

# Laboratorio Acceleratori 2024 • Controls

Sessioni: 3 e 10 Aprile 2023

Docenti: Alessandro Stecchi (Istituto Nazionale di Fisica Nucleare), Giampiero Di Pirro (Istituto Nazionale di Fisica Nucleare)

## Spunti di riflessione

Si ritiene utile lasciare agli studenti di questo corso una traccia sintetica di quanto detto e fatto nel corso della giornata di studio e laboratorio.

Questa traccia contiene degli spunti di riflessione che ogni studente potrà usare come crede: per ricordare meglio quello che è stato detto nella presentazione introduttiva<sup>1</sup>, o per approfondire qualche argomento per curiosità personale o anche – in futuro – per approcciare un problema relativo all'IT (*Information Technology*) con una diversa forma mentale.

In ogni caso, questo è da considerarsi un documento informale e non è richiesta in alcun modo la sua lettura o l'utilizzo dei suoi contenuti per la stesura della relazione di attività.

### Come cambia il modo di realizzare i sistemi informatici

Nello sviluppo del software e nella realizzazione delle infrastrutture hardware attualmente si dispone di una tecnologia molto evoluta anche solo rispetto a pochi anni fa e questo modifica radicalmente sia il modo in cui si scrivono le applicazioni sia come si progettano le infrastrutture e le piattaforme di calcolo.

**Keywords:** [virtual machines](#), [dockers](#), [Kubernetes](#), [orchestrators](#), [high throughput networks and storage](#).

### Come cambia il metodo di lavoro

Le moderne metodiche nel campo dello sviluppo software adottano pratiche innovative volte ad accelerare il rilascio di funzionalità al cliente. Per esempio l'approccio *AGILE*<sup>2</sup> consente di lavorare per iterazioni successive (*sprints*) con il rilascio di nuove funzionalità alla fine di ogni iterazione. Questo metodo è nella maggior parte dei casi vincente rispetto al tradizionale approccio *waterfall* nel quale il prodotto viene rilasciato solo alla fine di un unico processo di sviluppo costituito da una serie di azioni reciprocamente propedeutiche. Tutto questo non si applica solo allo sviluppo SW ma in generale alla realizzazione di qualsiasi cosa: un impianto, una attività, etc.

**Keywords:** [AGILE](#), [Continuous Integration & Continuous Delivery](#), [Project Tracking](#), [DevOps](#), [Quality](#), [Documentazione](#), [Asset Management](#), [OOD](#), [OOP](#)

---

<sup>1</sup> <https://agenda.infn.it/event/35503/contributions/195800/attachments/104175/145963/Controlli.pdf>

<sup>2</sup> <https://www.agilealliance.org/agile-essentials/>

## Caratteristiche dei sistemi di controllo

Nell'ambito di questo corso, ci siamo focalizzati sui sistemi di controllo di installazioni scientifiche, basati sull'impiego di processi di controllo realizzati in software e operanti tramite computers. Abbiamo anche detto che – in sostanza – il controllo di un acceleratore di particelle non differisce da quello di un impianto industriale o medicale o di altra natura. D'altra parte abbiamo anche visto che ci sono delle necessità diverse in questi diversi contesti che richiedono una ottimizzazione del progetto di controllo finalizzata a renderlo adatto ad operare nel contesto specifico.

**Keywords:** prognostica AD (Anomaly Detection), affidabilità, scalabilità, single point of failure, sicurezza, flessibilità.

## Descrizione dei sistemi di controllo

Abbiamo visto come negli impianti che si estendono su aree medio/vaste si tende ad adottare architetture di sistemi distribuiti. In questi casi, i processi di controllo sono impiantati su computers fisici posti in prossimità (o anche all'interno) dei dispositivi ove sia richiesta una bassa latenza di controllo e su macchine remote – sia fisiche sia virtuali – ove i canali di comunicazione e la rete garantiscano la banda e la latenza necessarie.

Abbiamo visto come l'architettura di un sistema di controllo che deve operare su un orizzonte temporale lungo deve essere il più possibile indipendente da ogni specifico hardware e/o da quelli che al momento sono considerati *standards affidabili* dato che sono tutti comunque destinati all'obsolescenza.

**Keywords:** bus di acquisizione, VME, cPCI,  $\mu$ TCA (microTCA), bus di campo, ethernet, embedded computers, virtual machines, machine learnig, machine protection, AI.

## Flusso e ritenzione dei dati

In un sistema costituito da tanti processi di controllo da un lato (p.es. l'area di un acceleratore di particelle) e da tante postazioni operatore dall'altro (p.es. una sala controllo) ci sono molti modi per organizzare il traffico dati e l'inoltro dei comandi.

In particolare è rilevante la differenza fra il metodo basato su *queries* (dalle console ai processi di controllo) e quello basato sull'utilizzo di una *cache* (area di memoria volatile ove i processi di controllo scrivono e le *consoles* leggono). Si è anche descritta la differenza fra i database relazionali e quelli non relazionali.

Oltre a questi aspetti architettureali, abbiamo anche parlato di quanto sia importante passare per un processo di virtualizzazione volto a uniformare famiglie di dispositivi diversi per marca e/o modello ma che assolvono alla stessa funzione. Al termine di questo processo, il sistema si troverà ad operare solo con rappresentazioni virtuali di oggetti (classi) che avranno set di caratteristiche (proprietà) e di azioni possibili (metodi) uniformi.

Parlando di come si possono rappresentare i dati, abbiamo visto come le tecniche di serializzazione (p.es JSON) consentono di rappresentare un qualsiasi set di dati tramite una sequenza di bytes (da cui *serializzazione*) che, oltre a contenere il valore dei dati, ne descrive anche il formato.

Infine abbiamo visto quale è la differenza fra i database relazionali ed quelli non-relazionali e di come questi ultimi siano particolarmente adatti ad accogliere dati serializzati.

**Keywords:** Relational Database, memcached<sup>3</sup>, object DB, key-value DB, data serialization, JSON, BSON, XML, YAML

## Interfaccia utente

L'interazione degli utenti (operatori di macchina, fisici di macchina, utenti esterni, etc.) con il Sistema di Controllo deve essere il più possibile intuitiva e aggiornata rispetto alle interfacce WEB e a quelle degli *smartphones* che utilizziamo quotidianamente. Questo è uno degli aspetti più impegnativi nella realizzazione di un sistema di controllo in quanto la produzione di un'interfaccia uomo-macchina ergonomica e moderna è estremamente dispendiosa in termini di tempo. D'altra parte questo campo è in rapida evoluzione (fuori dalla comunità scientifica) e cominciano ad essere disponibili ambienti di presentazione grafica di dati molto belli e funzionale (p. es. nei LNF si utilizza molto il software *open source* Grafana<sup>4</sup>).

**Keywords:** WEB interface, GUI.

## Attività di laboratorio

Per l'attività di laboratorio abbiamo lavorato su postazioni individuali costituite da Mini PC Intel® NUC con processore i5 quad core, SSD 500 GB, RAM 16 GB, monitor 27" con ris. 2K. Le postazioni di lavoro erano connesse in rete tramite WiFi con un bitrate  $\geq 100$  Mbps, sufficiente per le attività previste.

L'ambiente di programmazione utilizzato è stato LabVIEW<sup>5</sup> v 2023 Q1 *professional edition*. Abbiamo utilizzato un *template* di *main program* basato sull'impiego di eventi già pronto. L'esperienza ha consistito nell'accedere ad una immagine catturata dal monitor di luce di sincrotrone dell'anello di elettroni di DAFNE e nell'elaborarla. L'immagine era costantemente aggiornata ad una frequenza di  $\sim 2$ Hz nella cache reale del Sistema di Controllo DANTE.

(la descrizione dell'esperienza non viene riportata).

**Keywords:** falsi colori, memcached.

Alessandro Stecchi e Giampiero Di Pirro  
(I docenti della sessione "Controls")

---

<sup>3</sup> <https://memcached.org/>

<sup>4</sup> <https://grafana.com/>

<sup>5</sup> <https://www.ni.com/it-it/shop/labview.html>