ricerca di fornitori e presentazione di offerte
 - 10.8.4 Assegnazione del contratto
 - 10.8.5 Gestione del contratto
 - 10.8.6 I contratti
 - 10.8.7 Tipi di contratto
 - 10.8.8 Negoziazione del contratto
- 10.9. Attuazione, monitoraggio, controllo e chiusura
 - 10.9.1 I gruppi dei processi
 - 10.9.2 Attuazione del progetto
 - 10.9.3 Monitoraggio e controllo del progetto
 - 10.9.4 Chiusura del progetto



10.10. Responsabilità professionale

10.10.1 Responsabilità professionale

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 Vantaggi della professionalizzazione

06 Tirocinio

Una volta completato il periodo teorico online, questo Master Semipresenziale in Ingegneria del Software Avanzata prevede un tirocinio presso un'istituzione di riferimento. Durante questo periodo, gli studenti saranno affiancati da un tutor che li aiuterà a trarre il massimo da questa esperienza. Di conseguenza, gli informatici acquisiranno competenze avanzate per fare un salto di qualità significativo nel loro percorso professionale.



“

*Svolgerai lo stage in un'azienda leader,
dove avrai il supporto di rinomati
professionisti dell'Ingegneria del
Software Avanzata”*

Il periodo di formazione pratica di questo programma in Ingegneria del Software Avanzata consiste in un soggiorno di 3 settimane in un'azienda rinomata, dal lunedì al venerdì, con 8 ore consecutive di formazione pratica con un tutor specializzato.

In questa proposta formativa, di natura completamente pratica, le attività sono finalizzate a sviluppare e perfezionare le competenze necessarie per l'ideazione e lo sviluppo di servizi informatici di Ingegneria del Software Avanzata, nonché le condizioni che richiedono un alto livello di qualificazione, orientato alla formazione specifica per l'esercizio dell'attività.

Questa è un'ottima opportunità per gli informatici di immergersi in un ambiente di lavoro reale, dove faranno parte di un team di professionisti specializzati in questo campo. In questo modo, gli studenti parteciperanno a progetti relativi a temi quali la sicurezza dei sistemi informativi, il riutilizzo del software o la progettazione di applicazioni web.

La parte pratica si svolgerà con la partecipazione attiva dello studente che svolge le attività e le procedure di ogni area di competenza (imparare a imparare e imparare a fare), con l'accompagnamento e la guida di insegnanti e altri partner formativi che facilitano il lavoro di gruppo e l'integrazione multidisciplinare come competenze trasversali per la prassi di Informatico (imparare a essere e imparare a relazionarsi).

Le procedure descritte di seguito costituiscono la base della parte pratica della formazione e la loro realizzazione è subordinata alla disponibilità dell'istituto e al suo carico di lavoro. Le attività proposte sono le seguenti:





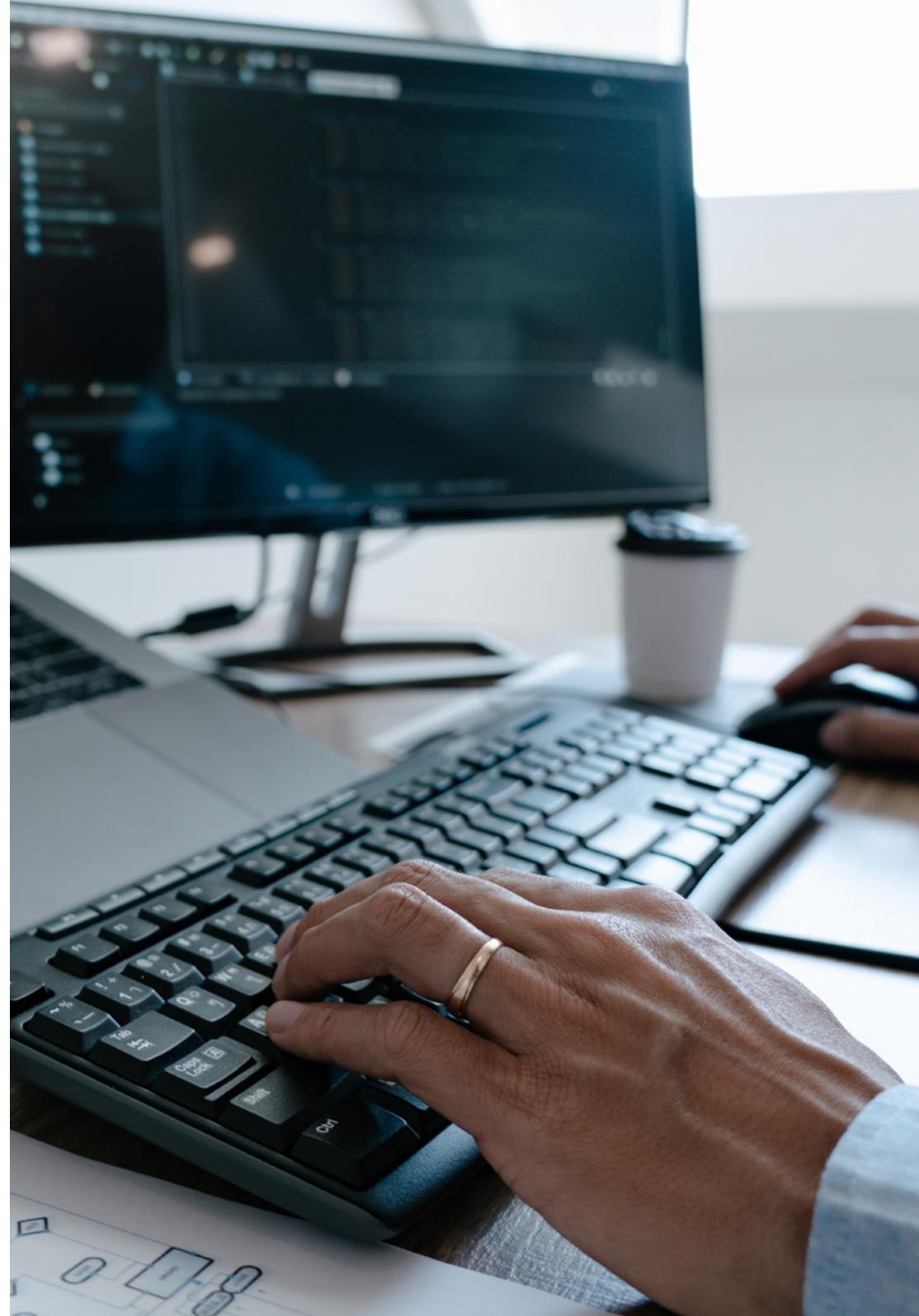
Modulo	Attività Pratica
Sviluppo di Software Avanzato	Progettare architetture software scalabili, affidabili e di facile manutenzione
	Utilizzare tecniche di modellazione come l' <i>Unified Modeling Language</i> per rappresentare la struttura e il comportamento del sistema prima dell'implementazione
	Scrivere codici efficienti e puliti in vari linguaggi di programmazione
	Utilizzare metodologie agili per pianificare, eseguire e monitorare progetti software
Gestione dei requisiti	Analizzare l'ambiente dell'utente e studiare il dominio applicativo per identificare i problemi che il software deve risolvere
	Utilizzo di tecniche di modellazione (come diagrammi UML o casi d'uso e scenari) per rappresentare i requisiti in modo strutturato
	Redigere i documenti di specifica dei requisiti con gli <i>stakeholder</i> per garantire che i requisiti acquisiti siano corretti
	Sviluppare prototipi e simulazioni del sistema per convalidare i requisiti con gli utenti
Gestione dei requisiti	Stabilire standard di qualità per la manutenzione dei sistemi informativi, sulla base del <i>framework</i>
	Eseguire test funzionali, di prestazione, di sicurezza e di <i>usability</i>
	Identificare e valutare i rischi associati ai sistemi informativi
	Implementare <i>pipeline</i> di integrazione e distribuzione continua per garantire che le nuove release del software siano sviluppate, testate e distribuite in modo efficiente
Connessione di sistemi	Approfondire i sistemi per identificare le potenziali sfide di integrazione e pianificare soluzioni efficaci
	Creare modelli di dati che definiscano come i dati saranno strutturati e gestiti tra i sistemi integrati
	Progettare l'architettura di integrazione, selezionando i modelli e i metodi più appropriati (ad esempio, integrazione basata sui servizi, messaggistica, API)
	Configurare le interfacce che consentono la comunicazione e il trasferimento di dati tra i sistemi <i>embedded</i>

Assicurazione di responsabilità civile

La preoccupazione principale di questa istituzione è quella di garantire la sicurezza sia dei tirocinanti sia degli altri agenti che collaborano ai processi di tirocinio in azienda. All'interno delle misure rivolte a questo fine ultimo, esiste la risposta a qualsiasi incidente che possa verificarsi durante il processo di insegnamento-apprendimento.

A tal fine, questa entità didattica si impegna a stipulare un'assicurazione di responsabilità civile per coprire qualsiasi eventualità possa verificarsi durante svolgimento del tirocinio all'interno del centro di collocamento.

La polizza di responsabilità civile per i tirocinanti deve garantire una copertura assicurativa completa e deve essere stipulata prima dell'inizio del periodo di tirocinio. Grazie a questa garanzia, il professionista non avrà alcuna preoccupazione nel caso di eventuali situazioni impreviste che possano insorgere durante il tirocinio e potrà godere di una copertura assicurativa fino al termine dello stesso.



Condizioni generali del tirocinio

Le condizioni generali dell'accordo di tirocinio per il programma sono le seguenti:

1. TUTORAGGIO: durante il Master Semipresenziale agli studenti verranno assegnati due tutor che li seguiranno durante tutto il percorso, risolvendo eventuali dubbi e domande. Da un lato, lo studente disporrà di un tutor professionale appartenente al centro di inserimento lavorativo che lo guiderà e lo supporterà in ogni momento. Dall'altro lato, allo studente verrà assegnato anche un tutor accademico che avrà il compito di coordinare e aiutare lo studente durante l'intero processo, risolvendo i dubbi e fornendogli tutto ciò di cui potrebbe aver bisogno. In questo modo, il professionista sarà accompagnato in ogni momento e potrà risolvere tutti gli eventuali dubbi, sia di natura pratica che accademica.

2. DURATA: il programma del tirocinio avrà una durata di tre settimane consecutive di preparazione pratica, distribuite in giornate di 8 ore lavorative, per cinque giorni alla settimana. I giorni di frequenza e l'orario saranno di competenza del centro, che informerà debitamente e preventivamente il professionista, con un sufficiente anticipo per facilitarne l'organizzazione.

3. ASSENZE: in caso di mancata presentazione il giorno di inizio del Master Semipresenziale, lo studente perderà il diritto allo stesso senza possibilità di rimborso o di modifica di date. L'assenza per più di due giorni senza un giustificato motivo/certificato medico comporterà la rinuncia dello studente al tirocinio e, pertanto, la relativa automatica cessazione. In caso di ulteriori problemi durante lo svolgimento del tirocinio, essi dovranno essere debitamente e urgentemente segnalati al tutor accademico.

4. CERTIFICAZIONE: lo studente che supererà il Master Semipresenziale riceverà un certificato che attesterà il tirocinio svolto presso il centro in questione.

5. RAPPORTO DI LAVORO: il Master Semipresenziale non costituisce alcun tipo di rapporto lavorativo.

6. STUDI PRECEDENTI: alcuni centri potranno richiedere un certificato di studi precedenti per la partecipazione al Master Semipresenziale. In tal caso, sarà necessario esibirlo al dipartimento tirocini di TECH affinché venga confermata l'assegnazione del centro prescelto.

7. NON INCLUDE: il Master Semipresenziale non includerà nessun elemento non menzionato all'interno delle presenti condizioni. Pertanto, non sono inclusi alloggio, trasporto verso la città in cui si svolge il tirocinio, visti o qualsiasi altro servizio non menzionato.

Tuttavia, gli studenti potranno consultare il proprio tutor accademico per qualsiasi dubbio o raccomandazione in merito. Egli fornirà tutte le informazioni necessarie per semplificare le procedure.

07

Dove posso svolgere il tirocinio?

Fedele alla sua filosofia di fornire titoli universitari di alta qualità, TECH espande le opportunità accademiche per gli studenti e permette loro di svolgere la loro formazione pratica in diverse realtà di prestigio internazionale. Gli informatici hanno quindi l'opportunità perfetta per migliorare la propria qualità professionale lavorando con i migliori specialisti nel campo dell'Ingegneria del Software Avanzata.





“

Effettuerai il tirocinio in un'azienda di riferimento nel settore dell'Ingegneria del Software Avanzata”

tech 46 | Dove posso svolgere il tirocinio?



Gli studenti potranno svolgere il tirocinio di questo Master Semipresenziale presso i seguenti centri:



Informatica

Captia Ingeniería

Paese	Città
Spagna	Madrid

Indirizzo: Av. de las Nieves, 37, Bloque A Planta 1
Oficina E, 28935, Móstoles, Madrid

Azienda IT dedicata alla fornitura di soluzioni tecnologiche avanzate alle industrie

Tirocini correlati:

- Visual Analytics e Big Data
- Sviluppo di Software





“

Approfondisci la teoria più rilevante in questo campo, applicandola successivamente in un ambiente di lavoro reale”

08

Metodologia

Dieses Fortbildungsprogramm bietet eine andere Art des Lernens. Unsere Methodik wird durch eine zyklische Lernmethode entwickelt: **das Relearning**.

Dieses Lehrsystem wird z. B. an den renommiertesten medizinischen Fakultäten der Welt angewandt und wird von wichtigen Publikationen wie dem **New England Journal of Medicine** als eines der effektivsten angesehen.





Entdecken Sie Relearning, ein System, das das herkömmliche lineare Lernen hinter sich lässt und Sie durch zyklische Lehrsysteme führt: eine Art des Lernens, die sich als äußerst effektiv erwiesen hat, insbesondere in Fächern, die Auswendiglernen erfordern"

Fallstudie zur Kontextualisierung aller Inhalte

Unser Programm bietet eine revolutionäre Methode zur Entwicklung von Fähigkeiten und Kenntnissen. Unser Ziel ist es, Kompetenzen in einem sich wandelnden, wettbewerbsorientierten und sehr anspruchsvollen Umfeld zu stärken.

“

Mit TECH werden Sie eine Art des Lernens erleben, die an den Grundlagen der traditionellen Universitäten auf der ganzen Welt rüttelt"



Sie werden Zugang zu einem Lernsystem haben, das auf Wiederholung basiert, mit natürlichem und progressivem Unterricht während des gesamten Lehrplans.



Der Student wird durch gemeinschaftliche Aktivitäten und reale Fälle lernen, wie man komplexe Situationen in realen Geschäftsumgebungen löst.

Eine innovative und andersartige Lernmethode

Dieses TECH-Programm ist ein von Grund auf neu entwickeltes, intensives Lehrprogramm, das die anspruchsvollsten Herausforderungen und Entscheidungen in diesem Bereich sowohl auf nationaler als auch auf internationaler Ebene vorsieht. Dank dieser Methodik wird das persönliche und berufliche Wachstum gefördert und ein entscheidender Schritt in Richtung Erfolg gemacht. Die Fallmethode, die Technik, die diesem Inhalt zugrunde liegt, gewährleistet, dass die aktuellste wirtschaftliche, soziale und berufliche Realität berücksichtigt wird.

“ *Unser Programm bereitet Sie darauf vor, sich neuen Herausforderungen in einem unsicheren Umfeld zu stellen und in Ihrer Karriere erfolgreich zu sein* **”**

Die Fallmethode ist das am weitesten verbreitete Lernsystem an den besten Informatikschulen der Welt, seit es sie gibt. Die Fallmethode wurde 1912 entwickelt, damit Jurastudenten das Recht nicht nur auf der Grundlage theoretischer Inhalte erlernen. Sie bestand darin, ihnen reale komplexe Situationen zu präsentieren, damit sie fundierte Entscheidungen treffen und Werturteile darüber fällen konnten, wie diese zu lösen sind. Sie wurde 1924 als Standardlehrmethode in Harvard etabliert.

Was sollte eine Fachkraft in einer bestimmten Situation tun? Mit dieser Frage konfrontieren wir Sie in der Fallmethode, einer handlungsorientierten Lernmethode. Während des gesamten Kurses werden die Studenten mit mehreren realen Fällen konfrontiert. Sie müssen ihr gesamtes Wissen integrieren, recherchieren, argumentieren und ihre Ideen und Entscheidungen verteidigen.

Relearning Methodology

TECH kombiniert die Methodik der Fallstudien effektiv mit einem 100%igen Online-Lernsystem, das auf Wiederholung basiert und in jeder Lektion verschiedene didaktische Elemente kombiniert.

Wir ergänzen die Fallstudie mit der besten 100%igen Online-Lehrmethode: Relearning.

*Im Jahr 2019 erzielten wir die besten
Lernergebnisse aller spanischsprachigen
Online-Universitäten der Welt.*

Bei TECH lernen Sie mit einer hochmodernen Methodik, die darauf ausgerichtet ist, die Führungskräfte der Zukunft zu spezialisieren. Diese Methode, die an der Spitze der weltweiten Pädagogik steht, wird Relearning genannt.

Unsere Universität ist die einzige in der spanischsprachigen Welt, die für die Anwendung dieser erfolgreichen Methode zugelassen ist. Im Jahr 2019 ist es uns gelungen, die Gesamtzufriedenheit unserer Studenten (Qualität der Lehre, Qualität der Materialien, Kursstruktur, Ziele...) in Bezug auf die Indikatoren der besten spanischsprachigen Online-Universität zu verbessern.



In unserem Programm ist das Lernen kein linearer Prozess, sondern erfolgt in einer Spirale (lernen, verlernen, vergessen und neu lernen). Daher wird jedes dieser Elemente konzentrisch kombiniert. Mit dieser Methode wurden mehr als 650.000 Hochschulabsolventen mit beispiellosem Erfolg in so unterschiedlichen Bereichen wie Biochemie, Genetik, Chirurgie, internationales Recht, Managementfähigkeiten, Sportwissenschaft, Philosophie, Recht, Ingenieurwesen, Journalismus, Geschichte, Finanzmärkte und -instrumente fortgebildet. Dies alles in einem sehr anspruchsvollen Umfeld mit einer Studentenschaft mit hohem sozioökonomischem Profil und einem Durchschnittsalter von 43,5 Jahren.

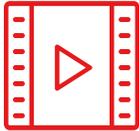
Das Relearning ermöglicht es Ihnen, mit weniger Aufwand und mehr Leistung zu lernen, sich mehr auf Ihre Spezialisierung einzulassen, einen kritischen Geist zu entwickeln, Argumente zu verteidigen und Meinungen zu kontrastieren: eine direkte Gleichung zum Erfolg.

Nach den neuesten wissenschaftlichen Erkenntnissen der Neurowissenschaften wissen wir nicht nur, wie wir Informationen, Ideen, Bilder und Erinnerungen organisieren, sondern auch, dass der Ort und der Kontext, in dem wir etwas gelernt haben, von grundlegender Bedeutung dafür sind, dass wir uns daran erinnern und es im Hippocampus speichern können, um es in unserem Langzeitgedächtnis zu behalten.

Auf diese Weise sind die verschiedenen Elemente unseres Programms im Rahmen des so genannten Neurocognitive Context-Dependent E-Learning mit dem Kontext verbunden, in dem der Teilnehmer seine berufliche Praxis entwickelt.



Dieses Programm bietet die besten Lehrmaterialien, die sorgfältig für Fachleute aufbereitet sind:



Studienmaterial

Alle didaktischen Inhalte werden von den Fachleuten, die den Kurs unterrichten werden, speziell für den Kurs erstellt, so dass die didaktische Entwicklung wirklich spezifisch und konkret ist.

Diese Inhalte werden dann auf das audiovisuelle Format angewendet, um die Online-Arbeitsmethode von TECH zu schaffen. All dies mit den neuesten Techniken, die in jedem einzelnen der Materialien, die dem Studenten zur Verfügung gestellt werden, qualitativ hochwertige Elemente bieten.



Meisterklassen

Die Nützlichkeit der Expertenbeobachtung ist wissenschaftlich belegt.

Das sogenannte Learning from an Expert festigt das Wissen und das Gedächtnis und schafft Vertrauen für zukünftige schwierige Entscheidungen.



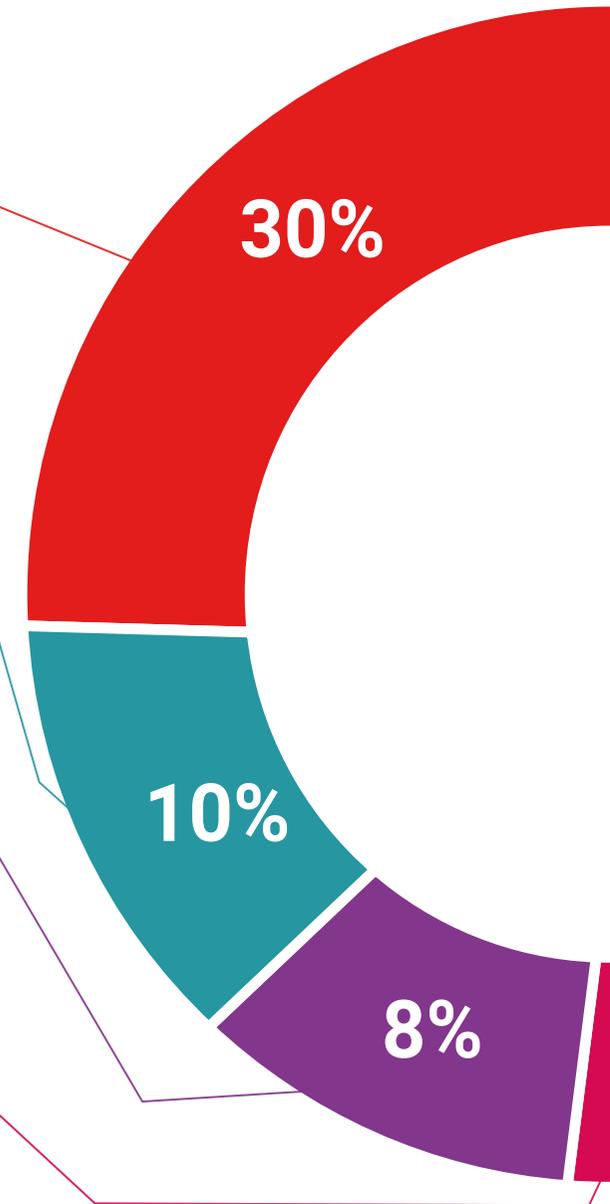
Übungen für Fertigkeiten und Kompetenzen

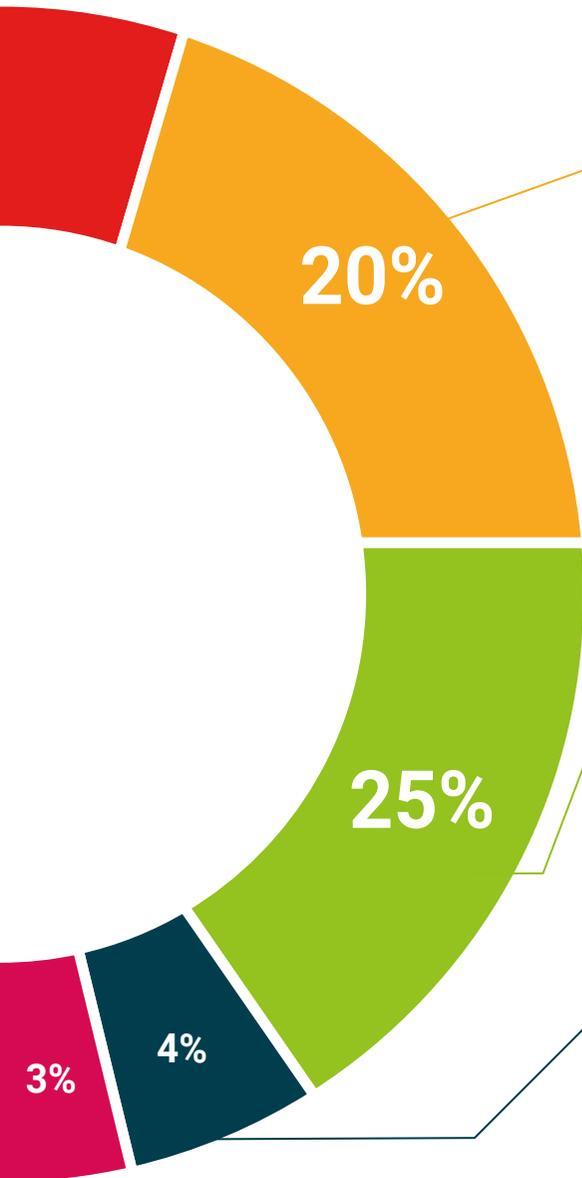
Sie werden Aktivitäten durchführen, um spezifische Kompetenzen und Fertigkeiten in jedem Fachbereich zu entwickeln. Übungen und Aktivitäten zum Erwerb und zur Entwicklung der Fähigkeiten und Fertigkeiten, die ein Spezialist im Rahmen der Globalisierung, in der wir leben, entwickeln muss.



Weitere Lektüren

Aktuelle Artikel, Konsensdokumente und internationale Leitfäden, u. a. In der virtuellen Bibliothek von TECH hat der Student Zugang zu allem, was er für seine Fortbildung benötigt.





Case Studies

Sie werden eine Auswahl der besten Fallstudien vervollständigen, die speziell für diese Qualifizierung ausgewählt wurden. Die Fälle werden von den besten Spezialisten der internationalen Szene präsentiert, analysiert und betreut.



Interaktive Zusammenfassungen

Das TECH-Team präsentiert die Inhalte auf attraktive und dynamische Weise in multimedialen Pillen, die Audios, Videos, Bilder, Diagramme und konzeptionelle Karten enthalten, um das Wissen zu vertiefen.

Dieses einzigartige Bildungssystem für die Präsentation multimedialer Inhalte wurde von Microsoft als "Europäische Erfolgsgeschichte" ausgezeichnet.



Testing & Retesting

Die Kenntnisse des Studenten werden während des gesamten Programms regelmäßig durch Bewertungs- und Selbsteinschätzungsaktivitäten und -übungen beurteilt und neu bewertet, so dass der Student überprüfen kann, wie er seine Ziele erreicht.



09

Certificazione

Il Master Semipresenziale in Ingegneria del Software Avanzata garantisce, oltre alla preparazione più rigorosa e aggiornata, il conseguimento di una qualifica di Master Semipresenziale rilasciata da TECH Università Tecnologica.



“

*Porta a termine questo programma e ricevi
il tuo titolo universitario senza spostamenti
o fastidiose formalità”*

Questo programma ti consentirà di ottenere il titolo di studio privato di **Master Semipresenziale in Ingegneria del Software Avanzata** possiede il programma più completo e aggiornato del mercato.

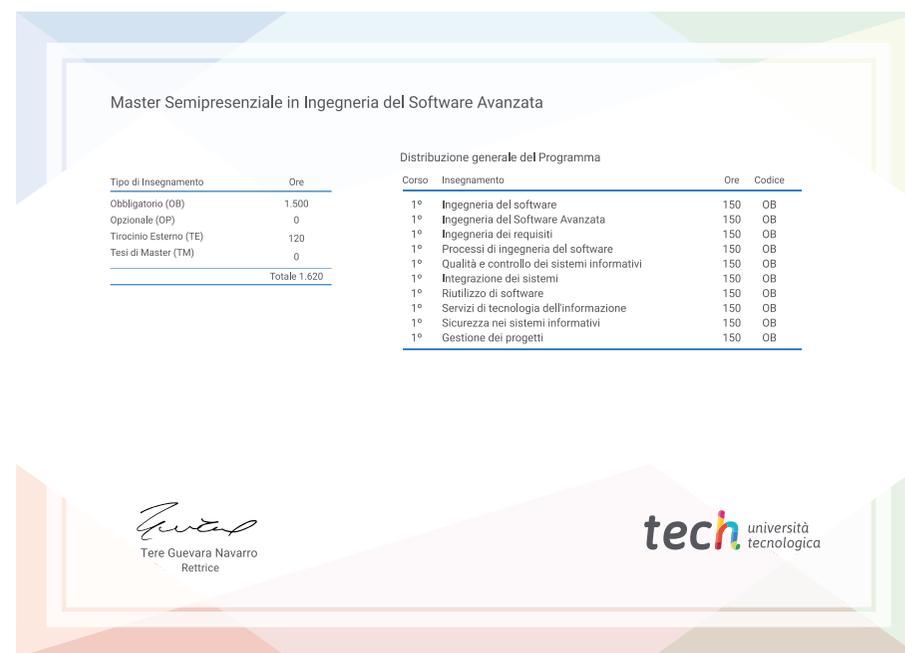
Dopo aver superato la valutazione, lo studente riceverà mediante lettera certificata* con ricevuta di ritorno, la sua corrispondente qualifica di **Master Semipresenziale** rilasciata da **TECH Università Tecnologica**.

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

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

Modalità: **Semipresenziale (Online + Tirocinio)**

Durata: **12 mesi**



*Apostille dell'Aia. 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 presento qualità
formazione online
sviluppo istituzioni
classe virtuale

tech università
tecnologica

Master Semipresenziale Ingegneria del Software Avanzata

Modalità: Semipresenziale (Online + Tirocinio)

Durata: 12 mesi

Certificazione: TECH Università Tecnologica

Crediti: 60 + 4 ECTS

Master Semipresenziale

Ingegneria del Software Avanzata