

Disciplina: INFORMATICA							
Livello scolastico	Nuclei fondanti della disciplina	Nuclei tematici disciplinari o temi anche di caratteri interdisciplinari	Conoscenze	Abilità	Competenze Specifiche	Attività didattiche possibili per lo sviluppo delle competenze	Competenze STEM
1° biennio	Architettura del computer	Informatica e computer Dato, informazione e conoscenza Le tecnologie dell'informazione e della comunicazione	Hardware Software Firmware Modello di Von Neumann CPU Gerarchia delle memorie Memorie di massa Periferiche Servizi e applicazioni dell'ICT Le professioni legate all'ICT	Saper identificare le caratteristiche tecniche e qualitative dei componenti Saper individuare i dispositivi congrui allo specifico elaboratore Identificare le periferiche e sfruttarne le funzionalità	Gestire efficacemente gli strumenti di lavoro comuni del computer e applicare i concetti di base ad essi collegati. Comprendere e distinguere le caratteristiche architettoniche di un computer, inclusi i concetti di hardware e software, codifica binaria, codici ASCII e Unicode. Essere in grado di identificare gli elementi funzionali della macchina di Von Neumann riconoscendo la differenza concettuale con un modello funzionale.. Acquisire il concetto di sistema operativo, delle sue funzionalità di base e le caratteristiche che contraddistinguono le varie tipologie di sistemi. Comprendere il	Attività pratica di familiarizzazione con gli strumenti di lavoro comuni del computer, come l'uso di sistemi operativi, software di produttività e strumenti di navigazione web. Introduzione ai concetti di base dell'architettura del computer, inclusi hardware e software, attraverso sessioni di laboratorio interattive. Creazione di una simulazione interattiva della Macchina di Von Neumann per comprendere visivamente i concetti di memoria, CPU e architettura di base del computer. Assegnazione di progetti per	Essere in grado di utilizzare in modo autonomo gli strumenti comuni del computer, applicando i concetti di base legati all'hardware e al software in contesti pratici. Dimostrare capacità di collaborare efficacemente con i pari nelle attività pratiche e di laboratorio, partecipando attivamente al lavoro di gruppo, al problem solving e alle discussioni per contribuire al processo di apprendimento reciproco. Sviluppare una competenza critica nell'uso di strumenti tecnologici e informatici, applicando tali strumenti in modo riflessivo per supportare processi cognitivi come
	I sistemi operativi	La classificazione dei software Introduzione ai sistemi operativi Le interfacce e le modalità di comunicazione Il file system Le licenze	Le fasi di installazione di un sistema operativo La configurazione dei driver Interfacce a riga di comando Interfacce GUI Tipologie di file Operare con i file Software libero e open source Le licenze Creative Commons	Sapere riconoscere le estensioni dei vari tipi di file. Sapere le differenze tra il software di base e il software applicativo. Sapere distinguere le principali funzioni di un sistema operativo. Sapere conoscere i diversi tipi di sistemi operativi in commercio.			
	I sistemi di numerazione e la rappresentazione digitale dell'informazione	La rappresentazione dei dati I sistemi di codifica Le conversioni Le operazioni tra numeri binari	Conversioni numeriche tra numeri binari, decimali, esadecimale Codifica delle informazioni Codifica binaria Codifica ASCII	Sapere come rappresentare e convertire i numeri nelle diverse basi di numerazione. Sapere come effettuare la conversione da decimale a binario, ottale e esadecimale. Sapere			

			Operazioni di somma, sottrazione tra numeri binari	come codificare le informazioni utilizzando la codifica binaria. Sapere e utilizzare le unità di misura, come bit, byte e loro multipli.	concetto di processo e la gestione della memoria, nonché le funzionalità dei file system come funzionalità di un sistema operativa	creare documenti elettronici. Attività di navigazione su Internet per comprendere la struttura e i servizi online. Progetti di ricerca che richiedono l'uso di motori di ricerca, l'accesso a risorse online e la presentazione dei risultati in formato elettronico.	investigare, esplorare modelli. Essere in grado di formulare soluzioni innovative per problemi reali utilizzando metodologie didattiche incentrate sull'apprendimento induttivo, e dimostrare abilità nell'analisi e nella risoluzione di problemi pratici attraverso esperienze di laboratorio o progetti applicativi. Essere in grado di navigare in Internet in modo consapevole, comprendendo la struttura della rete e riconoscendo i rischi associati, e dimostrare abilità nella comunicazione efficace online e nella conduzione di ricerche di informazioni, rispettando le regole legate all'uso di Internet.
Internet e il WWW	Le reti di computer Internet e la comunicazione La navigazione	Le reti di computer Caratteristiche delle reti di computer Il modello di internet Il modello client-server La comunicazione in rete L'indirizzo IP e le classi di indirizzi I rischi della rete La sicurezza in rete	Sapere individuare le tecnologie più recenti che permettono la comunicazione nel Web. Sapere riconoscere i rischi e i limiti nell'utilizzo della Rete.	Applicare le conoscenze sugli elementi costitutivi di un documento elettronico, con particolare attenzione ai fogli di calcolo, al fine di saperli utilizzare in modo competente.	Introduzione ai principi fondamentali dei linguaggi di programmazione attraverso esercizi pratici. Implementazione di algoritmi semplici in pseudo-codice e successivamente in un linguaggio di programmazione specifico, con particolare attenzione alla sintassi del linguaggio.	Assegnazione di un progetto di programmazione che richiede l'applicazione	
Il foglio di calcolo	Il foglio di calcolo Funzioni Grafici	L'ambiente del foglio di calcolo Operazioni sulle celle Riferimenti assoluti e relativi Formule matematiche e statistiche Formule logiche Formule sulle stringhe Grafici e stampa Ordinamenti e filtri	Saper operare nell'ambiente del foglio di calcolo. Sapere eseguire operazioni sulle celle. Sapere gestire riferimenti assoluti e relativi. Sapere formulare equazioni matematiche e statistiche. Sapere utilizzare formule logiche. Sapere applicare formule sulle stringhe. Sapere creare, modificare e stampare grafici. Sapere eseguire ordinamenti e filtri nei dati.	Comprendere i principi fondamentali dei linguaggi di programmazione, riconoscendone le diverse tipologie di linguaggi e comprendere il concetto di algoritmo come			
Algoritmi e linguaggi di programmazione	Problemi e algoritmi La soluzione ai problemi Algebra booleana e il ragionamento logico Le fasi di sviluppo Interazione macchina-utente	Dal dato, al problema, alla soluzione Algoritmi e linguaggi di programmazione Linguaggi a basso e alto livello	Sapere classificare un linguaggio di programmazione. Sapere identificare errori sia sintattici che semantici. Sapere individuare l'applicazione del				

		<p>Operare con i tipi di dato</p> <p>La selezione</p> <p>Le iterazioni</p>	<p>Linguaggi interpretati e compilati</p> <p>Le sintassi del linguaggio di programmazione</p> <p>Python</p> <p>I tipi di dato semplici</p> <p>numerici</p> <p>Le espressioni</p> <p>Funzioni predefinite</p> <p>Variabili</p> <p>Operazioni di input e output</p> <p>Operatori relazionali</p> <p>La selezione e le selezioni annidate</p> <p>Il costrutto match-case</p> <p>Le enumerazioni</p> <p>Algebra booleana, il dato bool, le espressioni booleane</p> <p>Ciclo while</p> <p>Ciclo for</p> <p>La struttura dati array e introduzione alle liste</p>	<p>software in varie attività.</p> <p>Sapere gestire l'output generato dal programma, secondo adeguate formattazioni.</p> <p>Sapere utilizzare le variabili per memorizzare dati in base al tipo di dato da gestire.</p> <p>Sapere inserire commenti nel codice per spiegare la logica dietro le istruzioni. con la consapevolezza del rendere agevole l'accesso alla comprensione del codice stesso. Sapere applicare gli operatori matematici e logici, tra cui quelli di confronto e modulo.</p> <p>Sapere comporre codice con istruzioni condizionali, anche sviluppando codice complesso con blocchi di istruzioni annidate.</p> <p>Sapere combinare più selezioni in un programma, tra i quali l'istruzione di selezione multipla. Sapere scrivere programmi che includano selezioni e iterazioni, individuando e utilizzando tra le diverse forme di iterazione quella più idonea. Sapere individuare e correggere errori presenti nel codice.</p>	<p>soluzione ad un problema posto.</p> <p>Essere in grado di implementare un algoritmo in pseudo-codice e/o in un linguaggio di programmazione specifico, applicando correttamente la sintassi del linguaggio affrontato.</p>	<p>pratica dei principi appresi.</p> <p>Sessioni di discussione sull'etica e le regole di utilizzo di Internet, con enfasi sulla comunicazione online responsabile e sulle problematiche legate alla privacy e alla sicurezza.</p>	<p>modo attivo e di implementare algoritmi in pseudo-codice o linguaggi specifici, applicando correttamente la sintassi e promuovendo la creatività nel risolvere problemi.</p>
--	--	--	--	---	---	--	---

2° biennio	Algoritmi e linguaggi di programmazione	Le strutture dati Le strutture dati avanzate Funzioni e sottoprogrammi	Cicli, successioni e liste La struttura dati tupla La struttura dati set I dizionari Le strutture dati avanzate Pila Coda Introduzione agli alberi	Saper sfruttare le strutture selettive e iterative come strumento nella gestione di dati strutturati. Saper identificare la tipologia adatta di struttura dati sulla base di casi proposti, operando quindi con tali strutture dati e con le loro funzioni. Saper predisporre soluzioni ai problemi operando con la varietà delle strutture dati proposte all'interno di casi reali.	Essere in grado di gestire dati strutturati anche in modo avanzato, utilizzando con competenza strutture selettive e iterative in funzione delle specifiche del software da implementare. Identificare e applicare la struttura dati adatta per risolvere scenari proposti, operando sulle strutture dati e specifiche tramite le relative funzioni. Predisporre soluzioni a problemi anche complessi, applicando la varietà di strutture dati in contesti che richiamano quelli reali. Utilizzare con competenza funzioni built-in e creare funzioni customizzate per adattarsi a esigenze specifiche emerse dalle analisi di problemi e progetti sottoposti. Comprendere e implementare in modo efficiente funzioni ricorsive, secondo le	Approccio pratico e orientato all'applicazione diretta. Risoluzione di problemi concreti consentirà loro di applicare le conoscenze acquisite in contesti reali, affinando la capacità di identificare e utilizzare le strutture dati più adeguate. Presentazione di scenari che richiedono l'integrazione di diverse strutture dati per una risoluzione efficace. Analisi critica e guidata delle situazioni proposte e nell'elaborazione di soluzioni strutturate ed efficienti. Progettazione e implementazione di database reali, sviluppando la capacità di applicare principi di progettazione logica e normalizzazione e	Dimostrare competenza nella gestione avanzata di dati strutturati, utilizzando con perizia strutture selettive e iterative in base alle specifiche del software, applicando soluzioni a problemi complessi con varietà di strutture dati in contesti reali. Utilizzare proficuamente le funzioni built-in, anche facendo riferimento ad un uso consapevole della documentazione ufficiale. Creare funzioni customizzate per rispondere alle esigenze specifiche emerse dall'analisi di problemi e progetti, comprendendo ed implementando in modo efficiente il codice secondo regole logiche, deduttive e matematiche quando necessario. Gestire con competenza le operazioni sui file, con l'obiettivo di elaborare i dati
	Algoritmi e linguaggi di programmazione	Le funzioni	Funzioni built-in Funzioni customizzate Funzioni ricorsive	Sapere utilizzare in modo competente le funzioni built-in, sapendo anche approcciare alla manualistica ufficiale reperibile in rete. Sapere creare e implementare funzioni customizzate per adattarsi alle necessità specifiche. Sapere comprendere e implementare funzioni ricorsive, sfruttando la capacità di richiamare sé stesse per risolvere problemi complessi in modo efficiente.			
	Strumenti avanzati di produzione dei documenti elettronici	Operare con i file	Accesso ai file Le operazioni su file: creazione ed eliminazione Operazioni di lettura da file Operazioni di scrittura su file	Sapere gestire l'accesso ai file. Sapere eseguire operazioni di creazione ed eliminazione dei file. Sapere effettuare operazioni di lettura da file, prevedendo manipolazioni adeguate			

			La modalità append Elaborazione di dati estrapolati dai file	anche in un ottica di analisi delle informazioni. Sapere eseguire operazioni di scrittura su file, anche in modalità append per aggiungere dati a un file esistente, ed elaborare i dati estratti da file per ulteriori manipolazioni e analisi.	specifiche regole logiche, deduttive e anche matematiche quando necessario. Saper gestire l'accesso ai file, prevedendo la corretta implementazione delle operazioni di creazione, eliminazione, lettura, scrittura e append, nell'ottica del saper elaborare i dati estratti dai file in funziona di ulteriori manipolazioni e analisi. Utilizzare HTML per la strutturazione di pagine web, gestendo adeguatamente le sezioni "head" e "body" e applicando le formattazioni con i fogli di stile. Creare elenchi, tabelle e gestire immagini. Implementare link interni ed esterni. Utilizzare JavaScript come linguaggio di scripting lato client per aggiungere dinamicità alle pagine web e gestirne gli eventi lato client.	acquisendo competenze pratiche nell'utilizzo di linguaggi di manipolazione dei database	estratti dai file per ulteriori manipolazioni e analisi, anche nell'ambito specifico delle discipline di indirizzo del corso di studi. Utilizzare i linguaggi del web come strumento per una comunicazione digitale in rete, secondo adeguate specifiche che consentano la fruizione di contenuti di valore scientifico. Dimostrare la capacità di definire classi per organizzare il codice, progettare e proporre l'interazione e la struttura di un sistema software, secondo le specifiche di sicurezza dei dati, sulla base di contesti reali e specialistici, in un'ottica applicativa e progettuale che miri alla solidità del software nelle sue fasi del ciclo di vita. Mostrare competenza nella gestione di database e sistemi di gestione dei database (DBMS), partendo da una progettazione
La progettazione web	La progettazione web Il linguaggio HTML I fogli di stile CSS Gli script lato client	Tag Head e body I metatag La formattazione dei testi e le proprietà delle pagine Elenchi puntati ed elenchi numerati Tabelle Immagini Link interni ed esterni Div e span Gli attributi id e class JavaScript Operare con variabili in JavaScript Espressioni e operatori Selezioni e switch case Le iterazioni con while e do-while Il ciclo for	Sapere utilizzare i tag HTML per la strutturazione di pagine web. Sapere gestire le sezioni "head" e "body" per organizzare il contenuto. Sapere applicare formattazioni e proprietà di pagina con fogli di stile. Sapere creare elenchi, tabelle, e gestire immagini. Sapere implementare link interni ed esterni. Sapere utilizzare i tag "div" e "span" per organizzare il layout. Sapere sfruttare gli attributi "id" e "class" per identificare e stilizzare elementi. Sapere introdurre JavaScript per la dinamicità delle pagine. Sapere operare con variabili, espressioni, e operatori in JavaScript. Sapere implementare con consapevolezza selezioni, switch case e cicli in JavaScript per				

				operare sulla base di considerazioni logiche.	Definire classi per organizzare il codice, gestendone attributi e metodi all'interno, utilizzando secondo le regole i costruttori e implementando l'ereditarietà. Utilizzare diagrammi UML dei casi d'uso e delle classi per visualizzare l'interazione e la struttura del sistema, applicando la corretta visibilità degli attributi e dei metodi nelle classi. Implementare la generalizzazione e utilizzare le associazioni per collegare le classi e le regole derivanti per progettare in maniera solida il software. Gestire database e sistemi di gestione dei database (DBMS)		che prende avvio da analisi induttive anche di ambiti settoriali, prevedendo l'interazione con le pagine web per ottenere una corretta proiezione dei dati estrapolati, che possa risultare uno strumento utile come competenza fruibile in una proiezione dello studente nel futuro contesto lavorativo.
	Implementazione di un linguaggio di programmazione, metodologie di programmazione, sintassi di un linguaggio orientato agli oggetti.	La programmazione orientata agli oggetti	Classi Attributi e metodi Costruttori Ereditarietà Incapsulamento Polimorfismo	Sapere definire classi per organizzare il codice in programmazione. Sapere distinguere tra attributi e metodi all'interno delle classi, operando e implementando correttamente tali costrutti. Sapere utilizzare costruttori per inizializzare gli oggetti delle classi. Sapere implementare ereditarietà per creare nuove classi basate su quelle esistenti. Sapere applicare l'incapsulamento per proteggere gli attributi e i metodi all'interno delle classi. Sapere comprendere e implementare il polimorfismo per consentire a oggetti di una stessa classe di specializzarsi tramite codifiche customizzate.			
	Introduzione al modello relazionale dei dati	La progettazione del software	Diagrammi dei casi d'uso delle classi Visibilità Dati derivati e dati di default Generalizzazione Associazioni Composizione e aggregazione Molteplicità degli attributi	Sapere utilizzare i diagrammi dei casi d'uso e delle classi in UML per visualizzare l'interazione e la struttura del sistema. Sapere gestire la visibilità degli attributi e dei metodi nelle classi. Sapere definire dati derivati e dati di default all'interno delle classi. Sapere implementare la	partendo da un'adeguata progettazione logica e applicando le regole della normalizzazione. Effettuare la progettazione fisica, considerando aspetti come l'archiviazione e l'accesso ai dati e utilizzando i		

			<p>Molteplicità delle relazioni Joint, disjoint e complete</p>	<p>generalizzazione per creare relazioni di ereditarietà. Sapere utilizzare le associazioni per collegare le classi secondo legami forti e deboli. Sapere distinguere tra composizione e aggregazione nelle relazioni tra classi. Sapere gestire la molteplicità degli attributi, specificando il numero di istanze connesse. Sapere gestire la molteplicità delle relazioni tra classi. Sapere utilizzare i concetti di joint, disjoint e complete per definire le relazioni tra classi in modo dettagliato.</p>	<p>linguaggi di manipolazione dei database, con particolare attenzione all'SQL, prevedendo se possibile l'interazione con le pagine web al fine di ottenere una corretta proiezione dei dati estrapolati.</p>		
Introduzione al modello relazionale dei dati, ai linguaggi di interrogazione e manipolazione dei dati	Database	<p>Database DBMS La progettazione logica La progettazione concettuale La normalizzazione La progettazione fisica I linguaggi di manipolazione I linguaggi DQL: l'SQL</p>	<p>Sapere gestire i database e i sistemi di gestione dei database (DBMS). Sapere eseguire la progettazione logica e fisica di un database richiamando contesti reali. Sapere applicare il processo di normalizzazione per ottimizzare la struttura del database. Sapere effettuare la progettazione fisica di un database, considerando aspetti come l'archiviazione e l'accesso ai dati. Sapere utilizzare i linguaggi di manipolazione dei</p>				

				database, al fine di strutturare adeguatamente la base di dati e procedere con query ben strutturate del linguaggio SQL (Structured Query Language).			
<b>5° anno</b>	Introduzione ai principi teorici della computazione	Sistemi e modelli Teoria degli automi Teoria della calcolabilità La complessità computazionale Intelligenza artificiale e reti neurali	Sistemi, modelli e loro caratteristiche Teoria degli automi Teoria della calcolabilità La Macchina di Turing e la Macchina di Turing universale La complessità computazionale Algoritmi e studio della complessità Classi di complessità AI debole e forte AI e robotica Sistemi esperti Componenti e approccio operativo delle reti neurali Algoritmi genetici Logica fuzzy	Sapere analizzare sistemi e modelli informatici, identificandone le caratteristiche distintive e le collocazioni delle varie tipologie all'interno di contesti reali.. Sapere applicare la teoria degli automi per comprendere i concetti di base nell'ambito computazionale e impostandone le corrette strutture progettuali. Sapere esplorare la teoria della calcolabilità, sulla base delle regole della Macchina di Turing (e la sua versione universale) come presupposto per le stime del costo algoritmico. Sapere valutare la complessità computazionale e comprendere gli algoritmi associati. Sapere distinguere e applicare classi di complessità in contesti computazionali. Sapere differenziare l'intelligenza artificiale debole e forte, nonché	Essere in grado di analizzare sistemi e modelli informatici, operando in maniera corretta e consapevole alla progettazione di automi individuati tra contesti reali e/o logico-matematici. Riconoscere le peculiarità dell'AI debole e forte, prevedendo l'individuazione di riscontri anche in contesti reali e richiami all'attualità., Saper identificare e comprendere vari tipi di reti informatiche, analizzando le varie topologie di rete e identificando quindi tecniche di commutazione volte alla gestione dei pacchetti nella rete. Gestire indirizzi IP su reti e contesti reali, anche nell'ottica della sicurezza dei dati	Realizzazione di laboratori con schede programmabili volti all'intercettazione di valori dal mondo fisico, comprendendo il concetto di sensore e della manipolazione da analogico a digitale dei valori intercettati. Richiami ad algoritmi precedentemente implementati al fine di valutare la complessità computazionale degli stessi, prevedendo sessioni di programmazione volta a verificare la corretta discriminazione del concetto di costo basato su tempo e su risorse, rispetto alla misura di	Comprendere sistemi e modelli informatici attraverso esperienze pratiche e laboratoriali, sviluppando competenze nella progettazione di automi e applicando le conoscenze a contesti reali e logico-matematici. Collaborare attivamente con i pari, contribuendo all'analisi e alla progettazione di automi, acquisendo una comprensione delle tecnologie implementate dall'Intelligenza Artificiale. Costruire conoscenze personali sfruttando strumenti tecnologici e informatici, esplorando risorse interattive e partecipando a esperienze di apprendimento pratiche.

				<p>riconoscerne le applicazioni, incluso il campo della robotica. Saper riconoscere le peculiarità dei sistemi esperti come strumento nell'ambito dell'intelligenza artificiale, anche come presupposto per i concetti di reti neurali, algoritmi genetici e logica fuzzy in contesti computazionali e decisionali.</p>	<p>che potrà avvenire tramite tecniche di crittografia e la valutazione della componente hardware e software delle reti, al fine di distinguere e conoscere le funzionalità disponibili degli apparati per l'interconnessione. Essere in grado di valutare la sicurezza in rete con crittografia simmetrica e asimmetrica. Saper riconoscere l'architettura per il web in funzione della struttura e presentazione, comprendendone le dinamiche relative alla pubblicazione di un sito web</p>	<p>qualità dell'algoritmo stesso. Richiamare i concetti della progettazione di rete implementandoli tramite software in cloud per la rappresentazione internetworking.</p>	<p>Affrontare in modo diretto questioni e problemi applicativi, collaborando in attività che collegano competenze tecniche alle conoscenze scientifico-tecnologiche e alla progettazione in contesti reali e attuali. Ottimizzare l'apprendimento personale, esplorando in maniera critica l'analisi di sistemi informatici, riconoscendo le peculiarità di specifici sistemi, anche valutando consapevolmente la sicurezza dei sistemi e della loro interconnessione in rete. Saper esplorare e ampliare gli interessi personali, acquisendo competenze trasversali che consentano anche di prevedere nuove implementazioni tecnologiche, derivanti dalla consapevolezza delle competenze stesse.</p>
	Reti di computer	Reti di elaboratori	<p>Tipi di rete Le topologie di rete</p>	<p>Saper identificare e comprendere i vari tipi di reti informatiche. Saper distinguere e analizzare le diverse topologie di rete anche sulla base di contesti reali descritti</p>			
	<p>Protocolli di rete Struttura di internet e dei servizi di rete</p>	<p>Tecniche di commutazione I modelli ISO/OSI e TCP/IP Gli indirizzi IP Tipi di rete Sicurezza dei dati e crittografia</p>	<p>La componente hardware La componente software Il modello ISO/OSI I sette livelli del modello Frame Gestione degli errori TCP/IP Apparati per l'interconnessione Indirizzi IP e classi Subnet Peer-to-peer Client-Server Concetto di sicurezza La sicurezza dei dati in rete Crittografia simmetrica</p>	<p>Saper riconoscere le tecniche di commutazione e descriverne gli aspetti tecnici, in funzione anche della comprensione dei modelli ISO/OSI e TCP/IP come strumenti per la comprensione delle dinamiche delle reti di elaboratori. Conoscere gli indirizzi IP e la corretta identificazione delle loro classi sulla base delle specifiche di reti descritte, prevedendo adeguate ipotesi sulla sicurezza dei dati nella sua</p>			

			<p>Crittografia asimmetrica          La firma digitale          Backup, firewall e proxy server</p>	<p>componente hardware e software. Sa riconoscere le impostazioni idonee per consentire l'interconnessione e pianificare adeguatamente indirizzamento IP e subnet. Conosce le caratteristiche di una rete client-server e valutare il concetto di sicurezza in rete, anche sulla base delle nozioni di crittografia e delle funzionalità di backup, firewall e proxy server.</p>			
I servizi di internet	Il livello applicativo	<p>Architettura per il web          Struttura e presentazione          Pubblicazione di un sito          CMS</p>	<p>Saper riconoscere l'architettura per il web considerandola in funzione della struttura e la presentazione. Sapere le dinamiche relative alla pubblicazione di un sito web, sviluppando criterio critico tra le opzioni di hosting e housing, anche in funzione del Content Management Systems (CMS)</p>				