

# Master Specialistico

## Ingegneria e Qualità del Software





**tech** università  
tecnologica

## Master Specialistico Ingegneria e Qualità del Software

- » Modalità: online
- » Durata: 2 anni
- » Titolo: TECH Università Tecnologica
- » Orario: a tua scelta
- » Esami: online

Accesso al sito web: [www.techitute.com/it/informatica/master-specialistico/master-specialistico-ingegneria-qualita-software](http://www.techitute.com/it/informatica/master-specialistico/master-specialistico-ingegneria-qualita-software)

# Indice

01

Presentazione del programma

---

*pag. 4*

02

Perché studiare in TECH?

---

*pag. 8*

03

Piano di studi

---

*pag. 12*

04

Obiettivi didattici

---

*pag. 34*

05

Opportunità professionali

---

*pag. 40*

06

Metodologia di studio

---

*pag. 44*

07

Personale docente

---

*pag. 54*

08

Titolo

---

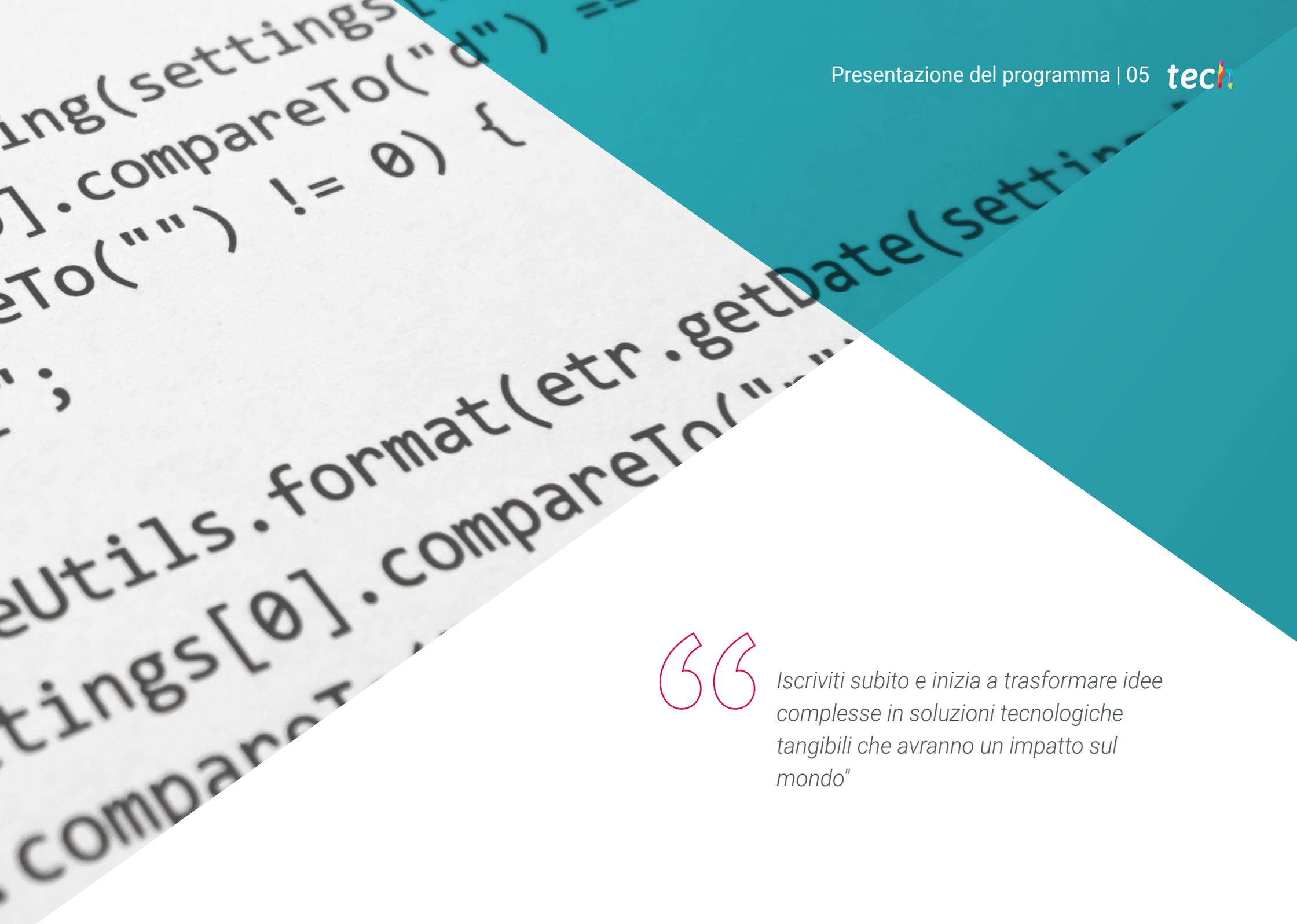
*pag. 60*

01

# Presentazione del programma

L'Ingegneria del Software è diventata il pilastro fondamentale della trasformazione digitale. Oggi tutti i settori dipendono dalle soluzioni tecnologiche per ottimizzare i processi, migliorare l'esperienza del cliente e rimanere competitivi. La qualità del Software, da parte sua, assicura che queste soluzioni siano affidabili, scalabili e sicure. Questa disciplina è un ramo dell'ingegneria che combina competenze tecniche e gestionali per garantire che i prodotti e i sistemi sviluppati siano funzionali e sostenibili. Per questo motivo, questo piano di studi va oltre la semplice programmazione, concentrandosi sull'intero ciclo di vita del Software, dalla concezione iniziale alla manutenzione e all'evoluzione del sistema. L'obiettivo principale è quello di offrire agli studenti un'opportunità accademica unica che fornisce loro informazioni all'avanguardia della tecnologia. TECH sviluppa questa qualifica multidisciplinare e 100% online che affronta dai fondamenti dell'ingegneria del Software fino alle ultime tendenze in metodologie agili.

```
... .getStr  
if (settings[0]  
name += "  
}  
name += Date  
if (set  
me.
```



“

*Iscriviti subito e inizia a trasformare idee complesse in soluzioni tecnologiche tangibili che avranno un impatto sul mondo"*

La qualità del Software garantisce che i sistemi non solo soddisfino i requisiti funzionali, ma siano anche intuitivi, sicuri e sostenibili a lungo termine. Questo aspetto è particolarmente importante in settori critici come quello finanziario, sanitario o dei trasporti, dove i guasti possono avere conseguenze gravi. Inoltre, dando priorità alla qualità, si assicura che le aziende possano adattarsi in modo agile ai continui progressi tecnologici e rispondere efficacemente alle crescenti richieste del mercato.

Utilizzando metodologie come lo sviluppo agile, *DevOps* e l'implementazione di standard internazionali di qualità, l'ingegneria del Software garantisce la consegna dei prodotti in tempi più brevi. Inoltre, i costi controllati e il livello di qualità che riduce al minimo gli errori critici sono stati ampliati con l'integrazione di tecnologie emergenti come l'intelligenza artificiale, il *cloud computing* e la cibersecurity. In questo contesto, il programma che TECH ha progettato è orientato a formare professionisti altamente qualificati in progettazione, sviluppo, gestione e garanzia della qualità del Software. Per acquisire le competenze necessarie, il piano di studi del Master Specialistico include i concetti più aggiornati sulla gestione dei progetti tecnologici e la direzione strategica. Questo approccio rappresenta un valore aggiunto sia per gli ingegneri che già occupano posizioni di responsabilità e desiderano aggiornare le loro conoscenze, sia per coloro che aspirano a guidare team e progetti per la prima volta in questo settore.

Uno dei principali vantaggi che offre questo programma è che sarà 100% online, eliminando la necessità di trasferimenti e di adattarsi a orari specifici. Inoltre, gli studenti avranno a disposizione il metodo di apprendimento *Relearning*, che si adatta al loro ritmo di studio. Questo approccio flessibile è molto utile, perché consente agli studenti di organizzare in modo efficiente i loro impegni quotidiani, sia professionali che familiari, raggiungendo così uno sviluppo completo.

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

- ♦ Sviluppo di casi di studio pratici presentati da esperti in informatica
- ♦ 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 settore dell'Ingegneria e della Qualità del Software
- ♦ Lezioni teoriche, domande all'esperto, forum di discussione su argomenti controversi e lavoro di riflessione individuale
- ♦ Disponibilità di accesso ai contenuti da qualsiasi dispositivo fisso o portatile dotato di connessione a Internet



*Grazie a TECH, non solo imparerai a sviluppare Software, ma a creare sistemi che fanno la differenza nella vita delle persone e delle aziende"*

“

*Padroneggia le competenze e gli strumenti più avanzati dell'ingegneria con la metodologia didattica più innovativa del panorama accademico attuale"*

Il personale docente del programma comprende rinomati specialisti del settore e altre aree correlate, che forniscono agli studenti le competenze necessarie a intraprendere un percorso di studio eccellente.

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. Lo studente potrà usufruire di un innovativo sistema di video interattivi creati da esperti di rinomata fama.

*Aumenta le tue aspettative professionali imparando in modo 100% online, senza interferire con i tuoi impegni personali e familiari.*

*Diventa un leader professionale in ingegneria, pronto a imparare da qualsiasi parte del mondo.*



02

# Perché studiare in TECH?

TECH è la più grande università digitale del mondo. Con un catalogo eccezionale di oltre 14.000 programmi accademici disponibili in 11 lingue, si posiziona come leader in termini di occupabilità, con un tasso di inserimento professionale del 99%. Inoltre, dispone di un enorme personale docente, composto da oltre 6.000 professori di altissimo prestigio internazionale.



“

*Studia presso la più grande università digitale del mondo e assicurati il successo professionale. Il futuro inizia con TECH"*

### La migliore università online al mondo secondo FORBES

La prestigiosa rivista Forbes, specializzata in affari e finanza, ha definito TECH «la migliore università online del mondo». Lo hanno recentemente affermato in un articolo della loro edizione digitale, che riporta il caso di successo di questa istituzione: «grazie all'offerta accademica che offre, alla selezione del suo personale docente e a un metodo innovativo di apprendimento orientato alla formazione dei professionisti del futuro».

**Forbes**  
Mejor universidad  
online del mundo

**Plan**  
de estudios  
más completo

### I piani di studio più completi del panorama universitario

TECH offre i piani di studio più completi del panorama universitario, con argomenti che coprono concetti fondamentali e, allo stesso tempo, i principali progressi scientifici nelle loro specifiche aree scientifiche. Inoltre, questi programmi sono continuamente aggiornati per garantire agli studenti l'avanguardia accademica e le competenze professionali più richieste. In questo modo, i titoli universitari forniscono agli studenti un vantaggio significativo per elevare le loro carriere verso il successo.

### Il miglior personale docente internazionale top

Il personale docente di TECH è composto da oltre 6.000 docenti di massimo prestigio internazionale. Professori, ricercatori e dirigenti di multinazionali, tra cui Isaiah Covington, allenatore dei Boston Celtics; Magda Romanska, ricercatrice principale presso MetaLAB ad Harvard; Ignacio Wistumba, presidente del dipartimento di patologia molecolare traslazionale di MD Anderson Cancer Center; o D.W Pine, direttore creativo della rivista TIME, ecc.

Profesorado  
**TOP**  
Internacional

La metodología  
más eficaz

### Un metodo di apprendimento unico

TECH è la prima università ad utilizzare il *Relearning* in tutte le sue qualifiche. Si tratta della migliore metodologia di apprendimento online, accreditata con certificazioni internazionali di qualità docente, disposte da agenzie educative prestigiose. Inoltre, questo modello accademico dirompente è integrato con il "Metodo Casistico", configurando così una strategia di insegnamento online unica. Vengono inoltre implementate risorse didattiche innovative tra cui video dettagliati, infografiche e riassunti interattivi.

### La più grande università digitale del mondo

TECH è la più grande università digitale del mondo. Siamo la più grande istituzione educativa, con il migliore e più ampio catalogo educativo digitale, cento per cento online e che copre la maggior parte delle aree di conoscenza. Offriamo il maggior numero di titoli di studio, diplomi e corsi post-laurea nel mondo. In totale, più di 14.000 titoli universitari, in dieci lingue diverse, ci rendono la più grande istituzione educativa del mondo.

**n°1**  
Mundial  
Mayor universidad  
online del mundo

### L'università online ufficiale dell'NBA

TECH è l'università online ufficiale dell'NBA. Grazie ad un accordo con la più grande lega di basket, offre ai suoi studenti programmi universitari esclusivi, nonché una vasta gamma di risorse educative incentrate sul business della lega e su altre aree dell'industria sportiva. Ogni programma presenta un piano di studi con un design unico e relatori ospiti eccezionali: professionisti con una distinta carriera sportiva che offriranno la loro esperienza nelle materie più rilevanti.

### Leader nell'occupabilità

TECH è riuscita a diventare l'università leader nell'occupabilità. Il 99% dei suoi studenti ottiene un lavoro nel campo accademico che hanno studiato, prima di completare un anno dopo aver terminato uno qualsiasi dei programmi universitari. Una cifra simile riesce a migliorare la propria carriera professionale immediatamente. Tutto questo grazie ad una metodologia di studio che basa la sua efficacia sull'acquisizione di competenze pratiche, assolutamente necessarie per lo sviluppo professionale.



### Google Partner Premier

Il gigante americano della tecnologia ha conferito a TECH il logo Google Partner Premier. Questo premio, accessibile solo al 3% delle aziende del mondo, conferisce valore all'esperienza efficace, flessibile e adattata che questa università offre agli studenti. Il riconoscimento non solo attesta il massimo rigore, rendimento e investimento nelle infrastrutture digitali di TECH, ma fa anche di questa università una delle compagnie tecnologiche più all'avanguardia del mondo.



### L'università meglio valutata dai suoi studenti

Gli studenti hanno posizionato TECH come l'università meglio valutata al mondo nei principali portali di recensioni, evidenziando il suo punteggio più alto di 4,9 su 5, ottenuto da oltre 1.000 recensioni. Questi risultati consolidano TECH come l'istituzione universitaria di riferimento a livello internazionale, riflettendo l'eccellenza e l'impatto positivo del suo modello educativo.



# 03

## Piano di studi

Il piano di studi del Master Specialistico in Ingegneria e Qualità del Software è progettato per fornire una specializzazione completa e avanzata in tutte le aree chiave dell'ingegneria del Software. I primi moduli si concentrano sui fondamenti, che vanno dalla progettazione Software e gestione dei requisiti alle architetture tecnologiche e metodologie agili. Man mano che il programma avanza, gli studenti si immergono in aree più specializzate come l'automazione dei test, l'integrazione continua e la garanzia della qualità. Inoltre, vengono inclusi corsi sulla gestione dei progetti tecnologici, in cui i partecipanti imparano a gestire team multidisciplinari.





“

*Questo Master Specialistico ti prepara a diventare l'esperto che fa la differenza nel settore dell'Ingegneria e della Qualità del Software"*

## Modulo 1. Qualità del Software: Livelli di sviluppo TRL

- 1.1. Elementi che influenzano la Qualità del Software (I). Il debito tecnico
  - 1.1.1. Il debito tecnico: Cause e conseguenze
  - 1.1.2. Qualità del Software: Principi generali
  - 1.1.3. Software senza principi e con principi di qualità
    - 1.1.3.1. Conseguenze
    - 1.1.3.2. La necessità di applicare i principi della qualità nel Software
  - 1.1.4. Qualità del Software: Tipologia
  - 1.1.5. Software di Qualità: Caratteristiche specifiche
- 1.2. Elementi che influenzano la Qualità del Software (II). Costi associati
  - 1.2.1. Qualità del Software: Elementi determinanti
  - 1.2.2. Qualità del Software: Idee sbagliate
  - 1.2.3. Qualità del Software: Costi associati
- 1.3. Modello di Qualità del Software (I). Gestione della conoscenza
  - 1.3.1. Modelli di qualità generali
    - 1.3.1.1. Gestione della qualità totale
    - 1.3.1.2. Modello Europeo di Eccellenza Aziendale (EFQM)
    - 1.3.1.3. Modello Six-Sigma
  - 1.3.2. Modelli di Gestione della Conoscenza
    - 1.3.2.1. Modello Dyba
    - 1.3.2.2. Modello Seks
  - 1.3.3. Esperienza di fabbrica e paradigma QIP
  - 1.3.4. Modelli di qualità d'uso (25010)
- 1.4. Modello di Qualità del Software (II). Qualità dei dati, dei processi e dei modelli SEI
  - 1.4.1. Modello di qualità dei dati
  - 1.4.2. Modello di processo del Software
  - 1.4.3. *Software & Systems Process Engineering Metamodel Specification (SPEM)*
  - 1.4.4. Modelli del SEI
    - 1.4.4.1. CMMI
    - 1.4.4.2. SCAMPI
    - 1.4.4.3. IDEAL
- 1.5. Standard ISO di Qualità del Software (I). Analisi degli standard
  - 1.5.1. Norme ISO 9000
    - 1.5.1.1. Norme ISO 9000
    - 1.5.1.2. Famiglia di standard di qualità ISO (9000)
  - 1.5.2. Altri standard ISO relativi alla Qualità
  - 1.5.3. Standard di Modellazione della Qualità (ISO 2501)
  - 1.5.4. Normativa di Misurazione della Qualità (ISO 2502n)
- 1.6. Standard ISO di Qualità del Software (II). Requisiti e valutazione
  - 1.6.1. Standard dei Requisiti di Qualità (2503n)
  - 1.6.2. Normativa sulla Valutazione della Qualità (2504n)
  - 1.6.3. ISO/IEC 24744: 2007
- 1.7. Livelli di sviluppo TRL (I). Livelli da 1 a 4
  - 1.7.1. Livelli TRL
  - 1.7.2. Livello 1: principi di base
  - 1.7.3. Livello 2: concetto e/o applicazione
  - 1.7.4. Livello 3: funzione analitica critica
  - 1.7.5. Livello 4: convalida dei componenti in ambiente di laboratorio
- 1.8. Livelli di sviluppo TRL (II). Livelli da 5 a 9
  - 1.8.1. Livello 5: convalida del componente in un ambiente pertinente
  - 1.8.2. Livello 6: modello di sistema/sottosistema
  - 1.8.3. Livello 7: dimostrazione in ambiente reale
  - 1.8.4. Livello 8: sistema completo e certificato
  - 1.8.5. Livello 9: successo in un ambiente reale
- 1.9. Livelli di sviluppo TRL: Usi
  - 1.9.1. Esempio di azienda con ambiente di laboratorio
  - 1.9.2. Esempio di azienda di R&S+I
  - 1.9.3. Esempio di azienda di R&S+I industriale
  - 1.9.4. Esempio di joint venture laboratorio-ingegneria
- 1.10. Qualità del Software: Dettagli principali
  - 1.10.1. Dettagli metodologici
  - 1.10.2. Dettagli tecnici
  - 1.10.3. Dettagli sulla gestione dei progetti Software
    - 1.10.3.1. Qualità dei sistemi informatici
    - 1.10.3.2. Qualità del Prodotto Software
    - 1.10.3.3. Qualità del Processo Software

**Modulo 2. Sviluppo di Progetti Software: Documentazione funzionale e tecnica**

- 2.1. Gestione dei progetti
  - 2.1.1. Gestione di progetti sulla Qualità del Software
  - 2.1.2. Gestione dei progetti: Vantaggi
  - 2.1.3. Gestione dei progetti: Tipologia
- 2.2. Metodologia nella gestione di progetti
  - 2.2.1. Metodologia nella gestione di progetti
  - 2.2.2. Metodologie di progetto: Tipologia
  - 2.2.3. Metodologia nella gestione di progetti: Applicazioni
- 2.3. Fase di identificazione dei requisiti
  - 2.3.1. Identificazione dei requisiti del progetto
  - 2.3.2. Gestione delle riunioni di progetto
  - 2.3.3. Documentazione da fornire
- 2.4. Modello
  - 2.4.1. Fase iniziale
  - 2.4.2. Fase di analisi
  - 2.4.3. Fase di costruzione
  - 2.4.4. Fase di test
  - 2.4.5. Consegna
- 2.5. Modello di dati da utilizzare
  - 2.5.1. Determinazione del nuovo modello di dati
  - 2.5.2. Identificazione del piano di migrazione dei dati
  - 2.5.3. Set di dati
- 2.6. Impatto su altri progetti
  - 2.6.1. Impatto di un progetto: Esempi
- 2.7. *MUST* del progetto
  - 2.7.1. *MUST* de progetto
  - 2.7.2. Identificazione dei *MUST* del progetto
  - 2.7.3. Identificazione dei punti di attuazione per la realizzazione di un progetto
- 2.8. Il team di costruzione del progetto
  - 2.8.1. Ruoli da svolgere in base al progetto
  - 2.8.2. Contatto con le risorse umane per il reclutamento
  - 2.8.3. Consegna dei prodotti e calendario del progetto

- 2.9. Aspetti tecnici di un progetto Software
  - 2.9.1. Architetto del progetto: Aspetti tecnici
  - 2.9.2. Leader tecnici
  - 2.9.3. Costruzione del progetto Software
  - 2.9.4. Valutazione della qualità del codice Sonar
- 2.10. Prodotti da consegnare al progetto
  - 2.10.1. Analisi funzionale
  - 2.10.2. Modelli di dati
  - 2.10.3. Diagrammi di stato
  - 2.10.4. Documentazione tecnica

**Modulo 3. *Testing* di Software: Automazione dei test**

- 3.1. Modello di qualità del Software
  - 3.1.1. Qualità del prodotto
  - 3.1.2. Qualità del processo
  - 3.1.3. Qualità d'uso
- 3.2. Qualità del processo
  - 3.2.1. Qualità del processo
  - 3.2.2. Modelli di maturità
  - 3.2.3. Normativa ISO 15504
    - 3.2.3.1. Propositi
    - 3.2.3.2. Contesto
    - 3.2.3.3. Fasi
- 3.3. Normativa ISO/IEC 15504
  - 3.3.1. Categorie di processo
  - 3.3.2. Processo di sviluppo: Esempio
  - 3.3.3. Frammento di profilo
  - 3.3.4. Fasi

- 3.4. CMMI (*Capability Maturity Model Integration*)
  - 3.4.1. CMMI: Integrazione dei modelli di maturità delle capacità
  - 3.4.2. Modelli e aree: Tipologia
  - 3.4.3. Aree di processo
  - 3.4.4. Livelli di capacità
  - 3.4.5. Gestione dei processi
  - 3.4.6. Gestione dei progetti
- 3.5. Gestione delle modifiche e del repository
  - 3.5.1. Gestione delle modifiche al Software
    - 3.5.1.1. Voce di configurazione: Integrazione continua
    - 3.5.1.2. Linee
    - 3.5.1.3. Diagrammi di flusso
    - 3.5.1.4. *Rami*
  - 3.5.2. Repository
    - 3.5.2.1. Controllo delle versioni
    - 3.5.2.2. Team di lavoro e utilizzo del repository
    - 3.5.2.3. Integrazione continua nel repository
- 3.6. *Team Foundation Server (TFS)*
  - 3.6.1. Installazione e configurazione
  - 3.6.2. Creazione di un progetto di squadra
  - 3.6.3. Aggiunta di contenuti al controllo del codice sorgente
  - 3.6.4. *TFS on Cloud*
- 3.7. *Testing*
  - 3.7.1. Motivazione per la realizzazione di test
  - 3.7.2. Test di verifica
  - 3.7.3. Test beta
  - 3.7.4. Implementazione e manutenzione
- 3.8. Implementazione e manutenzione
  - 3.8.1. *Load testing*
  - 3.8.2. Test con *LoadView*
  - 3.8.3. Test con *K6 Cloud*
  - 3.8.4. Test con *Loader*

- 3.9. Test unitari di stress e resistenza
  - 3.9.1. Motivazione dei test unitari
  - 3.9.2. Strumenti per *Unit Testing*
  - 3.9.3. Motivazione degli stress test
  - 3.9.4. Test con *StressTesting*
  - 3.9.5. Motivazione degli stress test
  - 3.9.6. Test con *LoadRunner*
- 3.10. La scalabilità: Progettazione di Software scalabile
  - 3.10.1. Scalabilità e architettura del Software
  - 3.10.2. Indipendenza tra gli strati
  - 3.10.3. Accoppiamento tra gli strati: Modelli di architettura

## Modulo 4. Metodologie di Gestione dei Progetti Software: Metodologie Waterfall vs. Metodologie Agili

- 4.1. Metodologia Waterfall
  - 4.1.1. Metodologia Waterfall
  - 4.1.2. Metodologia Waterfall: Influenza sulla Qualità del Software
  - 4.1.3. Metodologia Waterfall: Esempi
- 4.2. Metodologia Agile
  - 4.2.1. Metodologia Agile
  - 4.2.2. Metodologia Agile: Influenza sulla Qualità del Software
  - 4.2.3. Metodologia Agile: Esempi
- 4.3. Metodologia SCRUM
  - 4.3.1. Metodologia SCRUM
  - 4.3.2. Manifesto SCRUM
  - 4.3.3. Applicazione di SCRUM
- 4.4. Kanban board
  - 4.4.1. Metodo Kanban
  - 4.4.2. Kanban board
  - 4.4.3. Kanban board: Esempi di applicazione
- 4.5. Gestione del progetto con Waterfall
  - 4.5.1. Fasi di un progetto
  - 4.5.2. Visione in un progetto Waterfall
  - 4.5.3. Prodotti da prendere in considerazione

- 4.6. Gestione del progetto con SCRUM
    - 4.6.1. Fasi di un progetto SCRUM
    - 4.6.2. Visione in un progetto SCRUM
    - 4.6.3. Aspetti da considerare
  - 4.7. Waterfall vs SCRUM: Confronto
    - 4.7.1. Approccio al progetto pilota
    - 4.7.2. Il progetto applicando Waterfall: Esempio
    - 4.7.3. Il progetto applicando SCRUM: Esempio
  - 4.8. Visione del cliente
    - 4.8.1. Documenti in Waterfall
    - 4.8.2. Documenti in SCRUM
    - 4.8.3. Confronto
  - 4.9. Struttura di Kanban
    - 4.9.1. Storie degli utenti
    - 4.9.2. Backlog
    - 4.9.3. Analisi Kanban
  - 4.10. Progetti ibridi
    - 4.10.1. Costruzione del progetto
    - 4.10.2. Gestione dei progetti
    - 4.10.3. Aspetti da considerare
- 
- Modulo 5. TDD (*Test Driven Development*): Progettazione del Software guidata dai test**
- 5.1. TDD: *Test Driven Development*
    - 5.1.1. TDD: *Test Driven Development*
    - 5.1.2. TDD: Influenza del TDD sulla qualità
    - 5.1.3. Progettazione e sviluppo basati sui test: Esempi
  - 5.2. Ciclo TDD
    - 5.2.1. Scelta di un requisito
    - 5.2.2. Esecuzione di test: Tipologie
      - 5.2.2.1. Test unitari
      - 5.2.2.2. Test di integrità
      - 5.2.2.3. Test *end-to-end*
    - 5.2.3. Verifica del test: Errori
    - 5.2.4. Creazione dell'Implementazione
    - 5.2.5. Esecuzione di test automatizzati
    - 5.2.6. Eliminazione dei doppioni
    - 5.2.7. Aggiornamento dell'elenco dei requisiti
    - 5.2.8. Ripetizione del ciclo TDD
    - 5.2.9. Ciclo TDD: Esempio teorico e pratico
  - 5.3. Strategie di implementazione del TDD
    - 5.3.1. Implementazione di prova
    - 5.3.2. Implementazione triangolare
    - 5.3.3. Implementazione ovvia
  - 5.4. TDD: Uso, Vantaggi e svantaggi
    - 5.4.1. Vantaggi di uso
    - 5.4.2. Limitazioni d'uso
    - 5.4.3. Equilibrio qualitativo nell'implementazione
  - 5.5. TDD: Buone pratiche
    - 5.5.1. Regole TDD
    - 5.5.2. Regola 1: prima di codificare in produzione, eseguire un test precedente che fallisce
    - 5.5.3. Regola 2: non scrivere più di un test unitario
    - 5.5.4. Regola 3: non scrivere più codice del necessario
    - 5.5.5. Errori e anti-pattern da evitare in TDD
  - 5.6. Simulazione di un progetto reale per l'utilizzo di TDD (I)
    - 5.6.1. Panoramica del progetto (Azienda A)
    - 5.6.2. Implementazione della TDD
    - 5.6.3. Esercizi proposti
    - 5.6.4. Esercizi: *Feedback*
  - 5.7. Simulazione di un progetto reale per l'utilizzo di TDD (II)
    - 5.7.1. Descrizione generale del progetto (Azienda B)
    - 5.7.2. Implementazione della TDD
    - 5.7.3. Esercizi proposti
    - 5.7.4. Esercizi: *Feedback*

- 5.8. Simulazione di un progetto reale per l'utilizzo di TDD (III)
  - 5.8.1. Descrizione generale del progetto (Azienda C)
  - 5.8.2. Implementazione della TDD
  - 5.8.3. Esercizi proposti
  - 5.8.4. Esercizi: *Feedback*
- 5.9. Alternative al TDD: *Test Driven Development*
  - 5.9.1. TCR (*Test Commit Revert*)
  - 5.9.2. BDD (*Behavior Driven Development*)
  - 5.9.3. ATDD (*Acceptance Test Driven Development*)
  - 5.9.4. TDD: Confronto teorico
- 5.10. TDD TCR, BDD e ATDD: Confronto pratico
  - 5.10.1. Definizione del problema
  - 5.10.2. Risoluzione con il TCR
  - 5.10.3. Risoluzione con il BDD
  - 5.10.4. Risoluzione con il ATDD

## Modulo 6. DevOps: Gestione della Qualità del Software

- 6.1. DevOps: Gestione della Qualità del Software
  - 6.1.1. DevOps
  - 6.1.2. DevOps e Qualità del Software
  - 6.1.3. DevOps: Benefici della cultura DevOps
- 6.2. DevOps: Rapporto con Agile
  - 6.2.1. Consegna accelerata
  - 6.2.2. Qualità
  - 6.2.3. Riduzione dei costi
- 6.3. Implementazione di DevOps
  - 6.3.1. Identificazione di problemi
  - 6.3.2. Implementazione in un'azienda
  - 6.3.3. Metriche di implementazione
- 6.4. Ciclo di consegna del Software
  - 6.4.1. Metodi di progettazione
  - 6.4.2. Convenzioni
  - 6.4.3. Tabella di marcia

- 6.5. Sviluppo di codice privo di errori
  - 6.5.1. Codice mantenibile
  - 6.5.2. Modelli di sviluppo
  - 6.5.3. *Testing* del codice
  - 6.5.4. Sviluppo di software a livello di codice: Best practice
- 6.6. Automatizzazione
  - 6.6.1. Automatizzazione: Tipi di test
  - 6.6.2. Costo dell'automazione e della manutenzione
  - 6.6.3. Automatizzazione: Attenuare gli errori
- 6.7. Distribuzione
  - 6.7.1. Valutazione dell'obiettivo
  - 6.7.2. Progettazione di un processo automatico e adattato
  - 6.7.3. Feedback e capacità di risposta
- 6.8. Gestione degli Incidenti
  - 6.8.1. Preparazione agli incidenti
  - 6.8.2. Analisi e risoluzione degli incidenti
  - 6.8.3. Come evitare errori futuri
- 6.9. Automazione della distribuzione
  - 6.9.1. Preparazione per le distribuzioni automatiche
  - 6.9.2. Valutazione dello stato di salute del processo automatico
  - 6.9.3. Metriche e capacità di rollback
- 6.10. Buone pratiche: Evoluzione di DevOps
  - 6.10.1. Guida alle migliori pratiche applicando DevOps
  - 6.10.2. DevOps: Metodologia per il team
  - 6.10.3. Evitare le nicchie

## Modulo 7. DevOps e Integrazione Continua: Soluzioni pratiche avanzate nello sviluppo di Software

- 7.1. Flusso di consegna del Software
  - 7.1.1. Identificazione di attori e artefatti
  - 7.1.2. Progettazione del flusso di consegna del Software
  - 7.1.3. Flusso di consegna del Software: Requisiti delle varie fasi

- 7.2. Automazione dei processi
  - 7.2.1. Integrazione continua
  - 7.2.2. Distribuzione continua
  - 7.2.3. Configurazione degli ambienti e gestione dei segreti
- 7.3. Pipeline dichiarative
  - 7.3.1. Differenze tra pipeline tradizionali, simili al codice e dichiarative
  - 7.3.2. Pipeline dichiarative
  - 7.3.3. Pipeline dichiarative in Jenkins
  - 7.3.4. Confronto tra i fornitori di integrazione continua
- 7.4. Gateway di qualità e feedback arricchito
  - 7.4.1. Gateway di qualità
  - 7.4.2. Standard di Qualità con gateway di qualità: Manutenzione
  - 7.4.3. Requisiti aziendali sulle richieste di integrazione
- 7.5. Gestione degli artefatti
  - 7.5.1. Artefatti e ciclo di vita
  - 7.5.2. Sistemi di conservazione e gestione degli artefatti
  - 7.5.3. Sicurezza nella gestione degli artefatti
- 7.6. Distribuzione continua
  - 7.6.1. Distribuzione continua come contenitore
  - 7.6.2. Distribuzione continua con PaaS
- 7.7. Migliorare il runtime della pipeline: analisi statica e *Git Hooks*
  - 7.7.1. Analisi statica
  - 7.7.2. Regole di stile del codice
  - 7.7.3. *Git Hooks* e *Test unitari*
  - 7.7.4. L'impatto dell'infrastruttura
- 7.8. Vulnerabilità dei contenitori
  - 7.8.1. Vulnerabilità dei contenitori
  - 7.8.2. Scansione di immagini
  - 7.8.3. Rapporti e avvisi periodici

## Modulo 8. Progettazione di Database (DB): Normalizzazione e Rendimento, Qualità del Software

- 8.1. Progettazione di database
  - 8.1.1. Database: Tipologia
  - 8.1.2. Database attualmente utilizzati
    - 8.1.2.1. Relazionali
    - 8.1.2.2. Chiave-Valore
    - 8.1.2.3. Basati sulla rete
  - 8.1.3. Qualità del dato
- 8.2. Progettazione del modello entità-relazione (I)
  - 8.2.1. Modello entità-relazione: Qualità e documentazione
  - 8.2.2. Entità
    - 8.2.2.1. Entità forte
    - 8.2.2.2. Entità debole
  - 8.2.3. Attributi
  - 8.2.4. Insieme di relazioni
    - 8.2.4.1. 1 a 1
    - 8.2.4.2. 1 a molti
    - 8.2.4.3. Molti a 1
    - 8.2.4.4. Molti a molti
  - 8.2.5. Chiavi
    - 8.2.5.1. Chiave primaria
    - 8.2.5.2. Chiave esterna
    - 8.2.5.3. Chiave primaria dell'entità debole
  - 8.2.6. Restrizioni
  - 8.2.7. Cardinalità
  - 8.2.8. Ereditarietà
  - 8.2.9. Aggregazione
- 8.3. Modello entità-relazione (II). Strumenti
  - 8.3.1. Modello entità-relazione (III). Strumenti
  - 8.3.2. Modello entità-relazione (IV). Esempio pratico
  - 8.3.3. Modello entità-relazione fattibile
    - 8.3.3.1. Campione visivo
    - 8.3.3.2. Campione in rappresentazione tabellare

- 8.4. Standardizzazione dei database (DB) (I). Considerazioni sulla Qualità del Software
  - 8.4.1. Standardizzazione e qualità del DB
  - 8.4.2. Dipendenze
    - 8.4.2.1. Dipendenza funzionale
    - 8.4.2.2. Proprietà della dipendenza funzionale
    - 8.4.2.3. Proprietà desunte
  - 8.4.3. Chiavi
- 8.5. Standardizzazione dei database (DB) (II). Forme normali e regole di Codd
  - 8.5.1. Forme normali
    - 8.5.1.1. Prima forma normale (1FN)
    - 8.5.1.2. Seconda forma normale (2FN)
    - 8.5.1.3. Terza forma normale (3FN)
    - 8.5.1.4. Forma normale di Boyce-Codd (BCNF)
    - 8.5.1.5. Quarta forma normale (4FN)
    - 8.5.1.6. Quinta forma normale (5FN)
  - 8.5.2. Le regole di Codd
    - 8.5.2.1. Regola 1: Informazione
    - 8.5.2.2. Regola 2: accesso garantito
    - 8.5.2.3. Regola 3: Trattamento sistematico dei valori nulli
    - 8.5.2.4. Regola 4: descrizione del database
    - 8.5.2.5. Regola 5: Sottolinguaggio integrale
    - 8.5.2.6. Regola 6: aggiornamento della vista
    - 8.5.2.7. Regola 7: inserimento e aggiornamento
    - 8.5.2.8. Regola 8: indipendenza fisica
    - 8.5.2.9. Regola 9: indipendenza logica
    - 8.5.2.10. Regola 10: Indipendenza dall'integrità
      - 8.5.2.10.1. Regole di integrità
    - 8.5.2.11. Regola 11: Distribuzione
    - 8.5.2.12. Regola 12: Non sovrersione
  - 8.5.3. Esempio pratico
- 8.6. Memorizzazione di dati / sistema OLAP
  - 8.6.1. Memorizzazione di dati
  - 8.6.2. Tabella dei fatti
  - 8.6.3. Tabella delle dimensioni
  - 8.6.4. Creazione del sistema OLAP: Strumenti
- 8.7. Prestazioni del database (DB)
  - 8.7.1. Ottimizzazione dell'indice
  - 8.7.2. Ottimizzazione delle query
  - 8.7.3. Partizionamento delle tabelle
- 8.8. Simulazione di un progetto reale per il disegno di DB (I)
  - 8.8.1. Panoramica del progetto (Azienda A)
  - 8.8.2. Applicazioni della progettazione di database
  - 8.8.3. Esercizi proposti
  - 8.8.4. Esercizi proposti: *Feedback*
- 8.9. Simulazione di un progetto reale per il disegno di DB (II)
  - 8.9.1. Descrizione generale del progetto (Azienda B)
  - 8.9.2. Applicazioni della progettazione di database
  - 8.9.3. Esercizi proposti
  - 8.9.4. Esercizi proposti: *Feedback*
- 8.10. Importanza dell'ottimizzazione dei DB nella qualità del software
  - 8.10.1. Ottimizzazione del design
  - 8.10.2. Ottimizzazione del codice delle query
  - 8.10.3. Ottimizzazione del codice delle procedure memorizzate
  - 8.10.4. Influenza dei *trigger* sulla Qualità del Software: Raccomandazioni per l'uso

## Modulo 9. Progettazione di architetture scalabili: L'Architettura nel Ciclo di Vita del Software

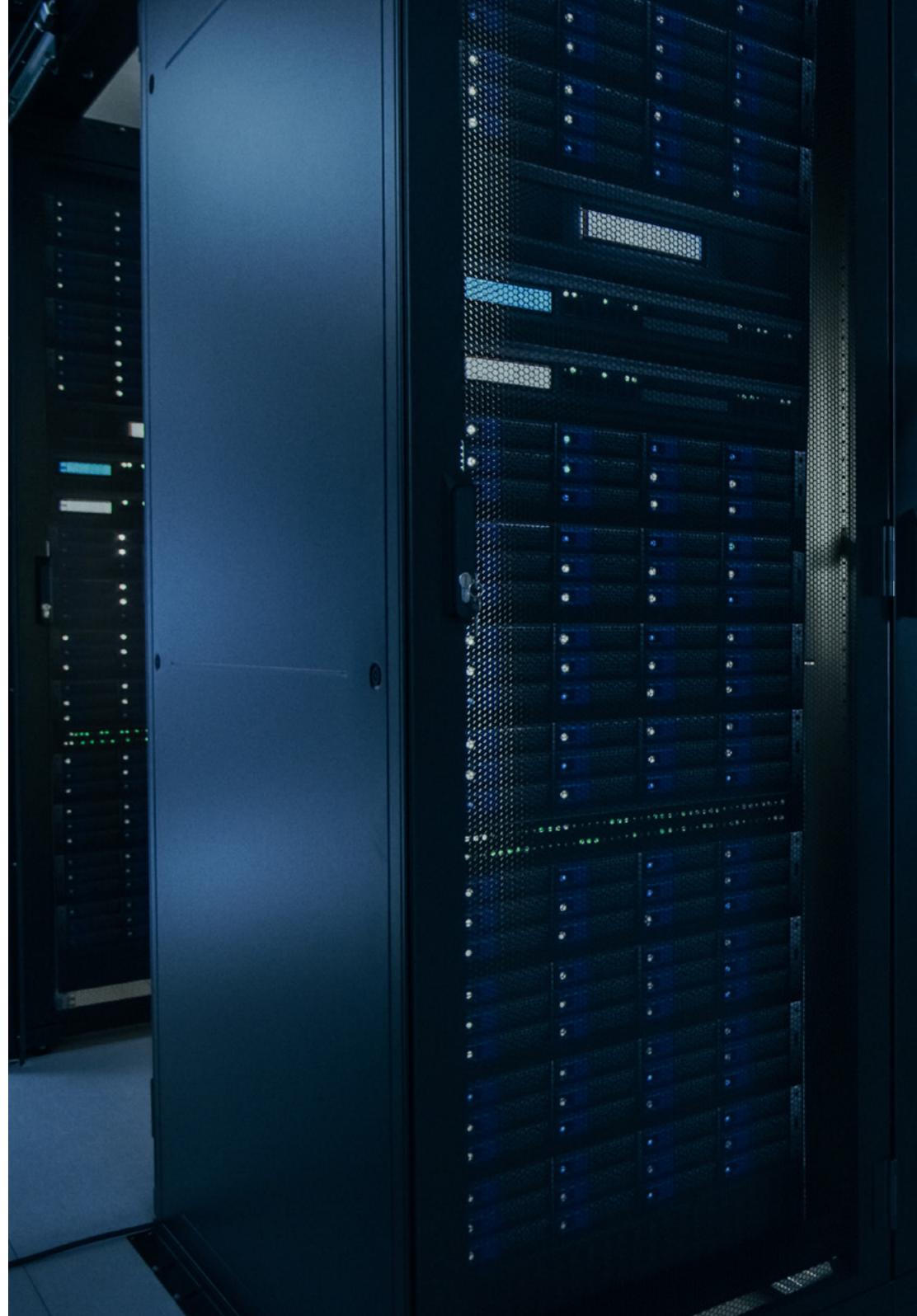
- 9.1. Progettazione di Architetture Scalabili (I)
  - 9.1.1. Architetture scalabili
  - 9.1.2. Principi di un'architettura scalabile
    - 9.1.2.1. Affidabile
    - 9.1.2.2. Scalabile
    - 9.1.2.3. Manutenibile
  - 9.1.3. Tipi di scalabilità
    - 9.1.3.1. Verticale
    - 9.1.3.2. Orizzontale
    - 9.1.3.3. Combinato

- 9.2. Architetture DDD (*Domain-Driven Design*)
  - 9.2.1. Il modello DDD: Orientamento al dominio
  - 9.2.2. Livelli, distribuzione delle responsabilità e modelli di progettazione
  - 9.2.3. Il disaccoppiamento come base per la qualità
- 9.3. Progettazione di Architetture Scalabili (II). Vantaggi, limiti e strategie di progettazione
  - 9.3.1. Architettura scalabile: Benefici
  - 9.3.2. Architettura scalabile: Limiti
  - 9.3.3. Strategie per lo sviluppo di architetture scalabili (tabella descrittiva)
- 9.4. Ciclo di vita del Software (I). Fasi
  - 9.4.1. Ciclo di vita del Software
    - 9.4.1.1. Fasi di pianificazione
    - 9.4.1.2. Fase di analisi
    - 9.4.1.3. Fase di progettazione
    - 9.4.1.4. Fase di implementazione
    - 9.4.1.5. Fase di test
    - 9.4.1.6. Fase di installazione/dispiegamento
    - 9.4.1.7. Fase di utilizzo e manutenzione
- 9.5. Modelli di ciclo di vita del Software
  - 9.5.1. Modello a cascata
  - 9.5.2. Modello ripetitivo
  - 9.5.3. Modello a spirale
  - 9.5.4. Modello Big Bang
- 9.6. Ciclo di vita del Software (II). Automatizzazione
  - 9.6.1. Cicli di vita dello sviluppo del Software: Soluzioni
    - 9.6.1.1. Integrazione continua e sviluppo continuo (CI/CD)
    - 9.6.1.2. Metodologia Agile
    - 9.6.1.3. DevOps / operazioni di produzione
  - 9.6.2. Tendenze future
  - 9.6.3. Esempi pratici
- 9.7. Architettura del Software nel suo ciclo di vita
  - 9.7.1. Benefici
  - 9.7.2. Limiti
  - 9.7.3. Strumenti
- 9.8. Simulazione di un progetto reale per il disegno dell'architettura del Software (I)
  - 9.8.1. Panoramica del progetto (Azienda A)
  - 9.8.2. Applicazioni della progettazione dell'architettura del Software
  - 9.8.3. Esercizi proposti
  - 9.8.4. Esercizi proposti. *Feedback*
- 9.9. Simulazione di un progetto reale per il disegno dell'architettura Software (II)
  - 9.9.1. Descrizione generale del progetto (Azienda B)
  - 9.9.2. Applicazioni della progettazione dell'architettura del Software
  - 9.9.3. Esercizi proposti
  - 9.9.4. Esercizi proposti: *Feedback*
- 9.10. Simulazione di un progetto reale per il disegno dell'architettura Software (III)
  - 9.10.1. Descrizione generale del progetto (Azienda C)
  - 9.10.2. Applicazioni della progettazione dell'architettura del Software
  - 9.10.3. Esercizi proposti
  - 9.10.4. Esercizi proposti: *Feedback*

## Modulo 10. Criteri di qualità ISO, IEC 9126: Metriche della Qualità del Software

- 10.1. Criteri di qualità: Normativa ISO, IEC 9126
  - 10.1.1. Criteri di qualità
  - 10.1.2. Qualità del Software: Giustificazione, Normativa ISO, IEC 9126
  - 10.1.3. Misurare la qualità del Software come indicatore chiave
- 10.2. Criteri di qualità del Software: Caratteristiche
  - 10.2.1. Affidabilità
  - 10.2.2. Funzionalità
  - 10.2.3. Efficienza
  - 10.2.4. Usability
  - 10.2.5. Mantenimento
  - 10.2.6. Portabilità
- 10.3. Normativa ISO, IEC 9126 (I). Presentazione
  - 10.3.1. Descrizione della Normativa ISO, IEC 9126
  - 10.3.2. Funzionalità
  - 10.3.3. Affidabilità
  - 10.3.4. Usability
  - 10.3.5. Mantenimento

- 10.3.6. Portabilità
- 10.3.7. Qualità in uso
- 10.3.8. Metriche della qualità del Software
- 10.3.9. Metriche di qualità in ISO 9126
- 10.4. Normativa ISO, IEC 9126 (II). Modelli McCall e Boehm
  - 10.4.1. Modello McCall: Fattori di qualità
  - 10.4.2. Modello Boehm
  - 10.4.3. Livello intermedio. Caratteristiche
- 10.5. Metriche della qualità del Software (I). Elementi
  - 10.5.1. Misura
  - 10.5.2. Metriche
  - 10.5.3. Indicatore
    - 10.5.3.1. Tipi di indicatori
  - 10.5.4. Misure e modelli
  - 10.5.5. Ambito di applicazione delle metriche del Software
  - 10.5.6. Classificazione delle metriche del Software
- 10.6. Misurazione della qualità del Software (II). Pratica di misurazione
  - 10.6.1. Raccolta dati metrici
  - 10.6.2. Misurazione degli attributi interni del prodotto
  - 10.6.3. Misurazione degli attributi esterni del prodotto
  - 10.6.4. Misurazione delle risorse
  - 10.6.5. Metriche per sistemi orientati agli oggetti
- 10.7. Progettazione di un unico indicatore di qualità del Software
  - 10.7.1. Singolo indicatore come qualificatore globale
  - 10.7.2. Sviluppo, giustificazione e applicazione degli indicatori
  - 10.7.3. Esempi di applicazione: Necessità di conoscere i dettagli
- 10.8. Simulazione di un progetto reale per la misurazione della qualità (I)
  - 10.8.1. Panoramica del progetto (Azienda A)
  - 10.8.2. Applicazione della misurazione della qualità
  - 10.8.3. Esercizi proposti
  - 10.8.4. Esercizi proposti: *Feedback*





10.9. Simulazione di un progetto reale per la misurazione della qualità (II)

- 10.9.1. Descrizione generale del progetto (Azienda B)
- 10.9.2. Applicazione della misurazione della qualità
- 10.9.3. Esercizi proposti
- 10.9.4. Esercizi proposti: *Feedback*

10.10. Simulazione di un progetto reale per la misurazione della qualità (III)

- 10.10.1. Descrizione generale del progetto (Azienda C)
- 10.10.2. Applicazione della misurazione della qualità
- 10.10.3. Esercizi proposti
- 10.10.4. Esercizi proposti: *Feedback*

**Modulo 11. Metodologie, sviluppo e qualità nell'Ingegneria dei Software**

11.1. Sviluppo Software basato su modelli

- 11.1.1. Necessità
- 11.1.3. Modellazione di oggetti
- 11.1.4. UML
- 11.1.5. Strumenti CASE

11.2. Modellazione di applicazioni e design pattern con UML

- 11.2.1. Modellazione avanzata dei requisiti
- 11.2.2. Modellazione statica avanzata
- 11.2.3. Modellazione dinamica avanzata
- 11.2.4. Modellazione dei componenti
- 11.2.5. Introduzione ai modelli di progettazione con UML
- 11.2.6. Adapter
- 11.2.7. Factory
- 11.2.8. Singleton
- 11.2.9. Strategy
- 11.2.10. Composite
- 11.2.11. Facade
- 11.2.12. Observer

- 11.3. Ingegneria guidata dai modelli
  - 11.3.1. Introduzione
  - 11.3.2. Metamodellazione dei sistemi
  - 11.3.3. MDA
  - 11.3.4. DSL
  - 11.3.5. Perfezionamento del modello con OCL
  - 11.3.6. Trasformazioni del modello
- 11.4. Ontologie nell'Ingegneria dei Software
  - 11.4.1. Introduzione
  - 11.4.2. Ingegneria ontologica
  - 11.4.3. Applicazione delle ontologie nell'Ingegneria dei Software

## Modulo 12. Gestione dei progetti Software

- 12.1. Gestione degli *stakeholder* e del campo di applicazione
  - 12.1.1. Identificare le parti interessate
  - 12.1.2. Sviluppare un piano per la gestione delle parti interessate
  - 12.1.3. Gestire il coinvolgimento delle parti interessate
  - 12.1.4. Controllare il coinvolgimento delle parti interessate
  - 12.1.5. L'obiettivo del progetto
  - 12.1.6. La gestione e il piano relativo al campo di applicazione
  - 12.1.7. Riunire i requisiti
  - 12.1.8. Definire l'ambito di applicazione
  - 12.1.9. Creare la WBS
  - 12.1.10. Verificare e controllare il campo di applicazione
- 12.2. L'ideazione della tabella di marcia
  - 12.2.1. La gestione del tempo e il relativo piano
  - 12.2.2. Definire le attività
  - 12.2.3. Sequenza delle attività
  - 12.2.4. Stima delle risorse delle attività
  - 12.2.5. Stimare la durata delle attività
  - 12.2.6. Sviluppo della tabella di marcia e calcolo del percorso critico
  - 12.2.7. Controllo della tabella di marcia
- 12.3. Sviluppo del budget e risposta ai rischi
  - 12.3.1. Fare una stima dei costi
  - 12.3.2. Definire il budget e la curva a S
  - 12.3.3. Controllo dei costi e metodo earned value
  - 12.3.4. I concetti di rischio
  - 12.3.5. Come fare un'analisi dei rischi
  - 12.3.6. Lo sviluppo del piano di risposta
- 12.4. Comunicazione e risorse umane
  - 12.4.1. Pianificare la gestione delle comunicazioni
  - 12.4.2. Analisi dei requisiti di comunicazione
  - 12.4.3. Tecnologia delle comunicazioni
  - 12.4.4. Modelli di comunicazione
  - 12.4.5. Metodi di comunicazione
  - 12.4.6. Piano di gestione delle comunicazioni
  - 12.4.7. Gestire le comunicazioni
  - 12.4.8. La gestione delle risorse umane
  - 12.4.9. Soggetti principali e il loro ruolo nei progetti
  - 12.4.10. Tipi di organizzazione
  - 12.4.11. Organizzazione del progetto
  - 12.4.12. Squadre di lavoro
- 12.5. Approvvigionamento
  - 12.5.1. Il processo di acquisto
  - 12.5.2. Pianificazione
  - 12.5.3. Ricerca di fornitori e presentazione di offerte
  - 12.5.4. Assegnazione del contratto
  - 12.5.5. Gestione del contratto
  - 12.5.6. I contratti
  - 12.5.7. Tipi di contratto
  - 12.5.8. Negoziazione del contratto
- 12.6. Attuazione, monitoraggio, controllo e chiusura
  - 12.6.1. I gruppi dei processi
  - 12.6.2. Attuazione del progetto
  - 12.6.3. Monitoraggio e controllo del progetto
  - 12.6.4. Chiusura del progetto

- 12.7. Responsabilità professionale
  - 12.7.1. Responsabilità professionale
  - 12.7.2. Caratteristiche della responsabilità sociale e professionale
  - 12.7.3. Codice etico del leader di progetto
  - 12.7.4. Responsabilità vs. PMP®
  - 12.7.5. Esempi di responsabilità
  - 12.7.6. Vantaggi della professionalizzazione

## Modulo 13. Piattaforme di sviluppo Software

- 13.1. Introduzione allo sviluppo di applicazioni
  - 13.1.1. Applicazioni per desktop
  - 13.1.2. Linguaggio di programmazione
  - 13.1.3. Ambienti di sviluppo integrato
  - 13.1.4. Applicazioni web
  - 13.1.5. Applicazioni mobili
  - 13.1.6. Applicazioni nel cloud
- 13.2. Sviluppo di applicazioni e interfaccia grafica in Java
  - 13.2.1. Ambienti di sviluppo integrati per Java
  - 13.2.2. Principali IDE per Java
  - 13.2.3. Introduzione alla piattaforma di sviluppo Eclipse
  - 13.2.4. Introduzione alla piattaforma di sviluppo NetBeans
  - 13.2.5. Controller View Model per le interfacce utente grafiche
  - 13.2.6. Progettare un'interfaccia grafica in Eclipse
  - 13.2.7. Progettare un'interfaccia grafica in NetBeans
- 13.3. Debug e test in Java
  - 13.3.1. Test e debug dei programmi in Java
  - 13.3.2. Debug in Eclipse
  - 13.3.3. Debug in NetBeans
- 13.4. Sviluppo di applicazioni e interfaccia grafica in .NET
  - 13.4.1. Net Framework
  - 13.4.2. Componenti della piattaforma di sviluppo .NET
  - 13.4.3. Visual Studio .NET
  - 13.4.4. Strumenti GUI .NET
  - 13.4.5. La GUI con Windows Presentation Foundation
  - 13.4.6. Debug e compilazione di un'applicazione WPF
- 13.5. Programmazione per reti .NET
  - 13.5.1. Introduzione alla programmazione di rete .NET
  - 13.5.2. Richieste e risposte .NET
  - 13.5.3. Utilizzo dei protocolli applicativi .NET
  - 13.5.4. Sicurezza nella programmazione di reti .NET
- 13.6. Ambienti di sviluppo di applicazioni mobili
  - 13.6.1. Applicazioni mobili
  - 13.6.2. Applicazioni mobile per Android
  - 13.6.3. Fasi dello sviluppo in Android
  - 13.6.4. L'IDE Android Studio
- 13.7. Sviluppo di applicazioni in ambiente Android Studio
  - 13.7.1. Installare e avviare Android Studio
  - 13.7.2. Lanciare un'applicazione Android
  - 13.7.3. Sviluppo di interfacce grafiche in Android Studio
  - 13.7.4. Avvio delle attività in Android Studio
- 13.8. Debug e pubblicazione di applicazioni Android
  - 13.8.1. Debug di un'applicazione in Android Studio
  - 13.8.2. Memorizzazione delle applicazioni in Android Studio
  - 13.8.3. Pubblicare un'applicazione su Google Play
- 13.9. Sviluppo di applicazioni per il cloud
  - 13.9.1. *Cloud computing*
  - 13.9.2. Tipi di *cloud*: SaaS, PaaS, IaaS
  - 13.9.3. Principali piattaforme di sviluppo nel cloud
  - 13.9.4. Riferimenti bibliografici
- 13.10. Introduzione a Google Cloud Platform
  - 13.10.1. Nozioni di base di Google Cloud Platform
  - 13.10.2. Servizi di Google Cloud Platform
  - 13.10.3. Strumenti di Google Cloud Platform

## Modulo 14. Informatica client web

- 14.1. Introduzione all'HTML
  - 14.1.1. Struttura di un documento
  - 14.1.2. Colore
  - 14.1.3. Testo
  - 14.1.4. Collegamenti ipertestuali
  - 14.1.5. Immagini
  - 14.1.6. Liste
  - 14.1.7. Tabelle
  - 14.1.8. *Frame*
  - 14.1.9. Formolari
  - 14.1.10. Elementi specifici per le tecnologie mobili
  - 14.1.11. Elementi in disuso
- 14.2. Fogli di stile web (CSS)
  - 14.2.1. Elementi e struttura di un foglio di stile
    - 14.2.1.1. Creazione di fogli di stile
    - 14.2.1.2. Applicazione degli stili: Selezionatori
    - 14.2.1.3. Eredità dello stile e applicazione a cascata
    - 14.2.1.4. Formattazione della pagina con gli stili
    - 14.2.1.5. Struttura della pagina con gli stili: Box-model
  - 14.2.2. Progettazione di stili per diversi dispositivi
  - 14.2.3. Tipi di fogli di stile: statici e dinamici. Pseudo-classi
  - 14.2.4. Pratiche corrette nell'uso dei fogli di stile
- 14.3. Introduzione e storia di JavaScript
  - 14.3.1. Introduzione
  - 14.3.2. Storia di JavaScript
  - 14.3.3. Ambiente di sviluppo da utilizzare
- 14.4. Nozioni di base di programmazione web
  - 14.4.1. Sintassi di base di JavaScript
  - 14.4.2. Tipi di dati primitivi e operatori
  - 14.4.3. Variabili e ambiti
  - 14.4.4. Stringhe di testo e *template literal*
  - 14.4.5. Numeri e booleani
  - 14.4.6. Confronti
- 14.5. Strutture JavaScript complesse
  - 14.5.1. Vettori o *array* e oggetti
  - 14.5.2. Insiemi
  - 14.5.3. Mappe
  - 14.5.4. Disgiuntive
  - 14.5.5. Loop
- 14.6. Funzioni e oggetti
  - 14.6.1. Definizione e invocazione delle funzioni
  - 14.6.2. Argomenti
  - 14.6.3. Funzioni a freccia
  - 14.6.4. Funzioni *callback*
  - 14.6.5. Funzioni di ordine superiore
  - 14.6.6. Oggetti letterali
  - 14.6.7. L'oggetto *this*
  - 14.6.8. Oggetti come spazi dei nomi: l'oggetto Math e l'oggetto Date
- 14.7. Il modello a oggetti del documento (DOM)
  - 14.7.1. Cos'è la DOM?
  - 14.7.2. Un po' di storia
  - 14.7.3. Navigazione e recupero degli elementi
  - 14.7.4. Un DOM virtuale con JSDOM
  - 14.7.5. Selettori *query* o *query selector*
  - 14.7.6. Navigazione attraverso le proprietà
  - 14.7.7. Assegnazione degli attributi agli elementi
  - 14.7.8. Creazione e modifica dei nodi
  - 14.7.9. Aggiornamento dello stile degli elementi DOM
- 14.8. Sviluppo web moderno
  - 14.8.1. Flusso basato su event *listener*
  - 14.8.2. *Toolkits* web moderni e sistemi di allineamento
  - 14.8.3. Modalità JavaScript rigorosa
  - 14.8.4. Ulteriori informazioni sulle funzioni
  - 14.8.5. Promesse e funzioni asincrone
  - 14.8.6. *Closures*
  - 14.8.7. Programmazione funzionale
  - 14.8.8. POO in JavaScript

- 14.9. Usabilità del web
  - 14.9.1. Introduzione all'usabilità
  - 14.9.2. Definizione di usabilità
  - 14.9.3. Importanza del web design incentrato sull'utente
  - 14.9.4. Differenze tra accessibilità e usabilità
  - 14.9.5. Vantaggi e problemi nel combinare accessibilità e usabilità
  - 14.9.6. Vantaggi e difficoltà nell'implementazione di siti web utilizzabili
  - 14.9.7. Metodi di usabilità
  - 14.9.8. Analisi dei requisiti dell'utente
  - 14.9.9. Principi del progettazione concettuale: Prototipazione orientata all'utente
  - 14.9.10. Linee guida per la creazione di siti web usabili
    - 14.9.10.1. Linee guida per l'usability di Jakob Nielsen
    - 14.9.10.2. Linee guida per l'usability di Bruce Tognazzini
  - 14.9.11. Valutazione dell'usability
- 14.10. Accessibilità web
  - 14.10.1. Introduzione
  - 14.10.2. Definizione di accessibilità del web
  - 14.10.3. Tipi di disabilità
    - 14.10.3.1. Disabilità temporanea o permanente
    - 14.10.3.2. Disabilità visiva
    - 14.10.3.3. Disabilità uditiva
    - 14.10.3.4. Disabilità motorie
    - 14.10.3.5. Disabilità neurologica o cognitiva
    - 14.10.3.6. Difficoltà legate all'invecchiamento
    - 14.10.3.7. Vincoli derivanti dall'ambiente
    - 14.10.3.8. Ostacoli all'accesso al web
  - 14.10.4. Ausili tecnici e prodotti di assistenza per superare gli ostacoli
    - 14.10.4.1. Ausili per non vedenti
    - 14.10.4.2. Ausili per ipovedenti
    - 14.10.4.3. Ausili per persone affette da daltonismo
    - 14.10.4.4. Ausili per non udenti
    - 14.10.4.5. Ausili per persone con disabilità motorie
    - 14.10.4.6. Ausili per persone con disabilità cognitive e neurologiche
  - 14.10.5. Vantaggi e difficoltà nell'implementazione dell'accessibilità del web
  - 14.10.6. Norme e standard di accessibilità del web
  - 14.10.7. Organismi di regolamentazione dell'accessibilità del web
  - 14.10.8. Confronto tra norme e standard
  - 14.10.9. Linee guida per la conformità normativa e standard
    - 14.10.9.1. Descrizione delle linee guida principali (immagini, link, video, ecc.)
    - 14.10.9.2. Linee guida per una navigazione accessibile
      - 14.10.9.2.1. Percepibilità
      - 14.10.9.2.2. Operatività
      - 14.10.9.2.3. Comprensibilità
      - 14.10.9.2.4. Robustezza
  - 14.10.10. Descrizione del processo di conformità all'accessibilità del web
  - 14.10.11. Livelli di conformità
  - 14.10.12. Criteri di conformità
  - 14.10.13. Requisiti di conformità
  - 14.10.14. Metodologia per la valutazione dell'accessibilità dei siti web

## Modulo 15. Informatica per Server Web

- 15.1. Introduzione alla programmazione sul server: PHP
  - 15.1.1. Concetti base di programmazione sul server
  - 15.1.2. Sintassi PHP di base
  - 15.1.3. Creazione di contenuti HTML con PHP
  - 15.1.4. Ambienti di sviluppo e test: XAMPP
- 15.2. PHP avanzato
  - 15.2.1. Strutture di controllo PHP
  - 15.2.2. Funzioni in PHP
  - 15.2.3. Gestione degli *array* in PHP
  - 15.2.4. Gestione delle stringhe con PHP
  - 15.2.5. Orientamento agli oggetti in PHP
- 15.3. Modelli di dati
  - 15.3.1. Concetto di dato: Ciclo di vita dei dati
  - 15.3.2. Tipi di dati
    - 15.3.2.1. Basici
    - 15.3.2.2. Registri
    - 15.3.2.3. Dinamici

- 15.4. Modello relazionale
  - 15.4.1. Descrizione
  - 15.4.2. Entità e tipi di entità
  - 15.4.3. Elementi dei dati: Attributi
  - 15.4.4. Relazioni: tipi, sottotipi, cardinalità
  - 15.4.5. Chiavi: Tipi di chiavi
  - 15.4.6. Normalizzazione: Forme normali
- 15.5. Costruzione del modello logico dei dati
  - 15.5.1. Specifiche della tabella
  - 15.5.2. Definizione delle colonne
  - 15.5.3. Specifiche chiave
  - 15.5.4. Conversione in forme normali: Dipendenze
- 15.6. Il modello fisico dei dati: File di dati
  - 15.6.1. Descrizione dei file di dati
  - 15.6.2. Tipi di file
  - 15.6.3. Modalità di accesso
  - 15.6.4. Organizzazione dei file
- 15.7. Accesso ai database da PHP
  - 15.7.1. Introduzione a MariaDB
  - 15.7.2. Lavorare con un database MariaDB: il linguaggio SQL
  - 15.7.3. Accesso al database MariaDB da PHP
  - 15.7.4. Introduzione a MySql
  - 15.7.5. Lavorare con un database MariaDB: il linguaggio SQL
  - 15.7.6. Accedere ai database MariaDB da PHP
- 15.8. Interazione con il cliente da PHP
  - 15.8.1. Formulario PHP
  - 15.8.2. Cookie
  - 15.8.3. Gestione di sessioni
- 15.9. Architettura delle applicazioni web
  - 15.9.1. Lo schema Modello-Vista-Controller
  - 15.9.2. Controller
  - 15.9.3. Modello
  - 15.9.4. View

- 15.10. Introduzione ai servizi web
  - 15.10.1. Introduzione a XML
  - 15.10.2. Architetture orientate ai servizi (SOA): servizi web
  - 15.10.3. Creare servizi web SOAP e REST
  - 15.10.4. Il protocollo SOAP
  - 15.10.5. Il protocollo REST

## Modulo 16. Gestione della sicurezza

- 16.1. Sicurezza delle informazioni
  - 16.1.1. Introduzione
  - 16.1.2. La sicurezza delle informazioni implica riservatezza, integrità e disponibilità
  - 16.1.3. La sicurezza è una questione economica
  - 16.1.4. La sicurezza è un processo
  - 16.1.5. La classificazione delle informazioni
  - 16.1.6. La sicurezza delle informazioni comporta la gestione del rischio
  - 16.1.7. La sicurezza è legata ai controlli di sicurezza
  - 16.1.8. La sicurezza è sia fisica che logica
  - 16.1.9. La sicurezza coinvolge le persone
- 16.2. Il professionista della sicurezza dell'informazione
  - 16.2.1. Introduzione
  - 16.2.2. La sicurezza informatica come professione
  - 16.2.3. Certificazioni (ISC)2
  - 16.2.4. Standard ISO 27001
  - 16.2.5. Pratiche di sicurezza nella gestione dei servizi informatici
  - 16.2.6. Modelli di maturità per la sicurezza delle informazioni
  - 16.2.7. Altre certificazioni, standard e risorse professionali
- 16.3. Controllo di accessi
  - 16.3.1. Introduzione
  - 16.3.2. Requisiti del controllo di accessi
  - 16.3.3. Meccanismi di autenticazione
  - 16.3.4. Metodi di autorizzazione
  - 16.3.5. Contabilità e revisione degli accessi
  - 16.3.6. Tecnologie Triple A

- 16.4. Programmi, processi e politiche di sicurezza delle informazioni
  - 16.4.1. Introduzione
  - 16.4.2. Programmi di gestione della sicurezza
  - 16.4.3. Gestione dei rischi
  - 16.4.4. Progettazione della politica di sicurezza
- 16.5. Piani di continuità aziendale
  - 16.5.1. Introduzione ai BCP
  - 16.5.2. Fase I e II
  - 16.5.3. Fase III e IV
  - 16.5.4. Manutenzione del BCP
- 16.6. Procedure per la corretta tutela dell'azienda
  - 16.6.1. Reti DMZ
  - 16.6.2. Sistemi di rilevamento delle intrusioni
  - 16.6.3. Liste di controllo di accessi
  - 16.6.4. Imparare dall'attaccante: Honeypot
- 16.7. Architettura di sicurezza: Prevenzione
  - 16.7.1. Visione generale: Attività e modello a strati
  - 16.7.2. Difesa perimetrale (Firewalls, WAFs, IPS, ecc.)
  - 16.7.3. Difesa degli endpoint (strumenti, server e servizi)
- 16.8. Architettura di sicurezza: Screening
  - 16.8.1. Panoramica di rilevamento e monitoraggio
  - 16.8.2. Registri, interruzione del traffico crittografato, registrazione e Siem
  - 16.8.3. Avvisi e informazioni
- 16.9. Architettura di sicurezza: Reazione
  - 16.9.1. Reazione: Prodotti, servizi e risorse
  - 16.9.2. Gestione degli Incidenti
  - 16.9.3. CERT e CSIRT
- 16.10. Architettura di sicurezza: Recupero
  - 16.10.1. Resilienza, concetti, requisiti aziendali e standard
  - 16.10.2. Soluzioni IT di Resilienza
  - 16.10.3. Gestione della crisi e Governance

## Modulo 17. Sicurezza nel Software

- 17.1. Problemi di sicurezza nel Software
  - 17.1.1. Introduzione al problema della sicurezza nel Software
  - 17.1.2. Vulnerabilità e la loro classificazione
  - 17.1.3. Proprietà di un Software sicuro
  - 17.1.4. Riferimenti
- 17.2. Principi di progettazione per la sicurezza del Software
  - 17.2.1. Introduzione
  - 17.2.2. Principi di progettazione per la sicurezza del Software
  - 17.2.3. Tipologie di S-SDLC
  - 17.2.4. Sicurezza del Software nelle fasi S-SDLC
  - 17.2.5. Metodologie e standard
  - 17.2.6. Riferimenti
- 17.3. Sicurezza del ciclo di vita del Software nelle fasi dei requisiti e di progettazione
  - 17.3.1. Introduzione
  - 17.3.2. Modellazione dell'attacco
  - 17.3.3. Casi di abuso
  - 17.3.4. Ingegneria dei requisiti di sicurezza
  - 17.3.5. Analisi dei rischi: Architettotnico
  - 17.3.6. Pattern di progettazione
  - 17.3.7. Riferimenti
- 17.4. Sicurezza nel ciclo di vita del Software durante le fasi di codifica, collaudo e funzionamento
  - 17.4.1. Introduzione
  - 17.4.2. Test di sicurezza basati sul rischio
  - 17.4.3. Revisione del codice
  - 17.4.4. Test di penetrazione
  - 17.4.5. Operazioni di sicurezza
  - 17.4.6. Revisione esterna
  - 17.4.7. Riferimenti

- 17.5. Codifica sicura di applicazioni I
  - 17.5.1. Introduzione
  - 17.5.2. Pratiche di codifica sicure
  - 17.5.3. Gestione e convalida degli input
  - 17.5.4. Overflow di memoria
  - 17.5.5. Riferimenti
- 17.6. Codifica sicura di applicazioni II
  - 17.6.1. Introduzione
  - 17.6.2. Integers overflow, errori di troncamento e problemi con le conversioni di tipo tra numeri interi
  - 17.6.3. Errori ed eccezioni
  - 17.6.4. Privacy e riservatezza
  - 17.6.5. Programmi privilegiati
  - 17.6.6. Riferimenti
- 17.7. Sicurezza nello sviluppo e nel cloud
  - 17.7.1. Sicurezza dello sviluppo: metodologia e pratica
  - 17.7.2. Modelli PaaS, IaaS, CaaS e SaaS
  - 17.7.3. Sicurezza nel cloud e per i servizi Cloud
- 17.8. Crittografia
  - 17.8.1. Fondamenti di Criptologia
  - 17.8.2. Crittografia simmetrica e asimmetrica
  - 17.8.3. Crittografia a riposo e in transito
- 17.9. Automazione e orchestrazione della sicurezza (SOAR)
  - 17.9.1. Complessità dell'elaborazione manuale: necessità di automatizzare le attività
  - 17.9.2. Prodotti e servizi
  - 17.9.3. Architettura SOAR
- 17.10. Sicurezza nel telelavoro
  - 17.10.1. Necessità e scenari
  - 17.10.2. Prodotti e servizi
  - 17.10.3. Sicurezza del telelavoro

## Modulo 18. Amministrazione di Server Web

- 18.1. Introduzione ai server web
  - 18.1.1. Che cos'è un server web?
  - 18.1.2. Architettura e funzionamento di un server web
  - 18.1.3. Risorse e contenuti in un server web
  - 18.1.4. Server di applicazioni
  - 18.1.5. Server proxy
  - 18.1.6. I principali server web presenti sul mercato
  - 18.1.7. Statistiche di utilizzo del server web
  - 18.1.8. Sicurezza del server web
  - 18.1.9. Bilanciamento del carico nei server web
  - 18.1.10. Riferimenti
- 18.2. Gestione del protocollo HTTP
  - 18.2.1. Funzionamento e struttura
  - 18.2.2. Descrizione delle richieste o request methods
  - 18.2.3. Codici di stato
  - 18.2.4. Intestazioni
  - 18.2.5. Codifica dei contenuti: Pagine di codice
  - 18.2.6. Esecuzione di richieste HTTP su Internet utilizzando un proxy, livehttpheader o metodi simili, analizzando il protocollo utilizzato
- 18.3. Descrizione delle architetture distribuite multi-server
  - 18.3.1. Modello a 3 strati
  - 18.3.2. Tolleranza ai guasti
  - 18.3.3. Condivisione del carico
  - 18.3.4. Archivi dello stato della sessione
  - 18.3.5. Archivi di cache
- 18.4. Internet Information Services (IIS)
  - 18.4.1. Cos'è IIS?
  - 18.4.2. Storia ed evoluzione dell'IIS
  - 18.4.3. Vantaggi e caratteristiche principali di IIS7 e successivi
  - 18.4.4. Architettura IIS7 e successive

- 18.5. Installazione, amministrazione e configurazione di IIS
    - 18.5.1. Premessa
    - 18.5.2. Installazione di Internet Information Services (IIS)
    - 18.5.3. Strumenti di amministrazione di IIS
    - 18.5.4. Creazione, configurazione e amministrazione di siti web
    - 18.5.5. Installazione e gestione delle estensioni di IIS
  - 18.6. Sicurezza avanzata di IIS
    - 18.6.1. Premessa
    - 18.6.2. Autenticazione, autorizzazione e controllo degli accessi in IIS
    - 18.6.3. Configurazione di un sito web sicuro su IIS con SSL
    - 18.6.4. Criteri di sicurezza implementati in IIS 8.x
  - 18.7. Introduzione ad Apache
    - 18.7.1. Che cos'è Apache?
    - 18.7.2. Principali vantaggi di Apache
    - 18.7.3. Caratteristiche principali di Apache
    - 18.7.4. Architettura
  - 18.8. Installazione e configurazione di Apache
    - 18.8.1. Installazione iniziale di Apache
    - 18.8.2. Configurazione di Apache
  - 18.9. Installazione e configurazione dei diversi moduli Apache
    - 18.9.1. Installazione dei moduli Apache
    - 18.9.2. Tipi di moduli
    - 18.9.3. Configurazione sicura di Apache
  - 18.10. Sicurezza avanzata
    - 18.10.1. Autenticazione, autorizzazione e controllo degli accessi
    - 18.10.2. Metodi di autenticazione
    - 18.10.3. Configurazione sicura di Apache con SSL
- Modulo 19. Audit sulla Sicurezza**
- 19.1. Introduzione ai sistemi informatici e al loro audit
    - 19.1.1. Introduzione ai sistemi informatici e al ruolo dell'auditing informatico
    - 19.1.2. Definizioni di Audit IT e controllo interno IT
    - 19.1.3. Funzioni e obiettivi dell'audit IT
    - 19.1.4. Differenze tra controllo interno e audit IT
  - 19.2. Controlli interni dei Sistemi Informatici
    - 19.2.1. Organigramma funzionale di un centro di elaborazione dati
    - 19.2.2. Classificazione dei controlli sui sistemi informatici
    - 19.2.3. La Regola d'Oro
  - 19.3. Il processo e le fasi dell'audit dei Sistemi Informatici
    - 19.3.1. Valutazione del rischio (RRA) e altre metodologie di audit IT
    - 19.3.2. Esecuzione di un audit dei Sistemi Informatici: Fasi di audit
    - 19.3.3. Competenze chiave dell'auditor di Sistemi Informatici
  - 19.4. Audit tecnico di sicurezza di sistemi e reti
    - 19.4.1. Audit tecnici di sicurezza: Test di intrusione, Concetti preliminari
    - 19.4.2. Audit di sicurezza del sistema: Strumenti di supporto
    - 19.4.3. Audit sulla sicurezza della rete: Strumenti di supporto
  - 19.5. Audit tecnico della sicurezza di Internet e dei dispositivi mobili
    - 19.5.1. Audit sulla sicurezza di Internet: Strumenti di supporto
    - 19.5.2. Audit di sicurezza dei dispositivi mobili: Strumenti di supporto
    - 19.5.3. Allegato 1. Struttura del rapporto esecutivo e del rapporto tecnico
    - 19.5.4. Allegato 2. Inventario degli strumenti
    - 19.5.5. Allegato 3. Metodologie
  - 19.6. Sistema di gestione della sicurezza delle informazioni
    - 19.6.1. Sicurezza IS: proprietà e fattori di influenza
    - 19.6.2. Rischio d'impresa e gestione del rischio: implementazione dei controlli
    - 19.6.3. Sistema di gestione della sicurezza delle informazioni (ISMS): concetto e fattori critici di successo
    - 19.6.4. ISMS - Modello PDCA
    - 19.6.5. ISMS ISO-IEC 27001: contesto organizzativo
    - 19.6.6. Contesto dell'Organizzazione
    - 19.6.7. Leadership
    - 19.6.8. Pianificazione
    - 19.6.9. Supporto
    - 19.6.10. Operazione
    - 19.6.11. Valutazione della prestazione
    - 19.6.12. Miglioramento
    - 19.6.13. Allegato alla ISO 27001/ISO-IEC 27002: obiettivi e controlli
    - 19.6.14. Audit ISMS

- 19.7. Realizzazione dell'audit
  - 19.7.1. Procedure
  - 19.7.2. Tecniche
- 19.8. Tracciabilità
  - 19.8.1. Metodologie
  - 19.8.2. Analisi
- 19.9. Tutela
  - 19.9.1. Tecniche
  - 19.9.2. Risultati
- 19.10. Reporting e presentazione delle prove
  - 19.10.1. Tipi di report
  - 19.10.2. Analisi dei dati
  - 19.10.3. Presentazione delle prove

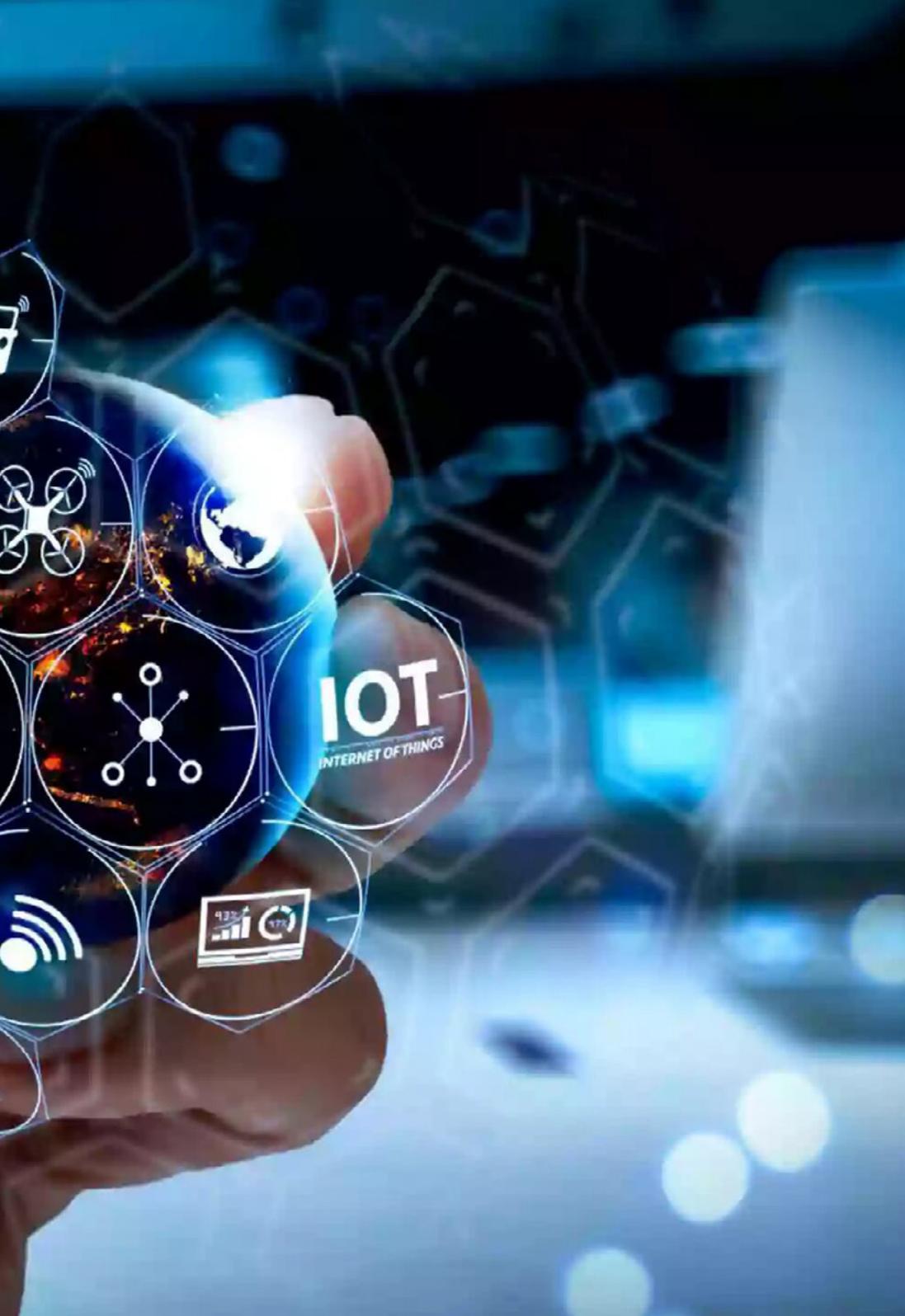
## Modulo 20. Sicurezza nelle Applicazioni Online

- 20.1. Vulnerabilità e problemi di sicurezza nelle applicazioni online
  - 20.1.1. Introduzione alla sicurezza nelle applicazioni online
  - 20.1.2. Vulnerabilità di sicurezza nella progettazione di applicazioni web
  - 20.1.3. Vulnerabilità di sicurezza nell'implementazione di applicazioni
  - 20.1.4. Vulnerabilità di sicurezza nella distribuzione di applicazioni web
  - 20.1.5. Elenchi ufficiali di vulnerabilità di sicurezza
- 20.2. Politiche e standard per la sicurezza delle applicazioni online
  - 20.2.1. I pilastri della sicurezza delle applicazioni online
  - 20.2.2. Politica di sicurezza
  - 20.2.3. Sistema di gestione della sicurezza delle informazioni
  - 20.2.4. Ciclo di vita dello sviluppo sicuro del Software
  - 20.2.5. Standard di sicurezza delle applicazioni
- 20.3. Sicurezza nella progettazione di applicazioni web
  - 20.3.1. Introduzione alla sicurezza delle applicazioni web
  - 20.3.2. Sicurezza nella progettazione di applicazioni web
- 20.4. Test di sicurezza e protezione online delle applicazioni web
  - 20.4.1. Analisi e test della sicurezza delle applicazioni web
  - 20.4.2. Sicurezza nell'implementazione e nella produzione di applicazioni web



- 20.5. Sicurezza dei servizi web
  - 20.5.1. Introduzione alla sicurezza dei servizi web
  - 20.5.2. Funzioni e tecnologie di sicurezza dei servizi web
- 20.6. Test di sicurezza e protezione online dei servizi web
  - 20.6.1. Valutazione della sicurezza dei servizi web
  - 20.6.2. Protezione online: Firewall e gateway XML
- 20.7. Hacking etico, malware e forensic
  - 20.7.1. Hacking etico
  - 20.7.2. Analisi del malware
  - 20.7.3. Analisi Forense
- 20.8. Le migliori pratiche per garantire la sicurezza delle applicazioni
  - 20.8.1. Manuale di pratiche corrette per lo sviluppo di applicazioni online
  - 20.8.2. Manuale di pratiche corrette per l'implementazione di applicazioni online
- 20.9. Errori comuni che minano la sicurezza delle applicazioni
  - 20.9.1. Errori di sviluppo comuni
  - 20.9.2. Errori comuni nell'hosting
  - 20.9.3. Errori comuni nella produzione

“ *Un programma completo che ti porterà a padroneggiare l'area dei Big Data e diventare un architetto di strategia aziendale di successo*”



# 04

## Obiettivi didattici

Il Master Specialistico in Ingegneria e Qualità del Software mira a formare professionisti altamente qualificati in progettazione, sviluppo e gestione di sistemi Software di alta qualità. Con particolare attenzione alla qualità e alla gestione strategica dei progetti tecnologici, il programma promuove anche la capacità di adattamento ai rapidi sviluppi del settore. In questo modo, gli studenti non solo saranno preparati per affrontare le sfide del futuro, ma saranno anche in grado di guidare l'innovazione nel campo del Software.



“

*Trasforma le idee in soluzioni tecnologiche efficaci con l'arte dell'ingegneria del Software"*



## Obiettivi generali

---

- ♦ Sviluppare competenze avanzate in progettazione, sviluppo e manutenzione di sistemi Software complessi e scalabili, applicando le migliori pratiche e metodologie di ingegneria del Software
- ♦ Specializzarsi nella garanzia della qualità del Software, ottenendo strumenti e tecniche per garantire l'affidabilità, la sicurezza e le prestazioni delle soluzioni tecnologiche
- ♦ Promuovere la leadership nella gestione dei progetti tecnologici, sviluppando competenze nella gestione di team multidisciplinari, pianificazione strategica e processo decisionale in ambienti dinamici
- ♦ Promuovere la capacità di adattamento ai rapidi progressi tecnologici, attraverso la specializzazione in nuovi strumenti, tecniche e tendenze che consentono di rimanere all'avanguardia nell'ingegneria del Software
- ♦ Sviluppare competenze nella gestione della qualità in tutto il ciclo di vita del Software, dalla pianificazione iniziale alla manutenzione e al miglioramento continuo dei sistemi
- ♦ Rafforzare le capacità di comunicazione e lavoro di squadra, essenziali per collaborare efficacemente con diversi stakeholder, gestire le aspettative e garantire il successo dei progetti tecnologici



*Migliora le tue capacità e diventa un leader nella creazione di soluzioni tecnologiche all'avanguardia"*





## Obiettivi specifici

---

### Modulo 1. Qualità del Software: Livelli di sviluppo TRL

- ♦ Comprendere i diversi livelli di maturità tecnologica e la loro relazione con la qualità del Software
- ♦ Valutare lo sviluppo del Software in ogni fase del TRL e il suo impatto sulla qualità finale del prodotto

### Modulo 2. Sviluppo di Progetti Software: Documentazione funzionale e tecnica

- ♦ Sviluppare le competenze per creare documentazione funzionale e tecnica chiara e dettagliata nei progetti Software
- ♦ Analizzare l'importanza di una documentazione accurata per la gestione dei progetti e la qualità del Software

### Modulo 3. Testing di Software: Automazione dei test

- ♦ Sviluppare competenze per progettare ed eseguire test automatizzati su applicazioni Software
- ♦ Implementare soluzioni di *testing* efficienti utilizzando strumenti di automazione dei test

### Modulo 4. Metodologie di Gestione dei Progetti Software: Metodologie Waterfall vs. Metodologie Agili

- ♦ Analizzare le differenze tra le metodologie Waterfall e Agile nella gestione dei progetti Software
- ♦ Valutare i vantaggi e i limiti di ogni metodologia in base al tipo di progetto

### Modulo 5. TDD (*Test Driven Development*): Progettazione del Software guidata dai test

- ♦ Sviluppare le capacità di scrivere test unitari prima di scrivere il codice di produzione
- ♦ Migliorare la qualità del Software implementando TDD nel processo di sviluppo

### Modulo 6. DevOps: Gestione della Qualità del Software

- ♦ Esplorare il concetto di DevOps e il suo impatto sul miglioramento continuo della qualità del Software
- ♦ Imparare a integrare le pratiche di sviluppo e operazioni per raggiungere un ciclo vita di Software più agile ed efficiente

### Modulo 7. DevOps e Integrazione Continua: Soluzioni pratiche avanzate nello sviluppo di Software

- ♦ Approfondire le tecniche avanzate di integrazione continua all'interno del quadro di DevOps
- ♦ Implementare soluzioni pratiche di integrazione continua per automatizzare il processo di sviluppo e distribuzione del Software

### Modulo 8. Progettazione di Database (DB): Normalizzazione e Rendimento, Qualità del Software

- ♦ Analizzare i principi di progettazione dei database, compresa la standardizzazione e l'ottimizzazione delle prestazioni
- ♦ Comprendere come la corretta progettazione di database contribuisce alla qualità del Software

### Modulo 9. Progettazione di architetture scalabili: l'Architettura nel Ciclo di Vita del Software

- ♦ Approfondire i principi di progettazione delle architetture scalabili e il loro impatto sulla qualità e le prestazioni del Software
- ♦ Valutare diversi modelli di architettura per applicazioni Software scalabili

#### **Modulo 10. Criteri di qualità ISO, IEC 9126: Metriche della Qualità del Software**

- ♦ Comprendere i criteri di qualità del Software secondo questi standard e come applicarli
- ♦ Implementare metriche di qualità per valutare e migliorare continuamente le applicazioni Software

#### **Modulo 11. Metodologie, sviluppo e qualità nell'Ingegneria dei Software**

- ♦ Approfondire le metodologie più utilizzate nell'Ingegneria del Software e la sua relazione con la Qualità
- ♦ Sviluppare un approccio integrato che combini sviluppo, test e qualità in progetti Software

#### **Modulo 12. Gestione dei progetti Software**

- ♦ Sviluppare competenze nella gestione dei progetti Software, dalla pianificazione alla realizzazione
- ♦ Gestire le risorse, i tempi e i rischi associati ai progetti di sviluppo Software

#### **Modulo 13. Piattaforme di sviluppo Software**

- ♦ Comprendere le diverse piattaforme di sviluppo Software e le loro caratteristiche
- ♦ Valutare le piattaforme di sviluppo in base alle loro capacità, flessibilità e compatibilità con diversi progetti

#### **Modulo 14. Informatica client web**

- ♦ Analizzare come viene eseguita la computing lato client nello sviluppo di applicazioni web
- ♦ Sviluppare applicazioni che sfruttano il computing online per migliorare l'interazione e le prestazioni





### **Modulo 15. Informatica per Server Web**

- ◆ Esplorare le tecnologie e tecniche utilizzate per il web server computing
- ◆ Comprendere la gestione dei dati, la logica di business e la gestione degli utenti sul server

### **Modulo 16. Gestione della sicurezza**

- ◆ Valutare i rischi di sicurezza nelle applicazioni e applicare misure preventive
- ◆ Implementare controlli di sicurezza in tutte le fasi del ciclo di vita del Software

### **Modulo 17. Sicurezza nel Software**

- ◆ Esplorare le migliori pratiche di sicurezza nello sviluppo Software
- ◆ Analizzare le vulnerabilità più comuni nel Software e imparare a mitigarle

### **Modulo 18. Amministrazione di Server Web**

- ◆ Comprendere il ruolo dei server web nello sviluppo e nella distribuzione delle applicazioni
- ◆ Sviluppare competenze nella gestione e nella manutenzione di server web

### **Modulo 19. Audit sulla Sicurezza**

- ◆ Valutare la sicurezza dei sistemi attraverso audit e penetration test
- ◆ Implementare processi di audit continui per migliorare la sicurezza del Software

### **Modulo 20. Sicurezza nelle Applicazioni Online**

- ◆ Implementare soluzioni per proteggere le applicazioni online da minacce esterne e interne
- ◆ Stabilire politiche di sicurezza e audit per garantire l'integrità delle applicazioni online

# 05

## Opportunità professionali

Lo studente sarà altamente qualificato per assumere ruoli di leadership in sviluppo, implementazione e gestione di Software di alta qualità. Grazie alla sua specializzazione avanzata in aree chiave come l'architettura del sSoftware, la sicurezza delle applicazioni online, la gestione dei progetti tecnologici e le metodologie agili, sarà in grado di guidare i team di sviluppo. Inoltre, la preparazione consentirà di occupare posizioni dirigenziali in progetti Software, mentre la sua capacità di innovare e guidare team multidisciplinari lo renderà in grado di affrontare le sfide dell'ambiente digitale e contribuire significativamente al successo di qualsiasi organizzazione.



“

*TECH ti offre l'opportunità di realizzare i tuoi sogni nella disciplina più appassionante, che trasforma le idee in prodotti tangibili in grado di migliorare la vita delle persone"*

### Profilo dello studente

Il profilo dello studente del Master Specialistico in Ingegneria e Qualità del Software è orientato a istruire professionisti altamente qualificati, capaci di guidare e gestire progetti tecnologici ad alto impatto. Garantendo qualità, sicurezza ed efficienza in tutte le fasi dello sviluppo del Software, padroneggerà sia le metodologie agili che quelle tradizionali. Inoltre, sarà in grado di progettare e sviluppare sistemi software scalabili, efficienti e sicuri, applicando standard internazionali di qualità e metodologie avanzate come DevOps e integrazione continua.

*Diventa un esperto che garantisce il successo delle aziende nella più grande università digitale del mondo.*

- ♦ **Sicurezza di Software e sistemi:** Competenza nell'implementazione di pratiche avanzate di sicurezza, tra cui la protezione dei dati e la gestione delle vulnerabilità nelle applicazioni online
- ♦ **Garanzia della qualità del Software:** Capacità di applicare standard internazionali (ISO, IEC 9126) e strumenti di test automatizzati per garantire l'affidabilità e le prestazioni del Software
- ♦ **Sviluppo di architetture scalabili:** Capacità di progettare e costruire sistemi Software che possono crescere e adattarsi alle richieste del mercato senza compromettere la qualità o la sicurezza
- ♦ **Integrazione continua e DevOps:** Capacità di implementare e gestire processi di integrazione continua, garantendo una consegna efficiente e senza interruzioni di nuove funzionalità del Software





Dopo aver completato il Master Specialistico potrai utilizzare le tue conoscenze e competenze nei seguenti ruoli:

1. **Direttore della Tecnologia (CTO):** Responsabile della direzione strategica della tecnologia in un'azienda, guidando team di sviluppo e supervisionando l'implementazione di soluzioni tecnologiche innovative.
2. **Responsabile della Qualità del Software:** Responsabile della supervisione e dell'assicurazione che i processi e i prodotti Software siano conformi agli standard di qualità stabiliti, guidando le iniziative di miglioramento continuo e test del Software
3. **Architetto del Software:** Progettista principale della struttura e dell'architettura dei sistemi Software complessi, assicurando che siano scalabili, sicuri ed efficienti
4. **Leader di Progetti Software:** Responsabile di pianificazione, esecuzione e consegna di progetti Software, gestendo team multidisciplinari e assicurando che i progetti siano completati entro il termine, il budget e con gli standard di qualità appropriati
5. **Specialista in Sicurezza Informatica:** Responsabile della protezione di applicazioni, infrastrutture e dati contro le minacce informatiche, implementando strategie e politiche di sicurezza efficaci
6. **Software Security Auditor:** Effettua audit approfonditi per identificare vulnerabilità nelle applicazioni e nei sistemi, proponendo miglioramenti e soluzioni per garantire la sicurezza del Software

“

*Se vuoi fare la differenza nel mondo digitale, scegli questo percorso che ti specializza come esperto nella creazione di Software di qualità”*

# 06

## Metodologia di studio

TECH è la prima università al mondo che combina la metodologia dei **case studies** con il **Relearning**, un sistema di apprendimento 100% online basato sulla ripetizione diretta.

Questa strategia dirompente è stata concepita per offrire ai professionisti l'opportunità di aggiornare le conoscenze e sviluppare competenze in modo intensivo e rigoroso. Un modello di apprendimento che pone lo studente al centro del processo accademico e gli conferisce tutto il protagonismo, adattandosi alle sue esigenze e lasciando da parte le metodologie più convenzionali.



“

*TECH ti prepara ad affrontare nuove sfide in ambienti incerti e a raggiungere il successo nella tua carriera"*

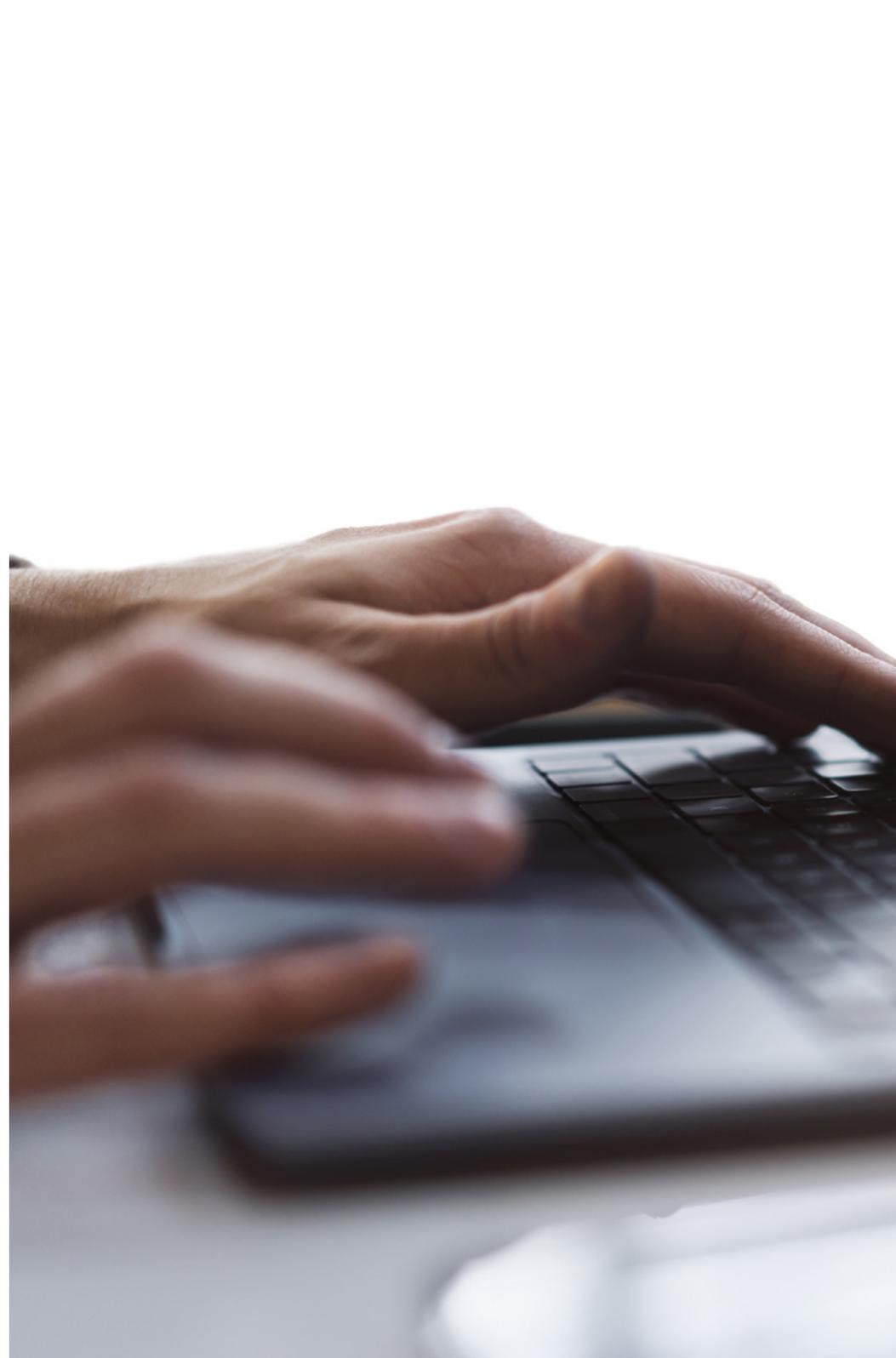
## Lo studente: la priorità di tutti i programmi di TECH

Nella metodologia di studio di TECH lo studente è il protagonista assoluto. Gli strumenti pedagogici di ogni programma sono stati selezionati tenendo conto delle esigenze di tempo, disponibilità e rigore accademico che, al giorno d'oggi, non solo gli studenti richiedono ma le posizioni più competitive del mercato.

Con il modello educativo asincrono di TECH, è lo studente che sceglie il tempo da dedicare allo studio, come decide di impostare le sue routine e tutto questo dalla comodità del dispositivo elettronico di sua scelta. Lo studente non deve frequentare lezioni presenziali, che spesso non può frequentare. Le attività di apprendimento saranno svolte quando si ritenga conveniente. È lo studente a decidere quando e da dove studiare.

“

*In TECH NON ci sono lezioni presenziali  
(che poi non potrai mai frequentare)”*



### I piani di studio più completi a livello internazionale

TECH si caratterizza per offrire i percorsi accademici più completi del panorama universitario. Questa completezza è raggiunta attraverso la creazione di piani di studio che non solo coprono le conoscenze essenziali, ma anche le più recenti innovazioni in ogni area.

Essendo in costante aggiornamento, questi programmi consentono agli studenti di stare al passo con i cambiamenti del mercato e acquisire le competenze più apprezzate dai datori di lavoro. In questo modo, coloro che completano gli studi presso TECH ricevono una preparazione completa che fornisce loro un notevole vantaggio competitivo per avanzare nelle loro carriere.

Inoltre, potranno farlo da qualsiasi dispositivo, pc, tablet o smartphone.

“

*Il modello di TECH è asincrono, quindi ti permette di studiare con il tuo pc, tablet o smartphone dove, quando e per quanto tempo vuoi”*

## Case studies o Metodo Casistico

Il Metodo Casistico è stato il sistema di apprendimento più usato nelle migliori facoltà del mondo. Sviluppato nel 1912 per consentire agli studenti di Giurisprudenza non solo di imparare le leggi sulla base di contenuti teorici, ma anche di esaminare situazioni complesse reali. In questo modo, potevano prendere decisioni e formulare giudizi di valore fondati su come risolverle. Nel 1924 fu stabilito come metodo di insegnamento standard ad Harvard.

Con questo modello di insegnamento, è lo studente stesso che costruisce la sua competenza professionale attraverso strategie come il *Learning by doing* o il *Design Thinking*, utilizzate da altre istituzioni rinomate come Yale o Stanford.

Questo metodo, orientato all'azione, sarà applicato lungo tutto il percorso accademico che lo studente intraprende insieme a TECH. In questo modo, affronterà molteplici situazioni reali e dovrà integrare le conoscenze, ricercare, argomentare e difendere le sue idee e decisioni. Tutto ciò con la premessa di rispondere al dubbio di come agirebbe nel posizionarsi di fronte a specifici eventi di complessità nel suo lavoro quotidiano.



## Metodo Relearning

In TECH i *case studies* vengono potenziati con il miglior metodo di insegnamento 100% online: il *Relearning*.

Questo metodo rompe con le tecniche di insegnamento tradizionali per posizionare lo studente al centro dell'equazione, fornendo il miglior contenuto in diversi formati. In questo modo, riesce a ripassare e ripete i concetti chiave di ogni materia e impara ad applicarli in un ambiente reale.

In questa stessa linea, e secondo molteplici ricerche scientifiche, la ripetizione è il modo migliore per imparare. Ecco perché TECH offre da 8 a 16 ripetizioni di ogni concetto chiave in una stessa lezione, presentata in modo diverso, con l'obiettivo di garantire che la conoscenza sia completamente consolidata durante il processo di studio.

*Il Relearning ti consentirà di apprendere con meno sforzo e più rendimento, coinvolgendoti maggiormente nella specializzazione, sviluppando uno spirito critico, difendendo gli argomenti e contrastando opinioni: un'equazione diretta al successo.*



## Un Campus Virtuale 100% online con le migliori risorse didattiche

Per applicare efficacemente la sua metodologia, TECH si concentra sul fornire agli studenti materiali didattici in diversi formati: testi, video interattivi, illustrazioni, mappe della conoscenza, ecc. Tutto ciò progettato da insegnanti qualificati che concentrano il lavoro sulla combinazione di casi reali con la risoluzione di situazioni complesse attraverso la simulazione, lo studio dei contesti applicati a ogni carriera e l'apprendimento basato sulla ripetizione, attraverso audio, presentazioni, animazioni, immagini, ecc.

Le ultime prove scientifiche nel campo delle Neuroscienze indicano l'importanza di considerare il luogo e il contesto in cui si accede ai contenuti prima di iniziare un nuovo apprendimento. Poter regolare queste variabili in modo personalizzato favorisce che le persone possano ricordare e memorizzare nell'ippocampo le conoscenze per conservarle a lungo termine. Si tratta di un modello denominato *Neurocognitive context-dependent e-learning*, che viene applicato in modo consapevole in questa qualifica universitaria.

Inoltre, anche per favorire al massimo il contatto tra mentore e studente, viene fornita una vasta gamma di possibilità di comunicazione, sia in tempo reale che differita (messaggistica interna, forum di discussione, servizio di assistenza telefonica, e-mail di contatto con segreteria tecnica, chat e videoconferenza).

Inoltre, questo completo Campus Virtuale permetterà agli studenti di TECH di organizzare i loro orari di studio in base alla loro disponibilità personale o agli impegni lavorativi. In questo modo avranno un controllo globale dei contenuti accademici e dei loro strumenti didattici, il che attiva un rapido aggiornamento professionale.



*La modalità di studio online di questo programma ti permetterà di organizzare il tuo tempo e il tuo ritmo di apprendimento, adattandolo ai tuoi orari"*

### L'efficacia del metodo è giustificata da quattro risultati chiave:

1. Gli studenti che seguono questo metodo non solo raggiungono l'assimilazione dei concetti, ma sviluppano anche la loro capacità mentale, attraverso esercizi che valutano situazioni reali e l'applicazione delle conoscenze.
2. L'apprendimento è solidamente fondato su competenze pratiche che permettono allo studente di integrarsi meglio nel mondo reale.
3. L'assimilazione di idee e concetti è resa più facile ed efficace, grazie all'uso di situazioni nate dalla realtà.
4. La sensazione di efficienza dello sforzo investito diventa uno stimolo molto importante per gli studenti, che si traduce in un maggiore interesse per l'apprendimento e in un aumento del tempo dedicato al corso.

## La metodologia universitaria più apprezzata dagli studenti

I risultati di questo innovativo modello accademico sono riscontrabili nei livelli di soddisfazione globale degli studenti di TECH.

La valutazione degli studenti sulla qualità dell'insegnamento, la qualità dei materiali, la struttura del corso e i suoi obiettivi è eccellente. A questo proposito, l'università è considerata la migliore per i suoi studenti nella piattaforma di valutazione Global score, ottenendo un 4,9 su 5.

*Accedi ai contenuti di studio da qualsiasi dispositivo con connessione a Internet (computer, tablet, smartphone) grazie al fatto che TECH è aggiornato sull'avanguardia tecnologica e pedagogica.*

*Potrai imparare dai vantaggi dell'accesso a ambienti di apprendimento simulati e dall'approccio di apprendimento per osservazione, ovvero Learning from an expert.*



In questo modo, il miglior materiale didattico sarà disponibile, preparato con attenzione:



#### Materiale di studio

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

Questi contenuti sono poi applicati al formato audiovisivo che supporterà la nostra modalità di lavoro online, impiegando le ultime tecnologie che ci permettono di offrirti una grande qualità per ogni elemento che metteremo al tuo servizio.



#### Capacità e competenze pratiche

I partecipanti svolgeranno attività per sviluppare competenze e abilità specifiche in ogni area tematica. Pratiche e dinamiche per acquisire e sviluppare le competenze e le abilità che uno specialista deve possedere nel mondo globalizzato in cui viviamo.



#### Riepiloghi interattivi

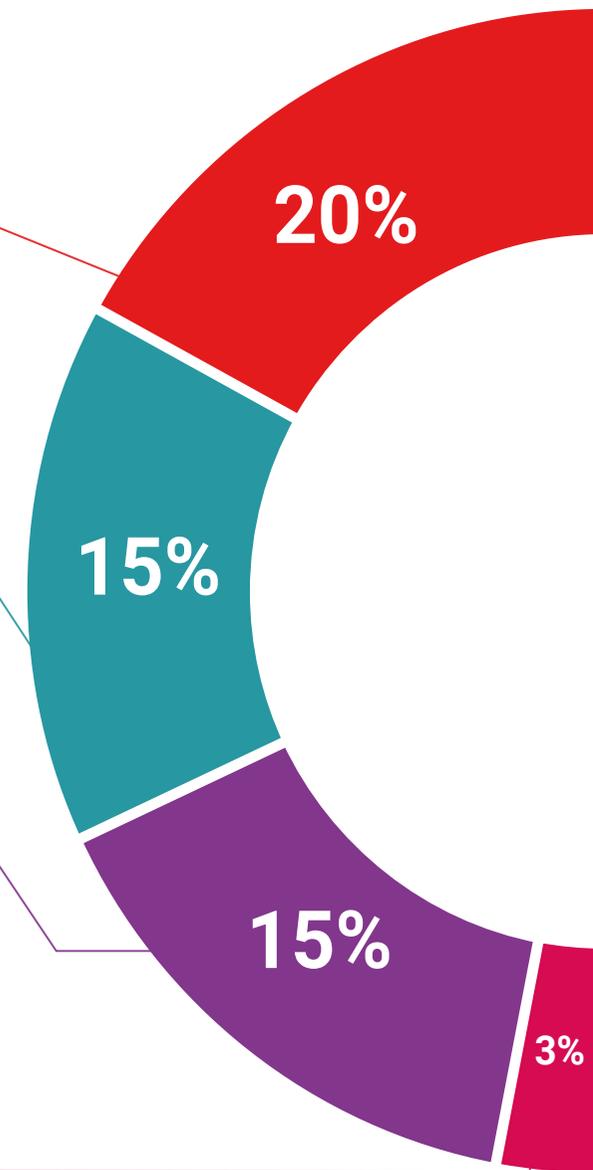
Presentiamo i contenuti in modo accattivante e dinamico tramite strumenti multimediali che includono audio, video, immagini, diagrammi e mappe concettuali per consolidare la conoscenza.

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



#### Letture complementari

Articoli recenti, documenti di consenso, guide internazionali... Nella nostra libreria virtuale avrai accesso a tutto ciò di cui hai bisogno per completare la tua formazione.





#### Case Studies

Completerai una selezione dei migliori *case studies* in materia. Casi presentati, analizzati e monitorati dai migliori specialisti del panorama internazionale.



#### Testing & Retesting

Valutiamo e rivalutiamo periodicamente le tue conoscenze durante tutto il programma. Tutto questo, su 3 dei 4 livelli della Piramide di Miller.



#### Master class

Esistono prove scientifiche su l'utilità d'osservazione di terzi esperti. Il cosiddetto *Learning from an Expert* rafforza le conoscenze e i ricordi, e genera sicurezza nel futuro processo decisionale.



#### Guide di consultazione veloce

TECH offre i contenuti più rilevanti del corso in formato di scheda o guida di facile consultazione. Un modo sintetico, pratico ed efficace per aiutare lo studente a progredire nel suo apprendimento.



# 07

## Personale docente

La direzione e l'insegnamento di questo Master Specialistico in Ingegneria e Qualità del Software è curata da un team di esperti di ingegneria con molti anni di esperienza nella gestione e nello sviluppo di progetti tecnici e specializzati. Il loro bagaglio professionale apporta a questa qualifica un plus di qualità che si rifletterà in una migliore contestualizzazione del contenuto da parte dello studente, nonché l'implementazione nell'esperienza accademica di casi pratici reali e simulati, ma sempre orientati a offrire un programma 100% online dinamico, all'avanguardia e basato sulla realtà immediata del settore.



“

*Partecipa alla gara più richiesta di esperti in ingegneria e qualità del Software e sfrutta questa opportunità per diventare uno di loro"*

## Direttore Ospite Internazionale

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

Ha ricoperto ruoli di alto livello nella sua carriera, tra cui quello di Chief Architect per le Soluzioni del Settore Pubblico presso Intel Corporation, dove ha promosso business moderni, processi e tecnologie per clienti, soci 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 network basato nel Software as a Service (SaaS), utilizzando le tecnologie di Big Data e Web 2.0.

Inoltre, ha lavorato in altre aziende, come Senior Director of Engineering, presso Dell Technologies, dove ha guidato la Business Unit Big Data nel Cloud, guidando i team negli Stati Uniti e in Cina per la gestione di progetti su larga scala e la ristrutturazione delle divisioni aziendali per il successo della loro integrazione. Ha lavorato anche come Direttore delle Tecnologie dell'Informazione (Chief Information Officer) presso XanGo, dove ha gestito progetti quali il supporto di 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 cibersicurezza, l'Intelligenza Artificiale Generativa, lo sviluppo di software, la tecnologia di rete, lo sviluppo cloud nativo e l'ecosistema dei container. Si tratta di conoscenze che ha condiviso attraverso il podcast e la newsletter settimanale "Embracing Digital Transformation", che ha prodotto e presentato personalmente, aiutando le organizzazioni a navigare con successo nella trasformazione digitale sfruttando le persone, i processi e la tecnologia.



## Dott. Pulsipher, Darren

---

- Architetto Responsabile di 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 Director di ingegneria presso Dell Technologies, Arkansas
- Direttore delle Tecnologie dell'Informazione (*Chief Information Officer*) presso XanGo, Utah
- Architetto Senior presso Cadence Design Systems, California
- Senior Project Process Manager presso Lucent Technologies, California
- Ingegnere del Software presso Cemax-Icon, California
- Ingegnere 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



*Grazie a TECH potrai apprendere dai migliori professionisti del mondo"*

## Direzione



### Dott. Molina Molina, Jerónimo

- ◆ Responsabile dell'Intelligenza Artificiale presso Helphone
- ◆ AI Engineer & Software Architect presso NASSAT, Internet Satellite in Movimento
- ◆ Consulente Senior presso Hexa Ingeniero
- ◆ Introduttore di Intelligenza Artificiale (ML e CV)
- ◆ Esperto di Soluzioni Basate sull'Intelligenza Artificiale nei settori della Computer Vision, ML/DL e NLP
- ◆ Esperto Universitario in Creazione e Sviluppo di Imprese presso Bancaixa e Fundeun
- ◆ Ingegnere Informatico presso l'Università di Alicante
- ◆ Master in Intelligenza Artificiale conseguito presso l'Università Cattolica di Ávila
- ◆ MBA Executive presso il Foro Europeo Business School

## Personale docente

### Dott.ssa Rodríguez Míguez, Cándida

- ◆ Junior Application Developer presso Getronics
- ◆ Co-fondatrice e City Leader della rete Galicia AI
- ◆ Ingegnere del Software Junior presso Indra
- ◆ Sviluppatrice web presso EDISA
- ◆ Laurea in Ingegneria Informatica presso l'Università di Vigo
- ◆ Master Universitario in Ingegneria Informatica presso l'Università di Vigo

### Dott. Pi Morell, Oriol

- ◆ Analista Funzionale presso Fihoca
- ◆ Product Owner di Hosting e posta elettronica in CDmon
- ◆ Analista Funzionale e Software Engineer in Atmira e CapGemini
- ◆ Docente in Capgemini, Forma Capgemini e Atmira
- ◆ Laurea in Ingegneria Tecnica in Gestione Informatica presso l'Università Autonoma di Barcellona
- ◆ Master in Intelligenza Artificiale conseguito presso l'Università Cattolica di Ávila
- ◆ MBA in Gestione e Amministrazione d'Impresa presso IMF Smart Education
- ◆ Master in Gestione dei Sistemi di Informazione con IMF Smart Education
- ◆ Corso di Specializzazione in Design Patterns dell'Universitat Oberta de Catalunya (UOC)

**Dott. Martínez Calvo, Francisco Javier**

- ♦ Ingegnere Tecnico Industriale Specializzato in Elettricità ed Elettronica
- ♦ Tecnico Software presso HEXA Ingenieros
- ♦ Senior Developer . Net/Solution Architect. Net su Everis
- ♦ Analista/Architetto *Software* presso LaLiga
- ♦ Ingegnere On-site di Microsoft presso BBVA
- ♦ Consulente Tecnico-Informatico *Freelance*
- ♦ Formatore in Visual Studio, SqlServer, CCNA (Routers y Switch Cisco), Programmazione Web PHP y .Net presso vari centri (Salesianos, Maforem, Dreamsoft)
- ♦ Ingegnere Tecnico Industriale Specializzato in Elettricità ed Elettronica Industriale
- ♦ Master in Informatica. NET, MCAD
- ♦ Master Eidos in Programmazione Avanzata, livello Esperto
- ♦ Master Web con Certificazioni Dreamweaver, Fireworks, Flash e ActionScript, Versioni MX

**Dott. Tenrero Morán, Marcos**

- ♦ Ingegnere DevOps presso Allot Communications
- ♦ Manager di Gestione del Ciclo di Vita delle Applicazioni presso Cegid Meta4
- ♦ Ingegnere dell'Automazione QA presso Cegid Meta4
- ♦ Master in Sviluppo di Applicazioni Professionali per Android presso l'Università Galileo, Guatemala
- ♦ Master in Sviluppo di Servizi Cloud, Node.js, JavaScript, HTML5 presso l'Università Politecnica di Madrid
- ♦ Sviluppo Web con Angular-CLI (4), Ionic e Node.js, Meta4 presso l'Università Rey Juan Carlos
- ♦ Laurea in Ingegneria dei Computer presso l'Università Rey Juan Carlos

**Dott.ssa Acebes Tamargo, Patricia**

- ♦ Consulente specializza in Big Data
- ♦ Dipartimento Operativo, lavora con Elasticsearch e Kivana a Sirt
- ♦ Ricercatrice online di *Human Factor* e *AI Applications* presso CTIC Centro Tecnológico
- ♦ Ricercatrice online di Unità di Business presso CTIC Centro Tecnológico
- ♦ Dipartimento di Salute Digitale e Invecchiamento Attivo presso CTIC Centro Tecnológico
- ♦ Dipartimento di Data Science presso CTIC Centro Tecnológico
- ♦ Dottorato in Informatica in Intelligenza Artificiale presso l'Università Politecnica di Valencia
- ♦ Laurea in Economia presso l'Università di Oviedo
- ♦ Master in Analisi dei Dati, UCJC
- ♦ Master in Ricerca sull'Intelligenza Artificiale presso la UNED
- ♦ Master in *Blockchain, Smart Contracts* e Criptovalute presso l'Università di Alcalá
- ♦ Corso Post-laurea in Ingegneria *Blockchain* presso EADA
- ♦ Master Universitario in Economia, Strumenti, Analisi Economica presso l'Università di Oviedo
- ♦ Master in Fiscalità presso il Collegio degli Economisti

# 08 Titolo

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



“

*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 e Qualità del Software** 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 Specialistico** 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 Specialistico in Ingegneria e Qualità del Software**

Modalità: **online**

Durata: **2 anni**



\*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.



## Master Specialistico Ingegneria e Qualità del Software

- » Modalità: online
- » Durata: 2 anni
- » Titolo: TECH Università Tecnologica
- » Orario: a tua scelta
- » Esami: online

# Master Specialistico

## Ingegneria e Qualità del Software

