

# Progetto di Tecnologie Web - A.A. 2023/24

## Appelli d'Esame

- **Giugno (metà):** Appello principale.
- **Luglio (inizio e metà):** Appelli successivi.
- **Settembre, Gennaio 2025, Febbraio 2025:** Ulteriori appelli.
- **Giugno 2025:** Solo esame scritto per chi è in debito dall'anno precedente.

## Valutazione

- **Scritto:** 50% del voto finale (in trentesimi).
- **Progetto:** 50% del voto finale (in trentesimi).
- **Progetto Base:** 18-21/30
- **+ I Modulo Aggiuntivo:** 18-24/30
- **+ I e II Moduli Aggiuntivi:** 18-27/30
- **+ Tutti e Tre i Moduli Aggiuntivi:** 18-33/30
- **Bonus:** Fino a +2 punti per scelte creative e funzionali nell'interfaccia (a discrezione del docente).

Il voto del progetto viene convertito in un voto pesato, secondo una tabella di conversione fornita.

## Struttura del Progetto

- **Obiettivo:** Creare un sistema reale e funzionante, con enfasi sull'approccio dichiarativo (documenti attivi) e sul mashup di tecnologie esistenti.
- **Team:** Lavoro di gruppo obbligatorio (2-3 persone, progetti 18-24: 1-2 persone, progetti 18-21 individuali). Il contributo individuale è fondamentale e verrà valutato all'esame.
- "Portatori di Pizza" Non sono ammessi.

## Teoria di Base: Strumenti per lo Sviluppo Web

### Framework

- Librerie che semplificano e arricchiscono l'uso di tecnologie (linguaggi server-side, client-side, CSS).
- **Server-side:** Facilitano la programmazione a tre livelli.
- **Client-side:** Sviluppatisi dal 2002, basati su CSS e Javascript, con vari scopi.

### API (Application Programming Interfaces)

- Librerie, protocolli e strumenti per utilizzare algoritmi e servizi di un software da parte di un altro software (non da esseri umani).
- **Delegazione:** Le API permettono di delegare aspetti come interazione, interfaccia, navigazione, ecc., fornendo solo il servizio "nudo".
- **Mashup:** Integrazione di più servizi tramite API per creare applicazioni più ricche e potenti.

### API RESTful

- Sfruttano HTTP e URI per fornire servizi.
- **Componenti:**
  - URI base.
  - Sintassi degli URI delle entità interrogabili/modificabili.
  - Media type (es. JSON, XML).
  - Semantica dei verbi HTTP (GET, PUT, POST, DELETE).

- **Documentazione:** Descrive URI, verbi e formati per ogni servizio.

## API di Servizi Locali

- Permettono alle applicazioni web di accedere a servizi del device (es. periferiche I/O).
- **Esempi:** Camera API, Speech API, Geolocation API, Web Audio API, Vibration API, Web Telephony API, ecc.
- Circa 80 API disponibili sulla maggior parte dei browser.

## Progetto: SELFIER

### Specifiche

- **Nero:** Requisiti obbligatori per tutte le consegne.
- **Arancione:** Requisiti per le estensioni del progetto (facoltativi).
- **Verde:** Vincoli obbligatori specifici per il progetto universitario.
- **Blu:** Esempi, non specifiche.
- **Modifiche:** Eventuali modifiche ai requisiti saranno pubblicate su Virtuale.

### Fondamenti

- **Applicazione per studenti UniBo:** Gestione di eventi, scadenze e impegni (personali, sociali, accademici).
- **Eventi:** Individuali o di gruppo, unici o ripetuti, semplici o complessi.
- **Uso:** Applicazione client+server utilizzabile sia da cellulare che da PC.
- **Moduli:** Moduli base (obbligatori) e moduli aggiuntivi per funzionalità estese.

### Architettura

#### 1. Progetto Base (18-21):

- Aggiunta, rimozione, modifica di eventi semplici dell'utente.
- Calendario con visualizzazioni giornaliere, settimanali e mensili.
- Timer (view Pomodoro) per lo studio.
- Editor di appunti.
- Sistema di navigazione temporale (Time Machine).

#### 2. Prima Estensione (18-24):

- Sistemi di notifica e geolocalizzazione.
- Notifiche calibrabili per testo, ripetizione, urgenza, ritardi, snooze.
- Geolocalizzazione per situare eventi in un luogo/fuso orario.

#### 3. Seconda Estensione (18-27):

- Eventi di gruppo: eventi appartenenti a più calendari collegati.
- Gestione della privacy.
- Integrazione con sistemi terzi (Google Calendar, Apple Calendar, iCalendar).

#### 4. Terza Estensione (18-33):

- Gestione di progetti complessi (es. studio di un esame).
- Fasi, attività, milestone.
- Attività individuali e di gruppo, sincronizzazione, dipendenze, monitoraggio.
- Diagramma di Gantt.

## Componenti dell'Applicazione

- **Tecnologie:**
  - **Client-side:** Javascript/Typescript.
  - **Server-side:** Node.js (obbligatorio, no altre tecnologie).
- **Parti:**
  - Home e accesso utente.
  - Calendario.
  - Note.
  - Applicazione Pomodoro.
  - Applicazione Gestione Progetti.
  - Applicazione Time Machine.

### Home Page e Accesso Utente

- **Accesso:** Account con nome utente e password.
- **Record Account:** Nome utente, password, nome vero, informazioni personali (opzionali).
- **Home Page:** Navigazione tra le view (Calendario, Pomodoro, Note, Progetti).
- **Preview:** Anteprime dei contenuti delle view.
- **Estensioni:**
  - **18-24:** Personalizzazione delle preview.
  - **18-27:** Invio di messaggi/notifiche ad altri utenti.
  - **18-33:** Mini-hub per chat con altri utenti.

### Calendario - Eventi

- **Creazione Eventi:** Appuntamenti con data, ora, durata, titolo.
- **Tipi:** Unici, ripetibili, lunga durata (giornate intere, più giorni).
- **Eventi Ripetibili:**
  - Frequenza (giornaliera, settimanale, mensile, ecc.).
  - Numero di ripetizioni (indefinite, N volte, fino a data).
- **Luogo:** Fisico o virtuale.
- **Estensioni:**
  - **18-24:** Notifiche per eventi imminenti (meccanismo, anticipo, ripetizione).
  - **18-27:** Inclusione di altri utenti, accettazione/rifiuto, intervalli non disponibili, integrazione iCalendar.
  - **18-33:** Risorse (stanze riunioni, apparecchiature), calendario risorse, scadenze progetti come eventi.

### Calendario - Attività

- **Creazione Attività:** Compiti di durata prolungata, non esclusivi, con scadenza (opzionale).
- **Completamento:** Esplicito o trascinato nei giorni successivi (attività in ritardo).
- **Visualizzazione:** Scadenza nel calendario e lista separata.
- **Estensioni:**
  - **18-24:** Notifiche di urgenza crescente.
  - **18-27:** Assegnazione a più persone.
  - **18-33:** Suddivisione in sotto-attività, correlazione a progetti.

### Note

- **Definizione:** Testo di lunghezza arbitraria con titolo, categorie, data di creazione/modifica.
- **Gestione:** View separata dal calendario.
- **Funzionalità:** Duplicazione, copia/incolla, cancellazione.

- **Home:** Preview delle note esistenti, aggiunta di nuove note.
- **Categorizzazione:** Ordine alfabetico, data, lunghezza.
- **Estensioni:**
  - **18-24:** Scrittura in Markdown.
  - **18-27:** Autore e lista di accesso (pubbliche, private, condivise).
  - **18-33:** Liste di cose da fare, aggiunta attività al calendario da list item.

### Pomodoro

- **Metodo Pomodoro:** Organizzazione del tempo in cicli studio-relax (30+5 minuti).
- **View:**
  - Form per scegliere tempo di studio/pausa.
  - Proposte di cicli studio/pausa da input (ore/minuti disponibili).
  - Bottoni: inizio forzato, ricomincia ciclo, fine ciclo.
  - Notifiche: inizio ciclo, passaggio fase, fine ciclo.
- **Animazioni:** CSS (obbligatorio) per studio e pausa.
- **Estensioni:**
  - **18-24:** Programmazione cicli su diverse giornate (evento su calendario), cicli non completati passati ai giorni successivi.
  - **18-27:** Notifica ad altro utente per studiare con stesse impostazioni.
  - **18-33:** Musica, video, modifica del tempo in corso.

### Gestione Progetti [18-33]

- **Progetto:** Lista di attività (sequenza/parallele) attribuite a uno o più attori.
- **Fasi e Sottofasi:** Raggruppamento delle attività.
- **Input/Output:** Attività con input (anche vuoto) e output (anche booleano).
- **Sincronizzazione:** Attività con output  $X = \text{input } Y$  (traslazione o contrazione).
- **Milestone:** Output importanti con date di conclusione non flessibili.
- **Note:** Descrizione progetto e attività (input/output come link a file online).
- **Stato Attività:**
  - Non attivabile, attivabile, attiva, conclusa, riattivata, in ritardo, abbandonata.
- **Eventi:** Inizio e fine attività come eventi sul calendario.
- **Visualizzazioni:** Lista (temporale/per attore) e GANTT.
- **Capo-Progetto:** Crea progetti, coinvolge utenti, modifica progetto, decide traslazione/contrazione.
- **Notifiche:** Attori coinvolti ricevono notifiche sulle decisioni del capo-progetto.
- **Esempio Gantt:** Fornito nel documento.

### Time Machine

- **Scopo:** Viaggiare nel tempo durante la presentazione.
- **Funzionamento:**
  - Ogni annotazione temporale (server/client) dipende da un servizio Time Machine.
  - Default: Allineato a data/ora del sistema operativo.
  - Modifica: Data/ora modificabile (avanti/indietro).
  - Interfaccia: Parte separata, sempre visibile, colori contrastanti.
  - View: Cambio immediato senza reload.
  - Notifiche: Attivazione notifiche del giorno specificato (non precedenti).
  - Pulsante: Ripristino data/ora del sistema operativo.

### Parte Facoltativa (per tutti)

- **Focus:** Semplicità e flessibilità nell'inserimento, spostamento e modifica di eventi/attività.
- **Criteri:**

- Efficacia (errori, precisione, multi-evento).
- Efficienza (tempo, azioni atomiche).
- Soddisfazione (grafica, chiarezza).
- **Hints:**
  - Ore del giorno non uguali (impegni diversi).
  - Allineamento eventi/attività a ore tonde (salvo richiesta specifica).
  - Notifiche attività in ritardo con urgenza crescente.
- **Documentazione:** Modalità di input speciale (se implementata) con documentazione e immagini.

## Requisiti di Progetto

- **Obbligatorie (Nero):**
  - Calendario, note, pomodoro: Mobile-first, framework Javascript/Typescript e CSS a scelta, usabilità.
  - Time Machine: Sempre visualizzata (PC), immediatamente accessibile (mobile).
  - Gestione progetti: PC-first, Javascript puro (Web Components ok), framework CSS a scelta, funzionalità adeguate su mobile.
  - Database: MongoDB sul server del dipartimento.
- **Presentazione:**
  - Database già popolati.
  - 4 account predefiniti: "fv1", "fv2", "fv3", "fvPM" (password "12345678").
  - Altri account predefiniti (nomi semplici, password "12345678").
  - Attività/eventi attribuiti agli account (scadenze ravvicinate per fv1).
  - Lista stampata dei prossimi eventi/attività.
  - Estensione 18-33: fvPM con progetti complessi (fasi, sottofasi, attività).
  - Varie note di diverse lunghezze/complessità.

## Criteri di Valutazione

1. **Generalità dei Tool:** Soluzioni per la compatibilità (scelte ottimali vs. forzature).
2. **Flessibilità:** Solidità, struttura, comprensibilità, estendibilità, adattabilità delle soluzioni tecniche.
3. **Usabilità:** Attenzione alle esigenze degli utenti (non esperti del modello).
4. **Sofisticazione Grafica:** Presentazione delle informazioni, rapporto dimensioni maschere/dati, label comprensibili, differenziazione tipi di dati.

## Vincoli Hard

1. **Tecnologie:**
  - **Back-office:** Node, MongoDB, vanilla Javascript (Express, moduli npm ok, NO php, perl, python, java, ruby, MySQL).
  - **Home + Calendario + Pomodoro + Note:** Framework a scelta (Angular, React, Vue, Svelte, ecc.).
  - **Gestione Progetto:** Senza framework (Web Components ok), vanilla Javascript.
2. **Framework Grafico:** Libero (Bootstrap, Tailwind, Foundation, ecc.), enfasi su sofisticazione, usabilità, eleganza.
3. **Deploy:** Due container Docker sul server del dipartimento.
4. **Database:** Presentati già popolati.
5. **Consegna Sorgenti:** Directory "sources" con tutti i sorgenti leggibili e README.txt (organizzazione del progetto).
6. **Modalità Input Speciale (opz):** Pagina HTML con documentazione e immagini.

## Progetti e Moduli (Riepilogo)

- **Installazione:** Docker sul server del dipartimento (nessuna eccezione).
- **Progetti 18-27 e 18-33:** Presentazione di persona (gruppi di 2-3).
- **Progetti 18-21 e 18-24:** Prevalutazione (possibile presentazione).
  - **18-21:** Individuali.
  - **18-24:** Gruppi di 1-2.
- **README.txt:** Nomi, cognomi, email (UniBo), contributo individuale, documentazione scelte implementative.

## Suggerimenti e Organizzazione Team

- Presentazione con progetto funzionante (obbligatorio).
- **Lavoro di Team:** Contributo determinante di ogni membro (no HTML/CSS o parti marginali).
- Distribuzione ideale dei compiti è per funzionalità.
- **Organizzazione Team:**
  - Decisione anticipata sulla sessione d'esame.
  - Team di 2-3 persone (no eccezioni, tranne casi singoli ben motivati).
  - Presentazione congiunta del progetto (dichiarazione contributo o interrogazione su tutto).

## Intelligenza Artificiale Generativa

- **Ammessa:** Uso di software AI come ausilio (problemi difficili/ripetitivi).
- **Documentazione:** Uso documentato esplicitamente.
- **Consapevolezza:** Membri del gruppo devono comprendere e poter modificare il codice.
- **Esame Scritto:** Competenze richieste per scrivere codice corretto (no ChatGPT).
- ChatGPT come copilota.

## Appello di Febbraio

- **31 Gennaio 2025:** Termine per presentare il progetto di quest'anno.
- **Febbraio:** Richieste ben motivate dell'ultimo minuto (riserva del docente).
- **Slot:** Richiesta entro il 31/12/2024 (no richieste dopo il 10/01/2024).
- **Non Ridursi all'Ultimo!**

## Flessibilità e Rigidità del Corso

- **Flessibilità:**
  - Prova scritta e progetto indipendenti.
  - Progetto sempre di gruppo (tranne eccezioni).
  - Scritto sempre individuale.
  - Possibilità di ripetere scritto e progetto.
  - Ritiro dalla presentazione del progetto possibile.
- **Rigidità (nessuna eccezione):**
  - Scritto su macchine di laboratorio.
  - Progetto funzionante secondo specifiche (possibili evoluzioni).
  - Progetto su Docker del dipartimento (codice + dati).
  - Installazione librerie/SW: Verifica eseguibilità e permessi.
  - Presentazione congiunta del gruppo (presenza o Teams).

Infine, un ultimo avvertimento: NON RIDURSI ALL'ULTIMO!