

Ministero dell'Istruzione e del Merito**Giacomo Leopardi – Battipaglia (SA)****Istituto Paritario di Istruzione Secondaria Superiore**

LICEO SCIENTIFICO nuovo ordinamento – IST. TECN. SETT. TECNOLOGICO INFORMATICA E TELECOMUNICAZIONI art. INFORMATICA

Via Belluno, 19 – Codice Meccanografico SAPS09500G – STAFVU500C

Ente gestore: DEA12 SRL Via Belluno, 19 - 84091 Battipaglia (SA) tel/fax 0828371134 mail:segreteria@dea12.it direzione@dea12.it

P.IVA 06196270653 www.dea12.it

UNITÀ DI APPRENDIMENTO DISCIPLINARI

ANNO SCOLASTICO 2024/2025

INDIRIZZO: ITI**DOCENTE: Andrea Di Sessa / Luca Bernasconi****CLASSE: 4 A****N. ALUNNI: 11**

DISCIPLINA: TPSIT

TITOLO UDA1	Sviluppare un software – settembre
LEZIONI UDA	<ol style="list-style-type: none">1. L'ingegneria del software2. Il ciclo di vita del software3. Progetti software e metodologie4. La gestione dei progetti
ATTIVITA' DIDATTICA	<ul style="list-style-type: none">• Lezioni di tipo interattivo• Lezioni di tipo frontale• Lezioni pratiche in laboratorio
CONOSCENZE E ABILITA' CHE L'UDA SI PREFIGGE DI SVILUPPARE	<ul style="list-style-type: none">• Comprendere i concetti fondamentali dell'ingegneria del software e delle sue fasi principali (analisi, progettazione, implementazione, testing e manutenzione).• Conoscere le fasi del ciclo di vita del software e le loro sequenze logiche, inclusi i tipi di manutenzione (correttiva, adattativa e migliorativa).• Conoscere i principi base di gestione dei progetti: pianificazione, monitoraggio, controllo, gestione delle risorse e strumenti di supporto.• Saper analizzare e documentare i requisiti di un progetto software.• Suddividere un progetto software in fasi di sviluppo, pianificando ogni fase del ciclo di vita e documentando le attività.

	<ul style="list-style-type: none"> • Documentare ogni fase di sviluppo e gestione del progetto, adeguando il piano in base ai cambiamenti o alle problematiche emergenti. • Collaborare in team, assegnando ruoli e responsabilità e coordinando le attività di sviluppo.
COMPETENZE CHE L'UDA SI PREFIGGE DI SVILUPPARE	<ul style="list-style-type: none"> • Analizzare e progettare un sistema software completo: dall'analisi dei requisiti fino alla manutenzione, selezionando la metodologia e il modello di sviluppo più adatti al contesto. • Gestire e monitorare un progetto software: pianificando le attività, controllando le risorse e monitorando l'avanzamento con strumenti di project management. • Lavorare in team per lo sviluppo di un progetto: gestendo la collaborazione e l'organizzazione delle attività per raggiungere gli obiettivi di progetto. • Adattare la gestione dei progetti alle esigenze emergenti, implementando una revisione continua delle attività e garantendo flessibilità nei processi.
VERIFICA E VALUTAZIONE	<ul style="list-style-type: none"> • Prove di verifica di fine modulo • Valutazione delle attività di laboratorio

TITOLO UDA2	Il WWW e le pagine Web – ottobre / dicembre
LEZIONI UDA	<ol style="list-style-type: none"> 1. Il World Wide Web 2. Gli elementi base di HTML 3. Le pagine web con link, immagini, elenchi e tabelle 4. Lo stile delle pagine web: CSS 5. Le classi gli identificatori e altri selettori 6. Il box model e i CSS layout
ATTIVITA' DIDATTICA	<ul style="list-style-type: none"> • Lezioni di tipo interattivo • Lezioni di tipo frontale • Lezioni pratiche in laboratorio
CONOSCENZE E ABILITA' CHE L'UDA SI PREFIGGE DI SVILUPPARE	<ul style="list-style-type: none"> • Comprendere il funzionamento del World Wide Web e i protocolli di base (HTTP, HTTPS). • Conoscere la struttura e gli elementi fondamentali di HTML per creare una pagina web (es. tag, attributi, elementi strutturali). • Conoscere i principali elementi di una pagina web come link, immagini, elenchi, tabelle e la loro sintassi in HTML. • Comprendere il ruolo dei fogli di stile CSS per definire l'aspetto di una pagina web e separare la struttura dal design. • Conoscere l'utilizzo di classi, identificatori e selettori avanzati in CSS per applicare stili specifici agli elementi. • Comprendere il box model in CSS e i concetti fondamentali per gestire layout di pagina (padding, margin, border) e organizzare contenuti con layout CSS (es. flexbox, grid).

	<ul style="list-style-type: none"> • Creare pagine web utilizzando HTML, inserendo link, immagini, elenchi e tabelle in modo corretto. • Utilizzare CSS per stilizzare le pagine web, applicando colori, font, bordi e background agli elementi. • Applicare correttamente classi e identificatori per strutturare il design e utilizzare selettori avanzati per stili specifici. • Implementare il box model per gestire margini, padding e bordi, migliorando la disposizione e la leggibilità degli elementi. • Utilizzare layout CSS come flexbox e grid per organizzare i contenuti in maniera responsiva e coerente. • Creare progetti semplici che combinano HTML e CSS, dimostrando padronanza nell'applicazione degli stili e nella costruzione di layout.
COMPETENZE CHE L'UDA SI PREFIGGE DI SVILUPPARE	<ul style="list-style-type: none"> • Progettare e realizzare pagine web statiche che rispettino le basi della sintassi HTML e applichino correttamente gli stili CSS. • Utilizzare classi, identificatori e selettori CSS per garantire un controllo preciso sull'aspetto e la disposizione dei contenuti in una pagina web. • Gestire il layout di una pagina utilizzando il box model e tecniche di layout CSS (flexbox, grid), creando una struttura visiva coerente e responsiva. • Integrare contenuti multimediali e strutturali (immagini, tabelle, elenchi) in una pagina web, garantendo funzionalità e accessibilità
VERIFICA E VALUTAZIONE	<ul style="list-style-type: none"> • Prove di verifica di fine modulo • Valutazione delle attività di laboratorio

TITOLO UDA3	Introduzione ai sistemi operativi – gennaio
LEZIONI UDA	<ol style="list-style-type: none"> 1. Il ruolo del sistema operativo 2. Il Kernel e la Shell 3. Il modello a ring con user mode e kernel mode 4. Tipologie di sistemi operativi 5. Prestazioni del sistema operativo 6. La gestione della CPU 7. La gestione degli interrupt
ATTIVITA' DIDATTICA	<ul style="list-style-type: none"> • Lezioni di tipo interattivo • Lezioni di tipo frontale • Lezioni pratiche in laboratorio
	<ul style="list-style-type: none"> • Comprendere il ruolo e le funzioni principali del sistema operativo, inclusa la gestione delle risorse hardware e software.

<p>CONOSCENZE E ABILITA' CHE L'UDA SI PREFIGGE DI SVILUPPARE</p>	<ul style="list-style-type: none"> • Conoscere la struttura del sistema operativo, inclusi i concetti di Kernel e Shell, e il loro ruolo nella comunicazione tra hardware e software. • Comprendere il modello a ring, con user mode e kernel mode, e la separazione tra modalità di esecuzione per sicurezza e controllo del sistema. • Conoscere le diverse tipologie di sistemi operativi • Comprendere i fattori che influenzano le prestazioni del sistema operativo, inclusi tempi di risposta, velocità di elaborazione e gestione delle risorse. • Conoscere i meccanismi di gestione della CPU, inclusi i processi di scheduling e le politiche di allocazione del processore. • Comprendere la gestione degli interrupt e il loro ruolo nel coordinamento delle attività tra hardware e software. • Descrivere il funzionamento e le componenti principali di un sistema operativo, spiegando il ruolo di Kernel e Shell nella gestione delle risorse. • Identificare e descrivere i vantaggi dell'uso del modello a ring, distinguendo tra user mode e kernel mode. • Confrontare le diverse tipologie di sistemi operativi e scegliere la più adatta a determinati contesti o esigenze. • Analizzare le prestazioni di un sistema operativo, valutando i principali fattori che influenzano la gestione delle risorse e il tempo di risposta. • Applicare concetti di gestione della CPU e spiegare il funzionamento degli algoritmi di scheduling più comuni (es. round-robin, priority scheduling). • Descrivere il funzionamento degli interrupt e il loro utilizzo per coordinare l'esecuzione tra processi e periferiche.
<p>COMPETENZE CHE L'UDA SI PREFIGGE DI SVILUPPARE</p>	<ul style="list-style-type: none"> • Analizzare il ruolo del sistema operativo nella gestione delle risorse, descrivendo come Kernel e Shell contribuiscano all'interazione tra hardware e software. • Interpretare il modello a ring e utilizzare i concetti di user mode e kernel mode per migliorare la comprensione della sicurezza e della gestione delle risorse. • Valutare e confrontare diverse tipologie di sistemi operativi in base al contesto d'uso, identificando le soluzioni più appropriate per specifiche esigenze operative. • Valutare le prestazioni di un sistema operativo in base alla gestione della CPU e delle risorse, applicando concetti di scheduling e ottimizzazione delle risorse. • Gestire e spiegare l'importanza degli interrupt per la sincronizzazione tra hardware e software, dimostrando la capacità di integrare questa conoscenza nella comprensione globale del sistema operativo.
<p>VERIFICA E VALUTAZIONE</p>	<ul style="list-style-type: none"> • Prove di verifica di fine modulo • Valutazione delle attività di laboratorio

TITOLO UDA4	Programmazione Web – febbraio / marzo
LEZIONI UDA	<ol style="list-style-type: none"> 1. I form HTML 2. Javascript 3. Javascript- I moduli 4. Javascript: Altre componenti 5. Javascript: Elaborazioni complesse
ATTIVITA' DIDATTICA	<ul style="list-style-type: none"> • Lezioni di tipo interattivo • Lezioni di tipo frontale • Lezioni pratiche in laboratorio
CONOSCENZE E ABILITA' CHE L'UDA SI PREFIGGE DI SVILUPPARE	<ul style="list-style-type: none"> • Comprendere la struttura e la funzione dei form HTML. • Conoscere i fondamenti del linguaggio JavaScript. • Capire il concetto di modulo in JavaScript e la sua utilità. • Esplorare le diverse componenti utilizzate nello sviluppo web (DOM, eventi, ecc.). • Comprendere come implementare algoritmi e strutture dati complessi in JavaScript. • Creare form HTML interattivi utilizzando vari tipi di input. • Manipolare il DOM con JavaScript per creare interfacce dinamiche. • Organizzare il codice JavaScript in moduli per una migliore gestione e riutilizzo. • Utilizzare le librerie JavaScript più comuni (jQuery, React, Angular, ecc.). • Risolvere problemi complessi di programmazione web.
COMPETENZE CHE L'UDA SI PREFIGGE DI SVILUPPARE	<ul style="list-style-type: none"> • Progettare interfacce utente intuitive e user-friendly. • Ottimizzare il codice JavaScript per migliorare le prestazioni. • Collaborare con altri sviluppatori per la creazione di progetti web. • Risolvere problemi di compatibilità tra diversi browser.
VERIFICA E VALUTAZIONE	<ul style="list-style-type: none"> • Prove di verifica di fine modulo • Valutazione delle attività di laboratorio

<p>TITOLO UDA5</p>	<p>Il sistema operativo come gestore di risorse – aprile</p>
<p>LEZIONI UDA</p>	<ol style="list-style-type: none"> 1. La gestione della memoria fisica e virtuale 2. L'allocazione contigua a partizioni 3. L'allocazione non contigua a paginazione dinamica 4. La memoria virtuale a paginazione dinamica 5. La memoria virtuale a segmentazione 6. La gestione delle informazioni: il file system
<p>ATTIVITA' DIDATTICA</p>	<ul style="list-style-type: none"> • Lezioni di tipo interattivo • Lezioni di tipo frontale • Lezioni pratiche in laboratorio
<p>CONOSCENZE E ABILITA' CHE L'UDA SI PREFIGGE DI SVILUPPARE</p>	<ul style="list-style-type: none"> • Comprendere i concetti fondamentali della gestione della memoria fisica e virtuale e l'importanza della gestione efficiente delle risorse di memoria. • Conoscere l'allocazione contigua della memoria e le diverse modalità di partizionamento (statico e dinamico), con i rispettivi vantaggi e limitazioni. • Comprendere l'allocazione non contigua della memoria tramite paginazione, distinguendo tra paginazione statica e paginazione dinamica. • Conoscere il funzionamento della memoria virtuale a paginazione dinamica, il processo di traduzione degli indirizzi e i vantaggi dell'uso della paginazione per la gestione della memoria virtuale. • Comprendere il funzionamento della memoria virtuale a segmentazione e la differenza rispetto alla paginazione, inclusi vantaggi e svantaggi della segmentazione. • Conoscere le basi della gestione delle informazioni tramite il file system, inclusi i concetti di struttura, organizzazione e gestione dei file e delle directory. • Descrivere e confrontare i concetti di memoria fisica e memoria virtuale, spiegando i vantaggi di un uso ottimizzato delle risorse di memoria. • Applicare i principi dell'allocazione contigua a partizioni e identificare i contesti in cui è più indicata, comprendendo limiti come la frammentazione. • Spiegare e applicare il funzionamento dell'allocazione non contigua a paginazione, descrivendo i processi di gestione della memoria e riduzione della frammentazione. • Descrivere il funzionamento della memoria virtuale a paginazione dinamica e applicare i concetti di traduzione degli indirizzi tra memoria virtuale e fisica. • Spiegare la segmentazione della memoria virtuale, descrivendo come il sistema operativo gestisca i segmenti per ottimizzare l'uso della memoria. • Gestire e descrivere il file system, identificando le principali operazioni di creazione, gestione e cancellazione di file e

	directory, e spiegando la loro organizzazione logica e fisica.
COMPETENZE CHE L'UDA SI PREFIGGE DI SVILUPPARE	<ul style="list-style-type: none"> • Gestire la memoria di sistema analizzando e scegliendo tra memoria fisica e virtuale in base alle esigenze operative e al contesto. • Applicare i principi di allocazione della memoria e adattare l'allocazione contigua o non contigua (paginazione o segmentazione) in funzione delle risorse e delle necessità applicative. • Ottimizzare la gestione della memoria virtuale con un uso appropriato di tecniche come la paginazione dinamica e la segmentazione, per migliorare la performance del sistema e ridurre i tempi di accesso. • Gestire le informazioni attraverso il file system organizzando i file e le directory secondo le specifiche richieste e ottimizzando l'organizzazione logica per un accesso efficiente e sicuro.
VERIFICA E VALUTAZIONE	<ul style="list-style-type: none"> • Prove di verifica di fine modulo • Valutazione delle attività di laboratorio

TITOLO UDA6	Progettare e sviluppare interfacce per il web – maggio / giugno
LEZIONI UDA	<ol style="list-style-type: none"> 1. Le interfacce 2. L'interazione con i siti web: Javascript e DOM 3. L'interfaccia utente con i form HTML 4. Il formato JSON e il linguaggio XML
ATTIVITA' DIDATTICA	<ul style="list-style-type: none"> • Lezioni di tipo interattivo • Lezioni di tipo frontale • Lezioni pratiche in laboratorio
CONOSCENZE E ABILITA' CHE L'UDA SI PREFIGGE DI SVILUPPARE	<ul style="list-style-type: none"> • Conoscere i concetti fondamentali delle interfacce utente e il ruolo delle interfacce nell'interazione tra l'utente e le applicazioni web. • Comprendere il funzionamento di JavaScript e del DOM (Document Object Model) per manipolare dinamicamente gli elementi di una pagina web. • Conoscere gli elementi di base per creare interfacce utente con i form HTML, inclusi campi di input, selezioni, pulsanti e i vari attributi per gestire l'interazione dell'utente. • Comprendere i formati JSON e XML come strutture per lo scambio di dati, conoscendone le caratteristiche, le sintassi e gli usi principali nelle applicazioni web. • Progettare interfacce utente per il web, applicando i principi di usabilità e di interazione efficace. • Utilizzare JavaScript e DOM per creare interazioni dinamiche nei siti web, manipolando contenuti ed elementi in risposta agli eventi dell'utente. • Creare e gestire form HTML per raccogliere dati dagli utenti,

	<p>validare le informazioni e inviarle per l'elaborazione.</p> <ul style="list-style-type: none"> • Applicare JSON e XML per rappresentare e scambiare dati tra applicazioni, strutturando e leggendo informazioni in questi formati per la gestione e lo scambio di dati.
<p>COMPETENZE CHE L'UDA SI PREFIGGE DI SVILUPPARE</p>	<ul style="list-style-type: none"> • Sviluppare interfacce utente intuitive che consentano agli utenti di interagire con i siti web in modo efficace e accessibile. • Implementare interazioni dinamiche con JavaScript e DOM, integrando funzionalità interattive che migliorano l'esperienza utente e rendono la navigazione più fluida. • Progettare form HTML completi e funzionali per raccogliere e convalidare dati degli utenti in maniera efficace e sicura. • Gestire formati JSON e XML per il trasferimento e la rappresentazione dei dati in modo strutturato, utilizzando tali formati in applicazioni che richiedono lo scambio di informazioni.
<p>VERIFICA E VALUTAZIONE</p>	<ul style="list-style-type: none"> • Prove di verifica di fine modulo • Valutazione delle attività di laboratorio

LUOGO E DATA

FIRMA DEL DOCENTE

