

Master Specialistico Ingegneria dei Software



Master Specialistico Ingegneria dei Software

- » Modalità: online
- » Durata: 2 anni
- » Titolo: TECH Global University
- » Accreditamento: 120 ECTS
- » Orario: a scelta
- » Esami: online

Accesso al sito web: www.techtute.com/it/informatica/master-specialistico/master-specialistico-ingegneria-software

Indice

01

Presentazione

pag. 4

02

Obiettivi

pag. 8

03

Direzione del corso

pag. 14

04

Competenze

pag. 18

05

Struttura e contenuti

pag. 22

06

Metodologia

pag. 44

07

Titolo

pag. 52

01

Presentazione

Negli ultimi anni la domanda di software ha registrato una forte crescita. Con l'emergere di nuove piattaforme digitali, di hardware più sofisticati e di una crescente virtualizzazione dei processi quotidiani, gli ingegneri del *software* si trovano continuamente ad affrontare nuove sfide. Il pubblico si sta abituando sempre di più alle nuove tecnologie e le esigenze sono sempre maggiori, motivo per cui gli esperti nello sviluppo di *software* devono adattarsi a queste richieste e creare prodotti all'altezza delle aspettative del mercato. Ciò richiede un elevato livello di competenza tecnica in diversi campi delle conoscenze informatiche.



“

Ricoprirai un ruolo chiave nel futuro tecnologico di molte aziende. Specializzati in Ingegneria dei Software e inizia a sviluppare i sistemi che faranno la differenza”

L'industria tecnologica è uno dei settori più rilevanti al giorno d'oggi, in quanto quasi tutti interagiscono quotidianamente con qualche tipo di dispositivo digitale. In questo contesto, gli ingegneri del software sono in prima linea nell'intero processo di sviluppo tecnologico, in quanto sono coloro che devono costantemente aggiornare i sistemi, svilupparne di nuovi e offrire soluzioni intelligenti ai problemi che si presentano. In questo senso, i professionisti dell'ingegneria informatica devono essere persone molto decise, con grandi conoscenze tecniche e un'eccezionale capacità di adattarsi a tutti i tipi di sviluppo e di contesto.

Con questo obiettivo, TECH ha progettato questo Master Specialistico in Ingegneria dei Software, che offre una preparazione completa e di alto livello a tutti gli sviluppatori che vogliono specializzare la propria carriera e indirizzarla verso la creazione di sistemi. Da un lato, il programma affronta le diverse metodologie per la creazione e la gestione di un progetto di sviluppo di software, nonché tutti gli aspetti da tenere in considerazione che riguardano l'informatica, i requisiti e le piattaforme. D'altra parte, si occupa anche della sicurezza del software stesso, dei sistemi informativi e dell'ambiente di lavoro utilizzati durante il processo. Al termine del corso, lo studente disporrà di tutte le conoscenze necessarie per diventare un esperto competente ed efficiente nell'ambito dell'Ingegneria dei Software.

Inoltre, uno dei principali vantaggi di questo programma è che si svolge al 100% online. Ciò significa che lo studente non deve adattarsi a orari fissi e non è obbligato a recarsi in un centro fisico specifico. In questo modo, lo studente ha la libertà di gestire il proprio studio, secondo i propri ritmi e tenendo conto dei propri impegni, pianificando gli orari come preferisce.

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

- ♦ Sviluppo di casi di studio presentati da esperti in sviluppo di software
- ♦ Contenuti grafici, schematici ed eminentemente pratici che forniscono informazioni scientifiche e pratiche sulle discipline essenziali per l'esercizio della professione
- ♦ Esercizi pratici che offrono un processo di autovalutazione per migliorare l'apprendimento
- ♦ Speciale enfasi sulle metodologie innovative nel campo dell'Ingegneria dei Software
- ♦ Lezioni teoriche, domande all'esperto, forum di discussione su questioni controverse e compiti di riflessione individuale
- ♦ Contenuti disponibili da qualsiasi dispositivo fisso o mobile dotato di connessione a internet



Riesci ad immaginare di essere coinvolto nello sviluppo di Netflix? È il momento di smettere di immaginare e di concentrare la tua carriera sui migliori progetti software"

“

La tua esperienza e le tue conoscenze possono fare la differenza in progetti di grandi dimensioni che richiedono molti requisiti. Non perdere l'opportunità di distinguerti professionalmente e iscriviti subito a questo Master Specialistico in Ingegneria dei Software”

Il personale docente del programma comprende rinomati professionisti in Ingegneria dei Software, che apportano la propria esperienza a questa preparazione, oltre a specialisti riconosciuti e appartenenti a società scientifiche e università di riferimento.

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 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. A tale fine, il professionista potrà usufruire di un innovativo sistema di video interattivi creati da esperti di rinomata fama.

L'obiettivo di TECH è farti diventare un grande ingegnere informatico. Viene garantito l'accesso al miglior materiale e al miglior insegnamento possibile.

Studia quando, dove e come vuoi. Il programma è online al 100% e si adatta alle tue esigenze, non il contrario.



02

Obiettivi

Questo Master Specialistico in Ingegneria dei Software è stato sviluppato con l'obiettivo di offrire a tutti i professionisti del settore informatico la specializzazione necessaria per focalizzare la propria carriera sullo sviluppo di software moderni e adatti alle nuove realtà mutevoli del mercato. Grazie alle conoscenze altamente tecniche impartite durante il corso, gli studenti aumenteranno notevolmente le loro possibilità di avanzamento professionale e di accesso a posti di lavoro in grandi aziende del settore.





“

Questo Master Specialistico sarà lo stimolo più grande e positivo che potrai dare alla tua carriera verso il successo professionale”

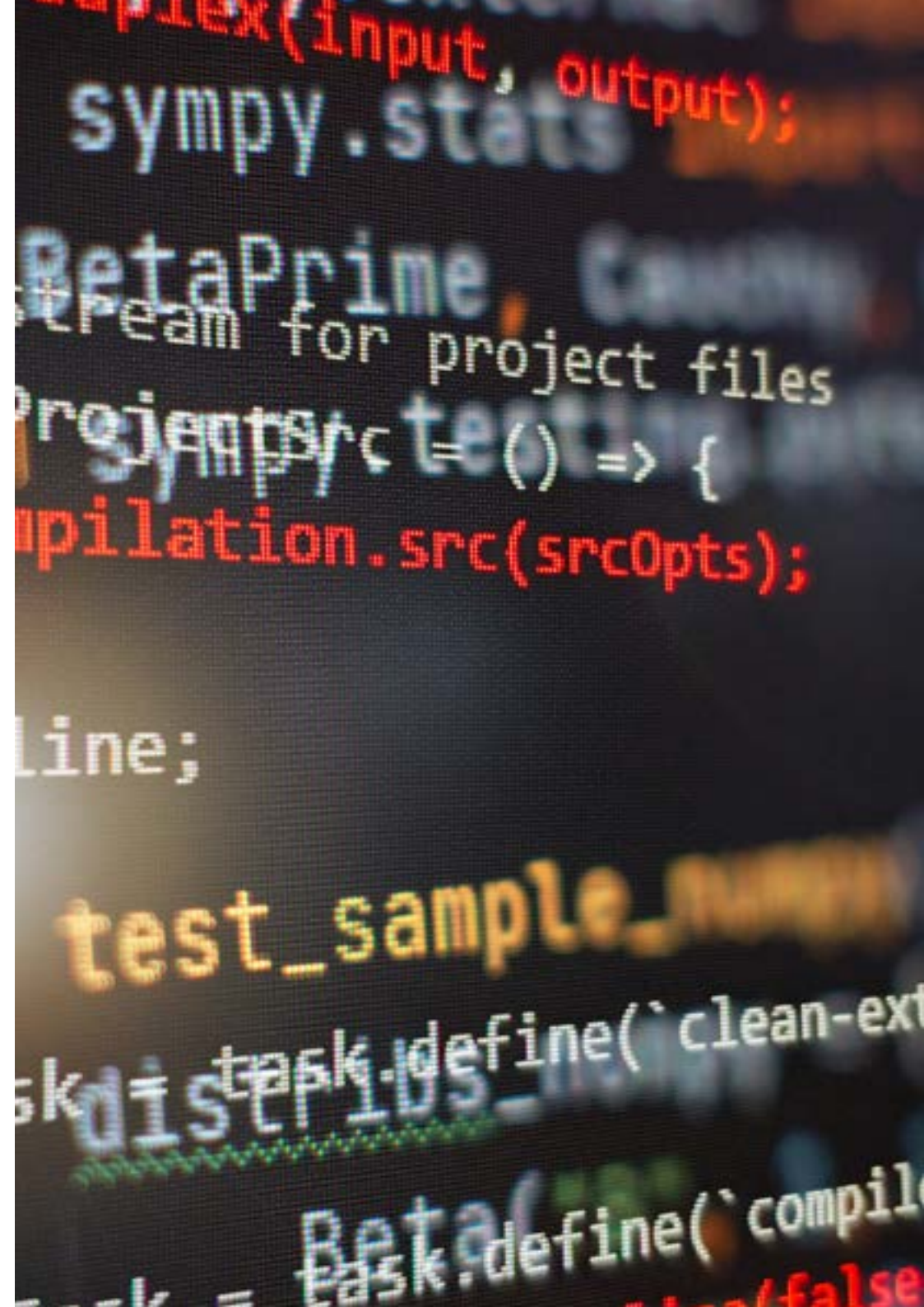


Obiettivi generali

- ♦ Acquisire le nuove competenze necessarie e richieste in termini di nuove tecnologie e ultimi sviluppi di software
- ♦ Completare le conoscenze acquisite con competenze nel campo dell'informatica e della struttura dei computer, comprese le basi matematiche, statistiche e fisiche essenziali nell'ingegneria
- ♦ Ampliare le conoscenze nel campo dell'Ingegneria dei Software e dei Sistemi Informatici con gli ultimi sviluppi e le metodologie più innovative
- ♦ Affrontare progetti e ambienti software complessi, sapendo fornire soluzioni intelligenti a problemi diversi



Una specializzazione che ti aiuterà a padroneggiare lo sviluppo di software con un insieme unico di competenze richieste da ogni azienda leader del settore





Obiettivi specifici

- ◆ Conoscere le basi dell'Ingegneria dei Software, nonché l'insieme di regole o principi etici e la responsabilità professionale durante e dopo lo sviluppo
- ◆ Comprendere il processo di sviluppo di software, i diversi modelli di programmazione e il paradigma di programmazione orientato agli oggetti
- ◆ Comprendere i diversi tipi di modellazione delle applicazioni e i modelli di progettazione nel linguaggio di modellazione unificato (UML)
- ◆ Comprendere i concetti fondamentali della gestione dei progetti e del loro ciclo di vita
- ◆ Comprendere il funzionamento della gestione della qualità nei progetti, compresi la pianificazione, la garanzia, il controllo, i concetti statistici e gli strumenti disponibili
- ◆ Acquisire le conoscenze essenziali relative alla responsabilità professionale derivante dalla gestione dei progetti
- ◆ Comprendere le diverse piattaforme di sviluppo software
- ◆ Acquisire le conoscenze necessarie per lo sviluppo di applicazioni e interfacce grafiche nei linguaggi *Java* e *.NET*
- ◆ Imparare gli ambienti di sviluppo delle applicazioni mobili *Android* e i processi di debug e pubblicazione
- ◆ Comprendere lo sviluppo di applicazioni basate sul cloud e determinare le procedure corrette per la sua implementazione
- ◆ Comprendere le procedure e le tecniche per migliorare l'aspetto di un documento scritto in HTML
- ◆ Acquisire le competenze necessarie per lo sviluppo di applicazioni web lato client
- ◆ Sviluppare applicazioni di strutture complesse, attraverso l'uso di diverse procedure, funzioni e oggetti che integrano *JavaScript*
- ◆ Imparare a utilizzare l'interfaccia di programmazione DOM per i documenti HTML e XML al fine di modificarne la struttura, lo stile e il contenuto
- ◆ Conoscere il concetto di usabilità del web, i suoi vantaggi, i principi, i metodi e le tecniche per rendere un sito web utilizzabile dall'utente
- ◆ Comprendere l'architettura software Model View Controller View (MVC) che separa i dati, l'interfaccia utente e la logica di controllo di un'applicazione in tre componenti distinti
- ◆ Acquisire le competenze per l'utilizzo dei servizi web, utilizzando XML, SOA e REST
- ◆ Imparare il processo di sicurezza delle informazioni, le sue implicazioni sulla riservatezza, l'integrità, la disponibilità e i costi economici
- ◆ Imparare l'uso di pratiche di sicurezza corrette nella gestione dei servizi informatici
- ◆ Acquisire le conoscenze per una corretta certificazione dei processi di sicurezza
- ◆ Comprendere i meccanismi e i metodi di autenticazione per il controllo degli accessi, nonché il loro processo di verifica
- ◆ Comprendere i programmi di gestione della sicurezza, la gestione del rischio e la progettazione delle politiche di sicurezza
- ◆ Imparare i piani di continuità aziendale, le loro fasi e il processo di manutenzione
- ◆ Comprendere le procedure per la corretta protezione dell'azienda attraverso reti DMZ, l'uso di sistemi di rilevamento delle intrusioni e altre metodologie
- ◆ Comprendere i problemi di sicurezza del software, le vulnerabilità e la loro classificazione
- ◆ Analizzare i diversi server web che si stanno diffondendo nel mercato odierno
- ◆ Comprendere il processo delle statistiche di utilizzo e del bilanciamento del carico nei server web
- ◆ Acquisire le conoscenze necessarie per la corretta esecuzione del processo di revisione e di controllo interno informatico

- ♦ 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
- ♦ Comprendere i diversi modelli di architetture di sistema e di progettazione del software, nonché l'architettura delle applicazioni cloud
- ♦ Approfondire la comprensione del miglioramento del processo di sviluppo del software e della sua qualità utilizzando gli standard ISO/IEC
- ♦ Comprendere l'importanza dell'ingegneria dei requisiti nel processo di sviluppo del software
- ♦ Approfondire la comprensione delle fonti dei requisiti e delle tecniche di elicitazione di questi, in quanto parte essenziale del processo
- ♦ Comprendere e applicare la prototipazione come parte essenziale del processo di sviluppo
- ♦ Gettare le basi per l'analisi forense nel mondo del software e del controllo informatico
- ♦ Comprendere i concetti fondamentali della gestione dei progetti e del loro ciclo di vita
- ♦ Imparare lo sviluppo della pianificazione per la gestione del tempo, lo sviluppo del budget e la risposta ai rischi
- ♦ Comprendere il funzionamento della gestione della qualità nei progetti, compresi la pianificazione, la garanzia, il controllo, i concetti statistici e gli strumenti disponibili



“

*Una preparazione completa che ti fornirà
le conoscenze necessarie per competere
con i migliori”*

03

Direzione del corso

Trattandosi di un'area estremamente specifica della modellazione 3D per i videogiochi, il personale docente incaricato della realizzazione di questo Corso Universitario conosce bene l'uso degli strumenti trattati nel corso del programma. Con contenuti aggiornati e una guida professionale personalizzata e orientata allo studente, questo programma è una delle migliori opzioni disponibili per i progettisti 3D che desiderano acquisire un riconoscimento speciale in questo campo. Grazie al contributo del personale docente, lo studente avrà a disposizione mezzi straordinari per rilanciare la propria carriera nel campo della progettazione di videogiochi tridimensionali.



“

Avrai a disposizione un personale docente che non solo chiarirà i tuoi dubbi, ma ti coinvolgerà nel compito di raggiungere il tuo più grande successo professionale creando le capigliature più spettacolari dei videogiochi"

Supervisore internazionale invitato

Darren Pulsipher è un **architetto di software** di grande esperienza, un innovatore con un notevole background internazionale nello **sviluppo di software e firmware**. In effetti, possiede competenze altamente sviluppate in **comunicazione, gestione di progetti e affari**, che gli hanno permesso di guidare importanti iniziative a livello globale.

Ha inoltre ricoperto incarichi di alto livello nel corso della sua carriera, come **Architetto Capo delle Soluzioni per il Settore Pubblico** presso Intel, dove ha promosso **attività, processi e tecnologie moderne** per clienti, partner e utenti del settore pubblico. Inoltre, ha fondato **Yoly Inc.**, dove ha anche ricoperto il ruolo di **CEO**, lavorando per sviluppare uno **strumento di aggregazione e diagnosi dei social media** basato sul **Software as a Service (SaaS)**, utilizzando tecnologie **Big Data e Web 2.0**.

Inoltre, ha lavorato in altre società, come **senior engineering director** presso **Dell Technologies**, dove ha diretto la **Business Unit Big Data Cloud**, guidando i team negli **Stati Uniti e in Cina** per la gestione di grandi progetti e la ristrutturazione delle divisioni aziendali per la loro integrazione di successo. Ha anche lavorato come **Chief Information Officer** presso **XanGo**, dove ha gestito progetti come il **supporto Help Desk**, il **supporto alla produzione** e lo **sviluppo di soluzioni**.

Tra le molteplici specializzazioni in cui è esperto, spiccano la tecnologia **Edge to Cloud**, la **sicurezza informatica**, l'**intelligenza artificiale generativa**, lo sviluppo software, la **tecnologia di rete**, lo **sviluppo nativo nel cloud** e l'**ecosistema dei container**. Conoscenze che ha condiviso attraverso il podcast e la **newsletter settimanale "Embracing Digital Transformation"**, che ha prodotto e presentato, aiutando le organizzazioni a navigare con successo nella trasformazione digitale sfruttando le **persone, i processi e la tecnologia**.



Dott. Pulsipher, Darren

- Architetto Capo delle Soluzioni per il Settore Pubblico presso Intel, California, USA
- Presentatore e produttore di "Embracing Digital Transformation", California
- Fondatore e CEO di Yoly Inc., Arkansas
- Senior Engineering Director presso Dell Technologies, Arkansas
- Chief Information Officer presso XanGo, Utah
- Architetto senior in Cadence Design Systems, California
- Senior Manager dei processi di progetto presso Lucent Technologies, California
- Ingegnere del software a Cemax-Icon, California
- Ingegnere del software presso ISG Technologies, Canada
- MBA in gestione della tecnologia presso l'Università di Phoenix
- Laurea in informatica e ingegneria elettrica presso la Brigham Young University

“

*Con TECH, potrete imparare
da alcuni dei migliori
professionisti del mondo”*

03

Competenze

Gli ingegneri del software devono aggiornare costantemente le loro conoscenze, poiché gli strumenti e le realtà in cui lavorano si modernizzano ogni giorno. Ciò richiede un processo di apprendimento regolare per il quale sono necessarie diverse competenze generali, sia in termini di puro sviluppo del software sia per altre discipline come la gestione del team. Comprendendo questa circostanza, TECH ha sviluppato il Master Specialistico in Ingegneria dei Software con l'obiettivo di fornire agli studenti tutte le competenze possibili e necessarie per rendere più leggero e automatico il proprio processo di apprendimento continuo.



```
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
if _operation == "MIRROR_Z":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = False  
    mirror_mod.use_z = True
```

```
#selection at the end -add back the deselected  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active = modifier_ob  
print("Selected" + str(modifier_ob.name))  
#mirror_ob.select = 0  
#context.scene.objects.active = mirror_ob
```

“

Grazie alle competenze acquisite con questo Master Specialistico in Ingegneria dei Software, sarai il miglior candidato per qualsiasi posizione lavorativa”



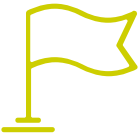
Competenze generali

- ♦ Sviluppare un sistema software tenendo conto di tutte le fasi di sviluppo, delle piattaforme di sicurezza e dei problemi di sicurezza
- ♦ Gestire in modo corretto e professionale tutti i dati generati durante lo sviluppo
- ♦ Applicare la migliore metodologia di lavoro in base al progetto o alle persone coinvolte
- ♦ Conoscere la realtà completa dell'Ingegneria del Software e prevenire eventuali rischi o problemi in modo rapido ed efficiente

“

Potrai compiere un passo in avanti verso un futuro lavorativo migliore. Iscriviti subito a questo Master Specialistico che ti darà accesso a molte opportunità professionali”





Competenze specifiche

- ◆ Comprendere i diversi tipi di modellazione delle applicazioni e i modelli di progettazione nel linguaggio di modellazione unificato (UML)
- ◆ Comprendere il funzionamento della gestione della qualità nei progetti, compresi la pianificazione, la garanzia, il controllo, i concetti statistici e gli strumenti disponibili
- ◆ Acquisire le conoscenze necessarie per lo sviluppo di applicazioni e interfacce grafiche nei linguaggi Java e .NET
- ◆ Comprendere le procedure e le tecniche per migliorare l'aspetto di un documento scritto in HTML
- ◆ Padroneggiare il processo di interazione con il cliente, utilizzando: moduli, cookie e gestione di sessioni
- ◆ Comprendere i meccanismi e i metodi di autenticazione per il controllo degli accessi, nonché il loro processo di verifica
- ◆ Comprendere l'applicazione della sicurezza nelle diverse fasi del ciclo di vita del software
- ◆ Comprendere il concetto, il funzionamento, l'architettura, le risorse e i contenuti di un server web
- ◆ Comprendere i diversi strumenti di supporto, le metodologie e le successive analisi durante i controlli di sicurezza su internet e sui dispositivi mobili
- ◆ Comprendere le politiche e gli standard di sicurezza da attuare per le applicazioni online
- ◆ Saper scrivere, pianificare, sviluppare e firmare progetti nel campo dell'ingegneria informatica finalizzati allo sviluppo o al funzionamento di sistemi, servizi e applicazioni informatiche
- ◆ Gestire le attività dei progetti informatici
- ◆ Saper definire, valutare e selezionare piattaforme hardware e software per lo sviluppo e l'implementazione di sistemi, servizi e applicazioni informatiche
- ◆ Saper sviluppare utilizzando le tecniche di *Scrum*, programmazione estrema e sviluppo software basato sul riuso
- ◆ Avere la capacità di progettare, sviluppare e mantenere sistemi, servizi e applicazioni informatiche utilizzando i metodi dell'ingegneria del software come strumento di garanzia di qualità
- ◆ Utilizzare i fondamenti della crittografia simmetrica e asimmetrica e i loro principali algoritmi
- ◆ Applicare i concetti essenziali relativi ai sistemi informativi in ambito aziendale e identificarne le opportunità e le esigenze
- ◆ Sapere come sviluppare il cronoprogramma per la gestione dei tempi, del budget e della risposta ai rischi
- ◆ Comprendere il funzionamento della governance e della gestione delle TIC, gli standard ISO/IEC che la regolano e le pratiche corrette da implementare
- ◆ Pianificare la gestione della sicurezza e gestire i principali meccanismi di protezione del patrimonio informativo

04

Struttura e contenuti

Il materiale didattico di questo Master Specialistico in Ingegneria dei Software è progettato per coprire tutti gli insegnamenti necessari e complementari alla materia, con le metodologie, gli strumenti e le conoscenze più aggiornate disponibili sul mercato. Un programma ampio ed esaustivo in cui vengono insegnati l'uso di linguaggi di programmazione e ambienti avanzati, l'amministrazione di server web e la loro integrazione nel processo di sviluppo o nella gestione vera e propria di un progetto. Questo garantisce allo studente la migliore opportunità di specializzarsi in Ingegneria dei Software e di eccellere rapidamente in questo campo.





“

Dimostra che le tue conoscenze sono affiancate dall'attitudine di essere il miglior professionista, e aggiungi grande valore al tuo CV grazie a questo Master Specialistico in Ingegneria dei Software”

Modulo 1. Metodologie, sviluppo e qualità nell'Ingegneria dei Software

- 1.1. Introduzione all'Ingegneria dei Software
 - 1.1.1. Introduzione
 - 1.1.2. La crisi del software
 - 1.1.3. Differenze tra ingegneria del software e informatica
 - 1.1.4. Etica e responsabilità professionale nell'ingegneria del software
 - 1.1.5. Industrie di software
- 1.2. Il processo di sviluppo del software
 - 1.2.1. Definizione
 - 1.2.2. Modello di processo del software
 - 1.2.3. Il processo unificato di sviluppo del software
- 1.3. Sviluppo di software orientato agli oggetti
 - 1.3.1. Introduzione
 - 1.3.2. Principi dell'orientamento agli oggetti
 - 1.3.3. Definizione di oggetto
 - 1.3.4. Definizione di classe
 - 1.3.5. Analisi orientata agli oggetti vs Disegno orientata agli oggetti
- 1.4. Sviluppo di software basato sui modelli
 - 1.4.1. La necessità di modellazione
 - 1.4.2. Modellazione di sistemi software
 - 1.4.3. Modellazione di oggetti
 - 1.4.4. UML
 - 1.4.5. Strumenti CASE
- 1.5. Modellazione di applicazioni e design pattern con UML
 - 1.5.1. Modellazione avanzata dei requisiti
 - 1.5.2. Modellazione statica avanzata
 - 1.5.3. Modellazione dinamica avanzata
 - 1.5.4. Modellazione dei componenti
 - 1.5.5. Introduzione ai modelli di progettazione con UML
 - 1.5.6. *Adapter*
 - 1.5.7. *Factory*
 - 1.5.8. *Singleton*
 - 1.5.9. *Strategy*
 - 1.5.10. *Composite*
 - 1.5.11. *Facade*
 - 1.5.12. *Observer*
- 1.6. Ingegneria guidata dai modelli
 - 1.6.1. Introduzione
 - 1.6.2. Metamodellazione dei sistemi
 - 1.6.3. MDA
 - 1.6.4. DSL
 - 1.6.5. Perfezionamento del modello con OCL
 - 1.6.6. Trasformazioni del modello
- 1.7. Ontologie nell'Ingegneria dei Software
 - 1.7.1. Introduzione
 - 1.7.2. Ingegneria ontologica
 - 1.7.3. Applicazione delle ontologie nell'Ingegneria dei Software
- 1.8. Metodologie agili per lo sviluppo del software, *Scrum*
 - 1.8.1. Cos'è l'agilità del software?
 - 1.8.2. Il manifesto agile
 - 1.8.3. La roadmap del progetto agile
 - 1.8.4. Il *Product Owner*
 - 1.8.5. Storie degli utenti
 - 1.8.6. Pianificazione e stima agile
 - 1.8.7. Misurazioni nello sviluppo agile
 - 1.8.8. Introduzione a Scrum
 - 1.8.9. I ruoli
 - 1.8.10. Il *Product Backlog*
 - 1.8.11. Lo *Sprint*
 - 1.8.12. Le riunioni
- 1.9. La metodologia di sviluppo di software *Lean*
 - 1.9.1. Introduzione
 - 1.9.2. *Kanban*

- 1.10. Qualità e miglioramento del processo software
 - 1.10.1. Introduzione
 - 1.10.2. Misurazione del software
 - 1.10.3. Test del software
 - 1.10.4. Modello di qualità di processo del software: CMMI

Modulo 2. Gestione dei progetti di software

- 2.1. Concetti fondamentali della gestione dei progetti e del loro ciclo di vita
 - 2.1.1. Cos'è un progetto?
 - 2.1.2. Metodologia comune
 - 2.1.3. Che cos'è la gestione/leadership di un progetto?
 - 2.1.4. Che cos'è un piano di progetto?
 - 2.1.5. Benefici
 - 2.1.6. Cicli di vita del progetto
 - 2.1.7. Gruppi di processo o ciclo di vita della gestione del progetto
 - 2.1.8. Rapporto tra gruppi di processi e aree di conoscenza
 - 2.1.9. Rapporto tra il ciclo di vita del prodotto e quello del progetto
- 2.2. Avvio e pianificazione
 - 2.2.1. Dall'idea al progetto
 - 2.2.2. Sviluppo del Project Charter
 - 2.2.3. Riunione di avvio del progetto
 - 2.2.4. Attività, conoscenze e competenze nel processo di avvio
 - 2.2.5. Il piano di progetto
 - 2.2.6. Sviluppo del piano base: Passaggi
 - 2.2.7. Attività, conoscenze e competenze nel processo di pianificazione
- 2.3. La gestione degli *stakeholders* e del percorso
 - 2.3.1. Identificare le parti interessate
 - 2.3.2. Sviluppare un piano di gestione degli stakeholder
 - 2.3.3. Gestire il coinvolgimento delle parti interessate
 - 2.3.4. Controllare l'impegno delle parti interessate
 - 2.3.5. L'obiettivo del progetto
 - 2.3.6. La gestione della percorso e il suo piano
 - 2.3.7. Raccogliere i requisiti
 - 2.3.8. Definire l'ambito di applicazione
 - 2.3.9. Creare la WBS
 - 2.3.10. Verificare e controllare il percorso
- 2.4. Lo sviluppo del cronoprogramma
 - 2.4.1. La gestione del tempo e il suo piano
 - 2.4.2. Definire le attività
 - 2.4.3. Stabilire la sequenza delle attività
 - 2.4.4. Stimare le risorse delle attività
 - 2.4.5. Stimare la durata delle attività
 - 2.4.6. Sviluppare il cronoprogramma e calcolare il percorso critico
 - 2.4.7. Controllare il cronoprogramma
- 2.5. Sviluppo del budget e risposta ai rischi
 - 2.5.1. Stimare i costi
 - 2.5.2. Sviluppare il budget e la curva a S
 - 2.5.3. Controllo dei costi e metodo earned value
 - 2.5.4. Concetti di rischio
 - 2.5.5. Come sviluppare un'analisi dei rischi?
 - 2.5.6. Lo sviluppo del piano di risposta
- 2.6. Gestione della qualità
 - 2.6.1. Pianificazione della qualità
 - 2.6.2. Garanzia di qualità
 - 2.6.3. Controllo di qualità
 - 2.6.4. Concetti statistici di base
 - 2.6.5. Strumenti di gestione della qualità
- 2.7. Comunicazione e risorse umane
 - 2.7.1. Pianificare la gestione delle comunicazioni
 - 2.7.2. Analisi dei requisiti di comunicazione
 - 2.7.3. Tecnologia delle comunicazioni
 - 2.7.4. Modelli di comunicazione
 - 2.7.5. Metodi di comunicazione
 - 2.7.6. Piano di gestione delle comunicazioni
 - 2.7.7. Gestione delle comunicazioni
 - 2.7.8. Gestione delle risorse umane
 - 2.7.9. Protagonisti e ruoli nei progetti
 - 2.7.10. Tipi di organizzazione
 - 2.7.11. Organizzazione del progetto
 - 2.7.12. Squadre di lavoro

- 2.8. Approvvigionamento
 - 2.8.1. Il processo di approvvigionamento
 - 2.8.2. Pianificazione
 - 2.8.3. Ricerca di fornitori e richiesta di offerte
 - 2.8.4. Concessione del contratto
 - 2.8.5. Amministrazione del contratto
 - 2.8.6. I contratti
 - 2.8.7. Tipi di contratto
 - 2.8.8. Negoziazione del contratto
- 2.9. Attuazione, monitoraggio, controllo e chiusura
 - 2.9.1. I gruppi dei processi
 - 2.9.2. Esecuzione del progetto
 - 2.9.3. Monitoraggio e controllo del progetto
 - 2.9.4. Chiusura del progetto
- 2.10. Responsabilità professionale
 - 2.10.1. Responsabilità professionale
 - 2.10.2. Caratteristiche della responsabilità sociale e professionale
 - 2.10.3. Codice etico del Project Manager
 - 2.10.4. Responsabilità vs PMP®
 - 2.10.5. Esempi di responsabilità
 - 2.10.6. Benefici della professionalizzazione

Modulo 3. Piattaforme di sviluppo del software

- 3.1. Introduzione allo sviluppo di applicazioni
 - 3.1.1. Applicazioni per desktop
 - 3.1.2. Linguaggio di programmazione
 - 3.1.3. Ambienti di sviluppo integrato
 - 3.1.4. Applicazioni web
 - 3.1.5. Applicazioni mobili
 - 3.1.6. Applicazioni nel cloud
- 3.2. Sviluppo di applicazioni e interfaccia grafica in Java
 - 3.2.1. Ambienti di sviluppo integrati per Java
 - 3.2.2. Principali IDE per Java
 - 3.2.3. Introduzione alla piattaforma di sviluppo Eclipse



- 3.2.4. Introduzione alla piattaforma di sviluppo *NetBeans*
- 3.2.5. Controller View Model per le interfacce utente grafiche
- 3.2.6. Progettare un'interfaccia grafica in *Eclipse*
- 3.2.7. Progettare un'interfaccia grafica in *NetBeans*
- 3.3. Debug e test in *Java*
 - 3.3.1. Test e debug di programmi in *Java*
 - 3.3.2. Debug in *Eclipse*
 - 3.3.3. Debug in *NetBeans*
- 3.4. Sviluppo di applicazioni e interfaccia grafica in .NET
 - 3.4.1. *Net Framework*
 - 3.4.2. Componenti della piattaforma di sviluppo .NET
 - 3.4.3. Visual Studio .NET
 - 3.4.4. Strumenti GUI .NET
 - 3.4.5. La GUI con *Windows Presentation Foundation*
 - 3.4.6. Debug e compilazione di un'applicazione WPF
- 3.5. Programmazione per reti .NET
 - 3.5.1. Introduzione alla programmazione di rete .NET
 - 3.5.2. Richieste e risposte .NET
 - 3.5.3. Utilizzo dei protocolli applicativi .NET
 - 3.5.4. Sicurezza nella programmazione di reti .NET
- 3.6. Ambienti di sviluppo di applicazioni mobili
 - 3.6.1. Applicazioni mobili
 - 3.6.2. Applicazioni mobili *Android*
 - 3.6.3. Passaggi per lo sviluppo in *Android*
 - 3.6.4. IDE *Android Studio*
- 3.7. Sviluppo di applicazioni nell'ambiente *Android Studio*
 - 3.7.1. Installare e avviare *Android Studio*
 - 3.7.2. Esecuzione di un'applicazione *Android*
 - 3.7.3. Sviluppo dell'interfaccia grafica in *Android Studio*
 - 3.7.4. Avvio di attività in *Android Studio*
- 3.8. Debug e pubblicazione di applicazioni *Android*
 - 3.8.1. Debug di un'applicazione in *Android Studio*
 - 3.8.2. Memorizzazione di applicazioni in *Android Studio*
 - 3.8.3. Pubblicità di un'applicazione in *Google Play*

- 3.9. Sviluppo di applicazioni per il cloud
 - 3.9.1. *Cloud computing*
 - 3.9.2. Livelli di *cloud*: SaaS, PaaS, IaaS
 - 3.9.3. Principali piattaforme di sviluppo nel cloud
 - 3.9.4. Riferimenti bibliografici
- 3.10. Introduzione a Google *Cloud Platform*
 - 3.10.1. Concetti di base di Google *Cloud Platform*
 - 3.10.2. Servizi di Google *Cloud Platform*
 - 3.10.3. Strumenti di Google *Cloud Platform*

Modulo 4. Informatica client web

- 4.1. Introduzione all'HTML
 - 4.1.1. Struttura di un documento
 - 4.1.2. Colore
 - 4.1.3. Testo
 - 4.1.4. Collegamenti ipertestuali
 - 4.1.5. Immagini
 - 4.1.6. Liste
 - 4.1.7. Tabelle
 - 4.1.8. Cornici (*Frames*)
 - 4.1.9. Formulare
 - 4.1.10. Elementi specifici per le tecnologie mobili
 - 4.1.11. Elementi in disuso
- 4.2. Fogli di stile web (CSS)
 - 4.2.1. Elementi e struttura di un foglio di stile
 - 4.2.1.1. Creazione di fogli di stile
 - 4.2.1.2. Applicazione degli stili: Selezionatori
 - 4.2.1.3. Ereditarietà di stili e applicazioni a cascata
 - 4.2.1.4. Formattazione della pagina con gli stili
 - 4.2.1.5. Struttura della pagina con gli stili: Il modello black box
 - 4.2.2. Progettazione di stili per diversi dispositivi
 - 4.2.3. Tipi di fogli di stile: statici e dinamici. Pseudo-classi
 - 4.2.4. Pratiche corrette nell'uso dei fogli di stile

- 4.3. Introduzione e storia di *JavaScript*
 - 4.3.1. Introduzione
 - 4.3.2. Storia di *JavaScript*
 - 4.3.3. Ambiente di sviluppo da utilizzare
- 4.4. Nozioni di base di programmazione web
 - 4.4.1. Sintassi di base di *JavaScript*
 - 4.4.2. Tipi di dati primitivi e operatori
 - 4.4.3. Variabili e ambiti
 - 4.4.4. Stringhe di testo e *Template Literals*
 - 4.4.5. Numeri e booleani
 - 4.4.6. Confronti
- 4.5. Strutture complesse in *JavaScript*
 - 4.5.1. Vettori o *Arrays* e oggetti
 - 4.5.2. Insiemi
 - 4.5.3. Mappe
 - 4.5.4. Disgiuntive
 - 4.5.5. Loop
- 4.6. Funzioni e oggetti
 - 4.6.1. Definizione e invocazione delle funzioni
 - 4.6.2. Argomenti
 - 4.6.3. Funzioni a freccia
 - 4.6.4. Funzioni di richiamo o *callback*
 - 4.6.5. Funzioni di ordine superiore
 - 4.6.6. Oggetti letterali
 - 4.6.7. L'oggetto *This*
 - 4.6.8. Oggetti come spazi dei nomi: l'oggetto *Math* e l'oggetto *Date*
- 4.7. Il modello a oggetti del documento (DOM)
 - 4.7.1. Cos'è il DOM?
 - 4.7.2. Un po' di storia
 - 4.7.3. Navigazione e recupero degli elementi
 - 4.7.4. Un DOM virtuale con *JSDOM*
 - 4.7.5. Selezionatori di interrogazione o *Query Selectors*
 - 4.7.6. Navigazione attraverso le proprietà
 - 4.7.7. Assegnazione degli attributi agli elementi
 - 4.7.8. Creazione e modifica dei nodi
 - 4.7.9. Aggiornamento dello stile degli elementi DOM
- 4.8. Sviluppo web moderno
 - 4.8.1. Flusso guidato dagli eventi e *Listeners*
 - 4.8.2. *Toolkits* web moderni e sistemi di allineamento
 - 4.8.3. Modo rigoroso di *JavaScript*
 - 4.8.4. Ulteriori informazioni sulle funzioni
 - 4.8.5. Promesse e funzioni asincrone
 - 4.8.6. *Closures*
 - 4.8.7. Programmazione funzionale
 - 4.8.8. OOP in *JavaScript*
- 4.9. Usabilità del web
 - 4.9.1. Introduzione all'usabilità
 - 4.9.2. Definizione di usabilità
 - 4.9.3. Importanza del web design incentrato sull'utente
 - 4.9.4. Differenze tra accessibilità e usabilità
 - 4.9.5. Vantaggi e problemi nel combinare accessibilità e usabilità
 - 4.9.6. Vantaggi e difficoltà nell'implementazione di siti web utilizzabili
 - 4.9.7. Metodi di usabilità
 - 4.9.8. Analisi dei requisiti dell'utente
 - 4.9.9. Principi del progettazione concettuale: Prototipazione orientata all'utente
 - 4.9.10. Linee guida per la creazione di siti web usabili
 - 4.9.10.1. Linee guida di usabilità di *Jakob Nielsen*
 - 4.9.10.2. Linee guida di usabilità di Bruce Tognazzini
 - 4.9.11. Valutazione dell'usabilità
- 4.10. Accessibilità web
 - 4.10.1. Introduzione
 - 4.10.2. Definizione di accessibilità del web
 - 4.10.3. Tipi di disabilità
 - 4.10.3.1. Disabilità temporanea o permanente
 - 4.10.3.2. Disabilità visiva
 - 4.10.3.3. Disabilità uditiva
 - 4.10.3.4. Disabilità motoria
 - 4.10.3.5. Disabilità neurologica o cognitiva
 - 4.10.3.6. Difficoltà legate all'invecchiamento
 - 4.10.3.7. Limiti relazionati con l'ambiente circostante
 - 4.10.3.8. Ostacoli all'accesso al web

- 4.10.4. Ausili tecnici e prodotti di assistenza per superare gli ostacoli
 - 4.10.4.1. Ausili per non vedenti
 - 4.10.4.2. Ausili per ipovedenti
 - 4.10.4.3. Ausili per persone affette da daltonismo
 - 4.10.4.4. Ausili per non udenti
 - 4.10.4.5. Ausili per persone con disabilità motorie
 - 4.10.4.6. Ausili per persone con disabilità cognitive e neurologiche
- 4.10.5. Vantaggi e difficoltà nell'implementazione dell'accessibilità del web
- 4.10.6. Norme e standard di accessibilità del web
- 4.10.7. Organismi di regolamentazione dell'accessibilità del web
- 4.10.8. Confronto tra norme e standard
- 4.10.9. Linee guida per la conformità normativa e standard
 - 4.10.9.1. Descrizione delle linee guida principali (immagini, link, video, ecc.)
 - 4.10.9.2. Linee guida per una navigazione accessibile
 - 4.10.9.2.1. Percezione
 - 4.10.9.2.2. Operatività
 - 4.10.9.2.3. Comprensibilità
 - 4.10.9.2.4. Robustezza
- 4.10.10. Descrizione del processo di conformità all'accessibilità del web
- 4.10.11. Livelli di conformità
- 4.10.12. Criteri di conformità
- 4.10.13. Requisiti di conformità
- 4.10.14. Metodologia per la valutazione dell'accessibilità dei siti web

Modulo 5. Informatica client web

- 5.1. Introduzione alla programmazione sul server: PHP
 - 5.1.1. Concetti base di programmazione sul server
 - 5.1.2. Sintassi PHP di base
 - 5.1.3. Creazione di contenuti HTML con PHP
 - 5.1.4. Ambienti di sviluppo e test: XAMPP
- 5.2. PHP avanzato
 - 5.2.1. Strutture di controllo PHP
 - 5.2.2. Funzioni in PHP
 - 5.2.3. Gestione di *Arrays* in PHP
 - 5.2.4. Gestione delle stringhe con PHP
 - 5.2.5. Orientamento agli oggetti in PHP
- 5.3. Modelli di dati
 - 5.3.1. Concetto di dato: Ciclo di vita dei dati
 - 5.3.2. Tipi di dati
 - 5.3.2.1. Base
 - 5.3.2.2. Registri
 - 5.3.2.3. Dinamici
- 5.4. Modello relazionale
 - 5.4.1. Descrizione
 - 5.4.2. Entità e tipi di entità
 - 5.4.3. Elementi dei dati: Attributi
 - 5.4.4. Relazioni: tipi, sottotipi, cardinalità
 - 5.4.5. Chiavi: Tipi di chiavi
 - 5.4.6. Normalizzazione: Forme normali
- 5.5. Costruzione del modello logico dei dati
 - 5.5.1. Specifiche della tabella
 - 5.5.2. Definizione delle colonne
 - 5.5.3. Specifiche chiave
 - 5.5.4. Conversione in forme normali: Dipendenze
- 5.6. Il modello fisico dei dati: File di dati
 - 5.6.1. Descrizione dei file di dati
 - 5.6.2. Tipi di file
 - 5.6.3. Modalità di accesso
 - 5.6.4. Organizzazione dei file
- 5.7. Accesso ai database da PHP
 - 5.7.1. Introduzione a *MariaDB*
 - 5.7.2. Lavorare con un database *MariaDB*: il linguaggio SQL
 - 5.7.3. Accedere ai database *MariaDB* da PHP
 - 5.7.4. Introduzione a *MySQL*
 - 5.7.5. Lavorare con un database *MariaDB*: il linguaggio SQL
 - 5.7.6. Accedere ai database *MariaDB* da PHP
- 5.8. Interazione con il cliente da PHP
 - 5.8.1. Formulario PHP
 - 5.8.2. *Cookie*
 - 5.8.3. Gestione di sessioni

- 5.9. Architettura delle applicazioni web
 - 5.9.1. Lo schema Modello-Vista-Controller
 - 5.9.2. Controller
 - 5.9.3. Modello
 - 5.9.4. Vista
- 5.10. Introduzione ai servizi web
 - 5.10.1. Introduzione all'XML
 - 5.10.2. Architetture orientate ai servizi (SOA): servizi web
 - 5.10.3. Creare servizi web SOAP e REST
 - 5.10.4. Il protocollo SOAP
 - 5.10.5. Il protocollo REST

Modulo 6. Gestione della sicurezza

- 6.1. Sicurezza delle informazioni
 - 6.1.1. Introduzione
 - 6.1.2. La sicurezza delle informazioni implica riservatezza, integrità e disponibilità
 - 6.1.3. La sicurezza è una questione economica
 - 6.1.4. La sicurezza è un processo
 - 6.1.5. La classificazione delle informazioni
 - 6.1.6. La sicurezza delle informazioni comporta la gestione del rischio
 - 6.1.7. La sicurezza è legata ai controlli di sicurezza
 - 6.1.8. La sicurezza è sia fisica che logica
 - 6.1.9. La sicurezza coinvolge le persone
- 6.2. Il professionista della sicurezza dell'informazione
 - 6.2.1. Introduzione
 - 6.2.2. La sicurezza informatica come professione
 - 6.2.3. Certificazioni (ISC)2
 - 6.2.4. Standard ISO 27001
 - 6.2.5. Pratiche di sicurezza nella gestione dei servizi informatici
 - 6.2.6. Modelli di maturità per la sicurezza delle informazioni
 - 6.2.7. Altre certificazioni, standard e risorse professionali

- 6.3. Controllo di accessi
 - 6.3.1. Introduzione
 - 6.3.2. Requisiti del controllo di accessi
 - 6.3.3. Meccanismi di autenticazione
 - 6.3.4. Metodi di autorizzazione
 - 6.3.5. Contabilità e revisione degli accessi
 - 6.3.6. Tecnologie "tripla A"
- 6.4. Programmi, processi e politiche di sicurezza delle informazioni
 - 6.4.1. Introduzione
 - 6.4.2. Programmi di gestione della sicurezza
 - 6.4.3. Gestione dei rischi
 - 6.4.4. Progettazione della politica di sicurezza
- 6.5. Piani di continuità aziendale
 - 6.5.1. Introduzione ai BCP
 - 6.5.2. Fase I e II
 - 6.5.3. Fase III e IV
 - 6.5.4. Manutenzione del BCP
- 6.6. Procedure per la corretta tutela dell'azienda
 - 6.6.1. Reti DMZ
 - 6.6.2. Sistemi di rilevamento delle intrusioni
 - 6.6.3. Liste di controllo di accessi
 - 6.6.4. Imparare dall'attaccante: *Honeypot*
- 6.7. Architettura di sicurezza: Prevenzione
 - 6.7.1. Visione generale: Attività e modello a strati
 - 6.7.2. Difesa perimetrale (*Firewalls*, *WAFs*, *IPS*, ecc.)
 - 6.7.3. Difesa degli endpoint (strumenti, server e servizi)
- 6.8. Architettura di sicurezza: Screening
 - 6.8.1. Panoramica di rilevamento e monitoraggio
 - 6.8.2. Registri, interruzione del traffico crittografato, registrazione e *Siem*
 - 6.8.3. Avvisi e informazioni
- 6.9. Architettura di sicurezza: Reazione
 - 6.9.1. Reazione: Prodotti, servizi e risorse
 - 6.9.2. Gestione degli incidenti
 - 6.9.3. CERT e CSIRT

- 6.10. Architettura di sicurezza: Recupero
 - 6.10.1. Resilienza, concetti, requisiti aziendali e standard
 - 6.10.2. Soluzioni IT di resilienza
 - 6.10.3. Gestione e governance delle crisi

Modulo 7. Sicurezza nei sistemi informativi

- 7.1. Panoramica sulla sicurezza, la crittografia e le crittoanalisi classiche
 - 7.1.1. Sicurezza informatica: prospettiva storica
 - 7.1.2. Ma cos'è esattamente la sicurezza?
 - 7.1.3. Storia della crittografia
 - 7.1.4. Cifrari sostitutivi
 - 7.1.5. Caso di studio: la macchina Enigma
- 7.2. Crittografia simmetrica
 - 7.2.1. Introduzione e terminologia base
 - 7.2.2. Crittografia simmetrica
 - 7.2.3. Modalità di funzionamento
 - 7.2.4. DES
 - 7.2.5. Il nuovo standard AES
 - 7.2.6. Crittografia del flusso
 - 7.2.7. Crittoanalisi
- 7.3. Crittografia asimmetrica
 - 7.3.1. Origini della crittografia a chiave pubblica
 - 7.3.2. Concetti di base e funzionamento
 - 7.3.3. L'algoritmo RSA
 - 7.3.4. Certificati digitali
 - 7.3.5. Conservazione e gestione delle chiavi
- 7.4. Attacchi in rete
 - 7.4.1. Minacce e attacchi alla rete
 - 7.4.2. Enumerazione
 - 7.4.3. Intercettazione del traffico: *Sniffers*
 - 7.4.4. Attacchi di negazione del servizio
 - 7.4.5. Attacchi ARP poisoning
- 7.5. Architetture di sicurezza
 - 7.5.1. Architetture di sicurezza tradizionali
 - 7.5.2. *Secure Socket Layer*: SSL
 - 7.5.3. Protocollo SSH
 - 7.5.4. Reti Private Virtuali (VPN)
 - 7.5.5. Meccanismi di protezione dell'unità di archiviazione esterna
 - 7.5.6. Meccanismi di protezione hardware
- 7.6. Tecniche di protezione del sistema e sviluppo sicuro del codice
 - 7.6.1. Sicurezza operativa
 - 7.6.2. Risorse e controlli
 - 7.6.3. Monitoraggio
 - 7.6.4. Sistemi di rilevamento delle intrusioni
 - 7.6.5. IDS di *Host*
 - 7.6.6. IDS di rete
 - 7.6.7. IDS basati sulla firma
 - 7.6.8. Sistemi di esche
 - 7.6.9. Principi di sicurezza di base nello sviluppo del codice
 - 7.6.10. Gestione dei guasti
 - 7.6.11. Nemico pubblico numero 1: buffer overflow
 - 7.6.12. Botch crittografici
- 7.7. *Botnet e Spam*
 - 7.7.1. Origine del problema
 - 7.7.2. Processo di spam
 - 7.7.3. Invio di spam
 - 7.7.4. Affinamento della mailing list
 - 7.7.5. Tecniche di protezione
 - 7.7.6. Servizio antispam offerto da terzi
 - 7.7.7. Casi di studio
 - 7.7.8. Spam esotico
- 7.8. Controllo e attacchi web
 - 7.8.1. Raccolta di informazioni
 - 7.8.2. Tecniche di attacco
 - 7.8.3. Strumenti

- 7.9. Malware e codice maligno
 - 7.9.1. Che cos'è il malware?
 - 7.9.2. Tipi di malware
 - 7.9.3. Virus
 - 7.9.4. Criptovirus
 - 7.9.5. Worm
 - 7.9.6. Adware
 - 7.9.7. Spyware
 - 7.9.8. Hoaxes
 - 7.9.9. Phishing
 - 7.9.10. Trojan
 - 7.9.11. L'economia del malware
 - 7.9.12. Possibili soluzioni
- 7.10. Analisi forense
 - 7.10.1. Raccolta di evidenze
 - 7.10.2. Analisi delle evidenze
 - 7.10.3. Tecniche anti-forensi
 - 7.10.4. Caso di studio pratico

Modulo 8. Sicurezza del software

- 8.1. Cos'è la sicurezza del software
 - 8.1.1. Introduzione al problema della sicurezza del software
 - 8.1.2. Vulnerabilità e la loro classificazione
 - 8.1.3. Proprietà del software sicuro
 - 8.1.4. Riferimenti
- 8.2. Principi di progettazione della sicurezza del software
 - 8.2.1. Introduzione
 - 8.2.2. Principi di progettazione della sicurezza del software
 - 8.2.3. Tipologie di S-SDLC
 - 8.2.4. Sicurezza del software nelle fasi S-SDLC
 - 8.2.5. Metodologie e standard
 - 8.2.6. Riferimenti

- 8.3. Sicurezza del ciclo di vita del software nelle fasi di progettazione e requisiti
 - 8.3.1. Introduzione
 - 8.3.2. Modellazione dell'attacco
 - 8.3.3. Casi di abuso
 - 8.3.4. Ingegneria dei requisiti di sicurezza
 - 8.3.5. Analisi dei rischi: Architettoneco
 - 8.3.6. Modelli di disegno
 - 8.3.7. Riferimenti
- 8.4. Sicurezza del ciclo di vita del software nelle fasi di codifica, prova e funzionamento
 - 8.4.1. Introduzione
 - 8.4.2. Test di sicurezza basati sul rischio
 - 8.4.3. Revisione del codice
 - 8.4.4. Test di penetrazione
 - 8.4.5. Operazioni di sicurezza
 - 8.4.6. Revisione esterna
 - 8.4.7. Riferimenti
- 8.5. Applicazioni di codifica sicura I
 - 8.5.1. Introduzione
 - 8.5.2. Pratiche di codifica sicure
 - 8.5.3. Gestione e convalida degli input
 - 8.5.4. Overflow di memoria
 - 8.5.5. Riferimenti
- 8.6. Applicazioni di codifica sicura II
 - 8.6.1. Introduzione
 - 8.6.2. *Integers Overflows*, errori di troncamento e problemi con le conversioni di tipo tra numeri interi
 - 8.6.3. Errori ed eccezioni
 - 8.6.4. Privacy e riservatezza
 - 8.6.5. Programmi privilegiati
 - 8.6.6. Riferimenti
- 8.7. Sicurezza nello sviluppo e nel cloud
 - 8.7.1. Sicurezza dello sviluppo: metodologia e pratica
 - 8.7.2. Modelli PaaS, IaaS, CaaS e SaaS
 - 8.7.3. Sicurezza nel cloud e per i servizi cloud

- 8.8. Automazione e orchestrazione della sicurezza (SOAR)
 - 8.9.1. Complessità dell'elaborazione manuale: necessità di automatizzare le attività
 - 8.9.2. Prodotti e servizi
 - 8.9.3. Architettura SOAR
- 8.9. Sicurezza del telelavoro
 - 8.9.1. Necessità e scenari
 - 8.9.2. Prodotti e servizi
 - 8.9.3. Sicurezza del telelavoro

Modulo 9. Qualità e controllo dei sistemi informativi

- 9.1. Introduzione ai Sistemi di Gestione della Sicurezza delle Informazioni
 - 9.1.1. Principi dei SGSI
 - 9.1.2. Regole d'oro dei SGSI
 - 9.1.3. Ruolo del controllo informatico nei SGSI
- 9.2. Pianificazione nella gestione della sicurezza
 - 9.2.1. Concetti relativi alla gestione della sicurezza
 - 9.2.2. Classificazione delle informazioni: obiettivi, concetti e ruoli
 - 9.2.3. Implementazione delle politiche di sicurezza: politiche, standard e procedure di sicurezza
 - 9.2.4. Gestione del rischio: principi e analisi del rischio degli asset informativi
- 9.3. Principali meccanismi di protezione del patrimonio informativo (I)
 - 9.3.1. Sintesi dei principali strumenti crittografici per la protezione della triade CID
 - 9.3.2. Considerazione della privacy, dell'anonimato e dei requisiti di gestione della tracciabilità degli utenti
- 9.4. Principali meccanismi di protezione del patrimonio informativo (II)
 - 9.4.1. Sicurezza delle comunicazioni: protocolli, dispositivi e architetture di sicurezza
 - 9.4.2. Sicurezza dei sistemi operativi
- 9.5. Controlli interni dei SGSI
 - 9.5.1. Tassonomia dei controlli SGSI: controlli amministrativi, logici e fisici
 - 9.5.2. Classificazione dei controlli in base al modo in cui affrontano la minaccia: controlli per la prevenzione, il rilevamento e la correzione della minaccia
 - 9.5.3. Implementazione dei sistemi di controllo interno nei SGSI

- 9.6. Tipi di revisione
 - 9.6.1. Differenza tra revisione e controllo interno
 - 9.6.2. Revisione interna vs esterna
 - 9.6.3. Classificazione della revisione in base all'obiettivo e al tipo di analisi
- 9.7. Sceneggiatore e sceneggiatura: soggetto e oggetto protetti dalla proprietà intellettuale
 - 9.7.1. Introduzione ai test di penetrazione e all'analisi forense
 - 9.7.2. Definizione e rilevanza dei concetti di *Fingerprinting* e *Footprinting*
- 9.8. Scansione delle vulnerabilità e monitoraggio del traffico di rete
 - 9.8.1. Strumenti di scansione delle vulnerabilità del sistema
 - 9.8.2. Principali vulnerabilità nel contesto delle applicazioni web
 - 9.8.3. Analisi dei protocolli di comunicazione
- 9.9. Il processo di controllo informatico
 - 9.9.1. Concetto di ciclo di vita dello sviluppo dei sistemi
 - 9.9.2. Monitoraggio delle attività e dei processi: raccolta ed elaborazione delle prove
 - 9.9.3. Metodologia di controllo informatico
 - 9.9.4. Processo di controllo informatico
 - 9.9.5. Individuazione dei principali reati e illeciti nel contesto informatico
 - 9.9.6. Indagine sui reati informatici: introduzione all'analisi forense e al suo rapporto con il controllo informatico
- 9.10. Piani di continuità operativa e di disaster recovery
 - 9.10.1. Definizione di piano di continuità aziendale e concetto di interruzione dell'attività
 - 9.10.2. Raccomandazione NIST sui piani di continuità aziendale
 - 9.10.3. Piano di disaster recovery
 - 9.10.4. Processo del piano di disaster recovery

Modulo 10. Amministrazione del server web

- 10.1. Introduzione ai server web
 - 10.1.1. Che cos'è un server web?
 - 10.1.2. Architettura e funzionamento di un server web
 - 10.1.3. Risorse e contenuti in un server web
 - 10.1.4. Server di applicazioni
 - 10.1.5. Server proxy
 - 10.1.6. I principali server web presenti sul mercato
 - 10.1.7. Statistiche di utilizzo del server web
 - 10.1.8. Sicurezza del server web
 - 10.1.9. Bilanciamento del carico nei server web
 - 10.1.10. Riferimenti

- 10.2. Gestione del protocollo HTTP
 - 10.2.1. Funzionamento e struttura
 - 10.2.2. Descrizione delle richieste o *Request Methods*
 - 10.2.3. Codici di stato
 - 10.2.4. Intestazioni
 - 10.2.5. Codifica dei contenuti: Pagine di codice
 - 10.2.6. Esecuzione di richieste HTTP su internet tramite proxy, *Live HTTP Headers* o metodo simile, analizzando il protocollo utilizzato
- 10.3. Descrizione delle architetture distribuite multi-server
 - 10.3.1. Modello a 3 strati
 - 10.3.2. Tolleranza ai guasti
 - 10.3.3. Condivisione del carico
 - 10.3.4. Archivi dello stato della sessione
 - 10.3.5. Archivi di cache
- 10.4. *Internet Information Services (IIS)*
 - 10.4.1. Cos'è IIS?
 - 10.4.2. Storia ed evoluzione dell'IIS
 - 10.4.3. Principali vantaggi e caratteristiche di IIS
 - 10.4.4. Architettura IIS
- 10.5. Installazione, amministrazione e configurazione di IIS
 - 10.5.1. Preambolo
 - 10.5.2. Installazione di *Internet Information Services (IIS)*
 - 10.5.3. Strumenti di amministrazione di IIS
 - 10.5.4. Creazione, configurazione e amministrazione di siti web
 - 10.5.5. Installazione e gestione delle estensioni di IIS
- 10.6. Sicurezza avanzata di IIS
 - 10.6.1. Preambolo
 - 10.6.2. Autenticazione, autorizzazione e controllo degli accessi in IIS
 - 10.6.3. Configurazione di un sito web sicuro su IIS con SSL
 - 10.6.4. Criteri di sicurezza implementati in IIS 10.x
- 10.7. Introduzione ad Apache
 - 10.7.1. Che cos'è Apache?
 - 10.7.2. Principali vantaggi di Apache
 - 10.7.3. Caratteristiche principali di Apache
 - 10.7.4. Architettura

- 10.8. Installazione e configurazione di Apache
 - 10.8.1. Installazione iniziale di Apache
 - 10.8.2. Configurazione di Apache
- 10.9. Installazione e configurazione dei diversi moduli Apache
 - 10.9.1. Installazione dei moduli Apache
 - 10.9.2. Tipi di moduli
 - 10.9.3. Configurazione sicura di Apache
- 10.10. Sicurezza avanzata
 - 10.10.1. Autenticazione, autorizzazione e controllo degli accessi
 - 10.10.2. Metodi di autenticazione
 - 10.10.3. Configurazione sicura di Apache con SSL

Modulo 11. Sicurezza nelle applicazioni online

- 11.1. Vulnerabilità e problemi di sicurezza nelle applicazioni online
 - 11.1.1. Introduzione alla sicurezza nelle applicazioni online
 - 11.1.2. Vulnerabilità di sicurezza nella progettazione di applicazioni web
 - 11.1.3. Vulnerabilità di sicurezza nell'implementazione di applicazioni
 - 11.1.4. Vulnerabilità di sicurezza nella distribuzione di applicazioni web
 - 11.1.5. Elenchi ufficiali di vulnerabilità di sicurezza
- 11.2. Politiche e standard per la sicurezza delle applicazioni online
 - 11.2.1. I pilastri della sicurezza delle applicazioni online
 - 11.2.2. Politica di sicurezza
 - 11.2.3. Sistema di gestione della sicurezza delle informazioni
 - 11.2.4. Ciclo di vita dello sviluppo del software sicuro
 - 11.2.5. Standard di sicurezza delle applicazioni
- 11.3. Sicurezza nella progettazione di applicazioni web
 - 11.3.1. Introduzione alla sicurezza delle applicazioni web
 - 11.3.2. Sicurezza nella progettazione di applicazioni web
- 11.4. Test di sicurezza e protezione online delle applicazioni web
 - 11.4.1. Analisi e test della sicurezza delle applicazioni web
 - 11.4.2. Sicurezza nell'implementazione e nella produzione di applicazioni web
- 11.5. Sicurezza dei servizi web
 - 11.5.1. Introduzione alla sicurezza dei servizi web
 - 11.5.2. Funzioni e tecnologie di sicurezza dei servizi web

- 11.6. Test di sicurezza e protezione online dei servizi web
 - 11.6.1. Valutazione della sicurezza dei servizi web
 - 11.6.2. Protezione online: *Firewalls* e *Gateways XML*
- 11.7. *Hacking* etico, malware e *Forensic*
 - 11.7.1. *Hacking* etico
 - 11.7.2. Analisi del malware
 - 11.7.3. Analisi forense
- 11.8. Risoluzione degli incidenti dei servizi web
 - 11.8.1. Monitoraggio
 - 11.8.2. Strumenti di misurazione delle prestazioni
 - 11.8.3. Misure di contenimento
 - 11.8.4. Analisi delle cause principali
 - 11.8.5. Gestione proattiva dei problemi
- 11.9. Le migliori pratiche per garantire la sicurezza delle applicazioni
 - 11.9.1. Manuale di pratiche corrette per lo sviluppo di applicazioni online
 - 11.9.2. Manuale di pratiche corrette per l'implementazione di applicazioni online
- 11.10. Errori comuni che minano la sicurezza delle applicazioni
 - 11.10.1. Errori di sviluppo comuni
 - 11.10.2. Errori comuni nell'hosting
 - 11.10.3. Errori comuni nella produzione

Modulo 12. Ingegneria dei Software

- 12.1. Introduzione all'ingegneria e alla modellazione di software
 - 12.1.1. La natura del software
 - 12.1.2. La natura unica del *Webapps*
 - 12.1.3. Ingegneria dei software
 - 12.1.4. Il processo del software
 - 12.1.5. La pratica dell'ingegneria del software
 - 12.1.6. Miti del software
 - 12.1.7. Come nasce tutto questo?
 - 12.1.8. Concetti orientati agli oggetti
 - 12.1.9. Introduzione all'UML
- 12.2. Il processo del software
 - 12.2.1. Un modello di processo generale
 - 12.2.2. Modelli di processo prescrittivi
 - 12.2.3. Modelli di processo specializzati
 - 12.2.4. Il processo unificato
 - 12.2.5. Modelli di processo personale e di gruppo
 - 12.2.6. Che cos'è l'agilità?
 - 12.2.7. Che cos'è un processo agile?
 - 12.2.8. *Scrum*
 - 12.2.9. Strumenti per il processo agile
- 12.3. Principi che guidano la pratica dell'ingegneria del software
 - 12.3.1. Principi che guidano il processo
 - 12.3.2. Principi che guidano la pratica
 - 12.3.3. Principi di comunicazione
 - 12.3.4. Principi di pianificazione
 - 12.3.5. Principi di modellazione
 - 12.3.6. Principi di costruzione
 - 12.3.7. Principi di implementazione
- 12.4. Comprensione dei requisiti
 - 12.4.1. Ingegneria dei requisiti
 - 12.4.2. Stabilire le basi
 - 12.4.3. Identificazione dei requisiti
 - 12.4.4. Sviluppo di casi d'uso
 - 12.4.5. Elaborazione del modello dei requisiti
 - 12.4.6. Negoziazione dei requisiti
 - 12.4.7. Convalida dei requisiti
- 12.5. Modellazione dei requisiti: scenari, informazioni e classi di analisi
 - 12.5.1. Analisi dei requisiti
 - 12.5.2. Modellazione basata su scenari
 - 12.5.3. Modelli UML che forniscono il caso d'uso
 - 12.5.4. Concetti di modellazione dei dati
 - 12.5.5. Modellazione basata sulle classi
 - 12.5.6. Diagrammi di classe

- 12.6. Modellazione dei requisiti: flusso, comportamento e modelli
 - 12.6.1. Strategie di modellazione dei requisiti
 - 12.6.2. Modellazione orientata al flusso
 - 12.6.3. Diagrammi di stato
 - 12.6.4. Creazione di un modello comportamentale
 - 12.6.5. Diagrammi di sequenza
 - 12.6.6. Diagrammi di comunicazione
 - 12.6.7. Modelli per la modellazione dei requisiti
- 12.7. Concetti di design
 - 12.7.1. Il design nel contesto dell'ingegneria del software
 - 12.7.2. Processo di design
 - 12.7.3. Concetti di design
 - 12.7.4. Concetti di design orientato agli oggetti
 - 12.7.5. Modello di design
- 12.8. Design dell'architettura
 - 12.8.1. Architettura del software
 - 12.8.2. Generi architettonici
 - 12.8.3. Stili architettonici
 - 12.8.4. Design architettonico
 - 12.8.5. Evoluzione dei progetti alternativi per l'architettura
 - 12.8.6. Mappatura dell'architettura utilizzando il flusso di dati
- 12.9. Progettazione a livello di componenti e basata su modelli
 - 12.9.1. Che cos'è un componente?
 - 12.9.2. Progettazione di componenti basata su classi
 - 12.9.3. Realizzazione della progettazione a livello di componente
 - 12.9.4. Design tradizionale dei componenti
 - 12.9.5. Sviluppo basato su componenti
 - 12.9.6. Modelli di disegno
 - 12.9.7. Progettazione software basata su modelli
 - 12.9.8. Modelli architettonici
 - 12.9.9. Modelli di progettazione a livello di componenti
 - 12.9.10. Modelli di progettazione dell'interfaccia utente

- 12.10. Qualità del software e gestione dei progetti
 - 12.10.1. Qualità
 - 12.10.2. Qualità del software
 - 12.10.3. Il dilemma della qualità del software
 - 12.10.4. Raggiungere la qualità del software
 - 12.10.5. Garanzia di qualità del software
 - 12.10.6. Lo spettro gestionale
 - 12.10.7. Il personale
 - 12.10.8. Il prodotto
 - 12.10.9. Il processo
 - 12.10.10. Il progetto
 - 12.10.11. Principi e pratiche

Modulo 13. Ingegneria dei Software Avanzata

- 13.1. Introduzione alle Metodologie Agili
 - 13.1.1. Modelli di processo e metodologie
 - 13.1.2. Agilità e processi agili
 - 13.1.3. Manifesto Agile
 - 13.1.4. Alcune metodologie agili
 - 13.1.5. Agile vs Tradizionale
- 13.2. Scrum
 - 13.2.1. Origini e filosofia di Scrum
 - 13.2.2. Valori di Scrum
 - 13.2.3. Flusso del processo Scrum
 - 13.2.4. Ruoli di Scrum
 - 13.2.5. Artefatti di Scrum
 - 13.2.6. Eventi di Scrum
 - 13.2.7. Storie degli utenti
 - 13.2.8. Estensioni Scrum
 - 13.2.9. Stime agili
 - 13.2.10. Scalabilità di Scrum

- 13.3. Programmazione estrema
 - 13.3.1. Motivazione e panoramica di XP
 - 13.3.2. Il ciclo di vita in XP
 - 13.3.3. I cinque valori fondamentali
 - 13.3.4. Le dodici pratiche di base di XP
 - 13.3.5. Ruoli dei partecipanti
 - 13.3.6. XP industriale
 - 13.3.7. Valutazione critica di XP
 - 13.4. Sviluppo del software basato sul riutilizzo
 - 13.4.1. Riutilizzo del software
 - 13.4.2. Livelli di riutilizzo del codice
 - 13.4.3. Tecniche di riutilizzo specifiche
 - 13.4.4. Sviluppo basato su componenti
 - 13.4.5. Benefici e problemi del riutilizzo
 - 13.4.6. Pianificazione del riutilizzo
 - 13.5. Architettura di sistema e modelli di progettazione del software
 - 13.5.1. Design architettonico
 - 13.5.2. Modelli architettonici generali
 - 13.5.3. Architetture con tolleranza ai guasti
 - 13.5.4. Architetture di sistemi distribuiti
 - 13.5.5. Modelli di progettazione
 - 13.5.6. Modelli gamma
 - 13.5.7. Modelli di progettazione dell'interazione
 - 13.6. Architettura di applicazioni nel cloud
 - 13.6.1. Fondamenti di *Cloud Computing*
 - 13.6.2. Qualità delle applicazioni nel cloud
 - 13.6.3. Stili di architettura
 - 13.6.4. Modelli di design
 - 13.7. Test del software: TDD, ATDD e BDD
 - 13.7.1. Verifica e convalida del software
 - 13.7.2. Test del software
 - 13.7.3. *Test Driven Development* (TDD)
 - 13.7.4. *Acceptance Test Driven Development* (ATDD)
 - 13.7.5. *Behavior Driven Development* (BDD)
 - 13.7.6. BDD e Cucumber
 - 13.8. Miglioramento dei processi software
 - 13.8.1. Miglioramento dei processi software
 - 13.8.2. Il miglioramento dei processi
 - 13.8.3. Modelli di maturità
 - 13.8.4. Il modello CMMI
 - 13.8.5. CMMI V13.0
 - 13.8.6. CMMI e Agile
 - 13.9. La qualità del prodotto software: *SQuaRE*
 - 13.9.1. La qualità del software
 - 13.9.2. Modello di qualità del prodotto software
 - 13.9.3. Famiglia ISO/IEC 13.000
 - 13.9.4. ISO/IEC 13.010: modello e caratteristiche di qualità
 - 13.9.5. ISO/IEC 13.0113: la qualità dei dati
 - 13.9.6. ISO/IEC 13.013: misurazione della qualità del software
 - 13.9.7. ISO/IEC 13.013, 13.013 e 13.013: metriche di qualità del software e dei dati
 - 13.9.8. ISO/IEC 13.040: valutazione del software
 - 13.9.9. Processo di certificazione
 - 13.10. Introduzione a *DevOps*
 - 13.10.1. Concetto di DevOps
 - 13.10.2. Pratiche principali
- Modulo 14. Ingegneria dei requisiti**
- 14.1. Introduzione all'ingegneria dei requisiti
 - 14.1.1. L'importanza dei requisiti
 - 14.1.2. Concetto di requisito
 - 14.1.3. Dimensioni dei requisiti
 - 14.1.4. Livelli e tipi di requisiti
 - 14.1.5. Caratteristiche dei requisiti
 - 14.1.6. Ingegneria dei requisiti
 - 14.1.7. Il processo di ingegneria dei requisiti
 - 14.1.8. *Frameworks* per ingegneria dei requisiti
 - 14.1.9. Pratica corretta in ingegneria dei requisiti
 - 14.1.10. Analista di business

- 14.2. Fonti dei requisiti
 - 14.2.1. Rete dei requisiti
 - 14.2.2. Gli *Stakeholders*
 - 14.2.3. Requisiti aziendali
 - 14.2.4. Documento di visione e scopo
- 14.3. Tecniche di elicitazione dei requisiti
 - 14.3.1. Elicitazione dei requisiti
 - 14.3.2. Problemi di elicitazione dei requisiti
 - 14.3.3. Contesti di scoperta
 - 14.3.4. Interviste
 - 14.3.5. Osservazione e "apprendimento"
 - 14.3.6. Etnografia
 - 14.3.7. *Workshops*
 - 14.3.8. *Focus groups*
 - 14.3.9. Questionari
 - 14.3.10. *Brainstorming* e tecniche creative
 - 14.3.11. Media del gruppo
 - 14.3.12. Analisi delle interfacce di sistema
 - 14.3.13. Analisi dei documenti e "archeologia"
 - 14.3.14. Casi d'uso e scenari
 - 14.3.15. Prototipi
 - 14.3.16. Ingegneria inversa
 - 14.3.17. Riutilizzo dei requisiti
 - 14.3.18. Pratiche corrette di elicitazione
- 14.4. Requisiti dell'utente
 - 14.4.1. Persone
 - 14.4.2. Casi di uso e storie dell'utente
 - 14.4.3. Scenari
 - 14.4.4. Tipi di scenari
 - 14.4.5. Come scoprire gli scenari?
- 14.5. Tecniche di prototipazione
 - 14.5.1. Prototipazione
 - 14.5.2. Prototipi in base al percorso
 - 14.5.3. Prototipi in base alla temporalità
 - 14.5.4. La fedeltà di un prototipo
 - 14.5.5. Prototipi di interfaccia utente
 - 14.5.6. Valutazione del prototipo
- 14.6. Analisi dei requisiti
 - 14.6.1. Analisi dei requisiti
 - 14.6.2. Pratica corretta di analisi dei requisiti
 - 14.6.3. Il dizionario dei dati
 - 14.6.4. Prioritarizzazione dei requisiti
- 14.7. Documentazione dei requisiti
 - 14.7.1. Il documento di specifica dei requisiti
 - 14.7.2. Struttura e contenuti di un SRS
 - 14.7.3. Documentazione in linguaggio naturale
 - 14.7.4. EARS: *Easy Approach to Requirements Syntax*
 - 14.7.5. Requisiti non funzionali
 - 14.7.6. Attributi e modelli in forma di tabella
 - 14.7.7. Pratiche corrette di specificazione
- 14.8. Convalida e negoziazione dei requisiti
 - 14.8.1. Convalida dei requisiti
 - 14.8.2. Tecniche di convalida dei requisiti
 - 14.8.3. Negoziazione dei requisiti
- 14.9. Modellazione e gestione dei requisiti
 - 14.9.1. Modellazione dei requisiti
 - 14.9.2. La prospettiva dell'utente
 - 14.9.3. La prospettive dei dati
 - 14.9.4. La prospettiva funzionale o orientata al flusso
 - 14.9.5. La prospettiva del comportamento
 - 14.9.6. La volatilità dei requisiti
 - 14.9.7. Processo di gestione dei requisiti
 - 14.9.8. Strumenti di gestione dei requisiti
 - 14.9.9. Pratica corretta per la gestione dei requisiti

- 14.10. Sistemi critici e specifiche formali
 - 14.10.1. Sistemi critici
 - 14.10.2. Specifiche basate sul rischio
 - 14.10.3. Specifica formale

Modulo 15. Processi di ingegneria del software

- 15.1. Struttura dell'ingegneria del software
 - 15.1.1. Caratteristiche del software
 - 15.1.2. Principali processi di ingegneria del software
 - 15.1.3. Modelli di processo di sviluppo software
 - 15.1.4. Quadro di riferimento standard per il processo di sviluppo del software: lo standard ISO/IEC 12207
- 15.2. Processo unificato di sviluppo software
 - 15.2.1. Processo unificato
 - 15.2.2. Dimensioni del processo unificato
 - 15.2.3. Processo di sviluppo guidato dai casi d'uso
 - 15.2.4. Flussi di lavoro unificati fondamentali per i processi
- 15.3. Pianificazione nel contesto dello sviluppo agile del software
 - 15.3.1. Caratteristiche dello sviluppo agile del software
 - 15.3.2. Diversi orizzonti temporali di pianificazione nello sviluppo agile
 - 15.3.3. Quadro di sviluppo Agile Scrum e pianificazione degli orizzonti temporali
 - 15.3.4. Storie dell'utente come unità di pianificazione e stima
 - 15.3.5. Tecniche comuni per ricavare una stima
 - 15.3.6. Scale di interpretazione delle stime
 - 15.3.7. *Planning Poker*
 - 15.3.8. Tipi di pianificazione comuni: pianificazione della consegna e dell'iterazione
- 15.4. Stili di progettazione del software distribuito e architetture software orientate ai servizi
 - 15.4.1. Modelli di comunicazione nei sistemi software distribuiti
 - 15.4.2. Livello intermedio o *Middleware*
 - 15.4.3. Modelli di architettura per sistemi distribuiti
 - 15.4.4. Processo generale di progettazione dei servizi software
 - 15.4.5. Aspetti di progettazione dei servizi software
 - 15.4.6. Composizione dei servizi
 - 15.4.7. Architettura dei servizi web
 - 15.4.8. Componenti infrastrutturali e SOA
- 15.5. Introduzione allo sviluppo di software guidato da modelli
 - 15.5.1. Il concetto di modello
 - 15.5.2. Sviluppo di software guidato da modelli
 - 15.5.3. Quadro di sviluppo guidato dal modello MDA
 - 15.5.4. Elementi di un modello di trasformazione
- 15.6. Progettazione dell'interfaccia grafica
 - 15.6.1. Principi di progettazione dell'interfaccia utente
 - 15.6.2. Modelli di progettazione architettonica per sistemi interattivi: Model-View-Controller (MVC)
 - 15.6.3. Esperienza dell'utente (UX *User Experience*)
 - 15.6.4. Design incentrato sull'utente
 - 15.6.5. Processo di analisi e progettazione dell'interfaccia utente
 - 15.6.6. Usabilità delle interfacce utente
 - 15.6.7. Accessibilità delle interfacce utente
- 15.7. Disegno delle applicazioni web
 - 15.7.1. Caratteristiche delle applicazioni web
 - 15.7.2. Interfaccia utente di un'applicazione web
 - 15.7.3. Design della navigazione
 - 15.7.4. Protocollo di interazione di base per le applicazioni web
 - 15.7.5. Stili di architettura per applicazioni web
- 15.8. Strategie e tecniche di test del software e fattori di qualità dello stesso
 - 15.8.1. Strategie di prova
 - 15.8.2. Progettazione dei casi di prova
 - 15.8.3. Relazione costo-qualità
 - 15.8.4. Modelli di qualità
 - 15.8.5. Famiglia di norme ISO/IEC 25000 (*SQuaRE*)
 - 15.8.6. Modello di qualità del prodotto (ISO 2501n)
 - 15.8.7. Modello di qualità dei dati (ISO 2501n)
 - 15.8.8. Gestione della qualità del software
- 15.9. Introduzione alle metriche nell'ingegneria del software
 - 15.9.1. Concetti di base: misure, metriche e indicatori
 - 15.9.2. Tipi di metriche nell'ingegneria del software
 - 15.9.3. Processo di misurazione
 - 15.9.4. ISO 250215: Metriche esterne e di qualità in uso
 - 15.9.5. Metrica orientata agli oggetti

- 15.10. Manutenzione e reingegnerizzazione del software
 - 15.10.1. Processo di manutenzione
 - 15.10.2. Quadro standard del processo di manutenzione: ISO/EIEC 115.64
 - 15.10.3. Modello di processo di reingegnerizzazione del software
 - 15.10.4. Ingegneria inversa

Modulo 16. Integrazione dei sistemi

- 16.1. Introduzione ai sistemi informativi aziendali
 - 16.1.1. Il ruolo dei sistemi informativi
 - 16.1.2. Che cos'è un sistema informativo?
 - 16.1.3. Dimensioni dei sistemi informativi
 - 16.1.4. Processi aziendali e sistemi informativi
 - 16.1.5. Il dipartimento dei sistemi informativi
- 16.2. Opportunità e necessità dei sistemi informativi nell'impresa
 - 16.2.1. Organizzazioni e sistemi informativi
 - 16.2.2. Caratteristiche delle organizzazioni
 - 16.2.3. Impatto dei sistemi informativi sull'impresa
 - 16.2.4. Sistemi informativi per il vantaggio competitivo
 - 16.2.5. Uso dei sistemi nell'amministrazione e nella gestione aziendale
- 16.3. Concetti di base dei sistemi e delle tecnologie dell'informazione
 - 16.3.1. Dati, informazioni e conoscenze
 - 16.3.2. Tecnologia e sistemi informativi
 - 16.3.3. Componenti tecnologici
 - 16.3.4. Classificazione e tipi di sistemi informativi
 - 16.3.5. Architetture basate su servizi e processi aziendali
 - 16.3.6. Forme di integrazione dei sistemi
- 16.4. Sistemi per la gestione integrata delle risorse aziendali
 - 16.4.1. Requisiti aziendali
 - 16.4.2. Un sistema informativo aziendale integrato
 - 16.4.3. Acquisizione vs Sviluppo
 - 16.4.4. Implementazione dell'ERP
 - 16.4.5. Implicazioni nella gestione
 - 16.4.6. Principali fornitori di ERP
- 16.5. Sistemi informativi per la gestione della supply chain e delle relazioni con i clienti
 - 16.5.1. Definizione di catena di approvvigionamento
 - 16.5.2. Gestione della catena di approvvigionamento
 - 16.5.3. Il ruolo dei sistemi informativi
 - 16.5.4. Soluzioni per la gestione della catena di approvvigionamento
 - 16.5.5. Gestione delle relazioni con i clienti
 - 16.5.6. Il ruolo dei sistemi informativi
 - 16.5.7. Implementazione di un sistema CRM
 - 16.5.8. Fattori critici di successo nell'implementazione del CRM
 - 16.5.9. CRM, e-CRM e altre tendenze
- 16.6. Processo decisionale sugli investimenti nelle TIC e nella pianificazione dei sistemi informativi
 - 16.6.1. Criteri per le decisioni di investimento nelle TIC
 - 16.6.2. Collegare il progetto al piano di gestione e di business
 - 16.6.3. Implicazioni nella gestione
 - 16.6.4. Riprogettazione dei processi aziendali
 - 16.6.5. Decisione della direzione sulle metodologie di implementazione
 - 16.6.6. Necessità di pianificazione dei sistemi informativi
 - 16.6.7. Obiettivi, partecipanti e tempistiche
 - 16.6.8. Struttura e sviluppo del piano di sistema
 - 16.6.9. Monitoraggio e aggiornamento
- 16.7. Considerazioni sulla sicurezza nell'uso delle TIC
 - 16.7.1. Analisi dei rischi
 - 16.7.2. Sicurezza nei sistemi informativi
 - 16.7.3. Consigli pratici
- 16.8. Fattibilità dell'implementazione di progetti TIC e aspetti finanziari dei progetti di sistemi informativi
 - 16.8.1. Descrizione e obiettivi
 - 16.8.2. Partecipanti all'EVS
 - 16.8.3. Tecniche e pratiche
 - 16.8.4. Struttura dei costi
 - 16.8.5. Proiezione finanziaria
 - 16.8.6. Preventivi

- 16.9. Business Intelligence
 - 16.9.1. Cos'è la Business Intelligence?
 - 16.9.2. Strategia e implementazione della BI
 - 16.9.3. Presente e futuro in BI
- 16.10. ISO/IEC 12207
 - 16.10.1. Cos'è «ISO/IEC 12207»?
 - 16.10.2. Analisi dei sistemi informativi
 - 16.10.3. Progettazione di un sistema informativo
 - 16.10.4. Implementazione e accettazione del sistema informativo

Modulo 17. Riutilizzo di software

- 17.1. Panoramica del riutilizzo di software
 - 17.1.1. Che cos'è il riutilizzo di software?
 - 17.1.2. Vantaggi e svantaggi del riutilizzo di software
 - 17.1.3. Principali tecniche di riutilizzo di software
- 17.2. Introduzione ai modelli di progettazione
 - 17.2.1. Che cos'è un modello di progettazione?
 - 17.2.2. Catalogo dei principali modelli di progettazione
 - 17.2.3. Come utilizzare i modelli per risolvere i problemi di progettazione
 - 17.2.4. Come selezionare il miglior modello di progettazione
- 17.3. Modelli di creazione
 - 17.3.1. Modelli di creazione
 - 17.3.2. Modello *Abstract Factory*
 - 17.3.3. Esempio di implementazione del modello *Abstract Factory*
 - 17.3.4. Modello *Builder*
 - 17.3.5. Esempio di implementazione del metodo *Builder*
 - 17.3.6. Modello *Abstract Factory* vs *Builder*
- 17.4. Modelli di creazione (II)
 - 17.4.1. Modello *Factory Method*
 - 17.4.2. *Factory Method* vs *Abstract Factory*
 - 17.4.3. Modello *Singleton*
- 17.5. Modelli strutturali (I)
 - 17.5.1. Modelli strutturali
 - 17.5.2. Modello *Adapter*
 - 17.5.3. Modello *Bridge*

- 17.6. Modelli strutturali (II)
 - 17.6.1. Modello *Composite*
 - 17.6.2. Modello *Decorator*
- 17.7. Modelli strutturali (III)
 - 17.7.1. Modello *Facade*
 - 17.7.2. Modello *Proxy*
- 17.8. Modelli di comportamento (I)
 - 17.8.1. Concetto di modelli di comportamento
 - 17.8.2. Modelli di comportamento: catena di responsabilità
 - 17.8.3. Modello di comportamento Ordine
- 17.9. Modelli di comportamento (II)
 - 17.9.1. Modello Interprete o *Interpreter*
 - 17.9.2. Modello di iterazione
 - 17.9.3. Modello di osservazione
 - 17.9.4. Modello di strategia
- 17.10. *Frameworks*
 - 17.10.1. Concetto di *Framework*
 - 17.10.2. Sviluppo mediante *Frameworks*
 - 17.10.3. Modello *Model View Controller*
 - 17.10.4. *Framework* per la progettazione di interfacce grafiche
 - 17.10.5. *Frameworks* per lo sviluppo di applicazioni web
 - 17.10.6. *Frameworks* per la gestione della persistenza degli oggetti nei database

Modulo 18. Servizi di tecnologia dell'informazione

- 18.1. La trasformazione digitale (I)
 - 18.1.1. Innovazione aziendale
 - 18.1.2. Gestione della produzione
 - 18.1.3. Gestione finanziaria
- 18.2. La trasformazione digitale (II)
 - 18.2.1. Marketing
 - 18.2.2. Gestione delle Risorse Umane
 - 18.2.3. Un sistema informativo integrato
- 18.3. Caso di studio
 - 18.3.1. Presentazione dell'azienda
 - 18.3.2. Metodologie di analisi degli acquisti IT
 - 18.3.3. Determinazione di costi, benefici e rischi
 - 18.3.4. Valutazione economica dell'investimento

- 18.4. Governance e gestione delle TIC
 - 18.4.1. Definizione di governance delle tecnologie e dei sistemi informativi
 - 18.4.2. Differenza tra governance e gestione delle TSI
 - 18.4.3. Quadri per la governance e la gestione delle TSI
 - 18.4.4. Governance e gestione delle TSI e standard
- 18.5. La governance aziendale delle TIC
 - 18.5.1. Che cos'è la buona governance aziendale?
 - 18.5.2. Background sulla governance delle TIC
 - 18.5.3. La norma ISO/IEC 318.00:2008
 - 18.5.4. Attuare una buona governance delle TIC
 - 18.5.5. Governance delle TIC e migliori pratiche
 - 18.5.6. La governance aziendale: Panoramica e tendenze
- 18.6. Obiettivi di controllo per le tecnologie informatiche e correlate (COBIT)
 - 18.6.1. Quadro di attuazione
 - 18.6.2. Dominio: pianificazione e organizzazione
 - 18.6.3. Dominio: acquisizione e implementazione
 - 18.6.4. Dominio: consegna e supporto
 - 18.6.5. Dominio: monitoraggio e valutazione
 - 18.6.6. Implementazione della guida COBIT
- 18.7. La Biblioteca dell'Infrastruttura Tecnologica dell'Informazione (ITIL)
 - 18.7.1. Introduzione all'ITIL
 - 18.7.2. Strategie di servizio
 - 18.7.3. Design del servizio
 - 18.7.4. Transizione del servizio
 - 18.7.5. Funzionamento del servizio
 - 18.7.6. Miglioramento del servizio



- 18.8. Il sistema di gestione dei servizi
 - 18.8.1. Principi fondamentali di UNE-ISO/IEC 20000-1
 - 18.8.2. La struttura della serie di norme ISO/IEC 20000
 - 18.8.3. Requisiti del Sistema di Gestione dei Servizi (SGS)
 - 18.8.4. Progettazione e transizione di servizi nuovi o modificati
 - 18.8.5. Processi di fornitura del servizio
 - 18.8.6. Gruppi di processo
- 18.9. Il sistema di gestione dei beni dei software
 - 18.9.1. Giustificazione della necessità
 - 18.9.2. Antecedenti
 - 18.9.3. Presentazione della norma 19770
 - 18.9.4. Attuazione della gestione
- 18.10. Gestione della continuità aziendale
 - 18.10.1. Piani di continuità aziendale
 - 18.10.2. Implementazione di un BCM



*Un programma completo
che sarà fondamentale per il
tuo sviluppo 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

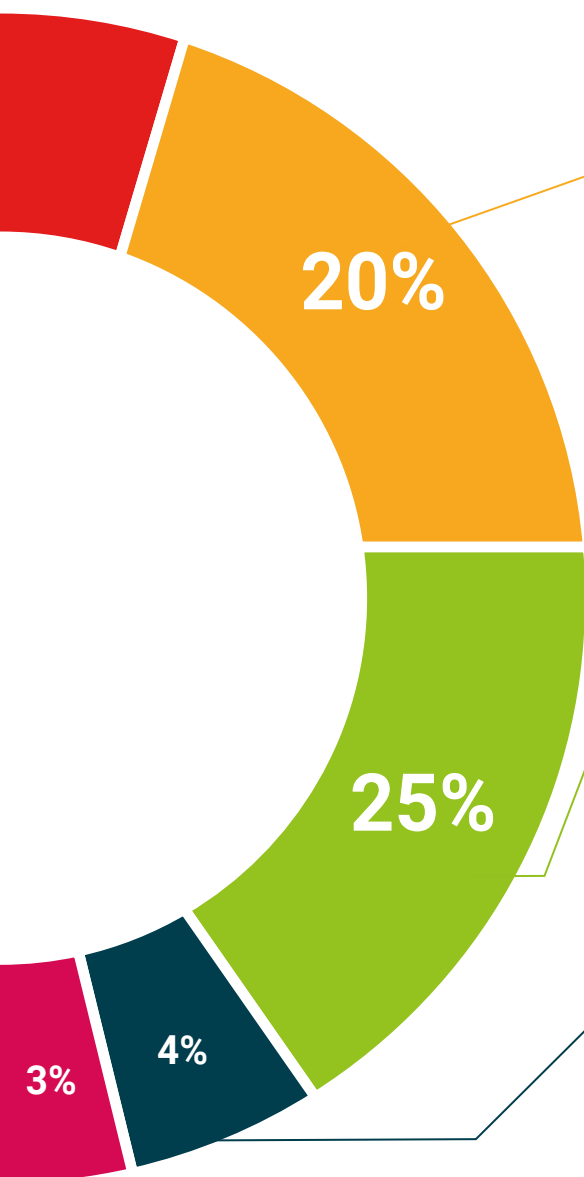
Svolgerai attività per sviluppare competenze e capacità specifiche in ogni area tematica. Pratiche e dinamiche per acquisire e sviluppare le competenze e le abilità che uno specialista deve sviluppare nel quadro della globalizzazione in cui viviamo.



Lettere 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 Specialistico in Ingegneria dei Software ti garantisce, oltre alla preparazione più rigorosa e aggiornata, l'accesso a una qualifica di Master Specialistico rilasciata da TECH Global University.



“

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

Questo programma ti consentirà di ottenere il titolo di studio di **Master Specialistico in Ingegneria dei Software** rilasciato da **TECH Global University**, la più grande università digitale del mondo.

TECH Global University è un'Università Ufficiale Europea riconosciuta pubblicamente dal Governo di Andorra ([bollettino ufficiale](#)). Andorra fa parte dello Spazio Europeo dell'Istruzione Superiore (EHEA) dal 2003. L'EHEA è un'iniziativa promossa dall'Unione Europea che mira a organizzare il quadro formativo internazionale e ad armonizzare i sistemi di istruzione superiore dei Paesi membri di questo spazio. Il progetto promuove valori comuni, l'implementazione di strumenti congiunti e il rafforzamento dei meccanismi di garanzia della qualità per migliorare la collaborazione e la mobilità tra studenti, ricercatori e accademici.

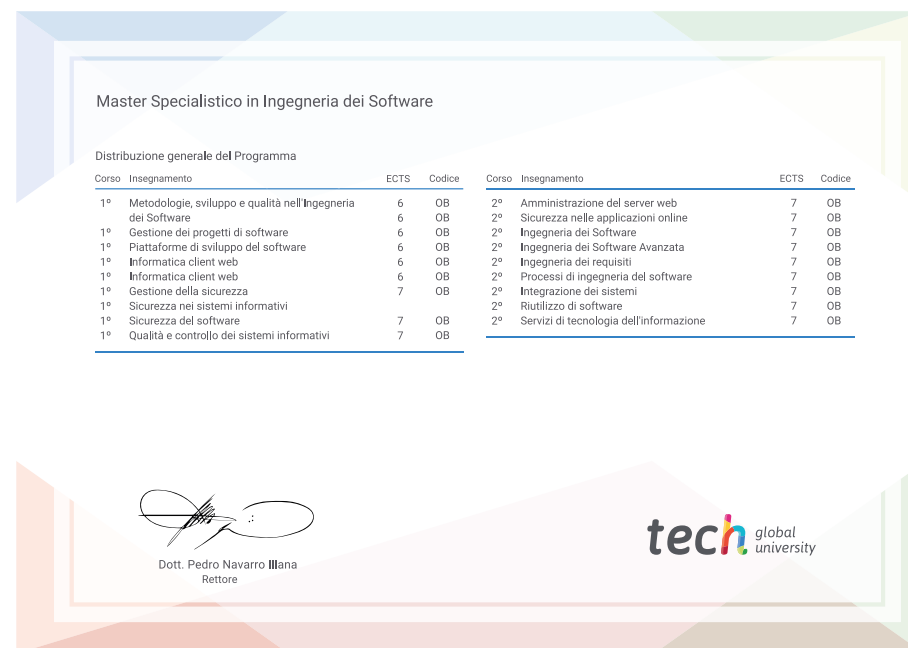
Questo titolo privato di **TECH Global University** è un programma europeo di formazione continua e aggiornamento professionale che garantisce l'acquisizione di competenze nella propria area di conoscenza, conferendo allo studente che supera il programma un elevato valore curriculare.

Titolo: **Master Specialistico in Ingegneria dei Software**

Modalità: **online**

Durata: **2 anni**

Accreditamento: **120 ECTS**



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 lingu



Master Specialistico
Ingegneria dei Software

- » Modalità: online
- » Durata: 2 anni
- » Titolo: TECH Global University
- » Accreditamento: 120 ECTS
- » Orario: a scelta
- » Esami: online

Master Specialistico Ingegneria dei Software