

Web & Git
trillaura.github.io
linkedin.com/in/trillaura
github.com/trillaura

**Mail &
Hangouts**
laura.trivelloni@
gmail.com

Tel
+39 331 4940058

Laura Trivelloni

Esperienza

03/18 - Ora **Tutor per il corso di Fondamenti di Informatica**

Università di Roma Tor Vergata, Roma

- Esercitazioni su implementazione di algoritmi di base in linguaggio **Jython**, usando *JES*
- Mantenere gruppo online del corso per interazione diretta con gli studenti
- Assistenza al Professore durante la prova pratica di programmazione dell'esame

03/17 - 02/18 **Tutor per il corso di Ingegneria degli Algoritmi**

Università di Roma Tor Vergata, Roma

- Esercitazioni su implementazione di strutture dati (liste, pile, code, code con priorità, alberi, grafi, tabelle hash) e algoritmi di ordinamento, selezione, visite di grafi, minimum spanning tree e cammini minimi in linguaggio **Python**, usando *PyCharm*
- Mantenere pagina online della parte pratica del corso
- Assistenza al Professore per la prova pratica dell'esame, progettando la traccia dei progetti d'esame

03/17- 02/18 **Tutor per il corso di Fondamenti di Informatica**

Università di Roma Tor Vergata, Roma

- Assistenza su installazione e configurazione ambiente **Linux**
- Esercitazioni su comandi base di Linux e su implementazione e debugging di algoritmi di base in linguaggio **C**, usando **GDB** e **Vim**
- Assistenza al Professore durante la prova pratica di programmazione dell'esame

08/16 - 09/16 **Web Developer**

Biotiful Srl, Velletri (RM)

- Progettazione interfaccia Web destinata agli amministratori dell'applicazione e agli utenti autorizzati per:
 - Gestire i dati contenuti nel database (MySQL) dell'applicazione tramite CRUD per ogni tabella
 - Gestire i dati contenuti nel database (MySQL) dell'applicazione visualizzando il risultato di diverse query
 - Salvare, modificare o eliminare foto contenute nell'applicazione (Amazon S3)
- Implementazione in **PHP**, usando il framework **Laravel**
- Usata l'SDK in PHP per interfacciarsi ai servizi di storage Cloud di **Amazon S3**
- Usato RDBMS **MySQL** per l'interazione con la base di dati

Istruzione

2016 - Ora **Laurea Magistrale in Computer and Information Engineering**

Università di Roma Tor Vergata, Roma

Curriculum Networking and Multimedia.

Materie principali: Architetture di sistemi distribuiti, Sicurezza informatica, Cloud computing, Programmazione multi/many-core, Sistemi operativi open-source embedded e real-time, Algoritmi per il Web.

2012 - 2016 **Laurea Triennale in Ingegneria Informatica**

Università di Roma Tor Vergata, Roma

Curriculum *Sistemi software e Web*.

Main subjects: Matematica e Fisica, Programmazione, Mobile programming, Sistemi Operativi, Operational Research, Ingegneria del Software, Ingegneria degli algoritmi e strutture dati, Progettazione Database, Web Applications, Sistemi di telecomunicazioni, Sistemi di controllo, Elettronica Digitale e Analogica.

2008 - 2012 **Diploma Liceo Scientifico PNI**

Liceo Scientifico, Velletri (RM)

Scientific Secondary School.

Main subjects: Mathematics, Physics, Computer Science.

Certificazioni

2012 **First Certificate in English FCE**

Livello B1

2011 **Patente Europea per il Computer - ECDL**

Progetti

- 2018 **Linux Mailslot Module**
Nell'ambito del corso *Programmazione di sistemi multi-core/many-core*, implementazione di un **driver Linux** per il kernel 4.13 chiamato mailslot, come il device file di Windows a cui questo progetto è ispirato.
- <https://github.com/trillaura/LinuxMailslotsModule>
- 2018 **ilpostuniversitario.it**
Un sito Web creato con Wordpress per dare uno spazio a iniziative, eventi, petizioni e scambi di materiale tra studenti.
- <https://ilpostuniversitario.it>
- 2018 **Sistema distribuito di stream processing per social network**
Nell'ambito del corso *Sistemi e architetture per Big Data*, sviluppo di un sistema di **processamento real-time** per l'analisi di dati provenienti da un social network. Il sistema è implementato in **Scala**, usando i seguenti framework:
- **Apache Kafka** per raccogliere dati in ingresso e in uscita al/dal sistema
 - **Apache Storm, Apache Flink e Apache Kafka Streams** come sistemi in parallelo per il processamento di dati
 - **Redis** per il sistema di memorizzazione dei risultati di processamento
- Il sistema è stato sia virtualizzato eseguendo ogni componente su container **Docker** sia caricato su **Google Cloud Platform**.
- https://github.com/trillaura/social_network_rt_analytics
- 2018 **Sistema distribuito di batch processing per prese intelligenti**
Nell'ambito del corso *Sistemi e architetture per Big Data*, sviluppo di un sistema di **processamento batch** per l'analisi di dati provenienti da un rete di sensori posti su prese intelligenti all'interno di case reali. Il sistema è implementato in **Scala** seguendo un modello di programmazione **MapReduce**, usando i seguenti framework:
- **Apache Nifi** per raccogliere dati in ingresso al sistema
 - **Apache Spark** per il processamento di dati con la libreria core e ad alto livello (SparkSQL)
 - **Alluxio** cache distribuita per il salvataggio dei risultati
 - **Apache Hadoop HDFS** file system distribuito per storage dati in ingresso e salvataggio dei risultati
 - **MongoDB** per storage persistente dei risultati su database distribuito
- Il sistema è stato sia virtualizzato eseguendo ogni componente su container **Docker** sia caricato su **Google Cloud Platform**.
- https://github.com/trillaura/smart_plugs
- 2018 **Euristica ASO**
Nell'ambito del corso *Teoria dei giochi e progetto di reti*, sviluppo di un algoritmo per la schedulazione degli ordini (ASO) come euristica di un modello di programmazione lineare nell'ambito dell'assegnazione di consegne di ordini a un numero prefissato di fattorini. La schedulazione delle consegne delle ordinazioni deve avvenire in base all'orario scelto dal cliente, minimizzando il costo dovuto ai ritardi registrati e alle eventuali cancellazioni. Implementazione in **Go**, usando l'IDE *GoLand*.
- <https://github.com/van-ema/ASO-Heuristic>

2017

CINI Smart Cities Challenge

Sviluppo di un progetto per Smart Cities per partecipare alla challenge **CINI**. Sistema real-time distribuito di monitoraggio e controllo dell'illuminazione pubblica lungo le strade attraverso dati provenienti da una rete di sensori, implementato in **Java** seguendo un'architettura MAPE (Monitor Analyze Plan Execute) e usando i seguenti framework:

- **RabbitMQ** per raccogliere dati in ingresso al sistema
- **Apache Storm** per il processamento di dati
- **Memcached** per il sistema di caching di processamento
- **Amazon SQS** per i dati di output del sistema

Il sistema è stato sia virtualizzato eseguendo ogni componente su container **Docker** sia caricato su istanze di **Amazon EC2**.

https://github.com/OviDanielB/CINI_SmartLightingSystem

2017

Next-Event Simulation

Nell'ambito del corso *Performance Modeling od Computer Systems and Networks*, sviluppo di un sistema per trovare, attraverso la next-event simulation, parametri ottimali per minimizzare il tempo di risposta e massimizzare il throughput di un sistema cloud e cloudlet variando la configurazione di capacità del sistema.

Implementato in **C** con una libreria diversa da quella di default per la generazione di numeri casuali. Usato **Python** (matplotlib and numpy libraries) per validazione dei risultati e visualizzazione dei risultati.

<https://github.com/OviDanielB/PMCSN>

2017

SOA System Development

Nell'ambito del corso *Ingegneria dei Sistemi Software e dei Servizi di Rete*, integrazione di sistemi implementati per la Fase 3 e 4 con le altre fasi del ciclo GQM+Strategies usando un Enterprise Service Bus condiviso. Il sistema espone una RESTful interface usata per il frontend e usa *Activiti* come business process manager. È stato necessario per tutti i gruppi **collaborare** per costruire un sistema completamente funzionante.

Le tecnologie principalmente utilizzate sono **Java Spring Framework** per la REST interface, **MongoDB** per storage e **AngularJS** per le chiamate REST.

2016

Web Server

Nell'ambito del corso *Ingegneria di Internet e Web*, sviluppo di un Web Server implementato in **C** usando le *Berkley Socket API*.

Il server principalmente restituisce immagini dinamicamente adattate al tipo di device che effettua la richiesta.

Le informazioni sul device sono ottenute dalla *WURFL API* e poi salvate in cache secondo le varie misure usando un database SQLite.

<https://github.com/OviDanielB/WebServer>

Competenze personali

Organizzazione e collaborazione

Competenze comunicative acquisite durante il ricoprimento del ruolo di **rap-presentante degli studenti** presso il Dipartimento di Ingegneria Informatica e presso il Consiglio degli Studenti dell'Università di Roma Tor Vergata.

Competenze organizzative e gestionali acquisite nella partecipazione al **collettivo universitario** nell'organizzazione di conferenze, incontri, raccolte di fondi/beni.

Disegno e illustrazione

Su foglio e in digitale.

- grafica vettoriale con *Inkscape* per creazione loghi da autodidatta
- premio speciale vinto al concorso di satira *Velletri Ridens 2015*
- elaborazione digitale con l'uso della tavoletta grafica, usando a livello base software *Corel Painter*, *ArtRage Studio Pro 4* e *Adobe Photoshop Elements 10*

Altre competenze informatiche

L^AT_EX

Per preparazione di slide, articoli,

Git

Sistema di controllo di versione di codice, da linea di comando e tramite IDE.

Altre informazioni

Si autorizza il trattamento delle informazioni contenute nel curriculum in conformità alle disposizioni previste dal d.lgs. 196/2003. Si dichiara altresì di essere consapevole che, in caso di dichiarazioni non veritiere, si è passibili di sanzioni penali ai sensi del DPR 445/00 oltre alla revoca dei benefici eventualmente percepiti.

Agosto 2018

Laura Trivelloni