

# Verifica di TPSIT - Classe 4AIQ - Compito A

Prof. Fedeli Massimo

23/02/2026

## Contesto

Dato il seguente schema di database svolgi a tua scelta l'esercizio 1 o l'esercizio 2

```
ABBONAMENTI(id PK, nome, tipo, durata_giorni, ingressi)
CLIENTI(id PK, cognome, nome, indirizzo, telefono, codice_fiscale,
numero_tessera, email, fototessera, data_creazione,
data_nascita, luogo_nascita, citta, cap, provincia, note)
CORSI(id PK, nome, descrizione, codice_tornello)
ISCRIZIONI(id PK, cliente_id FK, data_iscrizione,
data_scadenza_iscrizione, certificato_medico,
data_scadenza_certificato, file_certificato,
corso_id FK, limitazioni_accesso)
```

## Esercizio 1

### Obiettivo

Realizzare una piccola applicazione web che consenta di inserire un nuovo cliente nella tabella `CLIENTI` tramite una pagina html (client) e un endpoint PHP (backend) che salvi i dati su MySQL.

### Vincoli e requisiti tecnici

- Frontend: una pagina `index.html` con un form e uno script JavaScript. **3 pt**
- Backend: un file PHP (esempio: `api_clienti_create.php`) che riceve i dati e inserisce la riga nel database. **3 pt**
- NB. Comunicazione: il client deve inviare i dati al backend usando fetch in POST, in formato JSON.

- NB. Sicurezza: il backend deve usare query parametrizzate (PDO con prepared statements). Vietato concatenare stringhe in SQL.

## Esercizio 2

### Obiettivo

Realizzare una piccola applicazione web che consenta di visualizzare l'elenco delle iscrizioni di tutti i clienti, includendo anche le informazioni anagrafiche minime del cliente e, se presente, il corso associato.

### Vincoli e requisiti tecnici

- Frontend: una pagina HTML (può essere la stessa `index.html` o una pagina separata, ad esempio `iscrizioni.html`) che mostri i risultati in una tabella e uno script JavaScript. **4 pt**
- Backend: un file PHP (esempio: `api_iscrizioni_list.php`) che legga i dati dal database e restituisca un JSON. **4 pt**
- N.B. Comunicazione: il client deve richiedere i dati al backend usando `fetch` ricevere i dati in formato JSON.
- N.B. Query: il backend deve eseguire una `SELECT` con `JOIN`
- Sicurezza: il backend deve usare PDO e prepared statements anche per eventuali parametri di filtro.

### Ottimizzazione

Filtrare l'elenco per data di iscrizione e per tipologia di corso usando una select popolata tramite script php. **2 pt**

- Il frontend deve includere uno o due campi data (es. `data_da`, `data_a`) e un pulsante di ricerca.
- Il backend deve validare le date e applicare il filtro nella clausola `WHERE` tramite parametri bindati.
- Se i parametri sono assenti, devono essere restituite tutte le iscrizioni.