Modulo B - Conoscenza dell'area "Costruzione"

Sviluppo e integrazione dei sistemi informativi

(traduzione non ufficiale)

Syllabus. Le basi per i test del modulo B

Obiettivi del modulo B

È richiesto al candidato di comprendere lo sviluppo e l'implementazione di sistemi informativi.

Il candidato sarà in grado di:

- Comprendere gli aspetti tecnici della progettazione, dell'analisi delle specifiche, dello sviluppo, del testing, dell'integrazione e dello sviluppo dei sistemi IT.
- Comprendere il ciclo di vita dello sviluppo dei sistemi, il tipico processo di sviluppo ed è consapevole delle tendenze dei recenti sistemi di sviluppo.
- Conoscere i principi e gli usi dei database relazionali e dei data warehouse.
- Comprendere il modello relazione e i linguaggi di interrogazione. È consapevole dell'importanza dei temi dell'amministrazione e della sicurezza dei database
- Comprendere i metodi e le tecniche della progettazione del software, descrivere tipiche strutture di dati e algoritmi ed interpretare i costrutti della programmazione. Comprendere i principi della programmazione orientata agli oggetti.
- Conoscere il tema della manutenzione dei programmi, della documentazione e del testing dei sistemi software
- Conoscere i principi di progettazione relativi all'interfaccia utente, pagine web e ipermedia. Comprendere gli elementi base di HTML e XML e conoscere i diversi tipi di programmazione basata sul web.

Categoria	Argomento	Obiettivi
B.1 Processi e metodi dei sistemi di sviluppo	B.1.1 Software per i sistemi di trattamento dati	• Descrivere un sistema di elaborazione dati e una combinazione di hardware, firmware, sistema operativo, applicazioni, dati di configurazione e dati utente; • Riconoscere ed elencare esempi di sistemi operativi; • Riconoscere ed elencare esempi di applicazioni;
	B.1.2 Sistemi per il ciclo di sviluppo	• Descrivere le tipiche fasi dei sistemi di sviluppo; • Confrontare i diversi modelli classici dello sviluppo del ciclo di vita, come quello a cascata, a spirale, di prototipizzazione, di rilascio incrementale. • Descrivere il ciclo di vita di un sistema nel termini di analisi, sviluppo e distribuzione, uso e manutenzione, abbandono. • Individuare le specifiche richieste e di progettazione, come specifiche tecniche e di organizzazione.
	B.1.3 Strumenti di sviluppo del software	• Individuare l'uso di strumenti nei diversi stadi dello sviluppo, come un ambiente di sviluppo integrato CASE. • Individuare i punti di forza o di debolezza dei diversi ambienti di sviluppo software. • Individuare l'uso di semplici strumenti di sviluppo per la modifica, la compilazione, il testing e il debugging del software
	B.1.4 Sistemi di test e sviluppo	• Descrivere i diversi tipi di test e revisione che possono essere applicati durante il ciclo di sviluppo. • Descrivere i temi principali nelle fasi di sviluppo, come il rilascio del software agli utenti, la migrazione dei dati, l'addestramento degli utenti e il supporto iniziale. • Evidenziare i punti di forza e di debolezza dei diversi approcci di implementazione, come big bang, step by step e core model. • Elencare gli elementi tipici di un manuale utente dei documenti tecnici di riferimento

Categoria	Argomento	Obiettivi
	B.1.5 Sistemi di controllo e sicurezza	• Distinguere tra ambienti di sviluppo, test e produzione e comprendere l'importanza di un approccio strutturato al sistema di rilascio, come un Version Contro System e le procedure di distribuzione del software. • Riconoscere i rischi legati al fallimento di un sistema e evidenziare le misure per proteggere i dati sensibili a vari livelli, come quello fisico e procedurale. • Descrivere le procedure di sicurezza giornaliere in un sistema distribuito come quelle di back up e di controllo di accesso.
	B.1.6 Tendenze nei sistemi di sviluppo	• Descrivere gli approcci standard e innovativi e gli standard dello sviluppo come la metodologia agile, ISO12207, SEI/CMMI. • Comprendere l'impatto dell'architettura delle attuali tecniche di sviluppo, come la 2-tier o 3-tier nelle varianti client/server, la n-tier per il web, le architetture orientate ai servizi, le estensioni mainframe e l'integrazione sui sistemi di sviluppo. • Descrivere la complessità dei moderni "sistemi di sistemi" e degli approcci per gestirne la complessità, come quello dei sistemi autonomi.
B.2 Gestione dati e database	D.2.1 Dati e transazioni	• Riconoscere l'importanza della registrazione di dati permanenti per l'elaborazione di transazioni e sistemi di report . • Descrivere come atomicità, consistenza, isolamento e durabilità aiutino a garantire transazioni sicure. • Riconoscere i temi della progettazione e della manutenzione dei sistemi multiutente, come la ridondanza e l'inconsistenza dei dati, i problemi di integrità, la flessibilità dei dati, l'accesso concorrente e la sicurezza.

Categoria	Argomento	Obiettivi
	B.2.2 Struttura di database	• Distinguere tra un sistema di gestione dei file (FMS) e un sistema di gesrione di database (DBMS). • Descrivere gli elementi di un database, come i file di dati, il dizionario, gli indici, i dati statistici. • Riconoscere le aree di impiego dovei DBMS sono usati, e i benefici da essi offerti. • Riconoscere gli elementi di un DBMS, come il linguaggio di interrogazione, il generatore di report, gli strumenti di amministrazione, il controllo della concorrenza, la gestione delle transazioni, gli strumenti di backup e recupero. • Descrivere i differenti ruoli di un amministratore, di un progettatore/programmatore e di un utente di database.
	B.2.3 Modellazione dei dati	• Definire l'astrazione dei dati e descrivere la differenza tra livello fisico, livello concettuale (logico), e livello di vista utente. • Riconoscere la differenza tra gruppi di modelli di dati, come il modello logico orientato agli oggetti, il modello logico basato su record e campi, il modello fisico dei dati. • Descrivere i principi dei modelli logici, come il modello Gerarchico e il modello Reticolare. • Descrivere i principi dei modelli logici orientati agli oggetti, come il modello Relazionale e il modello orientato agli oggetti.
	B.2.4 Il modello Relazionale	• Riconoscere i benefici del modello Relazionale, come non-ridondanza, flessibilità, scalabilità. • Descrivere i termini importanti del modello Relazionale, come la relazione, la chiave primaria, la chiave esterna, la chiave alternativa, l'integrità referenziale. • Spiegare tramite semplici esempi i processi di normalizzazione nella prima, seconda e terza forma normale.

Categoria	Argomento	Obiettivi
	B.2.5 Linguaggi di interrogazione	• Riconoscere le differenze tra linguaggi di interrogazione procedurali e dichiarativi • Descrivere le operazioni fondamentali dell'algebra relazionale come selezione, proiezione, congiunzione, unione, differenza, prodotto cartesiano, rinomina. • Descrivere gli elementi del linguaggio SQL, come Data Definition Language (DDL), Data Manipulation Language (DML), Data Control Language (DCL). • Comprendere comandi DDL, come Create table, Drop table e Alter table. • Comprendere comandi DCL, come Grant e Revoke.
	B.2.6 Interrogazioni SQL	• Comprendere comandi DML, come Insert, Delete, Update, Select. • Comprendere le clausole SQL, come Where, Order by, Group by. • Riconoscere l'uso delle View a dei comandi SQL, come Commit e Rollback.
	B.2.7 Amministrazione e sicurezza dei database	• Descrivere le principali procedure di amministrazione di database, come definizione dello schema dei dati, organizzazione fisica dei dati, metodi e autorizzazioni per l'accesso ai dati. • Descrivere i problemi di integrità e di sicurezza coperti dall'acronimo CIA (Confidentiality, Integrity, Availability), come quello di vincolo di integrità, perdita accidentale di integrità, perdita accidentale di consistenza, accesso intenzionale indesiderato al database. • Riconoscere esempi di diverse politiche di sicurezza, come sicurezza dal punto di vista degli utenti, sicurezza fisica dei dati, sicurezza del sistema operativo, sicurezza del DBMS. • Descrivere procedure di recupero dati per diversi tipi di perdite dati, come errori logici, errori di sistema, crollo del sistema, errori sul disco.
	B.2.8 Data Warehousing e Data Mining	• Descrivere il concetto e gli elementi del sistema di data warehousing (DW). • Definire il concetto di data mining. • Conoscere i principali usi del sistema DW.

Categoria	Argomento	Obiettivi
B.3 Programmazione	B.3.1 Metodi e tecniche di progettazione software	• Riconoscere le principali caratteristiche dei diversi metodi di progettazione, come quello orientato agli oggetti (OO), il metodo top down, la programmazione strutturata. • Descrivere l'uso dell'astrazione come tecnica di problem solving e progettazione di software. • Riconoscere i bisogni specifici nella progettazione di programmi per i sistemi preesistenti, come la complessità della struttura, scarsa documentazione, sistemi fondamentali per gli scopi delle aziende. • Riconoscere la differenza tra lo sviluppo di software proprietario e open source. • Riconoscere le differenze tra la licenze di software proprietario, open source, free software (libero) e freeware.
	B.3.2 Strutture dati e algoritmi	• Descrivere tipi di dato strutturato e non, riconoscere le diverse strutture dati, come record, vettori, liste di puntatori. • Valutare l'adeguatezza dei tipici algoritmi di ricerca e ordinamento per le diverse strutture di dati.
	B.3.3 Linguaggi di programmazione	• Riconoscere le differenze e i vantaggi dei diversi tipi di linguaggi, come funzionali, procedurali, orientati agli oggetti. • Descrivere l'uso di funzioni, distinguendo tra chiamata per valore e per riferimento. • Definire il termine "sinstassi" e riconoscere la sua importanza nei linguaggi di programmazione. • Riconoscere la differenza tra linguaggi di programmazione compilati e interpretati.

Categoria	Argomento	Obiettivi
	B.3.4 Programmazione orientata agli oggetti (OO)	• Descrivere il concetto di progettazione orientata agli oggetti. • Descrivere il concetto diprogrammazione orientata agli oggetti. • Descrivere il concetto di classe, oggetto, istanza, metodo e delle loro relazioni nella programmazione OO. • Descrivere il concetto di ereditarietà e della sua influenza nella programmazione. • Descrivere il concetto di astrazione, incapsulamento e occultamento dell'informazione (information hiding) • Descrivere come il polimorfismo contribuisce all'efficienza nella progettazione attraverso lo sviluppo di componenti riutilizzabili.
	B.3.5 Costrutti elementari	• Interpretare e valutare istruzioni di input/output. • Interpretare e valutare istruzioni di controllo del flusso. • Interpretare e valutare operazioni aritimetiche e logiche.
	B.3.6 Testing	• Definire concetti base del testing, come errore, difetto, fallimento. Riconoscere i vari livelli di controllo, testing e debugging. • Descrivere i differenti scopi e ambiti dello unit testing, system testing, acceptance testing. • Riconoscere le differenze tra metodologie di test statico e dinamico, e fare esempi di strumenti per il test automatizzato.
	B.3.7 Documentazione e manutenzione	• Descrivere la tipica documentazione per lo sviluppo e la consegna del software, come l'inglese strutturato, gli alberi decisionali, il codice UML, il commento del codice, i diagrammi di flusso. • Descrivere l'importanza del codice ben strutturato e documentato. • Conoscere come documentare le modifiche al software e alla documentazione. • Descrivere metodi per ottenere una manutenzione di qualità, come l'ispezione del codice, le regole per commentare il codice, i documenti tecnici di riferimento.

Categoria	Argomento	Obiettivi
	B.3.8 Esempi di programmazione	• Interpretare piccoli segmenti di programmi costruiti per un dato caso di ipotesi. • Identificare errori o punti di debolezza nel codice e modificarli per rispondere alle richieste del problema.
B.4. Interfaccia per l'utente e web design	Interazione uomo-macchina: linee guida e standard	• Definire i concetti base della teoria della comunicazione come mittente, ricevente e messoggio. • Comprendere come la comunicazione è applicata agli esseri umani, e riconoscere i metodi efficaci per comunicare l'informazione. • Definire il concetto di User Interface ed elencare i diversi tipi di interfacce, come testuale, grafica e acustica. • Riconoscere le diverse tecnologie che comunicano l'informazione ai sensi umani, come tipi di suono, segni visivi, odori sintetici, percezioni tattili. • Elencare modelli per testare l'efficacia dell'interfaccia utente rispetto alle richieste e obiettivi di progetto.
	B.4.2 Progetto grafico	• Riconoscere i concetti di grafica e animazione (bitmap e vettoriale), video e audio digitale, e descrivere le loro differenze, gli utilizzi e i formati standard. • Descrivere i benefici dell'uso di disegni, immagini, colori, animazioni e applicare principi base della progettazione grafica, come bilanciamento, armonia, contrasto e varietà. • Usare strumenti tipici per la manipolazione di immagini, in termini di dimensione, forma, colore, contrasto, trasparenza.
	B.4.3 Web e ipermedia: potenzialità e limiti	• Descrivere la storia di Internet e del World Wide Web • Definire ipertesto e ipermedia e riconoscere la loro importanza della progettazione delle pagine web. • Riconoscere i componenti tipici usati nelle pagine web, come menu orizzontale, menu verticale, mappa del sito, contatti, ricerca, aiuto, ultime modifiche, icone. • Descrivere l'uso e l'importanza dei siti aziendali interni ed esterni.

Categoria	Argomento	Obiettivi
	B.4.4 Metodi e richieste della progettazione web	• Riconoscere i bisogni degli utenti per cui una pagina web è stata progettata. • Riconoscere i rischi dell'eccesso di messaggi in una pagina. • Riconoscere i problemi relativi alla corretta combinazione di colori. • Descrivere le linee guida per lo sviluppo di siti web facili da usare, come la leggibilità, il contenuto prioritario, la navigabilità, la coerenza della navigabilità, il conoscere la propria posizione attuale. • Descrivere i criteri generali di qualità per il testo sul web, come la validazione del codice HTML, la capacità dei browser, la sintesi dei contenuti, il controllo ortografico, le dimensioni leggere della grafica. • Spiegare il bisogno di una facile navigazione su un sito web. • Riconoscere gli strumenti per lo sviluppo di un sito web. • Riconoscere l'uso e lo scopo dei diagramma di struttura per i siti web. • Descrivere i principali metodi usati per la navigazione. • Descrivere alcuni approcci per la progettazione web e le tecniche più comuni come story boards, rough drafts.
	B.4.5 Progettazione di pagine web	• Riconoscere il concetto di linguaggio di segnatura e descrivere le principali caratteristiche del linguaggio HTML. • Usare comandi base del linguaggio HTML e intepretare comandi comandi, come tags,collegamenti, tabelle, liste, caratteri speciali (backgrounds?). • Riconoscere i principi base di grafica per il testo scritto, come dimensione e percentuale di spazio bianco. • Riconoscere gli elementi base del linguaggio XML, i suoi usi, l'evoluzione di HTML in XHTML. • Descrivere il concetto di foglio di stile, come Cascading Style Sheet (CSS) e Extensible Stylesheet Language (XSL). Comprendere il loro uso nella progettazione.

Categoria	Argomento	Obiettivi
	B.4.6 Programmazione basata sul web	• Distinguere tra programmazione lato client e lato server, riconoscere i diversi tipi di linguaggi. • Elencare alcune delle principali sfide nell'integrazione dei sistemi basati sul web con i sistemi esistenti.