

Proposta progetto Tecnologie Web

Observa

Giovanni Orlandi / A.A. 2024-2025

- **Obiettivo**
- **Targets della supervisione**
- **Tecnologie selezionate**
- **Livelli di Servizio**
 - Versione Free (Accesso gratuito previa registrazione)
 - Versione Pro (Accesso tramite abbonamento mensile o annuale)
 - Funzionalità comuni
- **Frontend**
- **Criticità e Strategie Realizzative**
 - Principali criticità individuate
 - Strategia proposta
 - Requisiti per l'Integrazione
- **(Facoltativo) Possibili Sviluppi Futuri**

Obiettivo

Il progetto ha l'obiettivo di sviluppare un servizio web che offre una panoramica delle attività di supervisione di server e/o servizi web, aggregando diverse metriche significative e monitorandone l'evoluzione nel tempo attraverso una visualizzazione a vari livelli di granularità.

Gli utenti target principali di questo progetto sono gli amministratori di sistema, ai quali viene fornito uno strumento essenziale per analizzare lo stato di salute dei server e facilitare il monitoraggio di un ampio numero di macchine e parametri all'interno di un'unica applicazione web.

Targets della supervisione

Le attività di supervisione, di seguito dettagliate, si applicano a due possibili entità:

- server remoti
- servizi web remoti

che potranno essere raggruppati per controllarne la visibilità da parte degli utenti autorizzati.

Le attività visualizzate dal sistema comprendono:

- rilevazione periodica e archiviazione del valore di parametri vitali del server monitorato, quali uso di RAM, CPU e Storage
- stato di avanzato dei backup periodici incrementali del server
- verifica della disponibilità del server/servizio web remoto
- misure dei tempi di risposta del server/servizio web remoto

come di seguito dettagliato.

Tutte queste grandezze dovranno essere consultabili sia in forma sintetica mediante grafici riassuntivi, sia in dettaglio mediante tabelle numeriche.

Tecnologie selezionate

Il progetto verrà realizzato utilizzando esclusivamente tecnologie open-source, riassunte nella seguente tabella.

Funzionalità	Tecnologia	Tipo di Software	Lingua di Programmazione	Utilizzo Tipico
Observa - backend	Django	Web Framework	Python	Framework per siti web e app web complesse
Observa - frontend	HTMX	Libreria frontend	HTML + JS	Web app dinamiche senza SPA, interazione AJAX
Raccolta e analisi di metriche in tempo reale	Prometheus	Sistema di monitoraggio	Go	Monitoraggio di sistemi, server e applicazioni
Statistiche e realizzazione di backup incrementale di dati	Restic	Strumento di backup	Go	Backup automatizzato di file e directory
Monitoraggio della disponibilità di siti e servizi web	Uptime Kuma	Monitoraggio dell'uptime	JavaScript (Node.js)	Ping di servizi web, notifiche di down

Livelli di Servizio

Il progetto prevede l'implementazione di due distinti livelli di utilizzo del servizio:

Versione Free (Accesso gratuito previa registrazione)

La versione gratuita consente agli utenti di accedere esclusivamente alla funzionalità di **monitoraggio della disponibilità di siti e servizi web**, con le seguenti caratteristiche:

- Definizione di un massimo di 10 end-point monitorati.
- Verifica periodica della raggiungibilità (risposta HTTP/HTTPS compresa tra 200 e 299).
- Misurazione del tempo medio di risposta.

In generale, le informazioni e le statistiche disponibili in questa modalità saranno limitate rispetto alla versione a pagamento.

Versione Pro (Accesso tramite abbonamento mensile o annuale)

La versione Pro, disponibile previa sottoscrizione di un abbonamento, estende le funzionalità disponibili, offrendo:

- **Monitoraggio avanzato di siti e servizi web**, con:

- Supporto per un numero maggiore di end-point.
- Verifica della presenza di keyword specifiche nella risposta.
- Configurazione della frequenza dei controlli.
- (Altre funzionalità soggette a conferma sulla base delle capacità effettive di Uptime Kuma.)
- Consultazione dello **stato di salute dei server assegnati**.
- Controllo anche sui backup con, ad esempio, verifica dei **timestamp** e delle **dimensioni**.

Funzionalità comuni

Indipendentemente dal livello di servizio sottoscritto, tutti gli utenti potranno:

- Configurare autonomamente la lista degli end-point web da monitorare
- Gestire il proprio profilo personale, inclusa la possibilità di caricare una foto profilo.
- Associare un'immagine rappresentativa (ad esempio un logo) a ciascun end-point monitorato, per facilitarne l'identificazione visiva.

Frontend

A livello di impostazione grafica e funzionale, il design del progetto si ispirerà, in parte, al modello presentato nel seguente esempio:

- [Demo Rudder.io](#)

Infatti, dopo aver visionato alcuni esempi ritengo questo un buon compromesso tra chiarezza, semplicità ed efficacia nella visualizzazione.

Ovviamente rimane solamente una fonte di ispirazione di partenza ed il layout finale sarà adattato alle specifiche del progetto.

Criticità e Strategie Realizzative

Il progetto si pone come obiettivo principale la realizzazione di una **visione sinottica** delle attività in corso, sfruttando appieno le funzionalità offerte dalle tecnologie selezionate (Prometheus, Restic e Uptime Kuma). A queste soluzioni verrà delegata completamente sia la raccolta delle metriche sia la loro archiviazione.

È una scelta progettuale consapevole quella di **evitare ogni tipo di duplicazione** di funzionalità o dati.

Tuttavia, l'interfaccia utente sarà progettata in modo da **mascherare l'eterogeneità tecnologica** sottostante, fornendo una rappresentazione dei dati coerente e normalizzata.

In quest'ottica, il progetto si configura principalmente come un **layer di presentazione**, mentre il database interno sarà utilizzato principalmente per la gestione degli utenti, delle relative permissions e per la configurazione di alcuni parametri operativi.

Principali criticità individuate

Le scelte progettuali sopra descritte comportano alcune criticità, tra cui:

- **Eterogeneità delle strategie di archiviazione:** ogni tecnologia adotta formati e logiche proprie per il salvataggio dei dati.
- **Diversità nei protocolli di comunicazione:** il dialogo con le tecnologie avverrà tramite protocolli differenti.
- **Forte dipendenza tecnologica:** la sostituzione di una delle tecnologie di monitoring comporterebbe la necessità di riscrivere anche la logica di presentazione dei dati.

Sebbene possa sembrare vantaggioso archiviare nel database del progetto tutte le metriche storiche, tale approccio introdurrebbe **duplicazione di dati** e **aumento della complessità**, in contrasto con i principi di leggerezza e manutenibilità prefissati.

Strategia proposta

Per ridurre l'impatto delle criticità sopra elencate, si propone di **introdurre un set di API interne**, responsabili di:

- Interrogare i diversi sistemi di monitoring utilizzando i protocolli nativi.
- Normalizzare i dati ricevuti in un formato unico e omogeneo.
- Fornire al frontend un'interfaccia dati stabile, indipendente dai dettagli tecnologici sottostanti.

Questa strategia consente di **disaccoppiare il livello di presentazione** dal livello di raccolta delle metriche, aumentando l'agilità e la futura manutenibilità del progetto.

Requisiti per l'Integrazione

L'accesso ai dati di monitoring richiede che i server da monitorare abbiano preventivamente installato e correttamente configurato i servizi necessari (**Prometheus**, **Restic** e **Uptime Kuma**). La predisposizione di tali servizi sarà **a carico dell'utente**.

Si assumerà pertanto che, al momento della registrazione di un nuovo endpoint da parte dell'utente, i dati di monitoring saranno già disponibili secondo le modalità previste.

Eventuali verifiche e definizioni più precise verranno effettuate durante le fasi di realizzazione del progetto.

Non è previsto dunque un servizio di assistenza di primo livello per la configurazione dei server. Questa scelta deriva dalla volontà di **concentrare il progetto esclusivamente sulla realizzazione e sull'utilizzo dell'applicazione web**, in coerenza con gli obiettivi didattici dell'esame di **Tecnologie Web**.

(Facoltativo) Possibili Sviluppi Futuri

- **Reportistica avanzata**

Consentire agli utenti della versione "Pro" di esportare report personalizzati in formato PDF, selezionando esplicitamente le metriche e le caratteristiche da includere.

- **Monitoraggio specifico per PostgreSQL**

Qualora i server monitorati utilizzino PostgreSQL, raccogliere metriche dedicate tramite l'integrazione con **PostgreSQL Exporter**.

- **Monitoraggio specifico per progetti Django**

Per i server che ospitano istanze Django, raccogliere metriche specifiche tramite l'integrazione con **django-prometheus**.

- **Miglioramento della sicurezza tramite autenticazione a due fattori (2FA)**

- **Possibilità di accesso tramite account Google (OAuth2)**

- **Gestione di gruppi di utenti**

Introdurre la possibilità di creare gruppi di utenti (ad esempio per rappresentare diverse aziende o team), con gestione centralizzata delle configurazioni e degli abbonamenti.

- **Gestione automatizzata dei pagamenti degli utenti pro**

In una fase iniziale nella quale il numero degli utenti sarà ragionevolmente ridotto la gestione amministrativa degli abbonamenti viene realizzata manualmente.