ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

# Il progetto del corso di Tecnologie Web - A.A. 2023/24

**Fabio Vitali, Angelo Di Iorio**
**Andrea Schimmenti**

Corso di laurea in Informatica
Alma Mater – Università di Bologna

# Appelli d'esame

- Uno a metà giugno *(in cui mi aspetto MOLTI di voi)*
- Uno a inizio luglio
- Uno a metà luglio *(nella settimana delle tesi)*
- Uno a settembre
- Uno a gennaio 2025
- Uno a febbraio 2025 *(non riducetevi all'ultimo!)*
- Uno a inizio giugno 2025 (solo esame scritto) riservato a chi è in debito dall'anno precedente.

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

# Valutazione delle prove

Scritti e progetti vengono valutati in trentesimi

- Lo scritto pesa il 50% del voto finale
- Il progetto pesa il 50% del voto finale

A partire da quest'anno il progetto ha valutazione come segue:

- Il progetto viene valutato in trentesimi.
- Il progetto base, realizzabile praticamente anche solo seguendo le esercitazioni, si converte in un voto da 18 a 21 (da mediare con il voto dello scritto)
- Il progetto con il primo modulo aggiuntivo si converte in un voto da 18 a 24 (da mediare con il voto dello scritto)
- Il progetto con il primo e il secondo modulo aggiuntivi si converte in un voto da 18 a 27 (da mediare con il voto dello scritto)
- Il progetto con tutti e tre i moduli aggiuntivi genera un voto da 18 a 33 (da mediare con il voto dello scritto)

# Valutazione delle prove

| voto progetto | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Progetto base | 18,00 | 18,25 | 18,50 | 18,75 | 19,00 | 19,25 | 19,50 | 19,75 | 20,00 | 20,25 | 20,50 | 20,75 | 21,00 |
| + I modulo | 18,00 | 18,50 | 19,00 | 19,50 | 20,00 | 20,50 | 21,00 | 21,50 | 22,00 | 22,50 | 23,00 | 23,50 | 24,00 |
| + I e II modulo | 18,00 | 18,75 | 19,50 | 20,25 | 21,00 | 21,75 | 22,50 | 23,25 | 24,00 | 24,75 | 25,50 | 26,25 | 27,00 |
| + I, II e III modulo | 18,00 | 19,25 | 20,50 | 21,75 | 23,00 | 24,25 | 25,50 | 26,75 | 28,00 | 29,25 | 30,50 | 31,75 | 33,00 |

Il voto finale va mediato (50%+50%) con il voto dello scritto.

Ad esempio:

- Voto scritto 26, voto progetto base 26 (cioè 20), voto finale 23
- Voto scritto 26, voto progetto + I modulo 26 (cioè 22), voto finale 24
- Voto scritto 26, voto progetto + I e II modulo 26 (cioè 24), voto finale 25
- Voto scritto 26, voto progetto completo 26 (cioè 28), voto finale 27

In aggiunta, a totale discrezione del docente, possono essere erogati fino a due punti aggiuntivi sul voto finale per scelte particolarmente

ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

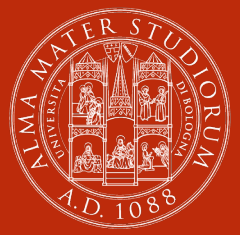
# Il progetto di fine corso

- Un sistema VERO, che funziona e fa cose utili
- Realizzabile sia in laboratorio che a casa.
- Enfasi in parte sulla programmazione (approccio procedurale) ma soprattutto sui documenti attivi (approccio dichiarativo)
- Enfasi sul mashup di tecnologie esistenti e sofisticate

# Il lavoro di team

- Tutti i membri dei team sono tenuti a lavorare e lavorare insieme.
- E' meglio essere parte attiva di un progetto mediocre che passiva di un progetto meraviglioso.
- Non saranno tollerati i portatori di pizze
- Mi riservo all'esame di scoprire il contributo individuale di ciascuno, indipendentemente dalla bontà del progetto consegnato.

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA




ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

# Un po' di teoria

# Background: strumenti per lo sviluppatore web negli anni '20

Lo sviluppo sw più sorprendente ed evidente degli ultimi anni è la nascita di innumerevoli strumenti per il programmatore, con cui realizzare servizi sofisticati

- Framework: invece di scegliere nuovi linguaggi di programmazione, si usano ora librerie per gli scopi più disparati, facilmente integrabili e mescolabili tra loro
- API (Application Programming Interfaces): invece di sviluppare applicazioni monolitiche che svolgono servizi complessi in un'unica maniera, si forniscono meccanismi di manipolazione delle strutture dati fondamentali e accesso agli algoritmi più sofisticati per applicazioni sviluppate dai clienti.

Questo permette incredibile sofisticazione, grande componibilità, e rapidità di sviluppo precedentemente irraggiungibili.

# I framework

- I framework sono librerie che rendono più ricco, sofisticato e semplice l'uso di una tecnologia, come un linguaggio server-side, un linguaggio client-side o le specifiche grafiche di una pagina web.
- Server-side esistono dalla fine degli anni novanta, e hanno reso la programmazione a tre livelli drasticamente più facile.
- Client-side si sono sviluppate a partire dal 2002, su CSS e Javascript, con scopi molto difformi.

# Application Programming Interface (API)

- Librerie, protocolli e strumenti per permettere di utilizzare gli algoritmi ed i servizi messi a disposizione da un software da parte di un altro software, invece che da esseri umani.
- Applicazioni e servizi che forniscono una API delegano ad un'applicazione terza aspetti come interazione, interfaccia, navigazione, ecc. e forniscono via API il solo servizio nudo.
- Uno degli scopi tipici di ricorso ad API è per integrare più servizi in un'applicazione più ricca e potente di quelle utilizzate come base.
- Questo si chiama *mashup* ed è una delle caratteristiche più evidenti di questo periodo storico: mescolare servizi di base per ottenere applicazioni impreviste dai fornitori dei servizi stessi.

# Le API REST

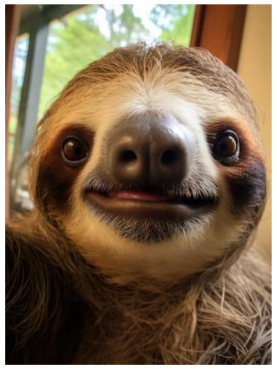
- Le API che ci interessano sono più concretamente quelle che vanno sotto il nome di RESTful API, ovvero che sfruttano al meglio la natura di HTTP e degli URI (i protocolli più importanti del web) per fornire i loro servizi.
- Una API RESTful fornisce:
  - Un URI base a cui accedere per ottenere i servizi
  - Una sintassi degli URI delle entità interrogabili e modificabili
  - Un media type attraverso cui ottenere e fornire dati da utilizzare nei servizi forniti (ad esempio XML, JSON, etc.)
  - Una semantica associata all'uso dei vari verbi HTTP (GET, PUT, POST, DELETE) attraverso i quali attivare e eseguire i servizi offerti.
- La documentazione di un'API tipicamente descrive nel dettaglio nomi (URI), verbi (comandi HTTP) e formati (Internet Media Type) per ogni servizio fornito.

# Le API di servizi locali

- Le seconde API che ci interessano sono quelle che permettono alle applicazioni web di accedere a speciali servizi offerti dal device su cui vengono eseguite (tipicamente accesso a periferiche di I/O particolari.
- La diffusione di TCP/IP su device mobili ha ampliato radicalmente la quantità e qualità di periferiche "non tradizionali" a cui l'applicazione può avere accesso:
  - Telecamera (Camera API)
  - Microfono e sintesi vocale (Speech API)
  - Geolocazione (via GPS o altro) (Geolocation API)
  - Suoni (Web Audio API)
  - Vibrazioni (Vibration API)
  - Telefono (Web telephony API),
  - ecc. ecc.
- Ci sono circa 80 API diverse al momento utilizzabili sulla maggior parte dei browser (purché il device offra la funzionalità)

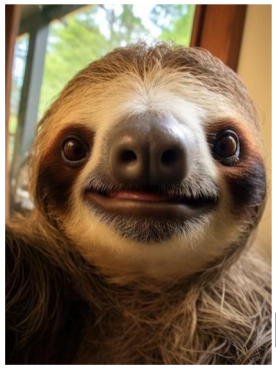

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

# SELFIE



# Ruolo di queste specifiche

- Questo documento contiene le specifiche fondamentali del progetto di fine corso.
- Quanto scritto in nero, salvo esplicite eccezioni, deve essere considerato requisito OBBLIGATORIO per TUTTE LE consegne.
  - Le frasi scritte in arancione si riferiscono a servizi relativi alle parti di estensione del progetto, e sono obbligatorie solo per chi ritiene di partecipare ai progetti estesi.
  - Le frasi in verde corrispondono a vincoli obbligatori introdotti solo per le esigenze del progetto universitario, non necessari o opportuni in un prodotto vero per il mercato esterno.
  - Le frasi in blu sono esempi, e non specifiche di progetto.
- Se una o più delle specifiche qui introdotte non funzionano, il progetto NON è considerato accettabile.
- Un'apposita pagina su Virtuale fornirà in maniera sempre aggiornata le eventuali modifiche ai requisiti.



# Selfie: fondamenti

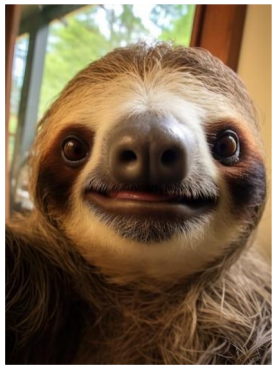
La vita di uno studente universitario è ricca e complessa, con eventi, scadenze, impegni che possono molto diversi tra loro:

- individuali o di gruppo,
- unici (un aperitivo con gli amici questo venerdì alle 19:00) o ripetuti (il calcetto del martedì alle 20:00),
- semplici (appello di TW il 12 giugno alle 11:00) o complessi e strutturati (dal 10 al 21 maggio studiare la teoria del progetto, dal 22 maggio al 5 giugno preparare il progetto, sera del 6 giugno demo con i compagni di gruppo, 7-11 debug del progetto, 7-11 studio dell'eame scritto, 12 esame scritto di TW), etc. Spesso anche sovrapposti.

Attraverso Selfie lo studente UniBo è in grado di programmarsi la propria vita, privata, sociale ed accademica, in maniera flessibile e completa.

Selfie è un'applicazione client + server usabile in maniera equivalente sia da cellulare che da PC, organizzata in moduli che forniscono supporto a tipi diversi di eventi, scadenze ed impegni della vita dello studente.

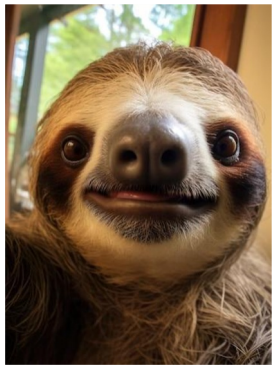
L'applicazione base (obbligatoria) è molto semplice, mentre i moduli aggiuntivi arricchiscono e integrano l'applicazione in un contesto molto più ampio.



# Selfie : Architettura (1)

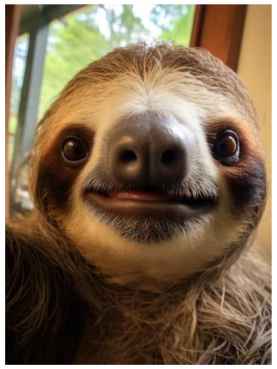
L'architettura di Selfie è basata su una struttura dati, l'*evento:*

- L'applicazione base (18-21) prevede di aggiungere, rimuovere, postare e modificare eventi semplici del solo utente, posizionati in un calendario e di durata nota (intervallo di ore, intere giornate, periodi più lunghi. Gli eventi si possono sovrapporre liberamente. Esistono visualizzazioni comode giornaliere, settimanali e mensili degli eventi inseriti.
- E' realizzato un timer (la view Pomodoro) per organizzare il passo dello studio.
- E' fornito un editor di appunti e note di studio.
- E' realizzato un sistema per navigare nel tempo e arrivare ad una data passata o futura come desiderato.



# Selfie : Architettura (2)

- La prima estensione (18-24) sono i ***sistemi di notifica e geolocalizzazione***: messaggia mentre sto usando l'app, ma anche mentre sto facendo qualcos'altro. La messaggistica deve calibrabile per testo, ripetizione, urgenza crescente, gestione dei ritardi e degli snooze. La geolocalizzazione permette di situare eventi e impegni in un luogo e/o fuso orario. Nessuna applicazione esistente, secondo me, soddisfa bene questo requisito.
- La seconda estensione (18-27) sono ***gli eventi di gruppo***: alcuni eventi e scadenze possono appartenere a più calendari collegati (io e i miei amici) e possono avere priorità e sovrapposizioni diverse. La corretta gestione della privacy è importante: i miei compagni di calcetto sanno solo del calcetto. Prevede anche *l'integrazione con sistemi terzi* (Google Calendar, Apple Calendar, lo standard iCalendar, etc.).
- La terza estensione (18-33) sono i ***sistemi di gestione di progetti complessi*** (ad esempio lo studio di un esame universitario). Ogni progetto è diviso in fasi, le fasi in attività, alla fine delle attività ci possono essere dei milestone. Un progetto organizzato tra più persone può avere attività individuali e di gruppo, periodi di grande autonomia e momenti di sincronizzazione, dipendenze reciproche e attività di monitoraggio e verifica. Un sistema noto di visualizzazione dei progetti si chiama *diagramma di Gantt.*

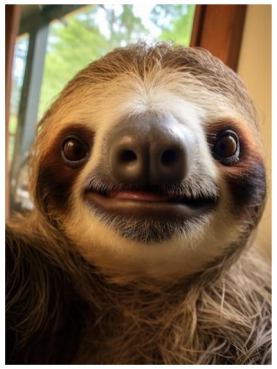


# Architettura dell'applicazione SELFIE

L'applicazione SELFIE prevede un'applicazione client-side in Javascript e/o Typescript collegata in maniera precisa con una parte server-side in Node.js (***non sono ammesse altre tecnologie server-side***).
Le funzionalità specifiche dell'applicazione dipendono dai servizi implementati che a loro volta dipendono dal livello del progetto. Tutti i livelli hanno parti client-side e parti server-side.

- Home e accesso utente
- Calendario
- Note
- Applicazione pomodoro
- Applicazione gestione progetti
- Applicazione Time Machine

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA



# Home page e Accesso Utente

L'utente accede all'app tramite account basato su nome utente e password.
Il record di un account contiene sicuramente nome utente, password e nome vero ed una quantità a piacere di informazioni personali (e.g. selfie personale, data del compleanno da aggiungere al calendario).
La Home page serve per navigare tra le view: Calendario, Pomodoro, Note, Progetti.
Nell'Home vengono mostrate preview dei contenuti delle singole view: ad esempio gli eventi della settimana/giorno corrente, l'ultima nota creata, le attività imminente, report sull'ultimo pomodoro svolto, scadenze imminenti dei progetti.

**Estensione 18-24**

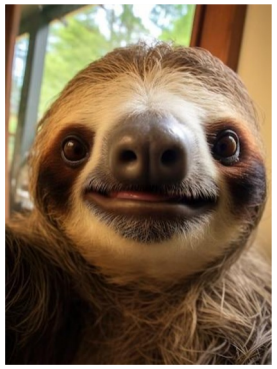
- Gli utenti hanno la possibilità di personalizzare il tipo di preview per ogni view.

**Estensione 18-27**

- Gli utenti possono mandare messaggi e notifiche ad altri utenti (ma non una chat a tutti gli effetti).

**Estensione 18-33:**

- Gli utenti hanno un mini hub dove poter chattare con gli altri utenti.



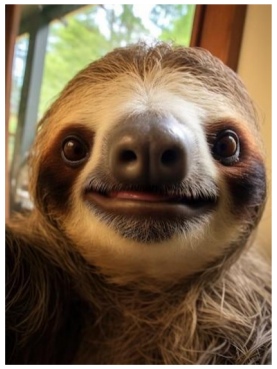
# Calendario - Eventi (1)

l calendario consente ad un utente di creare eventi: questi sono specifici appuntamenti o incontri che hanno una data, un'ora e una durata definite. Hanno un titolo.

- Possono essere una tantum, come un appuntamento medico, o ripetibili, come lezioni settimanali. Possono anche essere lunghi come tutta la giornata o più giorni.
- Gli eventi ripetibili hanno
  - una frequenza (es.: *tutti i giorni, tutti i martedì e giovedì, una volta la settimana, tutti i giorni 4 del mese, tutti i primi lunedì del mese, ecc.*)
  - un numero di ripetizioni (*ripeti indefinitamente, ripeti N volte, ripeti fino alla data XX/XX/XXXX*).
- Gli eventi possono anche includere un luogo fisico o virtuale dove si svolgono.

**Estensione 18-24**

- Gli utenti possono indicare meccanismo di notifica dell'imminente appuntamento
  - usando uno o più meccanismi (il servizio di notifica del sistema operativo e/o un alert, son una mail con un whatsapp, ecc.)
  - Con un certo anticipo (all'ora voluta, un minuto, cinque minuti, un'ora, due ore, un giorno, due giorni prima, ecc.)
  - con una certa ripetizione (ripeti tre volte, ripeti ogni minuto, ripeti ogni ora, ripeti fino a che non rispondo, ecc.).

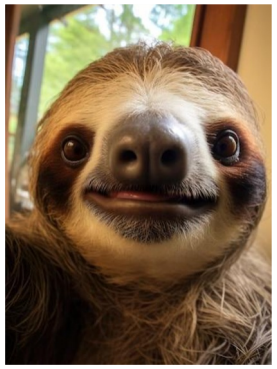


# Calendario - Eventi (2)

## Estensione 18-27

- Gli utenti possono includere altri utenti nei loro eventi
- Gli altri utenti possono accettare, rifiutare o rimandare l'accettazione dell'evento
- Gli utenti possono indicare certi intervalli di tempo (anche ripetuti) come non disponibili per eventi di gruppo
- Gli eventi devono essere integrati con lo standard iCalendar, ed essere liberamente importabili/esportabili. (Google Calendar, Apple Calendar, ecc. anche via mail.

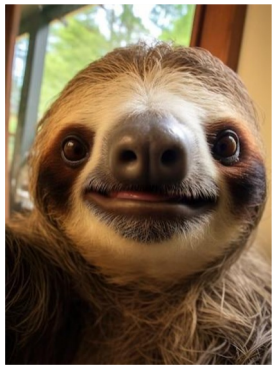
  Alcuni esempi di librerie che aiutano a gestire i file iCalendar:
  - https://www.npmjs.com/package/ical
  - https://www.npmjs.com/package/datebook
  - https://www.npmjs.com/package/@pgswe/ics.js



# Calendario - Eventi (3)

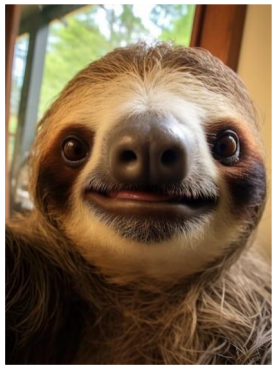
**Estensione 18-33:**

- Alcuni utenti sono risorse (ad esempio una stanza riunioni o un'apparecchiatura): posso includerla nell'evento solo se è libera e in quel caso l'evento viene automaticamente accettato.
- Un utente speciale ha la responsabilità di gestire, oltre al proprio calendario, anche quello di tutte le risorse. Il calendario delle risorse è liberamente esplorabile da chiunque.
- Le scadenze dei progetti creano eventi per specifici utenti che vengono automaticamente aggiunti ai calendari relativi.



# Calendario - Attività

- Il calendario consente ad un utente di creare attività: queste sono azioni o compiti di durata prolungata e non esclusiva che un utente deve completare mentre fa anche altre cose. Non necessariamente devono svolgersi in un momento specifico, ma possono avere una scadenza. Ad esempio, "completare la relazione" ed inviarla entro una certa data.
- In genere non sono associate ad un intervallo di tempo preciso, ma "da adesso ed entro una certa data".
- Il loro completamento deve essere esplicitamente specificato o si trascinano nei giorni successivi (attività in ritardo)
- Le attività devono essere visualizzate nel calendario come scadenza e separatamente come lista.



# Calendario - Attività

**Estensione 18-24**

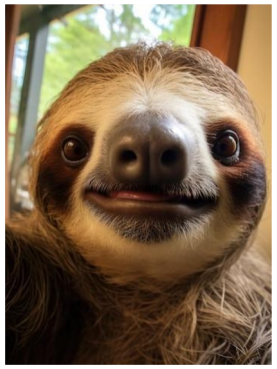
- Gli utenti possono ricevere notifica di urgenza crescente via via che ci si allontana dalla data di scadenza

**Estensione 18-27**

- Le attività possono essere assegnate a più persone in un ambiente di gruppo

**Estensione 18-33**

- Le attività possono essere suddivise in sotto-attività e possono essere correlate a progetti più grandi.

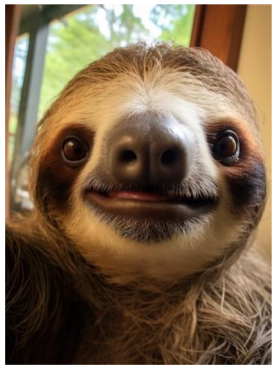


# Le Note

Una nota è un testo di lunghezza arbitraria (fate prove anche con testi molto lunghi) dotata di titolo, categorie (a scelta dell'utente, data di creazione, e data di ultima modifica.

Le note devono essere gestite in una view separata dal calendario. E' possibile duplicare note e copiare ed incollare il contenuto delle note. E' possibile cancellare note.

L'home delle note deve mostrare una preview delle note esistenti (i primi N caratteri, N almeno 200) e consentire di aggiungerne delle nuove. Possono essere categorizzate per ordine alfabetico (dal titolo), per data e per lunghezza del contenuto.



# Le Note

**Estensione 18-24**

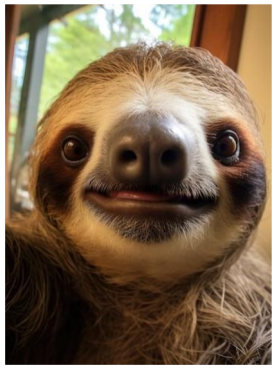
- Le note possono essere scritte in markdown. Si veda https://www.npmjs.com/package/marked

**Estensione 18-27**

- Le note sono dotate di autore e lista di accesso: aperte a tutti, aperte ad alcune specifiche persone, private.

**Estensione 18-33**

- Oltre alle note a testo libero si possono inserire anche liste di cose da fare con spunte cliccabili
- Aggiungendo una scadenza ad un list item si aggiunge automaticamente una attività nel calendario

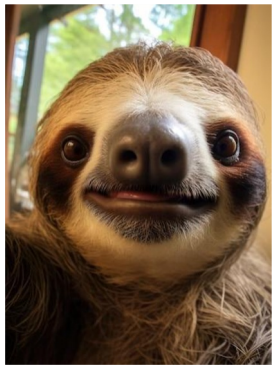


# Il pomodoro

La view Pomodoro gestisce il metodo di studio pomodoro per studiare. Organizza il tempo dello studente in cicli studio-relax secondo una tecnica fissa di 30+5 minuti.

La view è composta da:

- Form per scegliere il tempo di studio e di pausa. Quello standard è 5 cicli da 30 minuti di studio e 5 minuti di pausa.
  - Si possono anche inserire un totale di ore/minuti disponibili e si ottengono una o più proposte di cicli di studio/pausa, per esempio:
  - Input: 200 minuti; Output: 5 cicli da 35 minuti di studio e 5 minuti di pausa.
  - Input: 6 ore; output: 8 cicli da 35 minuti di studio e 10 minuti di pausa.
- Inizio del tempo di studio/pausa successivo forzato da bottone;
- Tasto ricomincia ciclo;
- Tasto fine ciclo.
- Notifica per inizio ciclo, passaggio da una fase alla successiva, fine ciclo.



# Il pomodoro

Si richiede un'animazione (OBBLIGATORIAMENTE fatta in CSS, non una gif) per lo studio e una per la pausa

**Estensione 18-24**

- Programmare cicli di studio su diverse giornate come Evento su calendario, i.e. cliccando sull'evento si viene rimandati alla view pomodoro;
- I cicli previsti e non completati vengono automaticamente passati alle giornate successive e si sommano a quelli previsti per quella giornata.

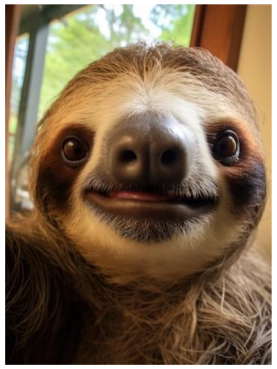
**Estensione 18-27**

- E' possibile mandare una notifica ad un altro utente che gli rende possibile studiare con le stesse impostazioni.

**Estensione 18-33**

- Musica, video su youtube (*music to study to?*), modifica del tempo una volta iniziato.

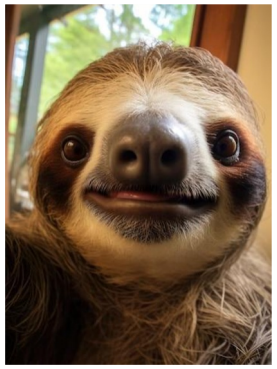
ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA



# La gestione progetti [18-33]

- Un progetto è una lista di attività in sequenza o parallele attribuite ad uno o più attori. Le attività sono raggruppate in fasi e sottofasi.
- Le attività possono partire quando è disponibile un input (anche vuoto) e producono un output (anche solo un booleano "attività completata"). Se l'output dell'attività X è l'input dell'attività Y, le due attività sono sincronizzate e i problemi di X ricadono sullo svolgimento di Y sotto forma di traslazione (le due settimane sono ancora due settimane ma finiscono dopo) o contrazione (le due settimane diventano una settimana e mezzo per rispettare la data di conclusione originaria).
- Alcuni output sono importanti e sono chiamati milestones. Le attività che generano milestones hanno delle date di conclusione precise e non flessibili (quindi non possono spostarsi ma solo contrarsi).
- Il progetto tutto, e ogni attività individuale, vengono descritti da note più o meno brevi. Gli input sono note (eventualmente un link ad un file online) e gli output sono una nota (eventualmente un link ad un file online).

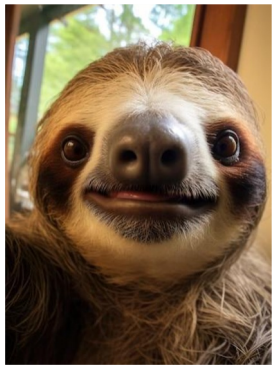
ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA



# La gestione progetti [18-33]

Ogni attività è associata ad un valore di status continuamente aggiornato e affidabile:

- Non attivabile (non è ancora disponibile l'input relativo)
- Attivabile (ma non attivata: l'input è presente ma l'attore non ha ancora dichiarato di averla iniziata)
- Attiva (l'attore ha dichiarato di averla iniziata)
- Conclusa (l'attore ha dichiarato di averla conclusa ed un output è disponbile)
- Riattivata (il capo-progetto ha rifiutato l'output e ha richiesto revisioni)
- In ritardo (la data di conclusione è passata ma l'output non è ancora disponibile)
- Abbandonata (la data di conclusione è passata da molto tempo, oppure l'attore ha dichiarato di non volersene più occupare)



# La gestione progetti [18-33]

Inizio e fine attività sono eventi che finiscono sul calendario con modalità di visualizzazione separate dagli eventi normali ed appropriate allo scopo.

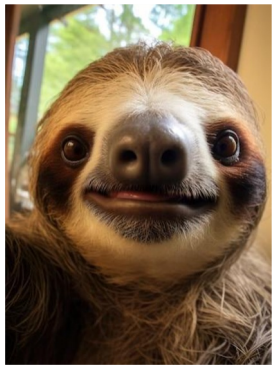
Due visualizzazioni speciali per i progetti sono sempre disponibili: una a lista (ordinata sia temporalmente sia per attore coinvolto) e una a GANTT.

Chiunque può creare progetti (e diventa il capo-progetto) e coinvolgere altri utenti.

Solo il capo-progetto può modificare il progetto, aggiungere, togliere o raggruppare attività e cambiare le sincronizzazioni tra le attività. Solo il capo-progetto in caso di ritardi può decidere se traslare o contrarre le attività sincronizzate a quella in ritardo.

Gli attori coinvolti ricevono una notifica sulla decisione del capo-progetto.

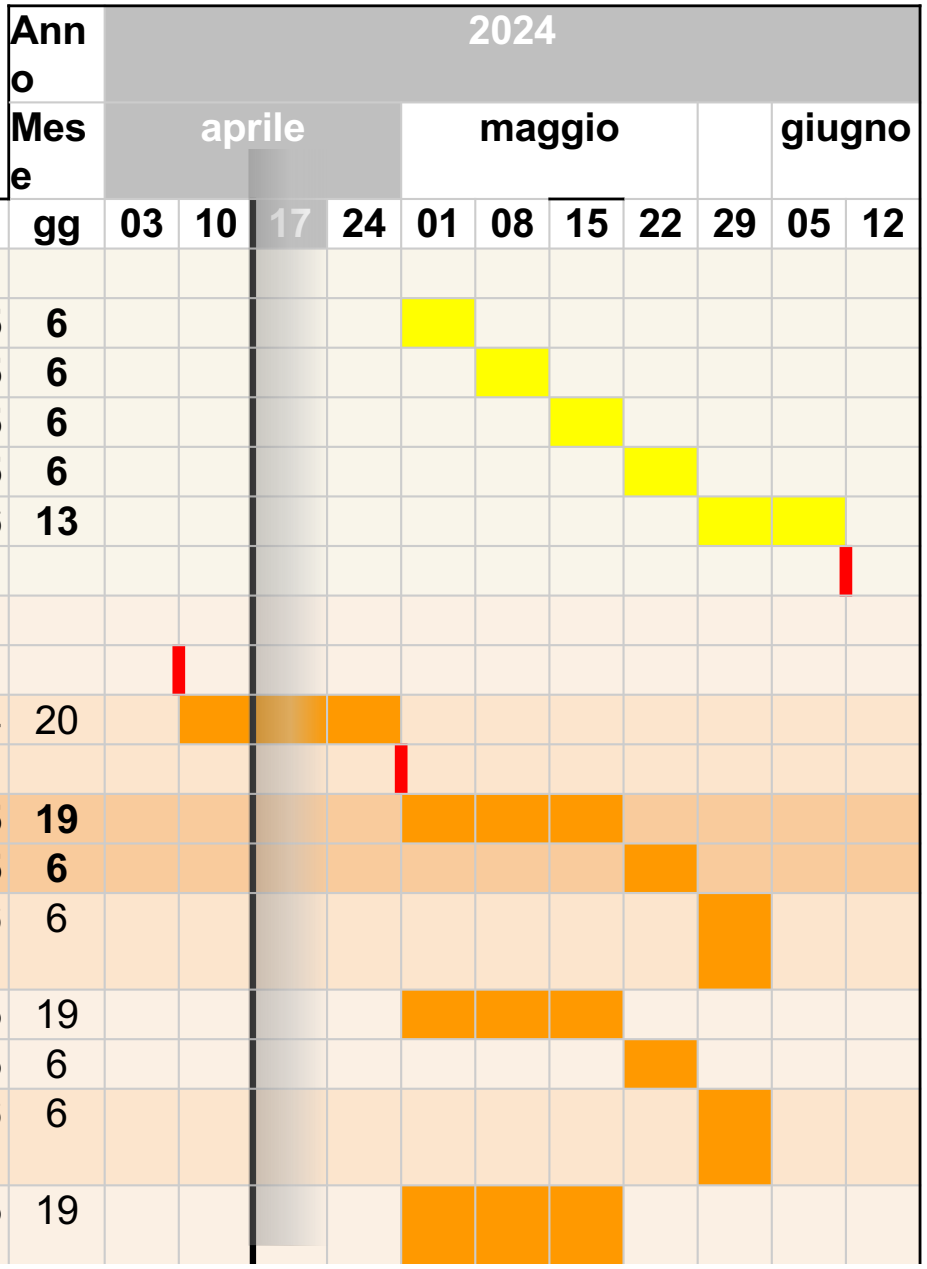
ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

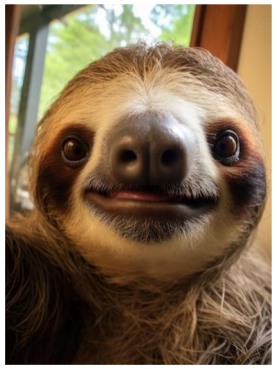


# La gestione progetti [18-33]

*Supponiamo oggi sia il 17 aprile 2024*

| Fase e attività | Attore | Inizio | Fine | gg |
|---|---|---|---|---|
| **Preparazione Esame scritto** | | | | |
| Studio teoria | **io** | **01/05** | **07/05** | **6** |
| Studio esercizi HTML | **io** | **08/05** | **14/05** | **6** |
| Studio esercizi Javascript | **io** | **15/05** | **21/05** | **6** |
| Studio esercizi Framework | **io** | **22/05** | **28/05** | **6** |
| Ripasso globale | **io** | **29/05** | **11/06** | **13** |
| *Esame scritto* | | *12/06* | | |
| **Preparazione Progetto** | | | | |
| *Ricezione specifiche* | | *10/04* | | |
| Progetto base | tutti | 10/04 | 30/04 | 20 |
| Test progetto base | tutti | 30/04 | | |
| **Realizzazione modulo I** | **io** | **02/05** | **21/05** | **19** |
| **Test modulo I** | **io** | **22/05** | **28/05** | **6** |
| Test integrazione base + mod I | tutti | 29/05 | 04/06 | 6 |
| Realizzazione modulo II | Andrea | 02/05 | 21/05 | 19 |
| Test modulo II | Andrea | 22/05 | 28/05 | 6 |
| Test integrazione base + mod II | tutti | 29/05 | 04/06 | 6 |
| Realizzazione modulo III | Beatrice | 02/05 | 21/05 | 19 |





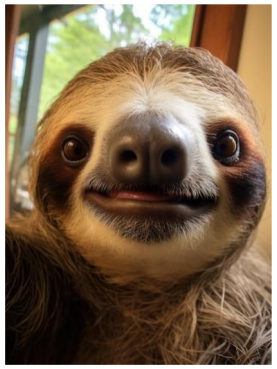
# La Time Machine

Durante la presentazione i docenti non vogliono aspettare tre mesi che si concluda una fase e arrivi la relativa notifica. Quindi vogliono poter viaggiare nel tempo.

Questo significa che OGNI ANNOTAZIONE TEMPORALE DI QUALUNQUE TIPO sia del server che del client non dipende dalla data e dall'ora del sistema operativo, ma dalla data e dall'ora di un apposito servizio chiamato Time Machine.

- per default è allineato alla data e all'ora del sistema operativo ma è possibile modificare data e ora in avanti ed indietro in qualunque momento.
- Una apposita parte dell'interfaccia, separata dalle altre view ma sempre visibile, e con colori contrastanti rispetto al resto dell'interfaccia, permette all'utente di cambiare data ed ora della Time Machine,
- Immediatamente e senza reload le view cambiano e riflettono la nuova data.
- Le notifiche relative al giorno appena specificato si attivano, ma non quelle dei giorni precedenti.
- Un semplice pulsante rimette a posto la view alla data e ora del sistema operativo.

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA



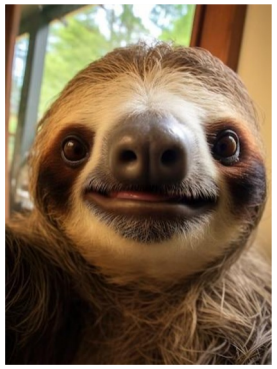
# La parte facoltativa [per tutti]

Questa parte è facoltativa per tutti e di grande difficoltà (creativa, non realizzativa).

Verrà fatta molta attenzione alla semplicità e flessibilità del meccanismo di inserimento, spostamento e modifica degli eventi e delle attività (punti in più nella valutazione a discrezione del docente).

- **Efficacia**: attenzione alla facilità a commettere errori, alla precisione, alla possibilità di lavorare su molti eventi insieme
- **Efficienza**: tempo e numero di azioni atomiche necessarie per richiedere la modifica ed effettuarla
- **Soddisfazione**: sofisticazione grafica e chiarezza di visualizzazione della modifica e degli eventi/attività modificate

Il vs. docente non è soddisfatto di nessuno dei metodi che ha sperimentato finora.

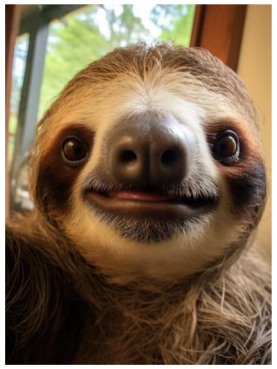


# La parte facoltativa [per tutti]

**Hints**

- Le ore del giorno non sono tutte uguali: Io non ho impegni tra mezzanotte e le otto, e dalle otto alle 18 sono per lo più impegni di lavoro (tipicamente cadenzati e regolari), dopo le 18 sono per lo più impegni del mio tempo libero (in generale eventi isolati e non ripetuti - ma attenzione alla serata calcetto). Stessa cosa per giorni lavorativi e weekend.
- Nessuno fissa appuntamenti alle 10:07 o alle 16:41. Gli eventi e le attività si allineano ad ore tonde (ore intere, mezz'ore, quarti d'ora, ecc.) A MENO CHE l'utente proprio non richieda un minuto particolare.
- Le notifiche delle attività in ritardo debbono prevedere un meccanismo crescente di urgenza e disturbo, in modo che diventi sempre maggiore la pressione sull'utente.

Se volete cimentarvi con idee nuove, dovete fare la doppia modalità di input, quella tradizioale e quella "speciale" e quella speciale dovete documentarla in maniera specifica.-



# Requisiti di progetto

Tutte le parti in nero sono obbligatorie

Tutte le parti in arancione appartengono ad una o più estensioni.

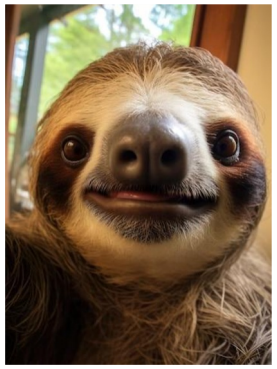
Le parti in verde sono necessarie solo per il progetto e sono obbligatorie.

Le parti calendario, note e pomodoro sono mobile first, sono realizzate con il framework Javascript/Typescript e CSS preferito, e sono pensate per essere usate velocemente e facilmente da tutti. Su PC è comunque possibile compiere in maniera adeguata le funzionalità previste.

La Time Machine è sempre visualizzata su PC e immediatamente accessibile (senza navigazione) su mobile.

La parte gestione progetto è PC-first, e sfrutta completamente e appropriatamente uno schermo grande. Deve essere fatta in Javascript puro (vanno bene Web Components) e con il framework CSS preferito. Device mobili debbono comunque permettere di compiere in maniera adeguata le funzionalità previste.

Tutti i dati vengono memorizzati su un DB Mongo sul server del dipartimento.



# Requisiti di progetto

All'atto della presentazione del progetto tutti i database sono già riempiti con un numero ragionevole di utenti, eventi, note, attività, progetti, ecc.

- Se l'applicazione lo prevede sono già creati 4 account: "fv1", "fv2", "fv3" e "fvPM", tutti con password "12345678". Sono poi anche già creati altri tre o quattro account con nomi semplici e password sempre "12345678". Altri utenti possono essere creati a vostro piacimento.
- Attività ed eventi sono liberamente attribuiti ad uno o più di questi utenti. Sicuramente fv1 ecc hanno degli eventi a scadenza ravvicinata vicino alla data dell'esame. Venite alla presentazione del progetto con una lista stampata dei prossimi eventi ed attività dei vostri utenti.
- In caso di [Estensione 18-33], fvPM ha creato uno o due progetti anche complessi composti di fasi e sottofasi e varie attività. Il giorno dell'esame dovremo essere a metà di una di queste fasi e vicino al completamento di qualche attività.
- Ci sono svariate note di varie lunghezze e complessità (soprattutto se usate Markdown)

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

# Valutazione del progetto

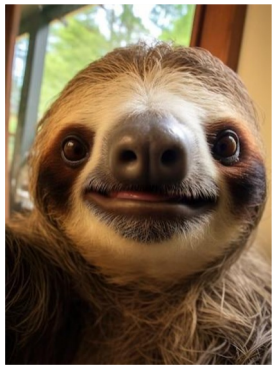
# Criteri di valutazione (1)

Criteri di valutazione saranno:

1. la generalità dei tool:
   - quanto le soluzioni per la compatibilità sono forzate e quanto sono frutto di scelte ottimali per framework, organizzazione del codice e uso corretto delle tecnologie disponibili
2. la flessibilità:
   - quanto le soluzioni tecniche adottate sono solide, strutturate, facilmente comprensibili, facilmente estendibili, facilmente adattabili a nuovi device / browser / sistemi operativi / modelli di dati / modelli di annotazione

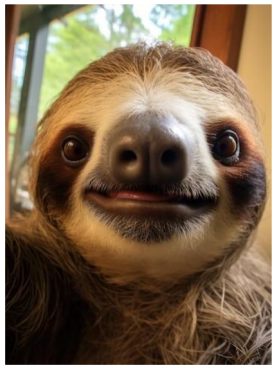
# Criteri di valutazione (2)

3. l'usabilità:
   - Quanta attenzione è data alle esigenze di utenti (sia giocatori, sia clienti, sia dipendenti che usano il back-office) che non conoscono i dettagli del modello di applicazione utilizzata.
4. la sofisticazione grafica
   - Quanta attenzione viene data alla presentazione delle informazioni, al rapporto tra dimensioni delle maschere e dimensioni dei dati da rappresentare, al rapporto tra label comprensibili e dati formalizzati, alla corretta differenziazione nei tipi di dati e di annotazioni.



# Vincoli hard

1. Le applicazioni sono fatte con tecnologie diverse:
   - il back-office è realizzato con Node, MongoDB e vanilla Javascript. Express e moduli ok. Moduli installabili con npm, ok. Assolutamente NO a php, perl, python, java, ruby, MySQL e altre tecnologie server side fuori dal mondo Node.
   - L'applicazione home + calendario + pomodoro + note è realizzata con un framework a scelta tra Angular, React, Vue, Svelte, etc.
   - L'applicazione gestione progetto è realizzata senza framework (vanno bene Web Components) e vanilla Javascript.
2. Il framework per la grafica è libero ma mi aspetto sofisticazione grafica, facilità d'uso e eleganza. Vanno bene Bootstrap, Tailwind, Foundation, ma anche altri a vostra scelta purché compatibili con gli altri vincoli.
3. Il deploy ***deve*** avvenire su due container docker sulle macchine del dipartimento.
4. Tutti i database vengono presentati già popolati.



# Vincoli hard

1. Oltre al deploy è necessario consegnare in apposita directory "sources" dentro alla vostra directory di progetto TUTTI I SORGENTI liberamente leggibili da chiunque, e un README.txt con una breve descrizione dell'organizzazione in file del progetto.
2. Se volete sottomettere una modalità di input speciale (parte facoltativa) create un'apposita pagina HTML con documentazione ed immagini esaurienti delle parti più importanti della vostra proposta.

# Progetti e moduli

- I progetti vengono installati su un docker (macchina virtuale) su un server del dipartimento, nessuna eccezione permessa.
- I progetti 18-27 e 18-33 sono presentati di persona su appuntamento.
  - Gruppi di 2-3 persone, mai più di 3, mai meno di 2.
- I progetti 18-21 e 18-24 vengono sottoposti a prevalutazione e potranno generare o una valutazione senza presentazione o una presentazione con tutti i membri del gruppo.
  - I progetti 18-21 sono individuali
  - I progetti 18-24 gruppi di 1-2 persone.
- Dentro al file README.txt includete sempre
  - Nomi, cognomi, email (unibo!) e contributo individuale
  - una (*breve)* documentazione scritta delle scelte implementative effettuate, in particolare delle tecnologie scelte.

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

# Suggerimenti per l'esame

- Venite alla presentazione con il progetto che funziona. Se non va io vi faccio tornare. Per questo preferisco
  - vedervi una settimana dopo l'appello con il progetto che funziona

  piuttosto che
  - perdere tempo con la presentazione di un progetto che non va,
  - mandarvi via in lacrime, e
  - vedervi una settimana dopo l'appello con il progetto che funziona

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

# Il lavoro di team

Tutti i membri dei team sono tenuti a lavorare e lavorare insieme.
E' meglio essere parte attiva di un progetto mediocre che passiva di un progetto meraviglioso.
Non saranno tollerati i portatori di pizze
Mi riservo all'esame di scoprire il contributo individuale di ciascuno, indipendentemente dalla bontà del progetto consegnato.

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

# Il contributo individuale

- Ogni membro di ogni team deve dimostrare di aver contribuito in maniera determinante alla realizzazione del progetto.
- Ad inizio della presentazione ogni membro dichiara che cosa ha realizzato, e il docente, in totale autonomia, decide se questo contributo è o non è sufficiente.
- Realizzare solo HTML e CSS non è sufficiente.
- Realizzare parti marginali del codice (login, logout, lettura delle preferenze, ecc.) non è sufficiente
- Un candidato ideale si è occupato sia della parte HTML/CSS, sia della parte di programmazione, sia client  sia server.
- La distribuzione ideale dei compiti è funzionale e non architetturale.

# Organizzazione dei team

- Ogni persona decide in anticipo se è interessata a sostenere l'esame in estate, autunno, sessione straordinaria o indeciso.
- Tutti gli studenti si dividono in team di 2-3 persone. Non sono accettati gruppi di più di 3 persone.
- Non sono accettati progetti singoli tranne eccezioni estreme e ben giustificate.
- Ogni team porta il progetto insieme (non ci sono eccezioni!). Il team dichiara in anticipo la natura del contributo di ciascun membro oppure accetta che chiunque sia interrogato (e nel dettaglio!) su tutto il progetto.
- Io non sono coinvolto nell'organizzazione dei team.

# Generative AI, una riflessione

- Di moda
- Competente
  - Addestrata anche su molti manuali di programmazione web
  - Passerebbe con un buon voto lo scritto di TW
- Allettante

**"E se facessi fare il progetto di TW a ChatGPT?"**

# Generative AI, una riflessione

***Sì, ma...***

- Ha ancora *(forse solo temporaneamente)* un pessimo modello grafico
  - oggetti mal posizionati, troppo piccoli, troppo grandi, marginature asimmetriche, sovrapposizioni, ecc.
- Non è *(forse solo temporaneamente)* in grado di gestire grandi progetti complessi
  - Fraintendimenti
  - Limiti nella lunghezza del prompt
  - Allucinazioni
- Ha *(credo permanentemente)* uno stile molto riconoscibile
  - Stile perfetto come è perfetto un fiore finto.

# Generative AI, una riflessione

Quindi:

- Da quest'anno accetto che si possano usare software di AI come ausilio nella realizzazione del progetto di TW.

Ma:

- L'uso deve essere limitato a problemi difficili e/o ripetitivi
- L'uso deve essere documentato esplicitamente. *Mi arrabbio altrimenti.*
- I membri del gruppo debbono comunque essere ***consapevoli*** e ***padroni*** del loro codice
  - capiscono cosa fa ogni singola riga del codice
  - sono in grado di mettervi mano e modificarlo a piacimento senza AI
- All'esame scritto ChatGPT non c'è, e verranno richieste competenze che richiedono di sapere scrivere codice corretto e in velocità.

ChatGPT, se lo usate, lo usate come *copilota*

# Attenzione all'appello di febbraio

- Quanto detto NON si applica all'appello di gennaio/febbraio.
- Il **31 gennaio 2025** si conclude la possibilità di presentare il progetto di quest'anno. Mi riservo di usare febbraio per eventuali richieste **ben motivate** dell'ultimo minuto.
- Non riducetevi all'ultimo, cercate di portare il progetto negli appelli estivi ed autunnali
- Gli slot a disposizione per presentare il progetto a gennaio non sono infiniti. Tutti gli anni questi slot (più del doppio degli altri appelli) si esauriscono molto presto.
- E' necessario fare richiesta di uno slot entro il **31/12/2024**. Non accetto richieste dopo il **10/1/2024**.
- Noi cerchiamo di ***non*** essere più esigenti a gennaio-febbraio, ma forse potrò dedicare meno attenzione al vostro ottimo progetto.
- Non riducetevi all'ultimo (l'ho già detto, lo so)

# Flessibilità del corso

- Prova scritta e prova di progetto sono indipendenti.
  - Potete provare prima lo scritto e poi il progetto (*anche se è peggio*), o prima il progetto e poi lo scritto (*che sarebbe meglio*).
  - Il progetto è sempre di gruppo tranne motivate eccezioni
  - Lo scritto è sempre individuale
- Potete provare lo scritto tutte le volte che volete
  - Il voto precedente verrà cancellato solo se consegnate un nuovo scritto
  - Gli scritti sono solo alle date degli appelli ufficiali
- Potete presentare il progetto tutte le volte che volete
  - Solo se lo decide consensualmente TUTTO IL GRUPPO
  - Potete ritirarvi dalla presentazione del progetto in qualunque momento e tornare una settimana o due dopo con le correzioni che ritenete opportune.

# Rigidità del corso
## *(nessuna eccezione per nessun motivo)*

- **Lo scritto avviene sulle macchine di laboratorio** (Ercolani, tipicamente) su rete parzialmente isolata: verificate di saper usare quelle macchine e il software che ci è installato.
- **Il progetto deve funzionare**. Completamente ed esattamente secondo specifiche. Nel corso del tempo queste specifiche possono anche evolvere.
- **Il progetto deve risiedere su un docker del dipartimento**. Questo include SIA il codice SIA tutti i dati del progetto. *Programmate la transizione con il dovuto anticipo.*
- **Potete installare librerie e SW per il progetto** a vostro piacimento, MA verificate prima che sia eseguibile sulle macchine e sui sistemi operativi offerti dal dipartimento e con i permessi d'uso di un utente normale,
- **Il progetto deve venire presentato da tutto il gruppo insieme**, in presenza oppure online su MS Teams. In nessun caso è accettabile che si presenti in una data una parte del gruppo e in una data diversa il resto del gruppo.

# Attenzione!

**Vi ho già detto di non ridurvi all'ultimo?**

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

**Fabio Vitali, Angelo Di Iorio,
Andrea Schimmenti**

Dipartimento di Informatica – Scienze e Ingegneria
Alma mater – Università di Bologna

www.unibo.it