

# Giorno 4

Giovedì 30 aprile 2026

Prima prova scritta - Esame di Stato Ingegneria Informatica / Ingegnere dell'Informazione

## Tema del giorno

Sistema informativo aziendale o pubblico: architettura, dati, cloud, sicurezza, backup e continuità operativa.

Obiettivo: imparare a scrivere una risposta completa senza bloccarsi, usando un esempio sede-neutrale e molto riutilizzabile.

## Promemoria strategico

Priorità assoluta: prima prova. Non stiamo preparando in modo pesante la seconda prova, la pratica o l'orale.

Però questo tema lascia basi utili anche dopo: architettura a livelli, dati, ruoli, sicurezza, backup, scalabilità e scelte progettuali.

Livello: ripartenza quasi da zero. Spiegazioni semplici, esempi guidati, niente teoria astratta non necessaria.

# 1. Riaggancio: dove siamo

Nei primi tre giorni hai iniziato a costruire il telaio mentale della prima prova.

Giorno	Tema	Cosa doveva fissare
1	Template base e sistema informativo scolastico	Scopo, attori, requisiti, architettura, dati, sicurezza, conclusione.
2	AI e machine learning	Tecnologie emergenti, dati, benefici, rischi e responsabilità.
3	Cybersecurity	Protezione di sistemi e dati: minacce, accessi, cifratura, backup, log, risposta agli incidenti.
4	Sistema informativo aziendale/pubblico	Mettere insieme architettura, database, cloud, backup, sicurezza e continuità operativa.

## Idea chiave di oggi

Molte tracce d'esame non chiedono codice. Chiedono di spiegare come un ingegnere organizza un sistema: quali utenti, quali dati, quali funzioni, quale architettura, quali rischi e quali misure di sicurezza.

Oggi lavoriamo su una traccia trasferibile: funziona per pubblica amministrazione, azienda, scuola, sanità digitale, sistema documentale o portale interno.

## 2. Piano di studio di oggi

Hai poco tempo e stai recuperando. Quindi non devi leggere tutto come se fosse un libro universitario. Usa questo PDF come guida pratica.

Tempo	Cosa fai	Risultato
0-15 min	Leggi la traccia e il glossario minimo	Capisci le parole base.
15-35 min	Studia il template del giorno	Sai l'ordine con cui scrivere.
35-60 min	Leggi la risposta svolta	Vedi un esempio completo.
60-85 min	Fai gli esercizi guidati	Produci materiale tuo.
85-100 min	Scrivi la mini-risposta finale	Me la incolli in chat per la correzione.

### Regola anti-blocco

Quando non ti viene in mente nulla, non partire da frasi perfette. Parti dalle domande: a cosa serve? chi lo usa? cosa deve fare? quali dati salva? come proteggo il sistema? cosa succede se si rompe? La risposta nasce dalle domande, non dalla memoria.

## 3. Traccia del Giorno 4

### Traccia sede-neutrale

Un'organizzazione pubblica o aziendale vuole digitalizzare la gestione dei propri processi, sostituendo archivi cartacei o applicazioni isolate con un sistema informativo integrato.

Il candidato illustri una possibile soluzione, descrivendo obiettivi, attori, requisiti, architettura, dati trattati, sicurezza, privacy, scalabilità, backup e continuità operativa.

Questa traccia può diventare:

- una pubblica amministrazione che gestisce pratiche e documenti;
- un'azienda che gestisce ordini, clienti, magazzino e fatture;
- un ente sanitario che gestisce prenotazioni, referti e dati clinici;
- una scuola che gestisce studenti, classi, voti e comunicazioni;
- un sistema interno per digitalizzare workflow e approvazioni.

Per il nostro esercizio useremo l'esempio di una pubblica amministrazione o azienda che gestisce pratiche, documenti, utenti e comunicazioni. È più informatico e meno biomedicale.

## 4. Cosa devi capire prima di scrivere

Non devi ricordare tecnologie complicate. Devi capire il senso del problema.

Domanda semplice	Risposta pratica
Che problema c'è?	I dati sono sparsi, cartacei, duplicati o gestiti con strumenti non integrati.
Cosa vuole l'organizzazione?	Un sistema unico o integrato per gestire dati, utenti, documenti e processi.
Chi lo usa?	Operatori, responsabili, amministratori, utenti esterni, tecnici.
Cosa deve fare?	Inserire dati, consultare pratiche, approvare richieste, notificare eventi, produrre report.
Cosa può andare storto?	Accessi non autorizzati, perdita dati, sistema non disponibile, errori, dati incoerenti.
Cosa deve dire l'ingegnere?	Architettura chiara, ruoli, sicurezza, backup, continuità, motivazione delle scelte.

### Frase mentale da ricordare

Un sistema informativo non è solo un programma: è un modo organizzato per gestire dati, persone, processi e tecnologie.

## 5. Glossario minimo - parte 1

Questi termini devono diventare familiari. Non devi impararli come definizioni universitarie: devi sapere come usarli in una risposta.

Termine	Spiegazione semplice	Frase utile da esame
Sistema informativo	Insieme di persone, dati, procedure e strumenti che permettono di gestire informazioni.	Il sistema informativo supporta la gestione integrata dei processi dell'organizzazione.
Sistema informatico	La parte tecnologica: software, server, database, rete, applicazioni.	Il sistema informatico costituisce il supporto tecnologico del sistema informativo.
Processo	Sequenza di attività per ottenere un risultato: per esempio aprire, valutare e chiudere una pratica.	La digitalizzazione permette di rendere i processi più tracciabili e controllabili.
Dato	Informazione elementare: nome, data, stato pratica, documento, importo, ruolo utente.	La qualità del dato è essenziale per evitare errori e decisioni non corrette.
Database	Archivio strutturato in cui i dati vengono memorizzati e interrogati.	Il database consente di memorizzare i dati in modo ordinato e coerente.

## 6. Glossario minimo - parte 2

Termine	Spiegazione semplice	Frase utile da esame
Frontend	Parte visibile all'utente: portale web, app, dashboard.	Il frontend consente agli utenti di interagire con il sistema.
Backend	Parte non visibile che esegue la logica: controlla dati, ruoli, operazioni e regole.	Il backend applicativo gestisce la logica del sistema e comunica con il database.
API	Interfaccia che permette a sistemi diversi di comunicare in modo controllato.	Le API consentono l'integrazione con sistemi esterni o applicazioni preesistenti.
Cloud	Uso di risorse informatiche remote, come server e database, gestite tramite servizi scalabili.	Una soluzione cloud può favorire scalabilità, disponibilità e gestione centralizzata.
On-premise	Sistema installato presso l'organizzazione, su server propri.	Una soluzione on-premise può essere scelta per vincoli specifici di controllo o normativa.
Dashboard	Schermata che mostra dati e indicatori in modo sintetico.	Le dashboard supportano il monitoraggio e le decisioni operative.

## 7. Glossario minimo - parte 3: continuità

Questa è una parte importante perché spesso distingue una risposta debole da una risposta ingegneristica.

Termine	Cosa significa davvero	Errore da evitare
Backup	Copia periodica dei dati, utile se qualcosa viene perso o danneggiato.	Pensare che il backup renda il sistema sempre disponibile.
Replica	Copia aggiornata del database o del sistema su un'altra macchina o sede.	Confondere replica e backup: la replica può copiare anche un errore.
Disaster recovery	Piano per ripristinare il servizio dopo guasti gravi, attacchi o incidenti.	Scrivere solo 'fare backup' senza spiegare il ripristino.
Monitoraggio	Controllo continuo di prestazioni, errori, accessi e disponibilità.	Accorgersi dei problemi solo quando l'utente li segnala.
Log	Registro degli eventi: accessi, modifiche, errori, operazioni importanti.	Usare i log senza proteggere dati sensibili o senza conservarli correttamente.
RTO/RPO	RTO: tempo massimo per ripartire. RPO: quantità massima di dati che si accetta di perdere.	Usare sigle senza spiegarle. Se le usi, spiega in parole semplici.

### Per oggi

Non devi diventare esperto di disaster recovery. Devi saper scrivere che un sistema serio prevede backup, procedure di ripristino, monitoraggio e continuità del servizio.

## 8. Analogia facile: ufficio, sportello, archivio

Immagina un ufficio che gestisce pratiche. Questa analogia ti aiuta a ricordare frontend, backend, database e sicurezza.

Nel mondo reale	Nel sistema informatico	Cosa fa
Sportello o modulo online	Frontend	Raccoglie richieste e mostra informazioni all'utente.
Impiegato che controlla le regole	Backend	Verifica permessi, controlla dati, applica procedure.
Archivio ordinato	Database	Conserva pratiche, documenti, utenti, stati e date.
Chiavi e autorizzazioni	Autenticazione e ruoli	Decidono chi può entrare e cosa può fare.
Registro delle operazioni	Log	Tiene traccia di accessi, modifiche e problemi.
Copia dell'archivio	Backup	Permette il recupero se qualcosa va perso.
Piano di emergenza	Disaster recovery	Spiega come ripartire dopo un problema serio.

### Memoria rapida

Frontend = sportello. Backend = regole. Database = archivio. Sicurezza = chiavi. Log = registro. Backup = copia. Recovery = piano per ripartire.

## 9. Template del Giorno 4

Per questa traccia useremo un template leggermente più ricco del Giorno 1, ma sempre semplice.

1. Introduzione
2. Analisi del problema e contesto
3. Attori e ruoli
4. Requisiti funzionali
5. Requisiti non funzionali
6. Architettura generale
7. Dati e database
8. Sicurezza e privacy
9. Backup, scalabilità e continuità operativa
10. Conclusione

Punto	Domanda da farti
Introduzione	Che cosa vuole ottenere l'organizzazione?
Analisi	Quale processo viene digitalizzato?
Attori	Chi usa il sistema e con quali permessi?
Funzioni	Cosa deve fare il sistema?
Qualità	Come deve comportarsi? Sicuro, disponibile, usabile, scalabile?
Architettura	Frontend, backend, database, API, cloud/on-premise?
Dati	Quali informazioni salvo e come le organizzo?
Sicurezza	Chi può vedere o modificare cosa?
Continuità	Cosa succede se il sistema si rompe o subisce un attacco?
Conclusione	Perché la soluzione è sensata?

## 10. Scaletta ragionata della risposta

Questa è la scaletta che puoi seguire in prova. Non serve inventare tutto: devi riempire i blocchi.

1. Introduzione - Dici che il sistema serve a digitalizzare e integrare processi e dati, riducendo errori e tempi.
2. Contesto - Scegli un esempio: PA che gestisce pratiche, oppure azienda che gestisce ordini e documenti.
3. Attori - Elenca utenti interni, responsabili, amministratori, cittadini/clienti, tecnici.
4. Requisiti funzionali - Inserire, consultare, modificare, approvare, notificare, generare report, gestire ruoli.
5. Requisiti non funzionali - Sicurezza, privacy, disponibilità, usabilità, scalabilità, manutenibilità, affidabilità.
6. Architettura - Frontend web, backend applicativo, database, API, eventuale cloud.
7. Dati - Utenti, pratiche, documenti, stati, date, autorizzazioni, log, notifiche.
8. Sicurezza - Autenticazione, autorizzazione, HTTPS, hashing password, log, controllo accessi.
9. Continuità - Backup, replica, monitoraggio, disaster recovery, test di ripristino.
10. Conclusione - La soluzione migliora efficienza, tracciabilità, sicurezza e qualità del servizio.

## 11. Come scrivere le prime righe

L'inizio deve essere semplice. Non deve sembrare una definizione imparata a memoria.

### Versione debole

Il sistema usa un database e un cloud per gestire dati e utenti.

### Versione migliore

Il sistema informativo proposto ha l'obiettivo di digitalizzare e integrare la gestione dei processi di un'organizzazione, centralizzando dati, documenti e operazioni in un ambiente accessibile in modo controllato dagli utenti autorizzati.

Perché la seconda è migliore? Perché dice: obiettivo, processi, dati, documenti, accesso controllato. In una sola frase hai già aperto molti punti della traccia.

Se non sai come iniziare, usa questa formula:

Il sistema informativo proposto ha lo scopo di [digitalizzare/centralizzare] la gestione di [processo/dati/documenti], consentendo a [attori] di svolgere [funzioni principali] in modo [sicuro/tracciabile/efficiente].

## 12. Attori, ruoli e requisiti

Gli attori servono a capire i permessi. I requisiti servono a capire cosa deve fare il sistema.

Attore	Cosa può fare	Permessi tipici
Utente esterno	Invia richieste, consulta lo stato, carica documenti.	Vede solo le proprie pratiche.
Operatore	Inserisce, aggiorna e verifica pratiche.	Gestisce pratiche assegnate al proprio ufficio.
Responsabile	Approva, valida, controlla report.	Vede pratiche del proprio settore.
Amministratore	Gestisce utenti, ruoli, configurazioni.	Ha privilegi tecnici ma deve essere tracciato.
Tecnico/IT	Manutiene sistema, backup, monitoraggio.	Accesso limitato e controllato ai dati.

### Requisiti funzionali tipici:

- gestione utenti e ruoli;
- inserimento e modifica controllata delle pratiche;
- caricamento e consultazione documenti;
- workflow di approvazione;
- notifiche automatiche;
- ricerca e filtraggio dei dati;
- report e statistiche;
- integrazione con sistemi esterni tramite API.

## 13. Requisiti non funzionali

Qui non dici cosa fa il sistema, ma come deve comportarsi.

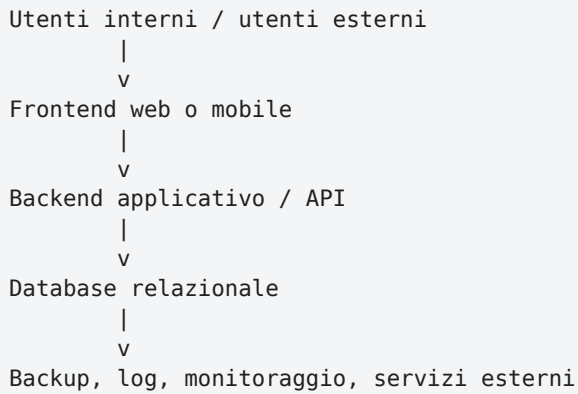
Requisito	Significato semplice	Come lo scrivo
Sicurezza	Proteggere dati e funzioni da accessi non autorizzati.	Accesso tramite autenticazione e autorizzazione basata sui ruoli.
Privacy	Trattare solo i dati necessari e proteggerli.	Minimizzazione dei dati, controllo accessi e tracciamento delle operazioni.
Disponibilità	Il sistema deve essere raggiungibile quando serve.	Ridondanza, monitoraggio e procedure di ripristino.
Usabilità	Gli utenti devono riuscire a usarlo senza difficoltà eccessiva.	Interfaccia chiara, percorsi guidati e messaggi comprensibili.
Scalabilità	Il sistema deve reggere crescita di utenti o dati.	Possibilità di aumentare risorse e separare i componenti.
Manutenibilità	Il sistema deve poter essere aggiornato senza rifarlo da zero.	Architettura modulare, documentazione e separazione dei livelli.
Affidabilità	Ridurre perdita dati, errori e interruzioni.	Backup, log, test e gestione degli errori.

### Frase da esame

I requisiti non funzionali sono fondamentali perché il sistema non deve solo funzionare, ma deve funzionare in modo sicuro, disponibile, affidabile e manutenibile nel tempo.

## 14. Architettura generale

Per oggi scegliamo una soluzione standard, chiara e difendibile: architettura a livelli.



Componente	Cosa fa	Perché serve
Frontend	Mostra schermate, moduli, dashboard.	Permette agli utenti di usare il sistema.
Backend	Controlla regole, permessi, validazioni e flussi.	Evita che la logica sia dispersa nel frontend.
API	Espongono funzioni in modo controllato.	Permettono integrazioni con altri sistemi.
Database	Memorizza dati e relazioni.	Garantisce ordine, coerenza e interrogazioni.
Cloud/Server	Esegue applicazione e conserva risorse.	Permette disponibilità, scalabilità e gestione centralizzata.
Monitoraggio	Controlla errori, prestazioni, accessi.	Serve per accorgersi dei problemi e reagire.

## 15. Cloud o server interno? Come parlarne senza complicarti

Non devi scegliere sempre cloud. Devi mostrare che sai valutare pro e contro.

Soluzione	Vantaggi	Attenzioni
Cloud	Scalabilità, accesso remoto, servizi gestiti, backup e monitoraggio più semplici da organizzare.	Contratti, localizzazione dati, sicurezza, dipendenza dal fornitore, costi nel tempo.
On-premise	Maggiore controllo diretto sull'infrastruttura e sui dati.	Richiede personale, manutenzione, aggiornamenti, protezione fisica e continuità.
Ibrida	Combina sistemi interni e servizi cloud.	Richiede integrazione, governance e gestione chiara delle responsabilità.

### Scelta consigliata per la traccia

Puoi proporre una soluzione cloud o ibrida, purché tu dica che sicurezza, ruoli, backup, monitoraggio e protezione dei dati devono essere progettati esplicitamente. Il cloud non è automaticamente sicuro: va configurato e governato.

## 16. Dati e database

Per una traccia di sistema informativo devi sempre dire quali dati gestisci. Non serve disegnare un database perfetto, ma devi mostrare ordine.

Entità / tabella	Esempi di campi	Relazioni
Utente	id, nome, email, ruolo, stato account	Un utente ha uno o più ruoli.
Ruolo	id, nome ruolo, permessi	Un ruolo abilita certe funzioni.
Pratica	id, oggetto, stato, data apertura, responsabile	Una pratica appartiene a un utente o ufficio.
Documento	id, nome file, tipo, data, pratica collegata	Un documento può essere allegato a una pratica.
Notifica	id, destinatario, messaggio, data, stato	Una notifica avvisa un utente.
Log	id, utente, azione, data, esito	Registra operazioni rilevanti.

### Perché database relazionale?

È una scelta adatta quando i dati sono strutturati e collegati tra loro: utenti, ruoli, pratiche, documenti, stati, date e autorizzazioni. Permette vincoli, relazioni e interrogazioni chiare.

#### Frase da esame

Un database relazionale è adatto perché consente di rappresentare in modo strutturato le relazioni tra utenti, ruoli, pratiche, documenti e stati del processo.

## 17. Sicurezza e privacy

Qui recuperi anche il Giorno 3. Sicurezza non significa solo password: significa proteggere dati, funzioni e continuità del servizio.

Misura	Cosa significa	Esempio nella traccia
Autenticazione	Verificare l'identità dell'utente.	Login con credenziali e, per ruoli critici, autenticazione a più fattori.
Autorizzazione	Stabilire cosa può fare ogni utente.	Un operatore vede solo pratiche del suo ufficio; l'amministratore gestisce utenti.
HTTPS	Cifrare la comunicazione tra utente e sistema.	Protegge i dati durante l'accesso via web.
Hashing password	Memorizzare le password in forma non leggibile.	Non si conservano password in chiaro.
Cifratura dati	Proteggere dati particolarmente sensibili anche a riposo.	Utile per documenti riservati o dati personali.
Log e audit	Registrare accessi e operazioni rilevanti.	Serve per controlli, responsabilità e ricostruzione incidenti.
Minimizzazione	Raccogliere solo dati necessari.	Evitare dati non utili al processo.

### Errore da evitare

Non scrivere 'password cifrate' come unica misura. Meglio: password memorizzate tramite funzioni di hashing sicure, comunicazioni protette tramite HTTPS e autorizzazione basata sui ruoli.

## 18. Backup, scalabilità e continuità operativa

Questa è la parte che fa sembrare la risposta più matura. Anche se non sai tutti i dettagli, devi dimostrare che pensi al sistema nel tempo.

Aspetto	Cosa scrivere
Backup	Prevedere backup periodici del database e dei documenti, conservati in modo sicuro e testati.
Ripristino	Definire procedure per ripristinare il servizio in caso di errore, guasto o attacco.
Replica	Per sistemi critici, usare replica del database o ridondanza dei componenti.
Monitoraggio	Controllare disponibilità, tempi di risposta, errori, accessi anomali e saturazione risorse.
Scalabilità	Separare frontend, backend e database permette di aumentare risorse o istanze quando gli utenti crescono.
Manutenzione	Aggiornamenti, patch di sicurezza, documentazione e test devono essere pianificati.

### Frase completa:

*Per garantire continuità operativa, il sistema deve prevedere backup periodici, procedure di ripristino, monitoraggio dello stato dei servizi e, per i componenti più critici, meccanismi di replica o ridondanza.*

## 19. Risposta svolta completa

Leggila lentamente. Non devi copiarla a memoria: devi capire come ogni paragrafo corrisponde a un punto del template.

Il sistema informativo proposto ha l'obiettivo di digitalizzare e integrare la gestione dei processi di un'organizzazione pubblica o aziendale, sostituendo archivi cartacei o applicazioni isolate con una piattaforma unica e controllata. In questo modo è possibile migliorare l'efficienza operativa, ridurre errori e duplicazioni, aumentare la tracciabilità delle operazioni e rendere più semplice l'accesso alle informazioni da parte degli utenti autorizzati.

Un possibile contesto applicativo è la gestione di pratiche, documenti e comunicazioni interne. Gli attori principali sono gli utenti esterni, che possono inviare richieste e consultare lo stato delle pratiche; gli operatori, che gestiscono e aggiornano le informazioni; i responsabili, che validano o approvano determinate attività; gli amministratori di sistema, che configurano utenti e ruoli; e il personale tecnico, che cura manutenzione, monitoraggio e continuità del servizio.

Tra i requisiti funzionali rientrano la gestione degli utenti e dei ruoli, l'inserimento e la modifica delle pratiche, il caricamento e la consultazione dei documenti, la ricerca dei dati, l'invio di notifiche, la produzione di report e l'integrazione con eventuali sistemi esterni tramite API. Il sistema deve inoltre permettere di tracciare lo stato di avanzamento dei processi, così da sapere chi ha svolto una determinata operazione e in quale momento.

I requisiti non funzionali sono altrettanto importanti. Il sistema deve essere sicuro, disponibile, usabile, scalabile, manutenibile e affidabile. In particolare, la sicurezza e la privacy sono centrali perché il sistema può trattare dati personali, documenti riservati e informazioni rilevanti per l'organizzazione. L'interfaccia deve essere semplice per gli utenti, mentre l'architettura deve consentire aggiornamenti e crescita nel tempo.

Dal punto di vista architetturale, una soluzione efficace può essere organizzata su più livelli: un frontend web o mobile per l'interazione con gli utenti, un backend applicativo che gestisce la logica del sistema, i controlli e le autorizzazioni, e un database relazionale per la memorizzazione strutturata dei dati. Le API possono consentire l'integrazione con altri sistemi, come servizi di autenticazione, protocolli di notifica o archivi documentali. L'infrastruttura può essere realizzata in cloud, on-premise o in modalità ibrida, valutando vincoli di sicurezza, costi, disponibilità e gestione dei dati.

Il database può contenere entità come Utente, Ruolo, Pratica, Documento, Notifica e Log. Un database relazionale è adatto perché consente di rappresentare in modo ordinato le relazioni tra utenti, pratiche, documenti, stati del processo e permessi. La qualità dei dati deve essere garantita attraverso controlli di validazione, vincoli di integrità e procedure di aggiornamento controllate.

La sicurezza deve essere progettata fin dall'inizio. L'accesso al sistema deve avvenire tramite autenticazione, preferibilmente con misure più robuste per gli utenti privilegiati. L'autorizzazione deve essere basata sui ruoli, in modo che ogni utente possa svolgere solo le operazioni consentite. Le comunicazioni devono essere protette tramite HTTPS, le password devono essere memorizzate tramite funzioni di hashing sicure e le operazioni rilevanti devono essere registrate tramite log e audit.

Per garantire affidabilità e continuità operativa, il sistema deve prevedere backup periodici, procedure di ripristino, monitoraggio dei servizi e gestione degli incidenti. Nei contesti più critici può essere opportuno introdurre replica del database, ridondanza dei componenti applicativi e test periodici delle procedure di disaster recovery. La scalabilità può essere favorita separando frontend, backend e database, così da poter aumentare le risorse in caso di crescita del numero di utenti o del volume dei dati.

In conclusione, un sistema informativo integrato consente di migliorare la gestione dei processi, aumentare la tracciabilità, ridurre errori e tempi operativi e garantire maggiore sicurezza nella gestione delle informazioni. Il ruolo dell'ingegnere è progettare una soluzione coerente con gli obiettivi dell'organizzazione, valutando non solo gli aspetti tecnologici, ma anche sicurezza, privacy, affidabilità, costi e sostenibilità nel tempo.

## 20. Perché questa risposta funziona

La risposta svolta non è perfetta perché usa parole difficili. Funziona perché è ordinata.

Parte della risposta	A cosa serve
Obiettivo e contesto	Fa capire subito il problema.
Attori	Mostra che hai capito chi usa il sistema.
Requisiti funzionali	Elenca le funzioni senza confondere il lettore.
Requisiti non funzionali	Dimostra maturità progettuale.
Architettura	Mostra una soluzione tecnica credibile.
Dati	Fa vedere che sai quali informazioni gestire.
Sicurezza	Risponde a privacy, accessi e responsabilità.
Continuità	Mostra che pensi anche a guasti e crescita.
Conclusione	Chiude spiegando il valore della soluzione.

### La regola

Una buona risposta d'esame non deve sembrare un trattato. Deve sembrare una soluzione pensata da un ingegnere: ordinata, motivata, realistica.

## 21. Versione compatta da ricordare

Questa versione è più breve. Ti serve quando vuoi ripassare velocemente o quando devi sbloccarti.

Il sistema informativo ha lo scopo di digitalizzare e integrare la gestione dei processi e dei dati di un'organizzazione.

Gli attori principali sono utenti esterni, operatori, responsabili, amministratori e personale tecnico.

Le funzioni principali sono inserimento, consultazione, modifica, approvazione, notifiche, report e gestione dei ruoli.

I requisiti non funzionali riguardano sicurezza, privacy, disponibilità, usabilità, scalabilità, manutenibilità e affidabilità.

L'architettura può essere organizzata su frontend, backend applicativo, API e database relazionale, eventualmente su infrastruttura cloud o ibrida.

La sicurezza si basa su autenticazione, autorizzazione per ruoli, HTTPS, hashing delle password, log, backup e controllo degli accessi.

La continuità operativa richiede backup, monitoraggio, procedure di ripristino e, per sistemi critici, replica o ridondanza.

In conclusione, la soluzione migliora efficienza, tracciabilità, qualità del servizio e protezione delle informazioni.

## 22. Frasi pronte da usare

1. Il sistema informativo proposto ha lo scopo di centralizzare e digitalizzare la gestione dei processi, riducendo duplicazioni, tempi operativi ed errori.
2. Gli attori del sistema devono essere associati a ruoli e permessi differenti, in modo da limitare l'accesso alle sole funzioni necessarie.
3. I requisiti funzionali descrivono le operazioni che il sistema deve consentire, come inserimento, consultazione, aggiornamento, approvazione e generazione di report.
4. I requisiti non funzionali riguardano qualità come sicurezza, privacy, disponibilità, usabilità, scalabilità, manutenibilità e affidabilità.
5. L'architettura può essere organizzata a livelli, separando frontend, backend applicativo, database e servizi di integrazione.
6. Un database relazionale è adatto quando i dati sono strutturati e collegati tra loro, come utenti, ruoli, pratiche, documenti e stati del processo.
7. La sicurezza deve essere garantita tramite autenticazione, autorizzazione basata sui ruoli, HTTPS, hashing delle password e tracciamento delle operazioni.
8. La continuità operativa richiede backup periodici, monitoraggio, procedure di ripristino e test delle misure adottate.
9. La scelta tra cloud, on-premise o soluzione ibrida deve essere motivata considerando costi, sicurezza, disponibilità, scalabilità e vincoli sui dati.
10. Il ruolo dell'ingegnere è progettare una soluzione coerente con gli obiettivi organizzativi, valutando anche rischi, responsabilità e sostenibilità nel tempo.

## 23. Sistema di memoria

Hai poca memoria: quindi non devi ricordare pagine. Devi fissare poche ancore.

### Ancora 1: la triade tecnica

Frontend = interfaccia. Backend = logica. Database = dati.

### Ancora 2: la triade di controllo

Autenticazione = chi sei. Autorizzazione = cosa puoi fare. Log = cosa hai fatto.

### Ancora 3: la triade di continuità

Backup = copia. Monitoraggio = controllo. Ripristino = ripartenza.

Se ti blocchi, scrivi queste tre triadi e poi le trasformi in frasi.

Triade tecnica: frontend - backend - database  
Triade di controllo: autenticazione - autorizzazione - log  
Triade di continuità: backup - monitoraggio - ripristino

### Formula completa da ricordare:

Obiettivo -> Attori -> Funzioni -> Qualità -> Architettura -> Dati  
-> Sicurezza -> Continuità -> Conclusione

## 24. Esercizio guidato 1 - niente pagina bianca

Completa gli spazi scegliendo un contesto: pubblica amministrazione, azienda, scuola, sanità o altro. Non cercare la perfezione.

Contesto scelto: \_\_\_\_\_

Il sistema informativo ha lo scopo di \_\_\_\_\_  
\_\_\_\_\_

Gli attori principali sono \_\_\_\_\_ e ciascuno ha permessi differenti in base al proprio ruolo.

Le funzioni principali sono:

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_
4. \_\_\_\_\_
5. \_\_\_\_\_

I dati principali da gestire sono \_\_\_\_\_  
\_\_\_\_\_

L'architettura può essere composta da \_\_\_\_\_  
\_\_\_\_\_

La sicurezza viene garantita tramite \_\_\_\_\_  
\_\_\_\_\_

La continuità operativa viene garantita tramite \_\_\_\_\_  
\_\_\_\_\_

## 25. Esercizio guidato 2 - mini-risposta finale

Questa è la risposta che devi produrre e incollarmi in chat. 10-12 righe bastano.

### Consegna

Scrivi una mini-risposta sulla traccia del Giorno 4 usando il template. Puoi copiare l'attacco e poi completare con parole tue.

### Schema da completare:

Il sistema informativo proposto ha lo scopo di \_\_\_\_\_  
\_\_\_\_\_

Gli attori principali sono \_\_\_\_\_  
\_\_\_\_\_

Tra i requisiti funzionali rientrano \_\_\_\_\_  
\_\_\_\_\_

I requisiti non funzionali principali sono \_\_\_\_\_  
\_\_\_\_\_

L'architettura può essere organizzata su tre livelli: \_\_\_\_\_  
\_\_\_\_\_

Il database può contenere dati come \_\_\_\_\_  
\_\_\_\_\_

La sicurezza deve essere garantita tramite \_\_\_\_\_  
\_\_\_\_\_

Per assicurare continuità operativa sono necessari \_\_\_\_\_  
\_\_\_\_\_

In conclusione, il sistema permette di \_\_\_\_\_  
\_\_\_\_\_

## 26. Parole da scegliere se ti blocchi

Quando non ti vengono le parole, scegli da queste liste. È un modo legittimo per sblocarti.

Blocco	Parole utili
Scopo	digitalizzare, centralizzare, integrare, semplificare, tracciare, ridurre errori
Attori	utente esterno, operatore, responsabile, amministratore, tecnico IT, dirigente
Funzioni	inserire, consultare, modificare, approvare, notificare, ricercare, generare report
Qualità	sicurezza, privacy, disponibilità, usabilità, scalabilità, affidabilità, manutenibilità
Architettura	frontend, backend applicativo, API, database relazionale, cloud, server
Dati	utente, ruolo, pratica, documento, stato, data, notifica, log
Sicurezza	autenticazione, autorizzazione, HTTPS, hashing, log, audit, backup
Continuità	backup periodici, replica, monitoraggio, ripristino, disaster recovery, ridondanza
Conclusione	efficienza, tracciabilità, qualità del servizio, protezione dei dati, riduzione tempi

## 27. Checklist di autovalutazione

Prima di mandarmi la risposta, controlla solo questi punti. Non serve riscrivere tutto.

- Ho scritto lo scopo del sistema?
- Ho indicato almeno 4 attori?
- Ho messo almeno 5 funzioni?
- Ho distinto requisiti funzionali e non funzionali?
- Ho scritto frontend, backend e database?
- Ho indicato quali dati vengono gestiti?
- Ho parlato di autenticazione e autorizzazione?
- Ho citato almeno backup e monitoraggio?
- Ho chiuso con una conclusione sul valore della soluzione?
- Le frasi sono abbastanza brevi?

### **Valutazione minima**

Se hai risposto sì ad almeno 7 punti su 10, la bozza è utile e me la puoi mandare. Non aspettare di renderla perfetta.

## 28. Errori tipici da evitare

Errore	Correzione
Partire subito da 'uso Java / Python / SQL'	Prima spiega problema, utenti, funzioni e dati. La tecnologia viene dopo.
Scrivere solo una lista di tecnologie	Ogni tecnologia va collegata a una funzione: database per dati, backend per logica, HTTPS per sicurezza.
Dire che il cloud risolve tutto	Il cloud aiuta, ma richiede configurazione, sicurezza, contratti e controllo dei dati.
Confondere backup e continuità	Il backup è una copia; la continuità include anche ripristino, monitoraggio e procedure.
Scrivere frasi troppo lunghe	Dividi in paragrafi: scopo, attori, funzioni, architettura, sicurezza.
Dimenticare i permessi	Ogni sistema informativo deve chiarire chi può vedere o modificare cosa.
Usare parole difficili senza spiegarle	Meglio parole semplici ma corrette: autenticazione, ruoli, database, backup, log.

## 29. Cosa ti resta utile per le prove successive

Senza esagerare: la prima prova resta la priorità. Però quello che hai studiato oggi ti aiuterà anche dopo.

Tema di oggi	Come torna utile dopo
Architettura a livelli	Può diventare uno schema nella prova pratica o una spiegazione all'orale.
Database relazionale	Può servire per schema dati, entità e relazioni.
Ruoli e permessi	Serve per sicurezza, privacy e responsabilità professionale.
Backup e ripristino	Serve per continuità operativa e affidabilità.
Cloud/on-premise	Serve per motivare scelte infrastrutturali.
Log e monitoraggio	Serve per audit, incidenti e gestione del sistema.

### Non approfondire ora

Non devi studiare nel dettaglio protocolli, Kubernetes, data warehouse o dimensionamento infrastrutturale. Per la settimana 1 basta saperli nominare solo se servono e spiegare il senso generale.

## 30. Pagina finale da memorizzare

Per oggi devi portarti via questa pagina. Il resto serve a spiegarti il perché.

TEMA: sistema informativo aziendale/pubblico

1. Scopo

Digitalizzare e centralizzare processi, dati e documenti.

2. Attori

Utenti esterni, operatori, responsabili, amministratori, tecnici IT.

3. Funzioni

Inserire, consultare, modificare, approvare, notificare, cercare, fare report.

4. Qualità

Sicurezza, privacy, disponibilità, usabilità, scalabilità, affidabilità.

5. Architettura

Frontend per gli utenti, backend per la logica, database per i dati, API per integrazione, cloud/server per l'infrastruttura.

6. Dati

Utenti, ruoli, pratiche, documenti, stati, notifiche, log.

7. Sicurezza

Autenticazione, autorizzazione per ruoli, HTTPS, hashing password, log, backup, controllo accessi.

8. Continuità

Backup, monitoraggio, ripristino, replica/ridondanza se il sistema è critico.

9. Conclusione

La soluzione migliora efficienza, tracciabilità, qualità del servizio e protezione dei dati.

### Cosa mandarmi

Quando sei pronto, incollami la mini-risposta finale del punto 25. Io la correggerò su chiarezza, completezza, stile da esame e precisione tecnica.

**Fine Giorno 4.**