

# **Programmazione dipartimento di Informatica**

## **Indirizzo Tecnico**

A.S. 2023-2024



IIS G. Marconi, Civitavecchia

# Introduzione

Il presente documento recepisce le Linee Guida Ministeriali per il passaggio al Nuovo Ordinamento (D.P.R. 15 marzo 2010, articolo 8, comma 3) e in particolare le Linee Guida per gli Istituti Tecnici, settore Tecnologico, indirizzo Informatica e telecomunicazioni, articolazione Informatica.

Le linee guida vengono declinate in base alle specifiche caratteristiche della scuola, in accordo con il PTOF 2022-2025 del nostro Istituto, quali ad esempio: Obiettivi da raggiungere, attrezzature dei laboratori, software e hardware messi a disposizione della scuola.

Nella presente programmazione si vuol fare inoltre riferimento alle competenze ed il know-how dei docenti di questa scuola, così come emerse durante gli anni e la riunione di dipartimento di inizio anno.

Per questo motivo ogni disciplina, per ogni anno, è divisa nelle seguenti sezioni:

**Finalità:** finalità generali della disciplina in quell'anno scolastico

**Obiettivi:** elenco degli Obiettivi da raggiungere alla fine dell'anno

**Contenuti:** elenco dettagliato dei contenuti della disciplina

**Esperienze e approfondimenti:** esperienze di laboratorio o in altre forme per consolidare l'apprendimento dei contenuti, con pratiche consolidate all'interno della scuola, anche in base ai principi dell'imparare facendo; in questa sezione sono anche consigliati possibili approfondimenti, da valutare in base alle proprie competenze e all'interesse della classe

**Strumenti di lavoro:** strumenti messi a disposizione dalla scuola, sia in termini di strumentazione hardware o software, sia in competenze degli altri docenti.

**Modalità di lavoro:** le modalità di lavoro consigliate in base ai contenuti e agli strumenti di laboratorio

**Tipologie di verifica:** tipologie di verifica consigliati per la valutazione degli studenti

**Criteri minimi per l'attribuzione della sufficienza:** i criteri ed i contenuti che tutti gli studenti devono raggiungere alla fine dell'anno per la sufficienza.

Per il **numero di verifiche** fare riferimento al verbale di dipartimento.

In caso di domande o richieste potete contattare il coordinatore di dipartimento.

# Tavola dei contenuti

<b>Introduzione</b>	<b>2</b>
<b>Tavola dei contenuti</b>	<b>4</b>
<b>Risorse e materiali</b>	<b>4</b>
Server studenti	4
Esercitazioni Linux	5
Sviluppo web	5
Database MySQL	5
Macchine virtuali	5
Crimpatrici, cavi Ethernet e terminali RJ-45	5
Arduino	6
Raspberry	6
Micro:Bit	6
Switch e Router	6
Realtà Virtuale	8
MARRtino Robot	8
<b>Servizi e Convenzioni</b>	<b>8</b>
CISCO NetAcademy	8
AWS Educate	8
GitHub Campus Program	8
Google Play Store	8
Divisione delle conoscenze e competenze per materie di indirizzo	8
Propedeuticità	9
Tabella riassuntiva argomenti minimi e propedeuticità	9
Biennio	9
Triennio	10
<b>Griglia di valutazione</b>	<b>10</b>
<b>Programmazione didattica</b>	<b>12</b>
Classe: 1a I.T. Tecnologico	12
Disciplina: TECNOLOGIE INFORMATICHE	12
Finalità	12
Obiettivi	13
Contenuti	13
Esperienze e approfondimenti	14
Strumenti di lavoro	14
Modalità di lavoro	14
Tipologie di verifica	14
Requisiti minimi per l'attribuzione della sufficienza	14

Classe: 2a I.T. Tecnologico	14
Disciplina: TECNOLOGIE ED APPLICAZIONI INFORMATICHE	14
Finalità	15
Obiettivi	15
Contenuti	15
Esperienze e approfondimenti	16
Classe: 3a I.T. Tecnologico	16
Disciplina: INFORMATICA	16
Finalità	16
Obiettivi	17
Contenuti	17
Esperienze e approfondimenti	19
Strumenti di lavoro	19
Modalità di lavoro	19
Tipologie di verifiche	19
Requisiti minimi per l'attribuzione della sufficienza	19
Disciplina: SISTEMI E RETI	19
Finalità	19
Obiettivi	19
Contenuti	20
Esperienze e approfondimenti	22
Riferimenti esterni	22
Attività interdisciplinari	22
Strumenti di lavoro	22
Modalità di lavoro	22
Tipologie di verifiche	23
Requisiti minimi per l'attribuzione della sufficienza	23
Disciplina: TECNOLOGIE E PROGETTAZIONE DI SISTEMI INFORMATICI E DI TELECOMUNICAZIONI	23
Finalità	23
Obiettivi	23
Contenuti	24
Esperienze e approfondimenti	25
Strumenti di lavoro	25
Modalità di lavoro	25
Tipologie di verifiche	26
Requisiti minimi per l'attribuzione della sufficienza	26
Classe: 4a I.T. Tecnologico	26
Disciplina: INFORMATICA	26
Finalità	26
Obiettivi	26
Contenuti	26

Esperienze e approfondimenti	26
Strumenti di lavoro	28
Modalità di lavoro	28
Tipologie di verifiche	28
Requisiti minimi per l'attribuzione della sufficienza	28
Disciplina: SISTEMI E RETI	28
Finalità	28
Obiettivi	28
Contenuti	28
Esperienze e approfondimenti	30
Risorse esterne	30
Strumenti di lavoro	30
Modalità di lavoro	30
Tipologie di verifiche	30
Requisiti minimi per l'attribuzione della sufficienza	30
Disciplina: TECNOLOGIE E PROGETTAZIONE DI SISTEMI INFORMATICI E DI TELECOMUNICAZIONI	31
Finalità	31
Obiettivi	31
Contenuti	31
Esperienze e approfondimenti	33
Strumenti di lavoro	33
Modalità di lavoro	33
Tipologie di verifiche	33
Requisiti minimi per l'attribuzione della sufficienza	33
Classe: 5a I.T. Tecnologico	33
Disciplina: INFORMATICA	33
Finalità	33
Obiettivi	33
Contenuti	34
Esperienze e approfondimenti	35
Strumenti di lavoro	35
Modalità di lavoro	35
Tipologie di verifiche	35
Requisiti minimi per l'attribuzione della sufficienza	35
Disciplina: SISTEMI E RETI	35
Finalità	35
Richiami di conoscenze utili allo svolgimento della seconda prova	36
Contenuti	37
Esperienze e approfondimenti	39
Risorse esterne	39
Strumenti di lavoro	39

Modalità di lavoro	39
Tipologie di verifiche	39
Disciplina: TECNOLOGIE E PROGETTAZIONE DI SISTEMI INFORMATICI E DI TELECOMUNICAZIONI	39
Finalità	39
Contenuti	40
Esperienze e approfondimenti	42
Strumenti di lavoro	42
Modalità di lavoro	42
Tipologie di verifiche	42
Disciplina: GESTIONE DEL PROGETTO E ORGANIZZAZIONE DI IMPRESA	42
Contenuti	42
Esperienze e approfondimenti	43
Strumenti di lavoro	43
Modalità di lavoro	43
Tipologie di verifiche	43

# Risorse e materiali

La scuola negli anni ha acquisito del materiale con cui è possibile svolgere delle attività.

Docenti teorici e ITP interessati nelle attività o nell'uso del materiale, possono rivolgersi al relativo docente referente del laboratorio in cui sono collocati.

Per maggiori dettagli sulle attività nelle singole materie nei diversi anni, consultare i relativi paragrafi della programmazione didattica.

## Server studenti

La scuola dispone di un server per le attività didattiche con gli studenti, raggiungibile all'indirizzo [studenti.marconicloud.it](http://studenti.marconicloud.it). Per coordinarsi sull'uso delle lettere degli utenti, consultare e modificare [questo](#) file.

Il vantaggio principale di usare il server rispetto allo sviluppo sulle macchine locali è di non dover installare nulla sulla propria macchina, si può quindi avere esattamente la stessa configurazione in qualsiasi macchina della scuola, a casa e anche con accesso da cellulare. Lo svantaggio è che se ci sono problemi di rete, il server potrebbe non essere raggiungibile o lento.

## Esercitazioni Linux

Ogni studente può collegarsi via SSH al server e fare degli esercizi sui comandi Linux. Bisogna tenere in considerazione che (ovviamente) gli studenti non hanno i permessi di amministratore. Guida per connettersi con SSH [qui](#) (attenzione che la porta è non standard).

## Sviluppo web

La home di ogni studente è servita all'URL [studenti.marconicloud.it/nome\\_utente](http://studenti.marconicloud.it/nome_utente) (es. [studenti.marconicloud.it/c\\_utente00](http://studenti.marconicloud.it/c_utente00)). Per modificare i file si può usare SSH oppure Tiny File Manager al link [studenti.marconicloud.it/x\\_utentefilem](http://studenti.marconicloud.it/x_utentefilem).

## Database MySQL

Alcuni utenti hanno anche un utente su mysql installato sulla macchina. La password è la stessa di quella dell'utente. C'è anche un'istanza di [PHPMyAdmin](#) per chi ne volesse fare uso. Per sapere quali utenti hanno anche l'utenza su mysql consultare [questo](#) file.

## Macchine virtuali

Sono possibili anche delle esercitazioni sulle macchine virtuali con VMWare. Per informazioni, rivolgersi al prof. Melita.

## Crimpatrici, cavi Ethernet e terminali RJ-45

### Attività

- Cablaggio strutturato, Sistemi e Reti, terzo e quarto anno

### Laboratorio

- Matematica (Fabrizio Manzella)

## Arduino

### Attività

- Introduzione alla programmazione, primo biennio
- Sistemi embedded a microcontrollore, Sistemi e Reti, terzo anno

### Laboratori

- Matematica, 15 unità (Fabrizio Manzella)
- Lab Inf. 1, 15 unità (Donatella Cappiello)
- Lab Inf. 3, 15 unità (Santo Dell'Omo)

### Risorse

- [Tinkercad Circuit](#) (simulatore di Arduino)
- [Circuito.io](#) (genera automaticamente circuiti e codice in base alle esigenze)

## Raspberry

### Attività

- configurazione di reti LAN, Sistemi e Reti, terzo e quarto anno
- sistemi operativi Linux, TPSI, terzo e quarto anno
- sistemi distribuiti, TPSI, quinto anno

### Laboratori

- Matematica, 15 unità (Fabrizio Manzella)
- Lab Inf. 1, 15 unità (Donatella Cappiello)
- Lab Inf. 3, 15 unità (Santo Dell'Omo)

## Micro:Bit

### Attività

- introduzione alla programmazione (primo biennio)
- utilizzabile anche in abbinamento allo smartphone con Bluetooth

### Laboratori

- Lab. Did. Innovativa, circa 20 unità (Claudio Capobianco)

## ESP32

## Switch e Router

La scuola possiede diversi dispositivi dismessi, tra cui switch e router, che possono essere usati per la didattica.

### Attività

- varie attività di Sistemi e Reti, soprattutto nel terzo e quarto anno

### Laboratori

- attualmente si trovano in sala server, in attesa di migliore collocazione; chiedere a Melita o Capobianco

## Realtà Virtuale

La scuola ha una tradizione ormai di diversi anni ed ha organizzato diversi festival della realtà virtuale. La scuola possiede 11 postazioni [Meta Quest 2](#) e 5 PC Alienware.

La collocazione è nel laboratorio Makerspace.

Referente: Claudio Capobianco

## MARRtino Robot

### Attività

- introduzione alla programmazione (primo biennio)
- utilizzabile anche in abbinamento allo smartphone con Bluetooth

### Laboratori

- Lab. Makerspace

## Servizi e Convenzioni

La scuola offre a docenti e studenti diversi servizi e convenzioni, di cui si può fare richiesta.

## CISCO NetAcademy

Il nostro istituto è una Local Academy CISCO, ed ha quindi la possibilità di offrire i seguenti servizi in modo completamente gratuito:

- per i docenti, la possibilità di conseguire certificazioni CCNA, scegliendo da un vasto catalogo, sia corsi base che avanzati
- per gli studenti, la possibilità di conseguire certificazioni CCNA all'interno di classi create dal proprio docente

Referente: Katuscia Veri

## AWS Educate

Il Marconi è accreditato come istituto presso [AWS Educate](#). Questo permette di avere del credito AWS sia per i docenti che per gli studenti, da spendere nei tantissimi servizi cloud offerti. Utile in particolare per TPSI nel quinto anno. Per l'iscrizione come docente, vedi l'[Appendice](#).

Referente: Claudio Capobianco

## GitHub Campus Program

Il Marconi è accreditato come "[GitHub Campus Program partner school](#)", che offre i seguenti vantaggi:

- Accesso automatico a [GitHub Student Developer Pack](#), che offre moltissimi servizi online gratuitamente o fortemente scontati
- Infiniti repository privati e la possibilità di fare l'upgrade di organizzazioni esistenti in modo gratuito
- Formazione gratuita su Git e GitHub

L'organizzazione GitHub ufficiale della scuola è [IIS G. Marconi Civitavecchia](#).

Referente: Claudio Capobianco.

## **Google Play Store**

Il Marconi ha un account sviluppatore sul Google Play Store, che permette alla nostra scuola di pubblicare applicazioni per Android.

Referente: Angelini, Capobianco.

## **Divisione delle conoscenze e competenze per materie di indirizzo**

Come linee guida generali, le materie di indirizzo nel triennio si dividono nel seguente modo:

- Informatica: programmazione stand-alone, client-server, e distribuiti;
- Sistemi e reti: funzionamento del singolo elaboratore e progettazione e gestione di una rete di uffici, gestione di sistemi hardware, i loro protocolli di comunicazione ;
- TPSI: astrazioni applicative nel singolo elaboratore (es. sistema operativo, multithreading, terminale) e in rete (es. servizi web), protocolli dal punto di vista applicativo

## **Propedeuticità**

Per coordinare la didattica di diverse materie di indirizzo che afferiscono ad un'unica competenza, si è reso necessario creare dei percorsi trasversali che dovranno essere svolti nei tempi esposti, con l'obiettivo di:

- permettere ai docenti nei diversi anni del triennio di fare affidamento a dei prerequisiti certi degli studenti
- proporre agli studenti delle attività di PCTO, sapendo che gli studenti hanno le conoscenze di base per poter affrontare tali attività

Per il quinto anno, le propedeuticità si intendono per l'esame di Stato.

## Tabella riassuntiva argomenti minimi e propedeuticità

### Biennio

	Informatica/Scienze Tecnologie applicate
<b>Primo anno</b>	<p>Riconoscere le caratteristiche architettoniche di un computer (calcolo, elaborazione, comunicazione)</p> <p>Conversione di numeri binari in decimale e viceversa</p> <p>Codifica di testo e immagini</p> <p>Riconoscere e utilizzare le funzioni di base di un sistema operativo</p> <p>Applicazioni elementari con foglio di calcolo</p> <p>Redigere un documento con un editor di testo</p> <p>Elementi base della programmazione (variabili, gestione I/O, selezione binaria)</p> <p><b>Laboratorio</b></p> <ul style="list-style-type: none"> <li>- uso di un editor di testo e del foglio di calcolo</li> <li>- AlgoBuild e/o Scratch</li> </ul>
<b>Secondo anno</b>	<p>Problemi e algoritmi</p> <p>Strutture della programmazione (sequenza, selezione, iterazione)</p> <p>Array</p> <p>Sviluppo e collaudo di un programma</p> <p>Problem solving</p> <p><b>Laboratorio (se possibile)</b></p> <ul style="list-style-type: none"> <li>- Python (opzionale, suggerito l'uso di replit.com)</li> <li>- AlgoBuild (opzionale)</li> <li>- C/C++ (opzionale)</li> </ul>

## Triennio

	Informatica	Sistemi e Reti	TPSI
<b>Terzo anno</b>	<p>Linguaggio compilato: cicli, selettori, array, pile e code.</p> <p><b>Laboratorio</b> Programmazione (C/C++)</p>	<p>Architettura dell'elaboratore.</p> <p>Topologia delle reti; basi ISO/OSI; mezzi trasmissivi bus.</p> <p><b>Laboratorio</b> Analisi componenti PC, analisi assembly, progetto Arduino/MicroBit, packet Tracer.</p>	<p>Sistemi di numerazione e codifica.</p> <p>Sistemi operativi: condivisione delle risorse.</p> <p>Linux, alfabetizzazione, basi del terminale.</p> <p>HTML e CSS.</p> <p><b>Laboratorio</b> Terminale Linux, pagina web statica.</p>
<b>Quarto anno</b>	<p>Linguaggio ad oggetti, composizione ed ereditarietà.</p> <p>JavaScript: validazione ingressi, creazione pagine dinamiche lato client.</p> <p><b>Laboratorio</b> Programmazione</p>	<p>IP Networking: ISO/OSI liv. 2,3,4; subnetting; switching &amp; routing statico e dinamico; TCP/IP.</p> <p>Cablaggio strutturato.</p> <p><b>Laboratorio</b> Packet Tracer, Raspberry.</p>	<p>Comandi Linux per la gestione delle risorse dell'OS.</p> <p>Automatizzazione dei comandi con Bash o Python.</p> <p>Multithreading.</p> <p><b>Laboratorio</b> Linux, scheduling</p>
<b>Quinto anno</b>	<p>Pagine dinamiche lato server, database relazionali (SQL)</p> <p><b>Laboratorio</b> Programmazione database, client/server.</p>	<p>VLAN, crittografia e autenticazione, firewall, VPN. Sistemi wireless.</p> <p>DHCP, DNS, HTTPS.</p> <p>Architetture IoT.</p> <p><b>Laboratorio</b> Packet Tracer, Cloud Services (es. AWS), Raspberry.</p>	<p>HTML/CSS/JS</p> <p>Servizi distribuiti, microservizi.</p> <p>Protocolli di rete (REST, WebSocket, MQTT)</p> <p><b>Laboratorio</b> Cloud Services (es. AWS), applicazioni distribuite.</p>

# Griglia di valutazione

Per la valutazione delle interrogazioni ci si atterrà allo schema seguente, che ha la funzione di correlare i voti assegnati ad un insieme di descrittori.

VOTO	PROFILO
1-2	Identifica una situazione di totale assenza di requisiti per ipotizzare il raggiungimento di qualsiasi obiettivo minimo disciplinare e/o comportamentale.
3	Identifica una situazione di carenze gravi nelle conoscenze, l'alunno è completamente disorientato, non evidenzia abilità e/o impegno che gli possono consentire di raggiungere le competenze minime.
4	Identifica una situazione di grave carenza nelle conoscenze; mostra un'assunzione mnemonica di informazioni o comunque una rielaborazione non sempre consapevole dei contenuti in modo che non si traduca in abilità.
5	Identifica una situazione di carenza o a livello delle conoscenze- anche a fronte di un impegno complessivamente sufficiente- o a livello delle competenze. L'alunno attiva qualche abilità ma in modo difficoltoso o impreciso
6	Identifica un livello minimo accettabile di conoscenze e competenze affiancato da una rielaborazione autonoma dei contenuti
7	Identifica un sicuro possesso delle conoscenze e la capacità di operare analisi e sintesi. Seria e responsabile la rielaborazione individuale e la partecipazione al dialogo scolastico.
8	Identifica il pieno possesso delle conoscenze e una notevole capacità di operare sintesi e analisi organiche. Matura e consapevolmente critica la rielaborazione individuale e sempre costruttiva la partecipazione al dialogo scolastico
9	Identifica un sicuro e completo possesso delle conoscenze e un'ottima capacità di operare sintesi ed analisi organiche e l'acquisizione consapevole di capacità e di competenze particolari in modo da riuscire a portare a termine con successo prove di particolare complessità, piccole tesi, ricerche bibliografiche..
10	Identifica livelli di eccellenza, prove sempre perfette con l'utilizzo anche di strategie personalizzate

# Programmazione didattica

## Classe: 1<sup>a</sup> I.T. Tecnologico

### Disciplina: TECNOLOGIE INFORMATICHE

#### Finalità

Scopo della disciplina “Tecnologie Informatiche” è implementare il raccordo tra sapere, metodo scientifico e tecnologia. Attraverso la didattica di laboratorio le abilità e le conoscenze già possedute vengono approfondite, integrate e sistematizzate. L'utilizzo del “problem-solving” permette di focalizzare l'attenzione degli allievi sul problema, sollecitando lo sviluppo di metodologie finalizzate all'approccio integrato del sapere scientifico e tecnico.

#### Obiettivi

Al termine della prima classe gli allievi dovranno raggiungere i seguenti obiettivi:

- riconoscere le caratteristiche architettoniche di un computer (calcolo, elaborazione, comunicazione)
- riconoscere e utilizzare le funzioni di base di un sistema operativo
- realizzare applicazioni elementari di calcolo
- saper utilizzare, in particolare, il foglio elettronico
- raccogliere, organizzare e rappresentare informazioni
- elementi base della programmazione (variabili, gestione I/O, selezione binaria)

#### Contenuti

- Le caratteristiche architettoniche di un computer
  - concetti di hardware e software
  - codifica binaria e codici ascii
  - sistemi di numerazione binario, esadecimale
- Conversione di basi
- Memorie, dischi, bus, periferiche
- Concetto di enunciato logico, operatori logici
- Il sistema operativo
  - il concetto di sistema operativo
  - il sistema operativo Windows
- Analisi e risoluzione di problemi
- Schemi risolutivi con il foglio elettronico
- Algoritmi e linguaggi di programmazione
- Analisi, comprensione, risoluzione dei problemi
- Il diagramma a blocchi su carta e sintesi al PC con Algobuild
- Il linguaggio di progetto (consigliati: Scratch, C, Python, Processing)
- Il programma e le variabili
- L'input e l'output dei dati
- Costrutto di selezione
- Cenni sui costrutti iterativi

### **Strumenti di lavoro**

1. Lavagna
2. Videoproiettore o LIM
3. Testo in adozione
4. Laboratorio di Informatica
5. Piattaforma e-learning (eventualmente)
6. Fotocopie di materiale didattico vario (appunti, schemi, ecc.)

### **Modalità di lavoro**

1. Spiegazioni in classe e/o in laboratorio
2. Utilizzo del laboratorio di Informatica
3. Studio e svolgimento di esercizi a casa
4. Recupero periodico

### **Tipologie di verifica**

1. Domande a risposta aperta/chiusa
2. Verifiche orali
3. Verifiche pratiche

### **Requisiti minimi per l'attribuzione della sufficienza**

Lo studente:

- possiede i contenuti minimi: basi dell'architettura di un computer, definizione di algoritmo, realizzazione di semplici diagrammi a blocchi, conoscenza e uso del SO
- applica le conoscenze acquisite per la soluzione di problemi noti (saper analizzare e risolvere semplici problemi legati all'uso delle tecnologie informatiche, saper passare da semplici problemi al diagramma a blocchi)
- si esprime in modo semplice ma con uso del linguaggio specifico della materia, cerca soluzioni anche precostituite

# **Classe: 2<sup>a</sup> I.T. Tecnologico**

## **Disciplina: TECNOLOGIE ED APPLICAZIONI INFORMATICHE**

### **Finalità**

- comprendere i principali fondamenti teorici delle scienze dell'informazione;
- acquisire la padronanza di strumenti dell'informatica;
- utilizzare tali strumenti per la soluzione di problemi significativi in generale, ma in particolare connessi allo studio delle altre discipline;
- acquisire la consapevolezza dei vantaggi e dei limiti dell'uso degli strumenti e dei metodi informatici e delle conseguenze culturali e sociali di tale uso.

### **Obiettivi**

Al termine della seconda classe gli allievi dovranno raggiungere i seguenti obiettivi:

- conoscere il concetto di algoritmo
- rappresentare algoritmi mediante flow-chart e linguaggio di progetto
- conoscere il concetto di programmazione strutturata
- conoscere le strutture della programmazione (sequenza, selezione, iterazione)
- conoscere e saper usare le variabili strutturate
- saper editare, testare e collaudare un programma (linguaggi consigliati: C, Python, Processing)

### **Contenuti**

- I traduttori: compilatori, interpreti e linkers
- Algoritmi e linguaggi di programmazione
- Introduzione ad un linguaggio di programmazione
- Struttura del programma e le variabili
- Ciclo di vita di una variabile
- L'input e l'output dei dati
- Operatori aritmetici e di confronto
- Incremento e decremento di una variabile
- La selezione a una via e a due vie
- Selezioni annidate, cascata di selezioni, selezioni multiple
- La selezione con gli operatori logici AND, OR e NOT
- Il ciclo a condizione in testa: while e for e suo diagramma di flusso
- Ciclo a condizione in coda: do-while e suo diagramma di flusso
- Variabili accumulatori e contatori e esempi di base
- Algoritmi iterativi definizione e esempi
- Calcolo di massimi e minimi
- Cicli di selezione e conteggio
- I vettori: dichiarazione, logica di accesso agli elementi
- Assegnazioni con i vettori
- Il ciclo for e i vettori: la variabile indice e algoritmi di visita sequenziale (diretta e inversa) per l'inserimento e il reperimento degli elementi
- Algoritmi di ricerca e selezione degli elementi di un vettore in base a condizioni su valori e indici

## **Esperienze e approfondimenti**

Sviluppo di semplici programmi. Linguaggi consigliati: C, C++, Python, Processing.

Applicazione per smartphone (Android, Apple) per introdurre i concetti della programmazione:

[Lightbot Code Hour](#).

Dispense dei docenti: <https://wbigger.github.io/book-programming/>

# Classe: 3<sup>a</sup> I.T. Tecnologico

## Disciplina: INFORMATICA

### Finalità

Il corso di Informatica ha come fine principale quello di mettere in grado lo studente di affrontare la soluzione di un problema posto dalla richiesta di un ipotetico committente, scegliendo le metodologie e gli strumenti software più idonei, e di fornirgli inoltre la formazione di base che gli consenta di seguire con una certa autonomia l'evoluzione delle tecnologie informatiche.

### Obiettivi

L'obiettivo della programmazione curricolare annuale è introdurre l'alunno alla programmazione evidenziando l'esistenza di diversi modi di intendere la programmazione stessa e abituardolo a ragionare per modelli.

Al termine della terza classe gli allievi dovranno raggiungere i seguenti obiettivi:

- analizzare criticamente le conoscenze pregresse di informatica
- impostare problemi, anche da un punto di vista non procedurale
- risolvere problemi, indipendentemente da un linguaggio di programmazione
- verificare la correttezza di una soluzione
- saper effettuare il debug di base di un programma
- leggere e interpretare descrizioni sintattiche in più notazioni
- applicare corretti stili di programmazione
- documentare il software a livello elementare
- usare con proprietà un linguaggio imperativo
- saper gestire allocazione, accesso e funzioni di base sulle principali strutture dati
- saper impostare la soluzione di semplici problemi mediante un approccio procedurale (top down e/o bottom up)
- riconoscere i limiti dell'approccio procedurale e i vantaggi di approcci con incapsulamento dei dati

Tali conoscenze permetteranno agli allievi di affrontare nelle due classi successive argomenti sempre più complessi

### Contenuti

- Programmazione in C, C++ e/o Python.
- Buone prassi per la generazione di codice ordinato e autoesplicativo, uso di commenti, indentazione e nomi significativi per le variabili.
- Generazione di numeri casuali, ruolo dell'operatore modulo.
- Riepilogo e approfondimento sui costrutti ciclici e di selezione.
- Vettori: concetti, assegnazione, iterazione.
- Algoritmi di ricerca (sequenziale e binaria), ordinamento (bubble, insertion e selection sort), merge e inserimento (in coda e ordinato) sui vettori.
- Le matrici: algoritmi di visita (riga, colonna, spirale, ecc) per inserimento, elaborazione e stampa di dati, operazioni statistiche sulle matrici (min, max, media), estrazione di sottomatrici e fusione di matrici, generazione di semplici figure.
- Le funzioni, ciclo di vita delle variabili interne ad una funzione, concetto di scope di una variabile, prototipi, definizione, parametri attuali e formali. Definizione ed invocazione di funzione.

- Parametri di IN, di OUT e di IN/OUT in teoria e loro realizzazione in C (passaggio per valore, riferimento e puntatore). Gestione di parametri statici. Documentazione dei parametri.
- Gestione dinamica dei dati, introduzione ai puntatori.
- Esempi su interazione utente lato server: realizzazione di menù a carattere con cicli e funzioni.
- Funzioni annidate e funzioni ricorsive e la gestione implicita dello stack di sistema.
- Gestione della pila e della coda mediante le funzioni. Ruolo dell'operatore modulo nella coda circolare.
- La gestione dinamica della memoria, differenza tra variabili statiche dinamiche e loro relazione con stack e heap. Strutture dati dinamiche e loro allocazione. Esempi di gestione dinamica dei dati (pila, coda).
- Le strutture dati non omogenee: le struct (C/C++), tuple e dizionari (Python).
- Gestione di strutture dati complesse.
- I file e la loro gestione (lettura, scrittura).

### **Esperienze e approfondimenti**

Corso CISCO IT Essentials: spiegazione ed esame dei punti critici dei capitoli del corso.

Uso di strumenti di condivisione del codice (es. Git, GitHub, etc.)

Esperienze di Intelligenza Artificiale con Python

Interdisciplinare: uso del linguaggio C/C++ per un progetto Arduino.

### **Strumenti di lavoro**

1. Lavagna
2. Videoproiettore o LIM
3. Testo in adozione
4. Laboratorio di Informatica
5. Piattaforma e-learning (eventualmente)
6. Fotocopie di materiale didattico vario (appunti, schemi, ecc.)

### **Modalità di lavoro**

1. Spiegazioni in classe e/o in laboratorio
2. Utilizzo del laboratorio di Informatica
3. Studio e svolgimento di esercizi a casa
4. Recupero periodico

### **Tipologie di verifiche**

1. Domande a risposta aperta/chiusa
2. Verifiche orali
3. Verifiche pratiche

### **Requisiti minimi per l'attribuzione della sufficienza**

Lo studente:

- possiede i contenuti minimi di teoria dell'informazione, basi dei linguaggi di programmazione, definizione di algoritmo, strutture di controllo, dati strutturati, funzioni e procedure.
- applica le conoscenze acquisite per la soluzione di problemi noti (saper analizzare e risolvere semplici problemi indipendentemente dal linguaggio di programmazione, saper sviluppare semplici programmi in almeno un linguaggio)
- si esprime in modo semplice ma con uso del linguaggio specifico della materia, cerca soluzioni anche precostituite

# Classe: 3<sup>a</sup> I.T. Tecnologico

## Disciplina: SISTEMI E RETI

### Finalità

Sistemi e reti concorre a far conseguire allo studente al termine del percorso quinquennale i seguenti risultati di apprendimento relativi al profilo educativo, culturale e professionale dello studente coerenti con la disciplina: cogliere l'importanza dell'orientamento al risultato, del lavoro per Obiettivi e della necessità di assumere responsabilità nel rispetto dell'etica e della deontologia professionale; orientarsi nella normativa che disciplina i processi produttivi del settore di riferimento, con particolare attenzione sia alla sicurezza sui luoghi di vita e di lavoro sia alla tutela dell'ambiente e del territorio; intervenire nelle diverse fasi e livelli del processo produttivo, dall'ideazione alla realizzazione del prodotto, per la parte di propria competenza, utilizzando gli strumenti di progettazione, documentazione e controllo; riconoscere gli aspetti di efficacia, efficienza e qualità nella propria attività lavorativa.

### Obiettivi

- Configurare, installare e gestire sistemi di elaborazione dati e reti;
- Saper riconoscere i componenti interni e i bus di un computer;
- Scegliere dispositivi e strumenti in base alle loro caratteristiche funzionali
- Descrivere e comparare il funzionamento di dispositivi e strumenti elettronici e di telecomunicazione;
- Saper realizzare o essere in grado di leggere un semplice programma in assembly;
- Saper comunicare con una periferica con le tecniche del polling e dell'interrupt;
- Saper misurare il link budget di un collegamento in filo
- Saper progettare e documentare il cablaggio strutturato di un edificio in una situazione elementare;
- Saper comunicare con una periferica di Arduino con le tecniche del polling e dell'interrupt.

### Contenuti

#### Architettura del PC

- Architettura dei sistemi di elaborazione secondo Von Neumann e descrizione in dettaglio delle sue parti:
- CPU: struttura di una CPU e suoi elementi, concetto di modello di programmazione, ciclo di lavoro della CPU, i registri.
- Cenni sull'architettura Harvard.
- Il BUS: il bus come canale di comunicazione broadcast e sue proprietà (modalità di TX, arbitraggio), i registri come porte di accesso al bus, struttura base e proprietà di carico (dispositivi three state), definizione di ciclo macchina, operazioni elementari di una istruzione tipo.
- La RAM: schema logico e schema fisico, funzioni, prestazioni e segnali tipici, spazio di indirizzamento, cenni su circuito di decodifica dell'indirizzo.
- Cenni sulle moderne architetture non Von Neumann.
- Formato delle istruzioni, sequenza di controllo cenni su istruzione MOV ad accesso diretto.
- Il modello di programmazione della CPU 8086: registri principali e gestione della memoria
- Cenni sulle istruzioni di controllo in assembly.
- Cenni sulle modalità di indirizzamento: immediato, diretto e indiretto (a registro e in memoria)

#### Gestione delle periferiche

- Elementi base di una interfaccia di periferica

- Tecniche di comunicazione con una periferica: polling ed interrupt.
- Fasi di un generico protocollo di interrupt, la ISR e il vettore delle interruzioni.
- Problemi generali nella gestione di un interrupt (riconosc. periferica, gestione priorità, gestione richieste annidate).
- Riconoscimento tramite polled interrupt e tramite interrupt vettorizzato.
- Tecniche principali di interrupt vettorizzato: daisy-chain, PIC.

#### Sistemi embedded a microcontrollore

- I collegamenti elettrici necessari per la predisposizione della board (da concordare con il docente di TLC)
- L'IDE di Arduino: codifica su PC e trasferimento del programma sulla scheda.
- Il linguaggio processing: costrutti fondamentali, comandi gestione dell'I/O.
- Cenni sulla gestione delle periferiche con Arduino: elementi essenziali di una periferica, gestione polled I/O, gestione interrupted I/O con Arduino.
- Esempi di utilizzo notevoli.
- Cenni sul dispositivo raspberry: preparazione della sdcard, accesso headless al dispositivo mediante porta di debug.
- Cenni sull'utilizzo delle porte GPIO in esempi notevoli (sysfs).

#### Reti di computer

- Definizione, scopo e classificazione delle reti di computer per grandezza e topologia.
- Le risorse trasmissive di una rete: nodi di elaborazione e mezzi trasmissivi
- Classificazione dei tipi di canale (half-duplex, full-duplex, simplex), definizione di canale broadcast.
- Modalità di allocazione dei mezzi trasmissivi (assoluta, statica e dinamica) con esempi tecnologici reali
- Tecniche di multiplazione: statica (SDM, FDM, TDM, CDM) e statistica (STDM).
- Definizione di protocollo di comunicazione e classificazione dei protocolli.
- Reti a circuito, a commutazione di messaggio e a circuito virtuale.
- Architettura di un commutatore a pacchetto e descrizione delle sue parti.
- Definizione di instradamento di un messaggio tra nodi e ruolo locale della commutazione.
- I doppini e le fibre ottiche, proprietà elettriche e loro misura secondo standard EIA/TIA 568.
- Cenni sul link budget di una comunicazione in db.
- Il cablaggio strutturato degli edifici secondo gli standard EIA/TIA 568, ISO/IEC 11801, EN 50173
- Architettura del cablaggio strutturato: materiali e modelli di connessione
- Documentazione del progetto di cablaggio strutturato con nozioni di dimensionamento
- Architettura generale degli apparati attivi.
- Presentazione di una metodologia a fasi per la progettazione e la documentazione di un cablaggio strutturato col dimensionamento accurato fino al livello L2.
- conoscere i più diffusi bus e avere un'informazione sulle funzioni di alcune schede di espansione e di interfaccia e relative compatibilità e standardizzazioni;

#### La pila ISO/OSI

- la pila ISO/OSI completa (7 livelli) e ridotta TCP/IP (4 livelli), scopo di un'architettura a strati, funzioni essenziali di ogni livello
- il canale virtuale di livello N e la tecnica dell'imbustamento multiplo definizione di PDU, PCI e SDU
- interfacce e il canale reale di livello N definizione di primitive e SAP
- descrizione sommaria delle funzioni di ogni livello
- Funzioni del livello 1
  - Canali seriali e paralleli, proprietà di una codifica di canale ideale, classificazione delle codifiche in uso, esempi di codifiche (RZ, NRZ, manchester)
  - Definizione di bus, HUB e repeater

- Sincronismo di bit e tecniche per il suo recupero
- Funzioni del livello 2
  - Definizione di switch, confronto con HUB
  - Schema di principio di un commutatore
  - I processi di ingress, forwarding ed egress, tipologie di switch
  - COP e BOP definizione di trama
  - Formato delle trame MAC, LLC e PPP.
  - Controllo di flusso e controllo di errore nel protocollo MAC.
  - Il protocollo CSMA/CD.
  - Schema essenziale di una connessione wireless.

### **Esperienze e approfondimenti**

Corso CISCO IT Essentials: spiegazione ed esame dei punti critici dei capitoli del corso. Presentazione di lezioni del corso CISCO IT Essential afferenti ai moduli precedenti.

Esperienze di laboratorio con il simulatore PacketTracer

Esempi di progettazione di reti LAN con PacketTracer.

Esperienze con dispositivi Arduino e Raspberry.

Basi sulla logica di utilizzo e programmazione di Arduino e Raspberry.

Uso delle porte del dispositivo Arduino per creare esperienze sulla gestione della comunicazione con le periferiche tramite polling e interrupt.

Uso delle porte GPIO di Raspberry per esperienze su comunicazioni con dispositivi esterni (interfaccia sysfs in linux o C e/o librerie python).

Eventuali esperienze di comunicazione con sensori mediante bus standard di livello 1 (1-Wire, 2-Wire, 3-Wire, 4-Wire).

Approfondimenti su assembly 8086 con Arduino o Raspberry: schema base delle parti di un programma, le variabili in assembly, istruzioni base dell'assembly 8086 per assegnamenti e gestione I/O (INT21H)

Generazione di codice Assembly a partire da un programma in C attraverso l'IDE o con il comando "gcc -S"

Realizzazione di parte del bus Daisy Chaining con un simulatore di porte logiche (vedi riferimenti esterni)

### **Riferimenti esterni**

CISCO Packet Tracer:

<https://www.netacad.com/courses/packet-tracer>

Controllo di un LED con Arduino:

<https://cfpmanfredini.wordpress.com/2013/01/14/controllo-di-un-led-con-tecnica-pwm/>

Somma di numeri in binario:

[http://chortle.ccsu.edu/assemblytutorial/Chapter-08/ass08\\_1.html](http://chortle.ccsu.edu/assemblytutorial/Chapter-08/ass08_1.html)

Raspberry Pi e bus:

<http://www.raspberry-pi-geek.com/Archive/2015/09/Getting-to-know-the-Raspberry-Pi-I2C-bus>

Simulatori online di gate logici:

<https://www.khanacademy.org/computer-programming/logic-gate-puzzler/1522357785> (online)

<https://play.google.com/store/apps/details?id=com.Suborbital.CircuitScramble&hl=it> (Android)

<http://www.smartlogicsimulator.com> (Android)

Net Analyzer:

<https://techet.net/netanalyzer> (iOS, Android)

Visualizzare il codice Assembly da un programma C:

<https://wbigger.github.io/book-arch-cpu/chap1/05-somma.html>

Tinkercad Circuits (simulatore Arduino)

<https://www.tinkercad.com/circuits>

Circuito.io (generatore automatico di circuiti e codice per Arduino)

<https://www.circuito.io/>

### **Attività interdisciplinari**

Collaborazione con i docenti di informatica e TPSI per la realizzazione di progetti didattici comuni riguardanti tecnologie di comunicazione client-server.

Proposta di progetti di IOT.

Collaborazione col docente di telecomunicazioni per la progettazione di un percorso didattico riguardante la misura di qualità di un collegamento in rame e in fibra ottica.

### **Strumenti di lavoro**

1. Lavagna
2. Videoproiettore o LIM
3. Testo in adozione
4. Laboratorio di Informatica
5. Piattaforma e-learning (eventualmente)
6. Fotocopie di materiale didattico vario (appunti, schemi, ecc.)

### **Modalità di lavoro**

1. Spiegazioni in classe e/o in laboratorio
2. Utilizzo del laboratorio di Informatica
3. Studio e svolgimento di esercizi a casa
4. Recupero periodico

### **Tipologie di verifiche**

1. Domande a risposta aperta/chiusa
2. Verifiche orali
3. Verifiche pratiche

### **Requisiti minimi per l'attribuzione della sufficienza**

Lo studente possiede i contenuti minimi:

- conosce gli elementi fondamentali che costituiscono gli elementi fondamentali di un sistema di elaborazione.
- conosce la classificazione delle reti
- conosce i protocolli di comunicazione standard
- si esprime in modo semplice ma con uso del linguaggio specifico della materia.

# Classe: 3<sup>a</sup> I.T. Tecnologico

## Disciplina: TECNOLOGIE E PROGETTAZIONE DI SISTEMI INFORMATICI E DI TELECOMUNICAZIONI

### Finalità

Tecnologie e progettazione di sistemi informatici e di telecomunicazioni concorre a far conseguire allo studente seguenti risultati di apprendimento:

- orientarsi nella normativa che disciplina i processi produttivi del settore di riferimento, con particolare attenzione sia alla sicurezza sui luoghi di vita e di lavoro sia alla tutela dell'ambiente e del territorio; intervenire nelle diverse fasi e livelli del processo produttivo, dall'ideazione alla realizzazione del prodotto, per la parte di propria competenza, utilizzando gli strumenti di progettazione, documentazione e controllo;
- riconoscere gli aspetti di efficacia, efficienza e qualità nella propria attività lavorativa; padroneggiare l'uso di strumenti tecnologici con particolare attenzione alla sicurezza nei luoghi di vita e di lavoro, alla tutela della persona, dell'ambiente e del territorio.

### Obiettivi

Al termine della prima classe gli allievi dovranno raggiungere i seguenti obiettivi:

- conoscere le più diffuse tecniche di codifica e compressione di dati, immagini e video;
- saper calcolare il peso di una immagine
- saper calcolare il bit rate di una sorgente campionata
- saper utilizzare alcuni comandi base di Linux per la gestione del filesystem, il montaggio e la gestione dei dischi e l'installazione e disinstallazione dei pacchetti.
- conoscere i principali problemi della trasmissione digitale dell'informazione sul versante della codifica
- conoscere scopo e risorse di un sistema operativo e del BIOS
- conoscere i principali algoritmi di scheduling
- conoscere le modalità di allocazione di un processo in memoria
- saper progettare e realizzare una semplice pagina web con elementi di contenuto e di stile
- saper gestire il layout di una pagina web tramite tabelle e div.

### Contenuti

Creazione di pagine web statiche

- il protocollo HTTP e le pagine WEB statiche
- le tabelle, le liste
- id e class applicati a tabelle e liste
- inserimento di immagini e loro centratura con il CSS
- il layout della pagina HTML

La codifica dell'informazione

- Definizione di informazione, elementi di informazione simbolici o fisici, proprietà di simboli e alfabeti.
- Definizione di segnale analogico, gli elementi di segnale analogico come particolari simboli.
- Le conversioni ADC e DAC scopo loro collocazione nella catena di trasmissione dei dati. Fasi della conversione ADC. Calcolo del bitrate data la frequenza del segnale e la precisione della quantizzazione.
- Definizione di codifica dell'informazione, tabelle di codifica statiche e dinamiche.
- La catena di trasmissione dell'informazione e le sue codifiche.

- Codifica per la rappresentazione delle sorgenti di informazione: codifica ASCII e Unicode, codici digitali pesati e non pesati, la codifica dei numeri.
- Codifica dei segnali video: quantizzazione spaziale vs codifica vettoriale. La percezione dei colori, la codifica dei colori additiva RGB e sottrattiva CMYK. Calcolo del peso in bit di una immagine.
- Codifica per la compressione dell'informazione: Huffman, cenni su RLE, Size-value e DPCM. Cenni su compressioni lossy e lossless.
- Le cause di degradazione del bit. La rigenerazione del segnale digitale. I vantaggi della trasmissione digitale su quella analogica.
- Codifica per la correzione d'errore: codifica di hamming.

Il sistema operativo.

- Schema delle risorse del sistema. Architettura a strati.
- Il BIOS: scopo e funzioni principali (POST, gestione scheda madre e terminal driver), La ROM e il BIOS.
- Definizione e scopo di un SO. Risorse del SO. Efficienza e condivisione delle risorse. Buffering e spooling.
- Architettura a buccia di cipolla di un SO. Definizione degli strati. Concetto di macchina virtuale e di virtualizzazione di una risorsa HW. Concetto di trasparenza e mediazione. Periferiche virtuali e architettura di gestione delle periferiche fisiche mediante drivers.
- La gestione dei processi. La multiprogrammazione e il multitasking, il time sharing. Lo schedulatore. Diagramma degli stati di un processo e il descrittore PCB. Il cambio di contesto. Parametri di uno schedulatore. Politiche e algoritmi di scheduling.
- Schema delle gerarchie di memorie e sue proprietà. Multiprogrammazione e gestione condivisa della memoria fisica. Modalità di allocazione dei processi (assoluta, statica, dinamica). Definizione di binding.
- Partizionamento fisso e variabile della memoria. Problematiche ed algoritmi. La tecnica dello swap.
- Le tecniche di paginazione, segmentazione, segmentazione mista a paginazione: elementi costitutivi e processo di binding.

Il sistema operativo Linux.

- La gestione del sistema operativo linux da shell: installazione e rimozione di pacchetti mediante i comandi apt-get (update, install, remove, ecc.)
- Struttura, realizzazione e gestione di un filesystem.
- Il filesystem di linux: Comandi base su copia, incolla e taglia di cartelle e files. Editor VI e editor NANO di Linux, logica di utilizzo e loro comandi base. Struttura ed esempi pratici di gestione dei dischi (comandi fdisk e mount).
- Cenni alle espressioni regolari (vedi anche [regex.101](http://regex.101))

## Esperienze e approfondimenti

Proposta di progetti di IOT.

Uso di strumenti di condivisione del codice (es. Git, GitHub, etc.).

Corso CISCO IT Essentials: spiegazione ed esame dei punti critici dei capitoli del corso.

Presentazione di lezioni del corso CISCO IT Essential afferenti ai moduli precedenti.

## Strumenti di lavoro

Lavagna

Videoproiettore o LIM

Testo in adozione

Laboratorio di Informatica

Piattaforma e-learning

Fotocopie di materiale didattico vario (appunti, schemi, ecc.)

Siti: [w3school](http://w3school) per la parte web

### **Modalità di lavoro**

Spiegazioni in classe e/o in laboratorio

Utilizzo del laboratorio di Informatica

Studio e svolgimento di esercizi a casa

Recupero periodico

### **Tipologie di verifiche**

Domande a risposta aperta/chiusa

Verifiche orali

Verifiche pratiche

### **Requisiti minimi per l'attribuzione della sufficienza**

Lo studente possiede i contenuti minimi:

- sa creare semplici pagine web statiche (propedeutico per gli anni successivi)
- conosce le funzionalità fondamentali di un sistema operativo
- è in grado rappresentare e gestire l'interazione di eventi concorrenti
- si esprime in modo semplice ma con uso del linguaggio specifico della materia.

# Classe: 4<sup>a</sup> I.T. Tecnologico

## Disciplina: INFORMATICA

### Finalità

Il corso di Informatica ha come fine principale quello di mettere in grado lo studente di affrontare la soluzione di un problema posto dalla richiesta di un ipotetico committente, scegliendo le metodologie e gli strumenti software più idonei, e di fornirgli inoltre la formazione di base che gli consenta di seguire con una certa autonomia l'evoluzione delle tecnologie informatiche.

### Obiettivi

L'obiettivo della programmazione curricolare annuale è far acquisire all'alunno le competenze per affrontare progetti di una certa dimensione anche con la metodologia orientata agli oggetti. Deve acquisire una certa flessibilità che gli consenta di analizzare e risolvere problemi di varia natura utilizzando il paradigma e gli strumenti più idonei alla natura del problema.

Al termine della quarta classe gli allievi dovranno raggiungere i seguenti Obiettivi

- Svolgere un'analisi qualitativa degli algoritmi.
- Applicare con proprietà tecniche di programmazione modulare.
- Elaborare file ad organizzazione sequenziale e random.
- Organizzazioni con archivi tradizionali - File sequenziali
- Definire e manipolare strutture di dati connesse da puntatori.
- Applicare il paradigma della programmazione orientata agli oggetti.
- Applicare il paradigma della programmazione guidata degli eventi.
- Progettare e costruire interfacce d'utente amichevoli.
- Conoscere ed usare classi notevoli di oggetti contenitori.
- Conoscere le basi della compilazione/interpretazione.
- Analizzare criticamente le conoscenze pregresse di informatica.

### Contenuti

La programmazione ad oggetti (Java e/o JavaScript)

- Classi e oggetti
- Attributi e metodi
- Variabili e metodi di classe, variabili locali e variabili istanza
- Incapsulamento e information hiding
- Ereditarietà e gerarchie
- Classi astratte e interfacce
- Polimorfismo e Casting
- Eccezioni

Metodologie di progettazione (es. Agile/Scrum, UML, etc.)

- Documentazione delle interazioni dell'utente con il sistema tramite scenari d'uso e casi d'uso.
- Determinazione dei requisiti funzionali e non funzionali.
- Proposta di una metodologia di progettazione di una applicazione ad oggetti
- Proposta di una metodologia di documentazione di una applicazione

Interfacce grafiche in Java

- La grafica e le interfacce utente grafiche (librerie grafiche AWT e Swing)
- La gestione degli eventi.
- La gestione delle eccezioni.

Il linguaggio JavaScript: introduzione al linguaggio, creazione di semplici applicazioni

## **Esperienze e approfondimenti**

Uso di strumenti di testing (es. JUnit per Java, Jest/Jasmine/Karma per JavaScript).

Uso delle lambda function (lambda expression in Java 8 e arrow function in

Cenni programmazione Android: eventuale programmazione in java e XML di un client Android (ad es. all'interno di un modulo compensativo, modulo pomeridiano, ecc)

Cenni di Javascript: validazione dei campi di un form, il DOM di una pagina (menu a tendina dinamici, tabelle dinamiche)

Approfondimenti sulla progettazione con Agile/Scrum con realizzazione di un semplice progetto

Approfondimenti sulla progettazione con UML con schede CRC, etc.

Esperienze interdisciplinari: collaborazione con i docenti di informatica e sistemi per la realizzazione di progetti didattici comuni riguardanti tecnologie di comunicazione client-server.

Proposta di progetti di IOT.

Uso di strumenti di condivisione del codice (es. Git, GitHub, etc.).

## **Strumenti di lavoro**

1. Lavagna
2. Videoproiettore o LIM
3. Testo in adozione
4. Laboratorio di Informatica
5. Piattaforma e-learning (eventualmente)
6. Fotocopie di materiale didattico vario (appunti, schemi, ecc.)

## **Modalità di lavoro**

1. Spiegazioni in classe e/o in laboratorio
2. Utilizzo del laboratorio di Informatica
3. Studio e svolgimento di esercizi a casa
4. Recupero periodico

## **Tipologie di verifiche**

1. Domande a risposta aperta/chiusa
2. Verifiche orali
3. Verifiche pratiche

## **Requisiti minimi per l'attribuzione della sufficienza**

Lo studente:

- possiede i contenuti minimi: strutture dati, file, programmazione a oggetti, basi del linguaggio Java, basi HTML e CSS;
- applica le conoscenze acquisite per la soluzione di problemi noti (saper sviluppare programmi in Java)

# Classe: 4<sup>a</sup> I.T. Tecnologico

## Disciplina: SISTEMI E RETI

### Finalità

Le finalità della disciplina comprendono due aree di interesse distinte:

- i sistemi per l'elaborazione (calcolatori) e la trasmissione delle informazioni (reti) dal punto di vista dell'architettura. In questo ambito la finalità dell'insegnamento è quella di contribuire alla formazione di un adeguato bagaglio di precise conoscenze tecniche per il futuro perito;
- le applicazioni nell'ambito industriale e dei servizi. La finalità dell'insegnamento è di tipo metodologico perché deve fornire anche generali capacità di analisi dei sistemi, di comprensione dei processi economici, di autoaggiornamento, oltre che metodologie specifiche di risoluzione dei problemi.

### Obiettivi

L'obiettivo della programmazione curricolare annuale è rendere in grado l'allievo di:

- installare, personalizzare e condurre la manutenzione di piccoli sistemi di elaborazione distribuiti
- assolvere con responsabilità ed autonomia compiti parziali nella gestione di grandi sistemi;
- conoscere le problematiche ed i prodotti software relativi all'implementazione dei vari livelli di un protocollo di rete
- conoscere approfonditamente le funzioni di un router e saper programmare correttamente il suo comportamento
- conoscere approfonditamente le funzioni di uno switch.
- conoscere le tecnologie WAN
- saper progettare e realizzare una rete LAN da ufficio in riferimento alla maggior parte degli aspetti che riguardano la completa connettività e interoperabilità ai livelli L1, L2 e L3
- saper valutare le problematiche principali relative alla gestione e configurazione di servizi di ISP

### Contenuti

Richiami sul cablaggio strutturato degli edifici

- Standards TIA/EIA 568B, ISO/IEC 11801, EN 50173 scopi e modello architettuale
- documentazione del progetto di un cablaggio strutturato: descrizione generale del progetto, planimetrie, albero degli apparati passivi, albero degli apparati attivi, tabella dorsali, definizione armadi

Funzioni del livello 2

- La commutazione di livello 2, descrizione di bridge e switch
- Il protocollo ARP per subnet direttamente connesse e non
- Il protocollo STP scopo e fasi di funzionamento

Funzioni del livello 3

- La commutazione di livello 3 descrizione di switch L3 e router
- Il subnetting IP, l'indirizzamento IP, classi di indirizzi e subnetting
- Metodologie ed esercitazioni per il calcolo del subnetting IPv4 secondo gli schemi FLSM classful, FLSM classless, VLSM, CIDR; subnetting delle dorsali
- Il routing statico, le tabelle di routing: proprietà e confronto con le tabelle di inoltro L2
- Processo di lookup di un indirizzo di subnet
- Modelli di connessione di una subnet al router (direttamente e indirettamente connessa) e routing nei due casi; metodo sistematico per la generazione manuale di una tabella di routing

statica

- Modelli di reti tramite grafi ed alberi
- Principio di ottimalità, algoritmo di Dijkstra. algoritmo di Bellmann-Ford; protocollo Link State Packet, protocollo Distance Vector; routing gerarchico; protocolli IGP (RIP e OSPF) ed EGP (EGP e BGP)

Funzioni del livello 4 (da rimandare in 5 classe se il tempo a disposizione non è sufficiente)

- Protocollo UDP e protocollo TCP
- Handshake del protocollo TCP e gestione dei flag SYN, ACK e FYN
- Problematiche di moltiplicazione, controllo di flusso e controllo di congestione

Funzioni di livello 7

- Protocolli DNS, DHCP e ICMP
- Tecnica del tunnelling e suoi impieghi notevoli

## **Esperienze e approfondimenti**

Descrizione del sistema CISCO IOS e delle peculiarità della sua shell.

Attribuzione di indirizzi IP alle interfacce di un router CISCO tramite shell.

Attribuzione di indirizzi IP alle interfacce di una macchina Linux.

L'interfaccia di configurazione web-based di un server Linux Webmin.

Simulazione con il CISCO PacketTracer di una LAN da ufficio composta switch e/o router.

Simulazione con il CISCO PacketTracer di scenari di subnetting.

Simulazione con il CISCO PacketTracer di scenari di routing statico.

Simulazione con il CISCO PacketTracer di scenari di routing RIP o OSPF.

Esempi con principali comandi della shell Linux e di Windows per la gestione delle reti.

Simulazione con il CISCO PacketTracer e/o Linux di scenari di configurazione di utilizzo di servizi DNS e DHCP.

Realizzazione di un router con Linux e Raspberry PI mediante i comandi `sysctl` e `ip`.

Simulazione di uno scenario di utilizzo di uno sniffer di rete con CISCO PacketTracer e/o Wireshark e/o Tcpdump.

Corso CISCO CCNA: spiegazione ed esame dei punti critici dei capitoli del corso.

Simulazioni Esami di Stato degli anni precedenti per la parte che riguarda subnetting e cablaggio strutturato.

Attività multidisciplinari: esercitazioni multidisciplinari tra TPSI, informatica e sistemi per lo svolgimento coordinato, ciascuna disciplina per la parte che le compete, di prove d'esame degli anni precedenti; collaborazione con i docenti di informatica e sistemi per la realizzazione di progetti didattici comuni riguardanti tecnologie di comunicazione client-server.

Proposta di progetti di IOT.

## **Risorse esterne**

CISCO Packet Tracer:

<https://www.netacad.com/courses/packet-tracer>

Net Analyzer:

<https://techet.net/netanalyzer> (iOS, Android)

Analisi del traffico di rete con Wireshark:

<https://www.wireshark.org/> (Linux, Windows, MacOS)

## **Strumenti di lavoro**

1. Lavagna

2. Videoproiettore o LIM
3. Testo in adozione
4. Laboratorio di Informatica
5. Piattaforma e-learning (eventualmente)
6. Fotocopie di materiale didattico vario (appunti, schemi, ecc.)

#### **Modalità di lavoro**

1. Spiegazioni in classe e/o in laboratorio
2. Utilizzo del laboratorio di Informatica
3. Studio e svolgimento di esercizi a casa
4. Recupero periodico

#### **Tipologie di verifiche**

1. Domande a risposta aperta/chiusa
2. Verifiche orali
3. Verifiche pratiche

#### **Requisiti minimi per l'attribuzione della sufficienza**

Lo studente:

- possiede i contenuti minimi: concetti di base delle reti , il modello ISO/OSI e TCP/IP, protocolli di rete, basi di routing, gestione problematiche di sicurezza nelle reti Intranet
- applica le conoscenze acquisite per la soluzione di problemi noti (utilizzo mezzi di trasmissione, routing, protocolli TCP/IP, gestione sicurezza)
- si esprime in modo semplice ma con uso del linguaggio specifico della materia, cerca soluzioni anche precostituite.

# Classe: 4<sup>a</sup> I.T. Tecnologico

## Disciplina: TECNOLOGIE E PROGETTAZIONE DI SISTEMI INFORMATICI E DI TELECOMUNICAZIONI

### Finalità

Le finalità della disciplina comprendono due aree di interesse distinte:

- una prima riguardante i sistemi per l'elaborazione (calcolatori) e la trasmissione delle informazioni (reti) dal punto di vista dell'architettura. In questo ambito la finalità dell'insegnamento è quella di contribuire alla formazione di un adeguato bagaglio di precise conoscenze tecniche per il futuro perito;
- una seconda riguardante le applicazioni nell'ambito industriale e dei servizi. La finalità dell'insegnamento è di tipo metodologico perché deve fornire anche generali capacità di analisi dei sistemi, di comprensione dei processi economici, di autoaggiornamento, oltre che metodologie specifiche di risoluzione dei problemi.

### Obiettivi

L'obiettivo della programmazione curricolare annuale è introdurre l'allievo alla conoscenza delle architetture dei sistemi per l'elaborazione e la trasmissione delle informazioni

Al termine della quarta classe gli allievi dovranno raggiungere i seguenti Obiettivi

- conoscere in modo approfondito una CPU;
- valutare comparativamente le architetture di diversi sistemi di elaborazione;
- conoscere i concetti di base su evoluzione e struttura dei sistemi operativi;
- conoscere i livelli più bassi di un sistema operativo: sincronizzazione di processi elementari, gestione delle interruzioni hardware e software;
- saper utilizzare i comandi e i servizi di base del sistema operativo Linux

### Contenuti

Basi teoriche di processi e thread

- Richiami sulla gestione dei processi (scheduler, PCB, ciclo di vita di un processo).
- Competizione e cooperazione tra i processi.
- Le risorse di un processo: gestione e classificazione. Grafi di Holt e loro riduzione.
- I thread: caratteristiche, ciclo di vita, classificazione, confronto con i processi, il TCB.

Principi di programmazione concorrente.

- Elaborazione sequenziale vs elaborazione concorrente. Definizione di sistema concorrente.
- Modellazione dei processi concorrenti: il grafo delle precedenze.
- Classificazione dei processi in base alla concorrenza (processi indipendenti, in competizione, cooperanti)
- Definizione di scomposizione sequenziale. Parallelismo fisico e parallelismo virtuale.
- Il costrutto fork-join senza count e con count e GOTO.
- Il costrutto cobegin-coend. Conversione da un costrutto all'altro. Il concetto di grafo strutturato.
- Concetto di precedenza implicita e semplificazione di un grafo delle precedenze.
- Il costrutto fork-join in C. Le fork semplici e quelle annidate in C.
- Esempi pratici di programmazione concorrente in C sulla shell di linux.
- I thread in Java. Ciclo di vita dei thread in java.
- Modelli implementativi dei thread in Java.

La cooperazione tra processi e tra thread.

- Modelli di comunicazione tra processi e tra thread e loro confronto.
- Concorrenza reale e simulata. Le condizioni di Bernstein.

- Gli errori dipendenti dal tempo: definizione e caratteristiche.
- I problemi della cooperazione tra thread o processi: mutua esclusione e sincronizzazione.
- Definizione di sezione critica e gestione della mutua esclusione. Concetti di starvation e deadlock.
- Gestione della zona critica con gli spin lock: primitive lock e unlock e loro realizzazione (alternanza semplice, interrupt, alternanza con primitiva TestAndSet).
- I semafori di Dijkstra: definizione, primitive PWait e VSignal, significato del contatore S.
- Applicazioni dei semafori a problemi di: sola mutua esclusione, di sola sincronizzazione, misti mutua esclusione-sincronizzazione in pseudocodice e in Java.
- Problemi canonici della programmazione concorrente risolti in pseudocodice e/o Java.

La programmazione della shell di linux

- Definizione di variabili e costrutti assegnazione, selezione ed iterazione nella shell di linux (linguaggi consigliati: BASH e/o Python).
- Creazione di menu a carattere lato server per la configurazione di servizi.
- L'interfaccia di gestione dei servizi di Linux systemd.

### **Esperienze e approfondimenti**

Uso di strumenti di testing (es. JUnit per Java, Jest/Jasmine/Karma per JavaScript).

Esperienze con dispositivi Arduino e Raspberry

Collaborazione con i docenti di informatica e sistemi per la realizzazione di progetti didattici comuni riguardanti tecnologie di comunicazione client-server.

Proposta di progetti di IOT.

Uso di strumenti di condivisione del codice (es. Git, GitHub, etc.).

Corso CISCO IT Essentials: spiegazione ed esame dei punti critici dei capitoli del corso.

### **Strumenti di lavoro**

1. Lavagna
2. Videoproiettore o LIM
3. Testo in adozione
4. Laboratorio di Informatica
5. Piattaforma e-learning (eventualmente)
6. Fotocopie di materiale didattico vario (appunti, schemi, ecc.)

### **Modalità di lavoro**

1. Spiegazioni in classe e/o in laboratorio
2. Utilizzo del laboratorio di Informatica
3. Studio e svolgimento di esercizi a casa
4. Recupero periodico

### **Tipologie di verifiche**

1. Domande a risposta aperta/chiusa
2. Verifiche orali
3. Verifiche pratiche

### **Requisiti minimi per l'attribuzione della sufficienza**

Lo studente: possiede i contenuti minimi ossia i concetti di base della struttura e dell'evoluzione dei sistemi operativi, delle basi del kernel di Linux e dei processi;

applica le conoscenze acquisite per la soluzione di problemi noti (gestione memoria, scheduling, sicurezza dei sistemi operativi, gestione dei file);

si esprime in modo semplice, usando linguaggio tecnico specifico, cercando soluzioni anche precostituite.

# Classe: 5<sup>a</sup> I.T. Tecnologico

## Disciplina: INFORMATICA

### Finalità

Il corso di Informatica ha come fine principale quello di mettere in grado lo studente di affrontare la soluzione di un problema posto dalla richiesta di un ipotetico committente, scegliendo le metodologie e gli strumenti software più idonei, e di fornirgli inoltre la formazione di base che gli consenta di seguire con una certa autonomia l'evoluzione delle tecnologie informatiche.

### Obiettivi

L'obiettivo della programmazione curricolare annuale è far acquisire all'alunno le competenze necessarie per rappresentare e gestire un insieme di informazioni, nonché per saper scegliere il tipo di organizzazione più adatto a seconda dell'applicazione. L'alunno dovrà saper gestire il progetto e la manutenzione di Sistemi Informativi per piccole realtà e conoscere i concetti e le tecniche fondamentali per la progettazione di basi di dati.

Al termine della quinta classe gli allievi dovranno raggiungere i seguenti obiettivi:

- Gestione di informazioni. Il progetto dei sistemi informativi. Modellazione concettuale di un sistema informatico (ad es. con il modello entità-associazioni).
- Basi di dati Concetti generali sulle basi di dati.
- Sistemi per la Gestione di Basi di Dati (DBMS).
- Il Modello relazionale e sue operazioni.
- Linguaggi di interrogazione non procedurali (SQL).
- Problematiche in multiutenza. Tecniche di accesso ai database in rete. Programmazione lato server (PHP).
- La produzione del software. Strumenti software di supporto allo sviluppo del software. Documentazione e presentazione del software.

### Contenuti

Il processo di raccolta dei requisiti e sua documentazione (es. Agile/Scrum, UML, etc)

- Tipi di raccolta dei requisiti, la fase di esplorazione, problemi della fase di esplorazione.
- Scenari d'uso e casi d'uso, tipi di scenari, descrizione dei casi d'uso, opzionalmente: diagramma di contesto.
- Documentazione dei casi d'uso.
- Documentazione del progetto informatico

Archivi e file

- Definizione di archivio
- Supporti di memoria
- Concetti di record e chiave
- Operazioni sugli archivi
- Archivi sequenziali
- Archivi non sequenziali

Archivi, basi di dati e DBMS

- Il sistema informativo aziendale. I sistemi informativi e la loro gestione automatica.
- Analisi dei problemi con archiviazione dati. Rappresentazione di situazioni reali e di processi aziendali.
- Caratteristiche del database. Vantaggi rispetto alle tecniche tradizionali.
- Modelli logici di rappresentazione dati: gerarchico, reticolare, relazionale.
- Modello relazionale e sue caratteristiche.

- Modellazione dei dati: entità e associazioni. Gli attributi. Le associazioni tra entità. Regole di lettura.
- Derivazione del modello logico dallo schema E-R.
- Introduzione al DBMS.
- Normalizzazione delle relazioni.
- Integrità referenziale.

#### Il linguaggio SQL

- Caratteristiche del linguaggio non procedurale.
- Principali identificatori e tipi di dati.
- Definizione e manipolazione di tabelle.
- Interrogazione del database.
- Operazioni relazionali nel linguaggio: proiezione, selezione, congiunzione (join).
- Tipi di join (inner, outer, self join).
- Interrogazioni nidificate e complesse con tabella di appoggio.
- Funzioni di aggregazione.
- Ordinamenti e raggruppamenti.
- Condizioni di ricerca (like, between... and, in, any, all).
- Query di servizio. Trigger. Grant e revoke.

#### Tecniche di accesso ai database in ambiente Internet

- Interfacciare un database in rete.
- Programmazione lato server e lato client.
- Problematiche relative allo sviluppo di applicazioni web.

#### La programmazione lato server

- La piattaforma Apache-PHP-MySQL.
- Il linguaggio PHP.
- Array associativi di sistema speciali SESSION E COOKIES.
- PHP e HTML.
- PHP e MySQL.

### **Esperienze e approfondimenti**

Ambienti consigliati per realizzare applicazioni client server: Docker o EasyPhp.

Approfondimenti: uso di nginx come alternativa leggera e ad alte prestazioni di Apache.

Attività multidisciplinari

Esercitazioni multidisciplinari tra TPSI, informatica e sistemi per lo svolgimento coordinato, ciascuna disciplina per la parte che le compete, di prove d'esame degli anni precedenti.

Collaborazione con i docenti di informatica e sistemi per la realizzazione di progetti didattici comuni riguardanti tecnologie di comunicazione client-server.

Proposta di progetti di IOT.

Uso di strumenti di condivisione del codice (es. Git, GitHub, etc.).

### **Strumenti di lavoro**

1. Lavagna
2. Videoproiettore o LIM
3. Testo in adozione
4. Laboratorio di Informatica
5. Piattaforma e-learning (eventualmente)
6. Fotocopie di materiale didattico vario (appunti, schemi, ecc.)

### **Modalità di lavoro**

1. Spiegazioni in classe e/o in laboratorio
2. Utilizzo del laboratorio di Informatica

3. Studio e svolgimento di esercizi a casa
4. Recupero periodico

### **Tipologie di verifiche**

1. Domande a risposta aperta/chiusa
2. Verifiche orali
3. Verifiche pratiche

### **Requisiti minimi per l'attribuzione della sufficienza**

Lo studente:

- possiede i contenuti minimi: conoscenza delle problematiche di archiviazione e delle basi di dati, le basi del modello E-R, modello relazionale, elementi di base del linguaggio SQL, la sicurezza dei dati, basi della programmazione lato server, basi del linguaggio PHP
- applica le conoscenze acquisite per la soluzione di problemi noti (uso del dello E-R, derivazione dello schema relazionale, realizzazione di semplici applicazioni con basi di dati, anche lato server, uso del linguaggio SQL, uso del linguaggio lato server PHP, gestione elementare delle problematiche di sicurezza )
- si esprime in modo semplice ma con uso del linguaggio specifico della materia, cerca soluzioni anche precostituite.

# Classe: 5<sup>a</sup> I.T. Tecnologico

## Disciplina: SISTEMI E RETI

### Finalità

Le finalità della disciplina comprendono due aree di interesse distinte:

- una prima riguardante i sistemi per l'elaborazione (calcolatori) e la trasmissione delle informazioni (reti) dal punto di vista dell'architettura. In questo ambito la finalità dell'insegnamento è quella di contribuire alla formazione di un adeguato bagaglio di precise conoscenze tecniche per il futuro perito;
- una seconda riguardante le applicazioni nell'ambito industriale e dei servizi. La finalità dell'insegnamento è di tipo metodologico perché deve fornire anche generali capacità di analisi dei sistemi, di comprensione dei processi economici, di autoaggiornamento, oltre che metodologie specifiche di risoluzione dei problemi.

### Richiami di conoscenze utili allo svolgimento della seconda prova

Richiamo cablaggio strutturato degli edifici

- Standards TIA/EIA 568B, ISO/IEC 11801, EN 50173 scopi e modello architettuale
- documentazione del progetto di un cablaggio strutturato: descrizione generale del progetto, planimetrie, albero degli apparati passivi, albero degli apparati attivi, tabella dorsali, definizione armadi.

Il processo di raccolta dei requisiti di un sistema informatico e sua documentazione

- Documentazione delle interazioni dell'utente con il sistema tramite scenari d'uso e casi d'uso definiti (es. con Agile/Scrum o UML).
- Determinazione dei requisiti non funzionali per la definizione delle prestazioni di apparati e collegamenti.

Richiamo dei prerequisiti al subnetting: confronto indirizzi IP e MAC, ripasso protocollo ARP, ripasso domini di collisione, definizione di dominio di broadcast.

Richiamo di subnetting IPv4: metodologie ed esercitazioni per il calcolo del subnetting IPv4 secondo gli schemi FLSM classful, FLSM classless, VLSM (calcolo di ottetto di lavoro, subnet mask, indirizzo di subnet, indirizzo di broadcast, indirizzo di gateway, rangehost) Il subnetting delle dorsali

Richiami di routing statico manuale

- Le tabelle di routing: proprietà e confronto con le tabelle di inoltro L2
- Modelli di connessione di una subnet al router (direttamente e indirettamente connessa) e routing nei due casi
- metodo sistematico per la generazione manuale di una tabella di routing e esercitazioni su generazione albero di instradamento

Attività multidisciplinari: esercitazioni multidisciplinari tra TPSI, informatica e sistemi per lo svolgimento coordinato, ciascuna disciplina per la parte che le compete, di prove d'esame degli anni precedenti.

### Contenuti

I protocolli TCP e UDP (se non trattati in quarta classe)

- Servizi e funzioni dello strato di trasporto
- Protocollo UDP e protocollo TCP.
- Handshake del protocollo TCP e gestione dei flag SYN, ACK e FIN.
- Problematiche di moltiplicazione, controllo di flusso e controllo di congestione.

Il protocollo HTTP

- Generalità sul protocollo

- Formato di una richiesta HTTP e della relativa risposta.

#### Le VLAN

- Definizione e scopi. Modelli realizzativi delle VLAN: VLAN untagged (portbased), VLAN tagged.
- Il protocollo 802.1q e il concetto di dorsale logica.
- Tipi di porte di uno switch (tagged, untagged e ibride) e le operazioni su di esse (ingresso, forwarding e egress)
- Modelli realizzativi dell'inter-VLAN Routing: inter-VLAN tradizionale, inter-VLAN "router-on-a-stick"
- Configurazione di interfacce di rete VLAN su macchine Linux.

#### Dispositivi NAT e Firewall

##### Definizione funzioni essenziali di un NAT

- Classificazione dei NAT: tipi di SNAT e tipi di DNAT e loro utilizzi principali
- Tabella degli stati di una connessione e sue proprietà
- Il port-forwarding
- I firewall scopi e funzioni essenziali.
- Classificazione dei firewall in base al tipo di impiego (perimetrali e personali) e in base al tipo di ispezione
- Caratteristiche di packetfilter e bastionhost
- ACL e ACE: logica di interpretazione e impostazioni canoniche anti-spoofing.
- Funzioni aggiuntive di un firewall (NAT, Proxy, logging, IDS)
- Modelli notevoli di utilizzo e installazione dei firewall in una subnet aziendale (un livello, due livelli con DMZ, tre o più livelli di protezione)
- Realizzazione delle impostazioni canoniche di firewall e NAT tramite tabelle e/o tramite comandi CISCO IOS.

##### Le applicazioni e i sistemi distribuiti

- Richiamo dei concetti di middleware e dei modelli client-server two-tier e three-tier
- Architetture sicure per sistemi web two-tier e three-tier protette tramite firewall
- Architetture sicure per sistemi web three-tier con ridondanza delle applicazioni
- Architettura sicura per una server farm web 5-tier
- Ridondanza ed affidabilità di applicazioni e dati: il cloning (RACS sharednothing e shared disk) e il partitioning (gracefuldegradation e RAPS)
- Cenni sui modelli distribuiti Windows (modello a workgroup e modello a dominio)
- Classificazione dei modelli informativi basati sul web

##### I protocolli di autenticazione e di crittografia.

- Riepilogo del funzionamento degli algoritmi di crittografia simmetrica (AES, DES) e asimmetrica (RSA, Diffie Hellman) più importanti e loro ambito specifico di utilizzo (autenticazione, integrità dei dati, confidenzialità, non ripudio)
- Integrità di un messaggio tramite algoritmi di hashing (SHA e MAC)
- Cifratura di un messaggio firmato e non firmato tramite protocolli SMIME e PGP
- Autenticazione di un messaggio tramite firma digitale
- Autenticazione di un utente: classificazione in base al tipo di segreto
- Tecniche di autenticazione debole per canali sicuri (PAP con salt), semiforte (CHAP con sfida) e forte (asimmetrica) per canali insicuri,
- Cenni su autenticazione e cifratura di connessioni wireless tramite protocollo WAP e server RADIUS.
- Tecnica del tunnelling e suoi impieghi notevoli. Realizzazione di una VPN.

##### I dispositivi wireless

- Modalità di funzionamento (AP, Client, Bridged) e architettura di rete
- Protocollo CSMA/CA
- Problema della stazione nascosta, modo RTS/CTS.

- Il protocollo di crittografia WPA2.
- Configurazione del WPA\_supplicant di linux, configurazione di Raspberry come AP e/o come Hot Spot.
- Installazione e configurazione di un server di autenticazione RADIUS.
- Realizzazione in un linguaggio di programmazione a piacere degli algoritmi ALOHA, CSMA, CSMA./CD, CSMA/CA;

Servizi di directory

- Servizi X.500 (LDAP)
- Servizi dei nomi (DNS)
- Servizio DHCP
- Esempi a scelta di configurazione di server in Linux (DNS, DHCP, SAMBA, LDAP, RADIUS, ecc.)

### **Esperienze e approfondimenti**

Simulazione con il CISCO PacketTracer di scenari di subnetting di una VLAN.

Simulazione con il CISCO PacketTracer di scenari di configurazione ed utilizzo di Firewall e NAT.

Esperienze di laboratorio con le IPTABLES di Linux.

Esperienze di laboratorio sulla realizzazione di cluster di DB con mysql (Maria DB e Galera cluster)

Esperienze di laboratorio sulla realizzazione di cluster di archivi di filescon Rsync o con Gluster FS.

Esperienze di laboratorio sulla installazione e configurazione di un server NFSlinux.

Simulazione con il CISCO PacketTracer di scenari di configurazione ed utilizzo di dispositivi VPN.

Realizzazione di un server VPN in Linux.

Simulazione con il CISCO PacketTracer di scenari di configurazione ed utilizzo di servizi DNS e DHCP.

Esperienze interdisciplinari: collaborazione con i docenti di informatica e sistemi per la realizzazione di progetti didattici comuni riguardanti tecnologie di comunicazione client-server.

Proposta di progetti di IOT.

Eventuali argomenti di potenziamento: protocolli e loro classificazione, la pila ISO/OSI completa e ridotta, scopo di un'architettura a strati, funzioni essenziali di ogni livello, il canale virtuale di livello N e la tecnica dell'imbustamento multiplo definizione di PDU, PCI e SDU, il multiplatore statistico, dispositivi di livello 1 hub e modem, la commutazione di livello 2, descrizione di bridge e switch, la commutazione di livello 3 descrizione di switch L3 e router, cenni su schemi di routing EGP e IGP e internetworking su reti eterogenee.

### **Risorse esterne**

CISCO Packet Tracer:

<https://www.netacad.com/courses/packet-tracer>

### **Strumenti di lavoro**

1. Lavagna
2. Videoproiettore o LIM
3. Testo in adozione
4. Laboratorio di Informatica
5. Piattaforma e-learning (eventualmente)
6. Fotocopie di materiale didattico vario (appunti, schemi, ecc.)

### **Modalità di lavoro**

1. Spiegazioni in classe e/o in laboratorio

2. Utilizzo del laboratorio di Informatica
3. Studio e svolgimento di esercizi a casa
4. Recupero periodico

### **Tipologie di verifiche**

1. Domande a risposta aperta/chiusa
2. Verifiche orali
3. Verifiche pratiche

Nel trimestre/pentamestre ogni studente è stato sottoposto mediamente a tre/quattro prove di verifica sommativa di tipologia varia.

# Classe: 5<sup>a</sup> I.T. Tecnologico

## Disciplina: TECNOLOGIE E PROGETTAZIONE DI SISTEMI INFORMATICI E DI TELECOMUNICAZIONI

### Finalità

Le finalità della disciplina comprendono due aree di interesse distinte:

- una prima riguardante i sistemi per l'elaborazione (calcolatori) e la trasmissione delle informazioni (reti) dal punto di vista dell'architettura. In questo ambito la finalità dell'insegnamento è quella di contribuire alla formazione di un adeguato bagaglio di precise conoscenze tecniche per il futuro perito;
- una seconda riguardante le applicazioni nell'ambito industriale e dei servizi. La finalità dell'insegnamento è di tipo metodologico perché deve fornire anche generali capacità di analisi dei sistemi, di comprensione dei processi economici, di autoaggiornamento, oltre che metodologie specifiche di risoluzione dei problemi.

### Contenuti

I sistemi distribuiti

- Definizione e vantaggi. Classificazione delle principali tipologie.
- I sistemi distribuiti per il calcolo parallelo: architetture SISD, MISD e MIMD. Cluster di PC e di una risorsa in genere.
- Il concetto di middleware: trasparenza di eterogeneità HW e SW, trasparenza del canale di comunicazione e dei dettagli della sua gestione (sicurezza, serializzazione).
- Canali di livello 4 e di livello applicativo.
- Il modello client-server, classificazione dei server in iterativi e concorrenti. Descrizione della modalità operativa di entrambi.
- architetture di riferimento one-tier, two-tier, three-tier. Definizione di vantaggi e svantaggi di ciascuna. Architetture two-tier, fat-client e thin-client. Ruolo dei middleware all'interno dell'architettura three-tier.
- Identificazione di una applicazione in rete, concetto di porta e di socket.
- Esempi di applicazioni client-server realizzate con i socket.
- Interazioni client server web based. L'hosting remoto.
- La virtualizzazione: definizione e classificazione.
- Definizione del formato di trasferimento JSON.
- Definizione del linguaggio XML e delle sue proprietà.
- Programmazione di rete di livello 7: la tecnologia ODBC e le primitive di connessione al database.

Installazione e configurazione di un server web

- Installazione e configurazione di server web su dispositivo Raspberry e/o un PC.
- Comandi Linux per la configurazione di web server e servizi di sistema (php.ini, httpd.conf).
- Applicazioni avanzate del PHP: PHP ad oggetti, gestione dei file, gestione mappe di Google.

L'autenticazione client server

- Come si effettua in PHP il login ad una pagina con l'uso delle sessioni lato server con confronto con utenti memorizzati in un array.
- Login ad una pagina con l'uso delle sessioni e dei cookies con confronto con utenti memorizzati in un array.
- Come sopra ma con utenti memorizzati in un DB.

Le connessioni monodirezionali request/response di livello applicativo

- Formato di una richiesta HTTP e della relativa risposta.

- Programmazione dinamica lato client: cenni di Javascript, la tecnologia AJAX.
- Client http responsive: schema base completo di una richiesta AJAX che aspetta una risposta di tipo testo (JSON e XML).
- I web services REST: realizzazione di applicazioni client e server in HTTP.
- Principi REST nella realizzazione di applicazioni distribuite, gli Open Data.
- Connessioni M2M: client REST PHP che si connette ad un web service PHP tramite la libreria CURL.
- Realizzazione di un client REST HTTP con Arduino per applicazioni sistemistiche (reti di sensori, domotica, servizi remoti).
- In alternativa
- Realizzazione di un client REST HTTP con Android per applicazioni sistemistiche (reti di sensori, domotica, servizi remoti).

Le connessioni bidirezionali di livello applicativo

- Realizzazione di server concorrenti ed iterativi in Java.
- I web socket HTTP: realizzazione di un client in javascript e di un server in PHP o in node.js
- Realizzazione di client websocket in JavaScript e di server in PHP e/o node.js.

### **Esperienze e approfondimenti**

Installazione server web Apache o Nginx su Raspberry

Utilizzo di Docker per l'installazione di un web server e interprete PHP-FastCGI

Utilizzo di Node-Red/JS come strumento di sviluppo di applicazioni distribuite.

Esperienze interdisciplinari: collaborazione con i docenti di informatica e sistemi per la realizzazione di progetti didattici comuni riguardanti tecnologie di comunicazione client-server.

Proposta di progetti di IOT.

Uso di strumenti di condivisione del codice (es. Git, GitHub, etc.).

### **Strumenti di lavoro**

1. Lavagna
2. Videoproiettore o LIM
3. Testo in adozione
4. Laboratorio di Informatica
5. Piattaforma e-learning (eventualmente)
6. Fotocopie di materiale didattico vario (appunti, schemi, ecc.)

### **Modalità di lavoro**

1. Spiegazioni in classe e/o in laboratorio
2. Utilizzo del laboratorio di Informatica
3. Studio e svolgimento di esercizi a casa
4. Recupero periodico

### **Tipologie di verifiche**

1. Domande a risposta aperta/chiusa
2. Verifiche orali
3. Verifiche pratiche

# Classe: 5<sup>a</sup> I.T. Tecnologico

## Disciplina: GESTIONE DEL PROGETTO E ORGANIZZAZIONE DI IMPRESA

### Contenuti

Elementi di economia e organizzazione aziendale

- L'Informazione e l'organizzazione (meccanismi di coordinamento)
- Micro e macro struttura (posizione, mansione e ruolo)
- Strutture organizzative

I processi aziendali

- Processi aziendali e processi di gestione del mercato
- Ciclo di vita del prodotto
- Prestazioni dei processi aziendali
- Informatizzazione dei processi aziendali e strategie di trasformazione dei processi (buy side, in side, sell side)
- Il ruolo delle tecnologie informatiche nella gestione per processi

La qualità totale: ciclo di miglioramento continuo e norme ISO

Principi e tecniche di project management

- Definizione di progetto, proprietà delle sue fasi e principi di buona gestione
- Gli Obiettivi di un progetto
- L'organizzazione dei progetti, ruoli e processi di gestione di un progetto
- Tecniche di pianificazione e controllo temporale dei progetti
- Tecniche di gestione dei rischi e della qualità di un progetto

Gestione dei progetti informatici

- Il processo di produzione del software
- Analisi dei requisiti: definizione e proprietà di requisiti funzionali, di dominio e non funzionali
- Modelli classici di sviluppo dei sistemi informatici, loro proprietà e ambiti di utilizzo (modello a cascata, a V, prototipale, evolutivo, incrementale, RAD, a spirale, agile e XP)
- Cenni sulle metriche del software sui metodi di valutazione di qualità e costi
- Cenni di sicurezza sul lavoro: il rischio, la sua gestione, i fattori che lo determinano

### Esperienze e approfondimenti

Simulazione di impresa usando Lean Business Canvas.

Progetti in collaborazione con il Talent Garden di Civitavecchia, gestito dalla Lazio Innova (Regione Lazio).

### Strumenti di lavoro

1. Lavagna
2. Videoproiettore o LIM
3. Testo in adozione
4. Laboratorio di Informatica
5. Piattaforma e-learning (eventualmente)
6. Fotocopie di materiale didattico vario (appunti, schemi, ecc.)

### Modalità di lavoro

1. Spiegazioni in classe e/o in laboratorio
2. Utilizzo del laboratorio di Informatica
3. Studio e svolgimento di esercizi a casa

#### 4. Recupero periodico

##### **Tipologie di verifiche**

1. Domande a risposta aperta/chiusa
2. Verifiche orali
3. Verifiche pratiche