
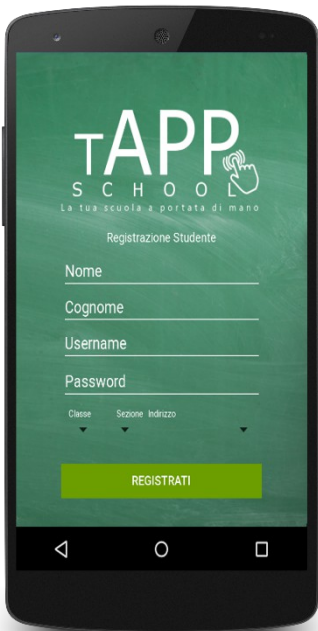
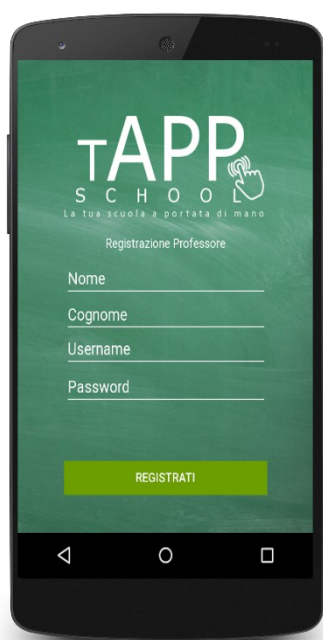


TAP SCHOOL

Schermata	Spiegazione
	Schermata dove l'utente effettua il login (studente/professore) fornendo le proprie credenziali quali username e password
	Schermata dove viene effettuata la registrazione di un nuovo studente



Schermata dove viene effettuata la registrazione di un nuovo professore



Schermata dove viene effettuata la modifica di uno studente



Schermata dove viene effettuata la modifica di un professore



Schermata dove viene effettuata la registrazione di un nuovo evento sul calendario



Schermata dove viene effettuata modifica di un evento sul calendario



Schermata dove viene effettuata l'eliminazione di un evento dal calendario

Casi d'uso

Professori e studenti devono tener traccia dei vari impegni settimanali, mensili, ecc. che possono essere già presenti in altri sistemi o comunicati a voce o tramite circolari. L'app permette di aggiungere gli impegni all'agenda recuperando i dati dagli altri sistemi o utilizzando le informazioni fornite dall'utilizzatore.

Caso d'Uso: Login Utente
ID: UC_LoginUtente
Attori: Professore Studente
Precondizioni: <ul style="list-style-type: none">Il sistema è stato avviatoIl professore o lo studente sono già registrati
Sequenza eventi: <ol style="list-style-type: none">Il caso d'uso inizia quando il professore o lo studente selezionano la voce "Login Utente"Il sistema mostra una form dove inserire i dati relativi al login (username, password)Il sistema controlla le credenziali nel databaseIl sistema invia una risposta affermativa o negativa
Sequenze alternative: <ol style="list-style-type: none">Lo studente (o il professore) può abbandonare la funzionalità in qualsiasi momento
Postcondizioni: <ul style="list-style-type: none">La Login è avvenuta con successo

Caso d'Uso: Registrazione Studente
ID: UC_RegistrazioneStudente
Attori: Studente
Precondizioni: <ul style="list-style-type: none">Il sistema è stato avviato
Sequenza eventi: <ol style="list-style-type: none">Il caso d'uso inizia quando lo studente che desidera registrarsi clicca sulla voce "Registrazione"Il sistema mostra una form dove inserire i dati relativi allo studente al fine di registrarsiIl sistema invia una risposta affermativa o negativa
Sequenze alternative:

1. Lo studente può abbandonare la funzionalità in qualsiasi momento

Postcondizioni:

- La Registrazione è avvenuta con successo

Caso d'Uso: Registrazione Professore

ID: UC_RegistrazioneProfessore

Attori:

Professore

Precondizioni:

- Il sistema è stato avviato

Sequenza eventi:

1. Il caso d'uso inizia quando il professore che desidera registrarsi clicca sulla voce "Registrazione"
2. Il sistema mostra una form dove inserire i dati relativi al professore al fine di registrarsi
3. Il sistema invia una risposta affermativa o negativa

Sequenze alternative:

1. Il professore può abbandonare la funzionalità in qualsiasi momento

Postcondizioni:

- La Registrazione è avvenuta con successo

Caso d'Uso: Modifica Studente

ID: UC_ModificaStudente

Attori: Studente

Precondizioni:

- Il sistema è stato avviato
- Lo studente si è registrato

Sequenza eventi:

1. Il caso d'uso inizia quando lo studente accede alla sua area personale e clicca la voce "modifica dati"
2. Il sistema mostra una form dove modificare i propri dati
3. Il sistema controlla che i nuovi dati siano pertinenti
4. Il sistema invia una risposta affermativa o negativa

Sequenze alternative:

1. Lo studente può abbandonare la funzionalità in qualsiasi momento
Postcondizioni:
<ul style="list-style-type: none"> La modifica è avvenuta con successo

Caso d'Uso: Modifica Professore
ID: UC_ModificaProfessore
Attori: Professore
Precondizioni:
<ul style="list-style-type: none"> Il sistema è stato avviato Il professore si è registrato
Sequenza eventi:
<ol style="list-style-type: none"> Il caso d'uso inizia quando il professore accede alla sua area personale e clicca la voce “modifica dati” Il sistema mostra una form dove modificare i propri dati Il sistema controlla che i nuovi dati siano pertinenti Il sistema invia una risposta affermativa o negativa
Sequenze alternative:
<ol style="list-style-type: none"> Il professore può abbandonare la funzionalità in qualsiasi momento
Postcondizioni:
<ul style="list-style-type: none"> La modifica è avvenuta con successo

Caso d'Uso: Gestione Calendari
ID: UC_GestioneCalendari
Attori:
Studente
Precondizioni:
<ul style="list-style-type: none"> Essersi collegato come utente registrato Aver cliccato sulla voce calendari
Sequenza eventi:
<ol style="list-style-type: none"> Il sistema visualizza l'elenco dei calendari già presenti nel sistema: pubblici e i privati di proprietà dell'utente L'utente clicca sul bottone aggiungi Il sistema visualizza l'activity per l'inserimento dei dati: nome calendario, eventuale breve descrizione, tipologia (pubblico, privato) L'utente inserisce i dati e clicca sul bottone di conferma

5. Il sistema conferma l'inserimento se non esiste un calendario con lo stesso nome
Sequenze alternative: <ol style="list-style-type: none"> L'utente seleziona uno o più calendari privati dall'elenco e clicca sul bottone cancella Il sistema conferma l'eliminazione L'utente seleziona un calendario in elenco di cui è proprietario e clicca sul bottone modifica Il sistema prosegue come dal punto 3 del flusso principale utilizzando i dati della selezione L'utente seleziona un calendario pubblico in elenco e clicca sul bottone sottoscrivi Il sistema conferma l'iscrizione al calendario
Postcondizioni: <ul style="list-style-type: none"> Il database contiene l'elenco aggiornato dei calendari

Caso d'Uso: Gestione Eventi
ID: UC_GestioneEventi
Attori: Professore
Precondizioni: <ul style="list-style-type: none"> Essersi collegato come utente registrato Aver cliccato sulla voce eventi
Sequenza eventi: <ol style="list-style-type: none"> Il sistema visualizza l'elenco dei calendari a cui l'utente si è iscritto o che sono di sua proprietà L'utente seleziona uno dei calendari Il sistema visualizza l'elenco degli eventi presenti nel calendario nel formato agenda L'utente clicca sul bottone aggiungi Il sistema visualizza l'activity per l'inserimento dell'impegno: data inizio, data fine, nel caso non sia un impegno giornaliero un'ora di inizio e un'ora di fine, il nome dell'evento, una eventuale breve descrizione, un eventuale luogo all'interno della scuola, la periodicità L'utente inserisce i dati e clicca su conferma Il sistema conferma l'inserimento
Sequenze alternative: <ol style="list-style-type: none"> L'utente seleziona una o più eventi da lui creati e clicca sul bottone cancella Il sistema conferma l'eliminazione L'utente seleziona un evento da lui creato e clicca sul bottone modifica Il sistema prosegue come dal punto 5 del flusso principale utilizzando i dati della selezione
Postcondizioni: <ul style="list-style-type: none"> Il database contiene l'elenco aggiornato degli eventi

Caso d'Uso: Importazione eventi da PDF/XML
ID: UC_ImportazioneEventiPDF

Attori: Studente Professore
Precondizioni: <ul style="list-style-type: none"> • Essersi collegato come utente registrato • Aver cliccato sulla voce importazione
Sequenza eventi: <ol style="list-style-type: none"> 1. Il sistema visualizza l'activity per l'inserimento del link al file che si vuole importare con dei radio button per la scelta del calendario in cui inserire gli eventi 2. L'utente inserisce il link al file PDF/XML 3. Il sistema legge il contenuto del file PDF/XML e lo interpreta in modo da recuperare per ogni evento almeno: <ul style="list-style-type: none"> ◦ Il calendario più adatto (personale, di classe, ...) ◦ Il nome dell'evento ◦ La data, l'orario di inizio, l'orario di fine ◦ Il luogo: lotto, aula/laboratorio 4. Il sistema controlla che non ci siano errori o conflitti (eventi sovrapposti) e conferma l'inserimento nel database con un messaggio
Sequenze alternative: <ol style="list-style-type: none"> 1. In caso di conflitti il sistema segnala la situazione all'utente e chiede che cosa fare 2. L'utente indica se aggiungere l'evento anche se sovrapposto o se tralasciarlo 3. Il sistema segue le indicazioni dell'utente
Postcondizioni: <ul style="list-style-type: none"> • Il database contiene l'elenco aggiornato con gli eventi importati

Caso d'Uso: Importazione eventi da file ics
ID: UC_ImportazioneEventiICS
Attori: Studente Professore
Precondizioni: <ul style="list-style-type: none"> • Essersi collegato come utente registrato • Aver cliccato sulla voce importazione
Sequenza eventi: <ol style="list-style-type: none"> 1. Il sistema visualizza l'activity per l'inserimento del link al file che si vuole importare con dei radio button per la scelta del calendario in cui inserire gli eventi

2. L'utente inserisce il link al file .ics
3. Il sistema legge il contenuto del file ics e lo interpreta in modo da recuperare per ogni evento almeno:
 - Il nome del calendario (personale, di classe, ...)
 - Il nome dell'evento
 - La data, l'orario di inizio, l'orario di fine
 - Il luogo: lotto, aula/laboratorio
4. Il sistema controlla che non ci siano errori o conflitti (eventi sovrapposti) e conferma l'inserimento nel database con un messaggio

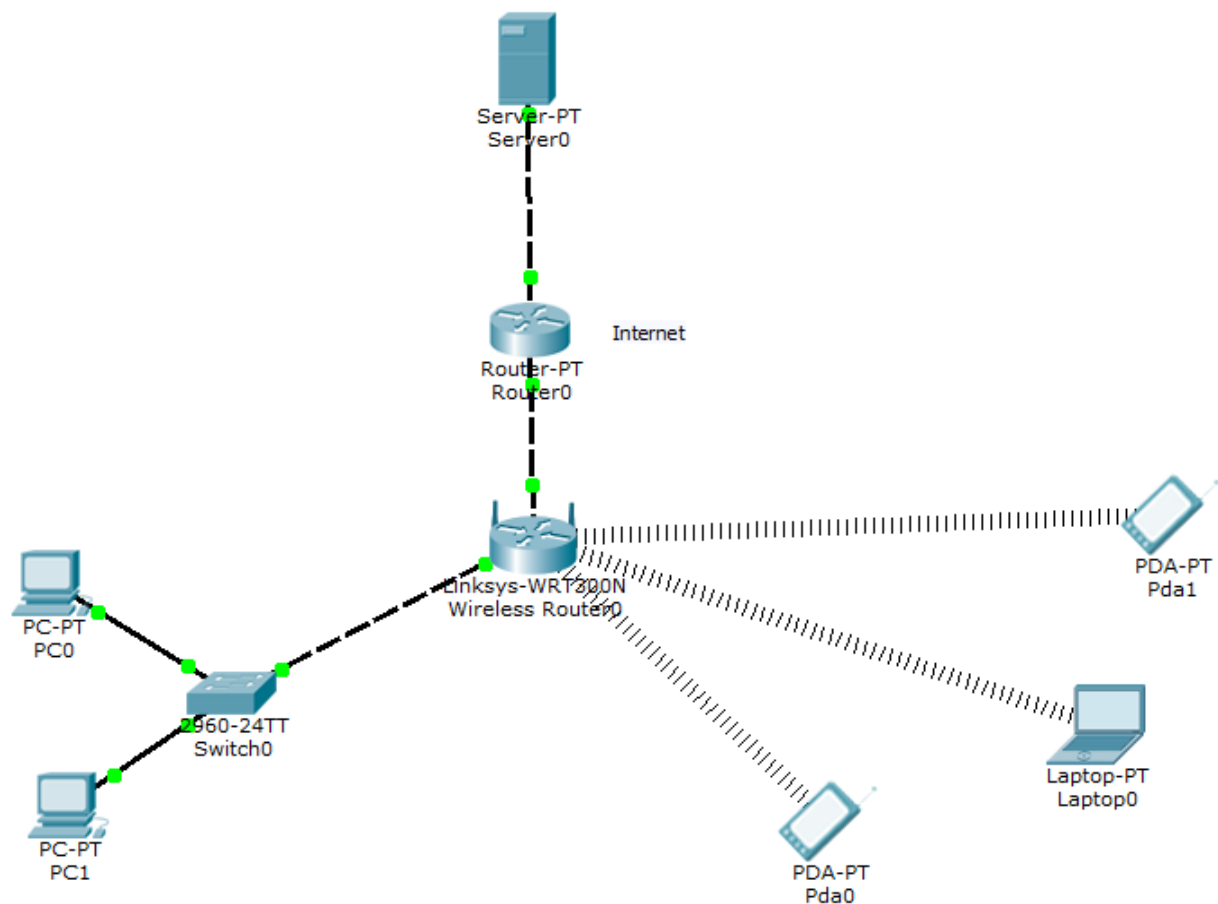
Sequenze alternative:

4. In caso di conflitti (eventi sovrapposti) il sistema segnala la situazione all'utente e chiede che cosa fare
5. L'utente indica se aggiungere l'evento anche se sovrapposto o se tralasciarlo
6. Il sistema segue le indicazioni dell'utente

Postcondizioni:

- Il database contiene l'elenco aggiornato con gli eventi importati

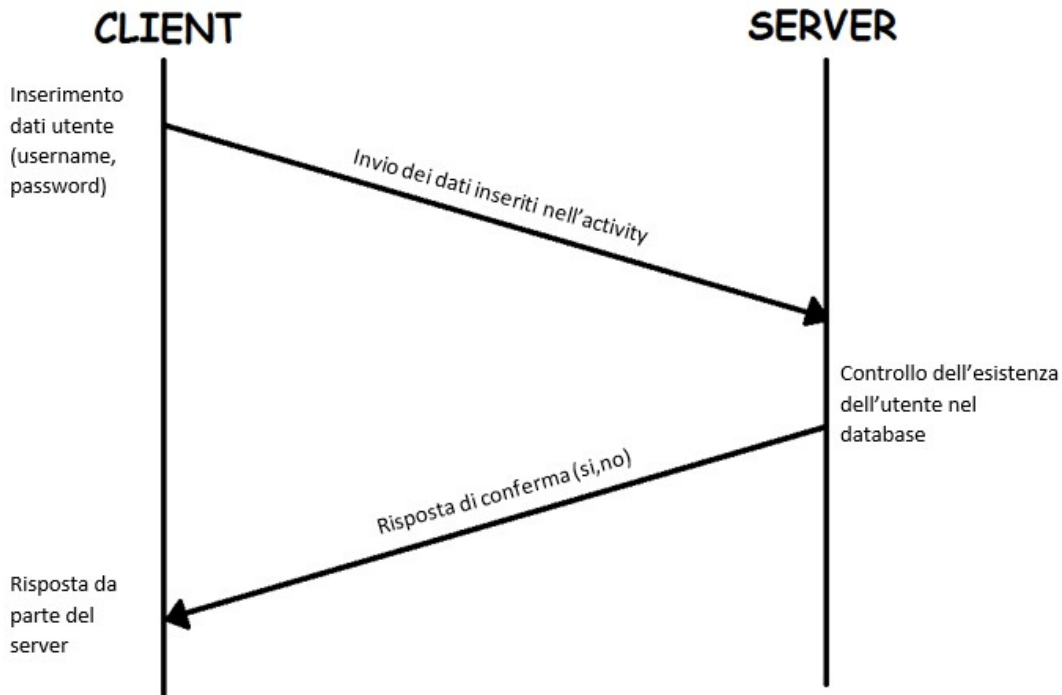
Architettura di rete



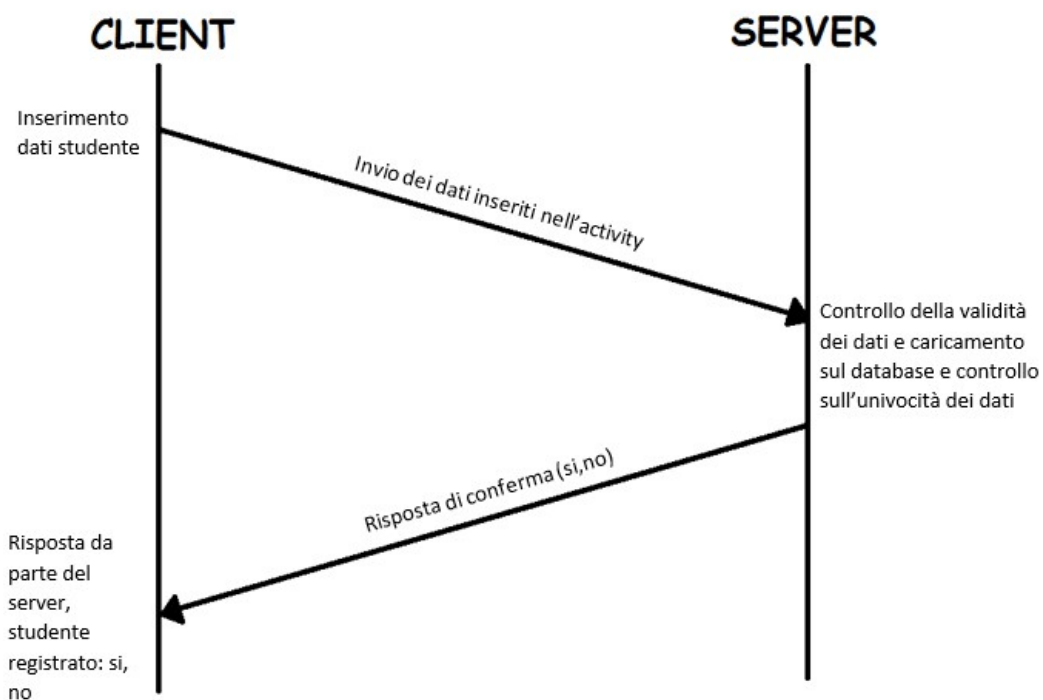
All'interno del server bisogna installare MYSQL che contiene i dati necessari al login e all'autorizzazione. Nei vari device bisognerà installare un browser per collegarsi ad Internet nel caso si usi un pc. Se si utilizza uno smartphone si deve scaricare l'applicazione. Sia il browser che l'applicazione fanno riferimento al server.

Protocolli casi d'uso

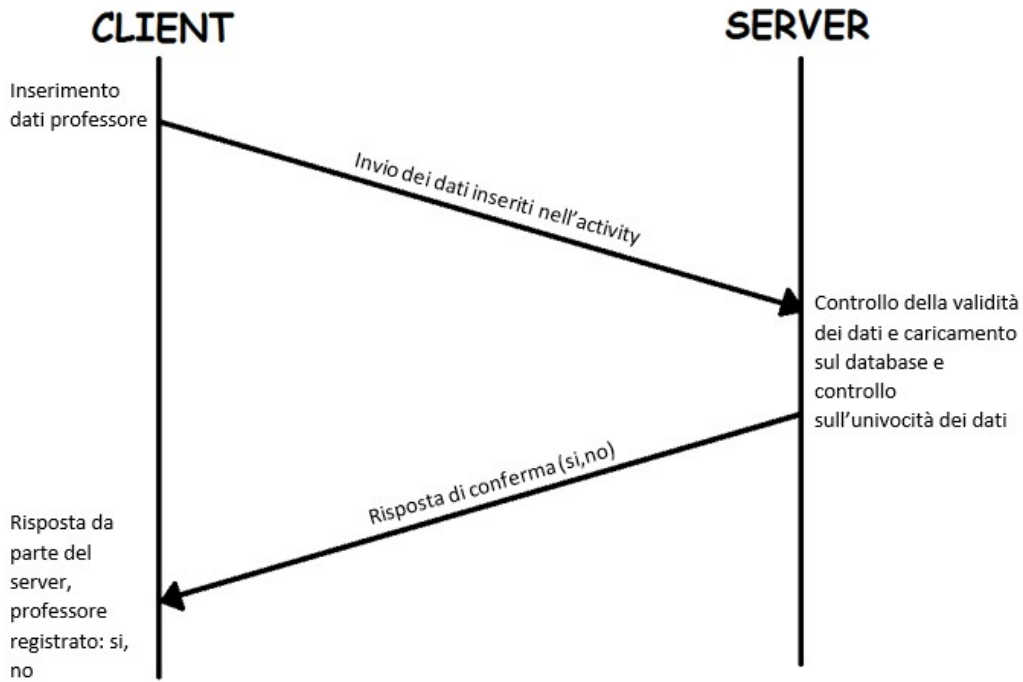
- Login



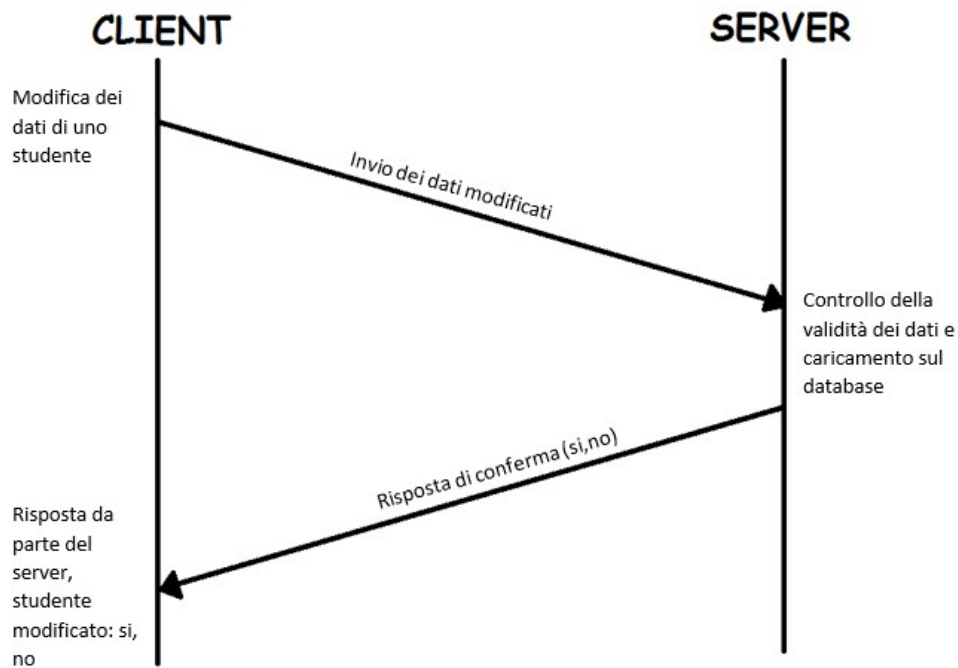
- **Registrazione Studente**



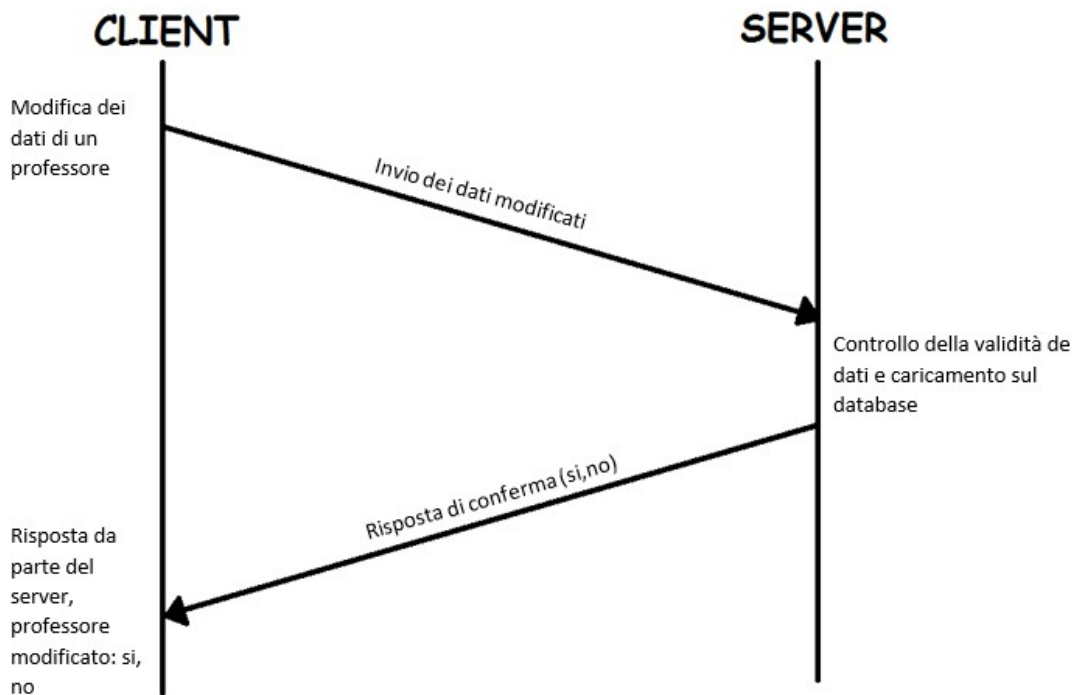
- **Registrazione Professore**



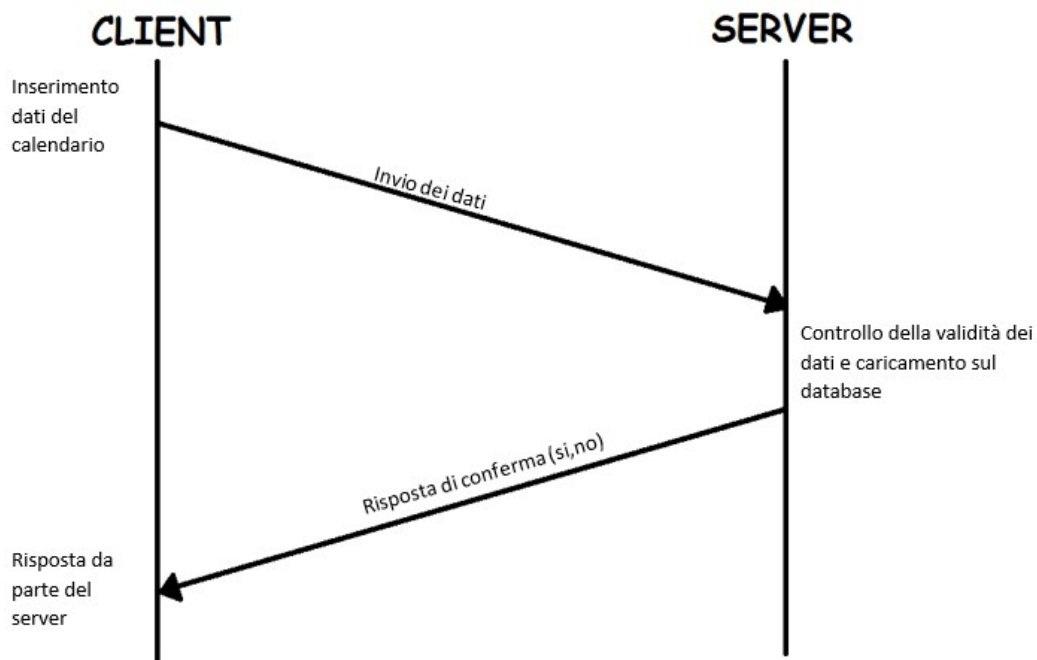
- **Modifica Studente**



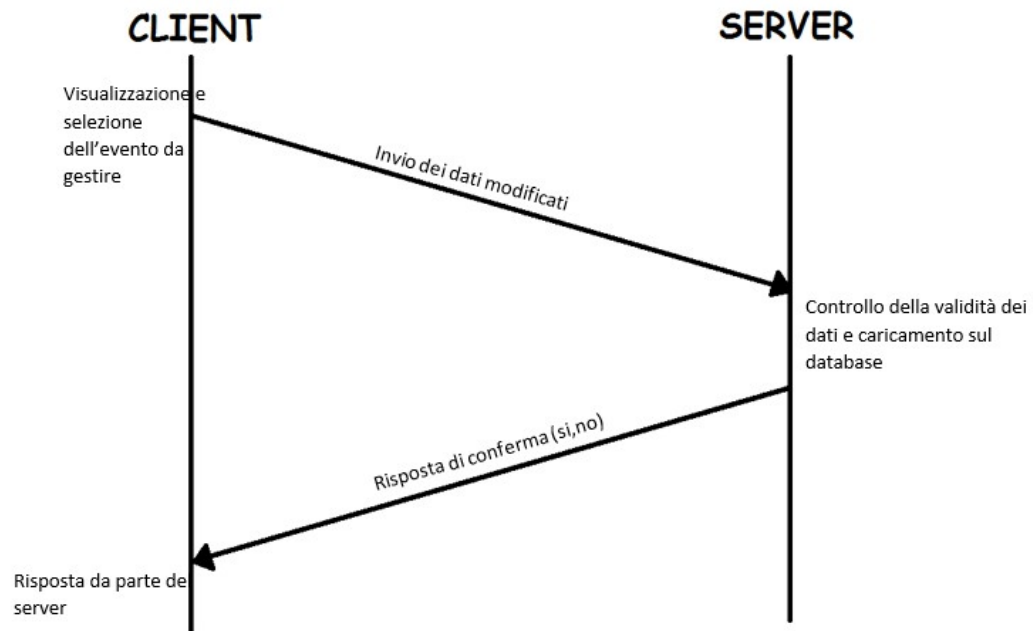
- **Modifica Professore**



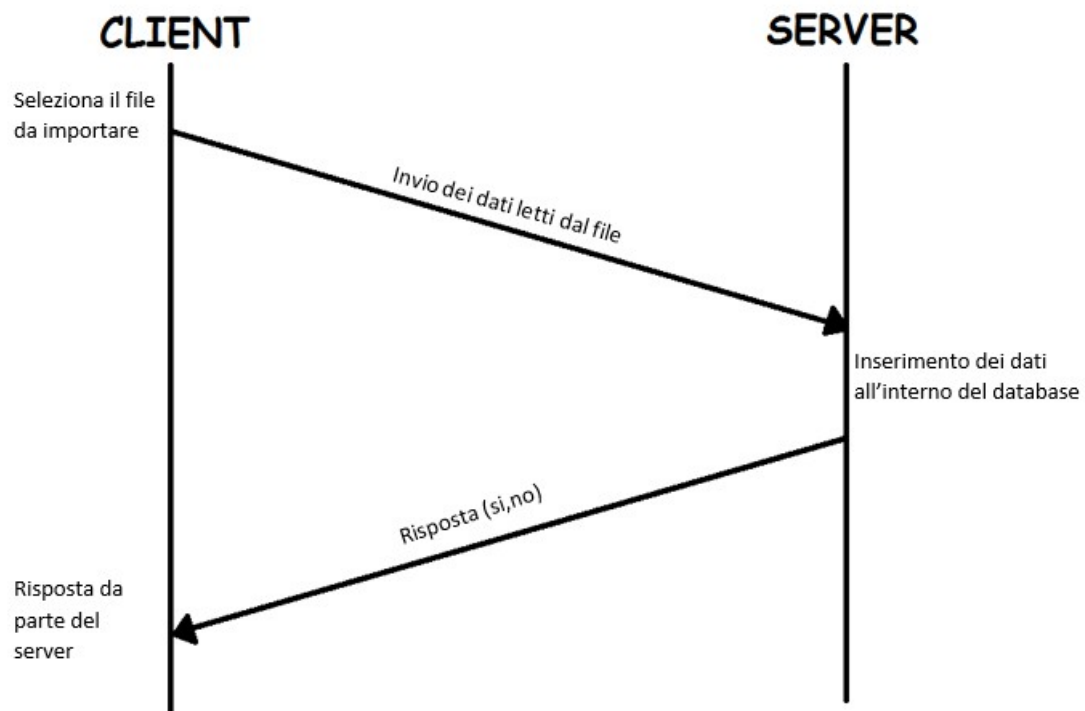
- **Gestione Calendari**



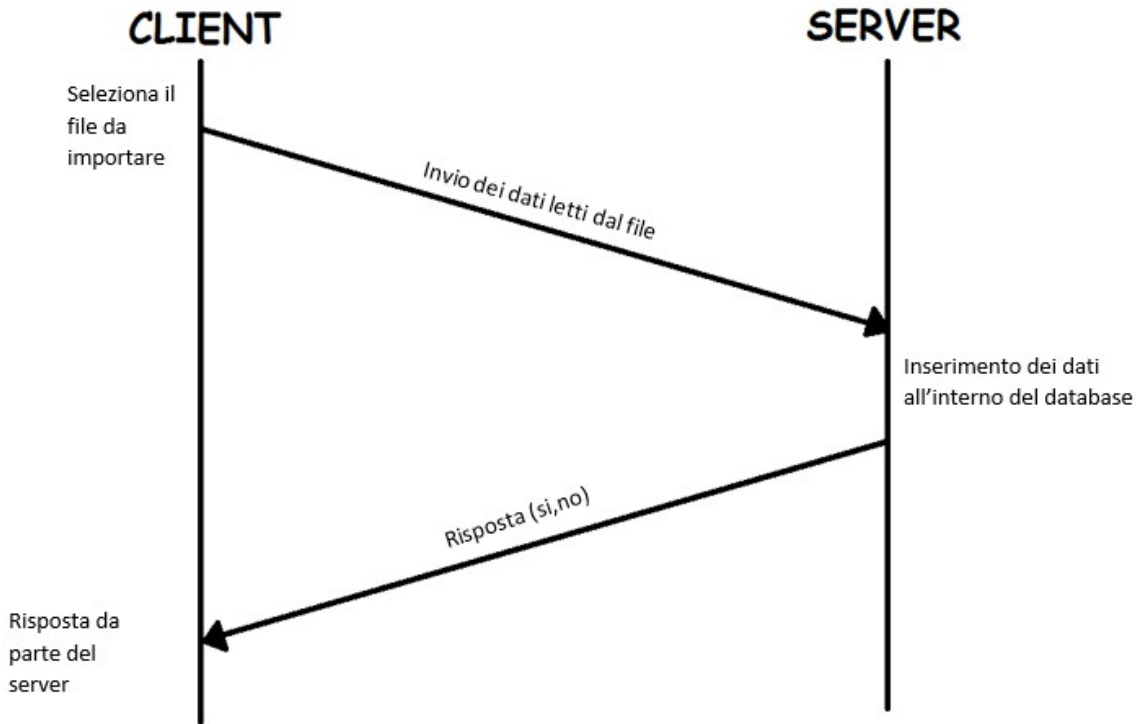
- **Gestione Eventi**



- **Importazione eventi da PDF/XML**



- Importazione eventi da file ics



Protocolli XML

- Login

```

<?xml version="1.0" encoding="UTF-8"?>
<client>
  <login> <!-- login da parte dell'utente fornendo username e password (utente che può essere studente o professore) -->
    <username> pippo_pluto </username>
    <password> password </password>
  </login>
</client>
  
```

- Registrazione Studente

```

<?xml version="1.0" encoding="UTF-8"?>
<client> <!-- lo studente ha effettuato il login -->
  <modificaStudente> <!-- modifica di tutti gli attributi di uno studente -->
    <nome> Pippo </nome>
    <cognome> Pluto </cognome>
    <username> pippo_pluto </username>
    <password> password </password>
    <classe> 5CI </classe>
  </modificaStudente>
</client>
  
```

- Registrazione Professore


```
<?xml version="1.0" encoding="UTF-8"?>
<client>
  <registraProfessore> <!-- inserimento di un nuovo professore all'interno del database -->
    <nome> Pippo </nome>
    <cognome> Pluto </cognome>
    <username> pippo_pluto </username>
    <password> password </password>
  </registraProfessore>
</client>
```

- **Modifica Studente**

```
<?xml version="1.0" encoding="UTF-8"?>
<client>
  <registraStudente> <!-- inserimento di un nuovo studente all'interno del database -->
    <nome> Pippo </nome>
    <cognome> Pluto </cognome>
    <username> pippo_pluto </username>
    <password> password </password>
    <classe> 5CI </classe>
  </registraStudente>
</client>
```

- **Modifica Professore**

```
<?xml version="1.0" encoding="UTF-8"?>
<client> <!-- il professore ha effettuato il login -->
  <modificaProfessore> <!-- modifica di tutti gli attributi di un professore -->
    <nome> Pippo </nome>
    <cognome> Pluto </cognome>
    <username> pippo_pluto </username>
    <password> password </password>
  </modificaProfessore>
</client>
```

- **Gestione Eventi (inserisci evento)**

```
<?xml version="1.0" encoding="UTF-8"?>
<client> <!-- l'utente ha effettuato il login -->
  <inserisciEvento> <!-- creazione di un nuovo evento, inserimento di tutti i dati dell'evento -->
    <data> 26/04/2017 </data>
    <ora> 08.00 </ora>
    <note> Gita a Verona </note>
  </inserisciEvento>
</client>
```

- **Gestione Eventi (modifica evento)**

```
<?xml version="1.0" encoding="UTF-8"?>
<client> <!-- l'utente ha effettuato il login -->
  <modificaEvento> <!-- modifica di tutti gli attributi dell'evento -->
    <data> 26/04/2017 </data>
    <ora> 08.00 </ora>
    <note> Gita a Verona </note>
  </modificaEvento>
</client>
```

- **Risposta Server**

```
<?xml version="1.0" encoding="UTF-8"?>
<server> <!-- il server risponde al client -->
    <invioRisposta risposta = "Yes">
    </invioRisposta>
</server>
```

Diagramma E/R e analisi

Analisi

Per gestire il progetto richiesto, abbiamo deciso di affidarci all'uso di entità e attributi. Di seguito ci sarà un elenco delle entità utilizzate e una spiegazione del loro impiego nel progetto con i rispettivi attributi.

- **Studente:** entità per gestire gli studenti;
 - ID: impostato come chiave primaria;
 - Nome: nome dello studente
 - Cognome: cognome dello studente
 - Username: username dello studente
 - Password: password studente
 - Classe: classe frequentata dallo studente
- **Professore:** entità per gestire i professori;
 - ID: impostato come chiave primaria;
 - Nome: nome del professore;
 - Cognome: cognome del professore;
 - Username: username del professore;
 - Password: password professore;
- **Evento:** entità per gestire gli eventi del calendario (gite, verifiche, interrogazioni...)
 - ID: chiave primaria per identificare ogni evento;
 - Data: data prevista;
 - Ora: ora prevista;
 - Note: eventuali note riguardo l'evento;
- **Classe:** entità classe;
 - ID: chiave primaria per identificare ogni classe;

- Classe: è il numero dell'anno della classe;
- Sezione: a, b, c...;
- Indirizzo: indirizzo della classe (es: informatica, meccanica...).
- **Materie:** entità per gestire ogni materia;
 - ID: chiave primaria per identificare ogni materia;
 - Nome: matematica, italiano...

Analizziamo ora le associazioni tra le entità, evidenziando la molteplicità che esiste.

1. **Tengono:** Professore – Classe (M: M), l'associazione è di tipo molti a molti in quanto più professori tengono più classi. Viceversa, più classe sono tenute da più professori.
2. **Insegnano:** Professore – Materie (1: M) l'associazione di tipo 1 a molti prevede infatti che un professore insegni una materia, e che una materia sia insegnata da più professori.
3. **Hannoprof:** Professore – Evento (M: M) l'associazione è di tipo molti a molti perché un professore programma molti eventi, e molti eventi sono stati programmati da molti professori.
4. **Hannostud:** Studente – Evento (M: M) quest'associazione molti a molti è così composta: uno studente ha in programma uno o più eventi e un evento coinvolge uno o più studenti.

Infine creiamo lo schema logico di ogni associazione.

1. Mapping associazione ISA:

- **Studente:** (ID; Nome; Cognome; Username, Password, Classe);
- **Professore:** (ID; Nome; Cognome; Username, Password).

2. Mapping associazione Hannoprof tra Professore ed evento:

- **Professore:** (ID; Nome; Cognome; Username, Password);
- **Evento:** (ID; Data; Ora; Note).

3. Mapping associazione Hannostud tra Studente ed evento:

- **Professore:** (ID; Nome; Cognome; Username, Password);
- **Evento:** (ID; Data; Ora; Note).

4. Mapping associazione Insegnano tra Professore e materie:

- **Professore:** (ID; Nome; Cognome; Username, Password);
- **Materie:** (ID; Nome).

5. Mapping associazione Tengono tra Professore e classe:

- **Professore:** (ID; Nome; Cognome: Username, Password);
- **Classe:** (ID; Classe, Sezione, Indirizzo).

Diagramma E/R

