

Danilo Carrabino

email: danilo.carrabino@gmail.com | LinkedIn: www.linkedin.com/in/danilocarrabino

Senior Software Engineer

Object-Oriented Analysis, Design and Development / Relational Database Systems / Web, Mobile, Stand-alone, Embedded systems Development

- Expert in advanced development methodologies, tools and processes contributing to the design and rollout of cutting-edge software applications.
- Troubleshooting skills – able to analyse code and engineer well-researched, cost-effective and responsive solutions, used to work both alone and in team with high-quality results.
- Lust for knowledge and for new technologies (**Node.js, Express, Typescript, React Js, MongoDB, QML, NPM, Javascript ES6, Asynchronous Programming, Promises, Lambda closures, Docker**)
- Top ranking in C++ practice on HackerRank: www.hackerrank.com/danilo_carrabino
- Always looking for the most appropriate algorithmic technique to accomplish the task

Objective

A challenging position as Software Engineer in a solid, friendly and demanding environment.

Skills and Expertise Summary

Expertise	Skills	Experience
Languages and Technologies	C, C++, Qt Framework, Javascript (ES6), Typescript, Python, Java, C#, Bash Shell Scripting, VBA	10+ years
Web Backend Technologies	PHP, Node.js	8-10 years
Operating Systems	Linux, Windows XP/7/8/10, HP-UX	8-10 years
Web Frontend Technologies	JavaScript, JQuery, React.js, Vue.js, HTML5, CSS, Bootstrap, EJS	6-8 years
Relational Database Technologies	MySQL, SQLite, PostgreSQL	8-10 years
Service-Oriented Architectures	SOA, Web Service, Micro Service, REST, JSON, JSONP, SOAP, XML	5-7 years
Network/Bus Protocols	TCP/IP, HTTP, FTP, XMPP, Web Socket, HDLC (Data Link layer), MIL-STD-1553	6-8 years
Version Control Software	Git, SVN, CVS	8-10 years

Danilo Carrabino

email: danilo.carrabino@gmail.com | LinkedIn: www.linkedin.com/in/danilocarrabino

Cloud architectures	MS Azure, Digital Ocean, Docker	1-2 years
Architecture Design	UML 2, Enterprise Architect	4-6 years
No-sql database	MongoDB	1-2 years
Mobile Technologies	Android, Web Apps	2-4 years
Web frameworks	Express, Mongoose, Hapi.js, Twisted Python, Doctrine, WordPress	1-2 years

Professional experience

05/2019 – present **Analyst Programmer** - Permanent contract @ MONK Software (Rome, Italy)

Full-stack Developer (Node.js, Javascript ES6, Typescript, Express, Redis Cluster, MongoDB, PostgreSQL, API, Micro services, Docker, Swagger, Vue.js, JSON-Schema)

Project 1: **BotManagement**

I designed and developed a micro service software architecture in order to create, publish and use custom chat-bots for multi-channels: the modules communicate through Redis Cluster queues, by using a library I personally wrote, wrapping [Node.js Bull](#) and other queues managers.

I wrote several modules for this software architecture:

- The *http API server* (Node.js, Javascript, Express, [Ajv – JSON Schema Validator](#)) to handle the bot configurations, and all the other information needed for the chat-bot system functionalities. It uses a PostgreSQL XL database and a Redis Cluster to save the users tokens.
A JWT (Json Web Token) is passed to every endpoints to guarantee the user authentication.
- The *Director module* (Node.js, Typescript) which uses a state-machine ([X-State](#)) in order to manage the transitions among bots' nodes, based on the clients input messages.
- A *wrapper library* (Node.js, Typescript) for Bull queue manager and other queues managers, used by all the other components to abstract the underlying transport layer.
- Part of the *Bot-designer portal* (Node.js, Vue.js) in order to offer a user experience to create, publish and manage the chat-bots.

Danilo Carrabino

email: danilo.carrabino@gmail.com | LinkedIn: www.linkedin.com/in/danilocarrabino

10/2018 – 05/2019 **Full-stack Developer** - Permanent contract @ DigitalGO (Rome, Italy)

Full-stack Developer (Node.js, Javascript, MongoDB, MS Azure, API, Microservices, Docker, Swagger)

Project 1: **Invictus Dashboard**

- I designed and developed a software architecture in order to provide web services and a dashboard to build and manage campaigns for the *Invictus* project. The system is composed of three main parts:
 - A database in *MongoDB* containing the campaign objects;
 - A REST API server in *Node.js, Express, Mongoose, Swagger*, which exposes API endpoints to manage the campaigns: a JWT (Json Web Token) is passed to every endpoints to guarantee the user authentication (only the service invoked by the *Invictus* landing pages is exposed without an authentication token);
 - A dashboard server in *Node.js, Express, Mongoose, Bootstrap 4, JQuery, Browserify*, is the access point for any user willing to create/update/delete/view campaigns. The dashboard is accessible only to users present in the *Azure Active Directory* of *DigitalGO*. Because the dashboard can read/write users information in the *MongoDB* database, it's also able to generate Json Web Tokens, necessary to invoke the endpoints exposed by the REST API server.
- The campaign data are shown by the *Invictus* landing pages, after invoking a specific API endpoint exposed by the REST API server, with the campaign identifier passed in the query parameters.

Project 2: **Guam Dashboard**

- I designed and developed a software architecture in order to provide web services and a dashboard to build and manage campaigns for the *Guam* project. The system is composed of three main parts:
 - A database in *MongoDB* containing the campaign objects;
 - A REST API server in *Node.js, Express, Mongoose, Swagger*, which exposes API endpoints to manage the campaigns: a JWT (Json Web Token) is passed to every endpoints to guarantee the user authentication (only the service invoked by the *Guam* landing pages is exposed without an authentication token);
 - A dashboard server in *Node.js, Express, Mongoose, Bootstrap 4, JQuery, Browserify*, is

Danilo Carrabino

email: danilo.carrabino@gmail.com | LinkedIn: www.linkedin.com/in/danilocarrabino

the access point for any user willing to create/update/delete/view campaigns. The dashboard is accessible only to users present in the *Azure Active Directory* of *DigitalGO*. Because the dashboard can read/write users information in the *MongoDB* database, it's also able to generate *Json Web Tokens*, necessary to invoke the endpoints exposed by the REST API server.

- The campaign data are shown by the *Guam* landing pages, after invoking a specific API endpoint exposed by the REST API server, with the campaign identifier passed in the query parameters.

Project 3: **Kidsapp**

- I am in charge of managing, improving and fixing bugs of the *Kidsapp* backend. The API server was developed with *Node.js*, *Hapi.js*, *Mongoose*, *Swagger*, and uses also third party services. There's also a dashboard to create/update/delete contents (it was developed in *Vue.js*) and a content proxy, which is responsible to stream videos, managing and caching images and other contents (it uses *Redis* as caching mechanism).

12/2015 – 10/2018 **Analyst and Software Developer Consultant** - Permanent contract @ SEGULA Technologies (Rome, Italy)

Analyst and Software Developer (C/C++ Qt framework and QML)

Project 1: **Realtime Embedded System Software** @ Elettronica S.p.A. (<http://www.elt-roma.com>).

- I designed and developed several requested enhancements and new features on a real-time embedded system software (C/C++), executed on a microcontroller of a physical Unit, which is part of the EW Self-Protection Suite. The software is designed and built to communicate with other hardware components through RS232 serial ports, an SDLC bus, and custom registers.

Project 2: **CSCI Loader** @ Elettronica S.p.A.

- I built a tool to enable the loading and verification of a CSCI software inside a physical Unit. It is a GUI application built in *C++* with the *Qt framework*, which features *custom widgets*, *Qt State Machines* to handle the operations, *message passing* and *multithreading*

Danilo Carrabino

email: danilo.carrabino@gmail.com | LinkedIn: www.linkedin.com/in/danilocarrabino

functionalities. It has been designed in accordance with various design patterns (*singleton, state machine, observer, façade*). It interfaces with an *RS232 serial port* to communicate with the physical Unit, and it is able to send and receive signals to and from a Digital I/O device of the Nation Instruments, in order to power on/off the Unit itself.

- The requirements and design part was realized in *UML 2* by using *Enterprise Architect* for:
 - i. Use Cases.
 - ii. Requirements definition.
 - iii. Activity Diagrams.
 - iv. Sequence Diagrams.
 - v. Class Diagrams.
 - vi. Package Diagrams.
- ***I was acknowledged in a letter of thanks from the British Ministry of Defence, for the outcome of a job which required the use of such software.***

Project 3: **Technique Manipulator** @ Elettronica S.p.A.

- I built from scratch a tool to perform function tests on a physical Unit by creating and manipulating the mission data: it is a GUI application built with the *Qt framework* in C++/QML for Windows XP/7/8/10 platforms. It has been designed in accordance with various design patterns (*factory, memento, MVC, observer, façade, dependency injection*). The app features a dynamic *QML ListView* interacting with a *custom AbstractListModel* through *Model-View-Controller pattern*, to dynamically generate and manipulate the mission data to pass to the Unit. Moreover the “technique generation” architecture was designed to be extremely reliable by implementing the *undo/redo* functionalities through the *Memento pattern*. The communication with the unit is performed by implementing a *multithreaded architecture* and the *signal and slots mechanism* of Qt.

Project 4: **ELT/160 Applications and tools** @ Elettronica S.p.A.

- The activity is about realizing support applications, tools and simulators porting for the ELT/160 RWR System (The ELT/160 is a family of light radar warning receivers). All the support applications, tools and simulators are realized in C++ with the Qt Framework and the bus Protocols used are: MIL-STD-1553, RS232, RS400.

Danilo Carrabino

email: danilo.carrabino@gmail.com | LinkedIn: www.linkedin.com/in/danilocarrabino

05/2013 – 12/2015 **Software Engineer** - Permanent contract @ Livebox S.r.l. (Rome, Italy)

Web and Standalone Software Engineer / Database Design and Administration

Project 1: **Design and development of the *private cloud system LiveBox Cloud***

- I developed the *standalone clients for the Windows and Mac Os platforms*, with the *Qt framework (with C/C++)*. The main purpose of these clients was to preserve the *synchronization of the user's data* between the server and any computer guesting the clients themselves.

I used several *design patterns* to plan the software functionalities: *singleton, observer* (implemented with the *signals and slots in Qt*), *composite*, which coupled to several *graph traversal algorithms*, helped me to handle the folders/files structure of the users.

The application orchestrates:

- i. the *file systems* (to handle folder and files' modifications and synchronization);
 - ii. the *SQLite* database (representing the state of the last synchronization);
 - iii. the communication with the *Web services* in both *polling and pushing modes* (in polling mode the requests are performed any x seconds from the client to the Web services, in push mode it's the server that communicates with the client through the *XMPP messaging protocol*).
- The *Livebox Agent Installer* for Windows OS platforms was built in C# with Visual Studio Installer, with the help of the so-called '*custom actions*' to register the folder/file *overlays* and show them through the explorer file manager.
 - I also developed the *backend core functionalities (PHP, and Doctrine* only in the first version of the software) and the database (*MySQL*) for the cloud system.
The server side development was oriented to the deployment of the Web services (*Rest/JSON*) to let the clients interact with the backend. The core logic functionalities interact with the MySQL database tables.
I built the system in order to communicate with *LDAP* and *Active Directory*, to guarantee the accessibility of already existing users to the cloud system, and with external SOAP Web Services to enrich the functionalities exposed.
 - Moreover, I developed some sections of the web portal with *JavaScript, JQuery, HTML5, Ajax, CSS* technologies.

Project 2: **GeoPy and GeoPHP.**

- The tool that I built in Python automatically downloads the most updated raw data from the Geonames portal, makes a selection of the data that will populate the database tables and

Danilo Carrabino

email: danilo.carrabino@gmail.com | LinkedIn: www.linkedin.com/in/danilocarrabino

creates a 'cache table' to speed up the queries (there are more than 10 million entries).

- I developed the backend software in PHP and MySQL in order to exhibit the Geonames functionalities to the rest of the software system. The PHP code takes advantage of the cache structures built by the Python software, and it is built with the use of different patterns, among which there's the *Abstract factory* pattern to hide the details of the system's main entities.

02/2013 – 05/2013 **Web Developer** - Permanent contract @ Immobiliare.it (Rome, Italy)

Development of several functionalities for the Immobiliare.it backoffice, in PHP, MySQL, HTML, Smarty, JavaScript, JQuery and Ajax.

12/2011 – 02/2013 **Mobile and Web Developer** -Annual contract @ AreaMobile & JDK (Rome, Italy)

Mobile and Web Developer, Android teacher

- I developed several apps for smartphone and tablet *Android*: one of them was the former *Euronics* app on the *Google Play Store*.
I've been involved in the design and the development of:
 - The web layer for the communication with the Web services (through 'Service' and 'Intent Service' components);
 - The presentation layer ('Activity' and 'Fragments' components);
 - The database layer (through 'Content provider' and direct queries to SQLite database) and more.

The Android framework offers several components implementing many useful design patterns, like the 'iterator' pattern (for the container objects) and the 'adapter' pattern to create a bridge between the data and the view for the user interaction.

- Backend (PHP) and Web portal (HTML5, JavaScript, JQuery, CSS, Bootstrap) development for the apps SFA and AMRevolution.
- I lectured in several courses of 'Introduction to Android development'.

10/2010 – 12/2011 **Software Developer** - Annual contract @ Proconsul Group (Rome, Italy)

Standalone Software Developer consultant for Rome HP Office

Danilo Carrabino

email: danilo.carrabino@gmail.com | LinkedIn: www.linkedin.com/in/danilocarrabino

Project 1: Custom Fraud Management System

- I built part of the custom framework and several application functionalities. The technologies used were C, C++ and Oracle Pro C on HP-UX, and the core functionalities of the custom framework were built to help the inter-process and inter-thread communications (posix threads). Moreover, the framework orchestrated the interactions with the database layer (Oracle DB) and with the input and output streams of the system.

05/2006 – 10/2010 **Web and Standalone developer** - Internship and Annual contracts @ CASPUR and “Sapienza” University of Rome

Web and stand-alone developer for several Bioinformatics applications.

- I started to work (and learn) at CASPUR (a Super Computing Facility for Universities and Research structures) now incorporated in a larger institution called CINECA (<http://www.cineca.it/>) during my master’s degree internship in 2005.
- I could interact and develop web and standalone applications with *computing clusters* (Linux SUSE sles with network file system), *stack of software* (XAMPP – PHP, MySQL, PhpMyAdmin...) and various kinds of technologies not very used at the university.
- I developed standalone tools and scripts in C/C++ (check my education section for the surface protein epitope scanning tool), java, *Bash Shell Scripting language*, to interact with file systems, databases, web communications and *distributed computing systems*. Along with the development of web portals and *service-oriented architectures* for the *Bioinformatics* (that interacted with my standalone scripts and tools) I had the pleasure to participate in several scientific publications (check my education section).
- I applied and expanded my knowledge of *algorithms and data structures* (*Greedy, Back Tracking, Divide et Impera, Dynamic Programming, distributed programming*), programming techniques and optimization techniques to improve the performances, reliability and usability of my software.

Danilo Carrabino

email: danilo.carrabino@gmail.com | LinkedIn: www.linkedin.com/in/danilocarrabino

Other relevant skills and competences

I developed several Open Source web and mobile applications which can be found on my [Github profile](#).

- [MyPortfolio](#) is a Web Application written in Node.js / Express, which allows users to create a modern resume (infos and projects) with great visual impact in few easy steps. The portfolios are sharable with the most important web socials.
Data are stored in MongoDB with Mongoose (an ORM used to interact with MongoDB).
The frontend is developed in *EJS*, *JQuery*, *Bootstrap*, *HTML5* and *CSS*.
Because of the asynchronous nature of Node Js, different backend sections followed an asynchronous approach by using *Promises* and *Async Parallel* strategy.
It is hosted by *Digital Ocean* and is available at this link address:
myportfolio.danilocarrabino.net.
- [Recurrency](#) is a REST API server written in Node.js / Express, which exposes services related to currency rates.
Responses are json encoded in order to be easily usable.
Data are stored in MongoDB with Mongoose (an ORM used to interact with MongoDB).
Because of the asynchronous nature of Node.js, different backend sections followed an asynchronous approach by using *Promises* and *Async Parallel* strategy.
It is hosted by *Digital Ocean* and is available at this link address:
recurrency.danilocarrabino.net.
- [PSighter](#) is a Node.js library, written in Typescript, with the purpose of downloading courses from Pluralsight portal: it uses the power of [Puppeteer](#) in order to scrape the Pluralsight portal pages.
- [React to Currency Exchange](#) is a Web Application built to convert currencies by using the latest rates powered by [Recurrency API](#).
It features the framework *React Js* with *React Bootstrap* module to allow a multi-screen compatibility. Moreover the module *Axios-JSONP* asynchronously invokes the [Recurrency API](#) endpoints, to download the latest currencies rates.
Thanks to create-react-app application that I used to bootstrap my web application, I easily developed the *Unit Tests* section and the *build/deployment* procedure (The application is simply deployed with: *npm run deploy*).
It is hosted on *Github pages* at the address evildead.github.io/reacttocurrex.
I developed and published also the *Android* application: [React-to-Curr-Exchange](#).
- [DSTD](#) is a custom *C++* library for *BigInt*, string manipulation and more.

Danilo Carrabino

email: danilo.carrabino@gmail.com | LinkedIn: www.linkedin.com/in/danilocarrabino

- [CMemento11](#) is a project showing how to use *Memento* and *Command* design patterns in *C++ 11*, to implement the undo/redo functionality on a *QML* listView.
- [TwistedDisposableChat](#) is a simple chat application (client/server) developed with the *Twisted Python* framework (server side) and *REDIS*. Any customer accessing the system, can chat in private or send a broadcast message to all the connected users. I used the *Web Socket* protocol to handle the client-server communication, and *REDIS* is used both to keep track of any user in the system and to notify messages among the users themselves (*REDIS* implements the publish-subscribe pattern to be a message broker system). Through this architecture I could make my system distributable on different nodes (you can interface it with a load balancer).
- [create-a-crud-app-with-node-and-mongodb](#) is a CRUD application built with *Node Js/Express/MongoDB* in a [Scotch.io](#) course.
- [react-weather-app](#) is a web application built with *React Js* in a [Scotch.io](#) course.

...And even more.

With *C* and *C++* I designed and developed many applications outside the working environment. During my operating system II laboratory I developed a Print Spooling System with inter-process and inter-thread communications on the CUPS protocol.

With *Visual C++ 6* I built visual applications for different private customers.

By using the *WordPress* framework I developed and deployed a site for a delegation of FISAR (Italian Federation of Sommelier).

With the *Visual Basic for Application* technology, I built or customized several tools for *MS Access* or *MS Excel* commissioned by private customers.

Playing electric and classical guitar is my greatest passion.

Danilo Carrabino

email: danilo.carrabino@gmail.com | LinkedIn: www.linkedin.com/in/danilocarrabino

Education and professional courses

Master's Degree in Computer Science, 2004 - 2007 (110/110) - "Sapienza" University of Rome

My Master's Degree thesis was about building up a *bioinformatics software system* to find the epitopes on the proteins surface. I designed and developed a tool in C++ which, starting from a 3D model of a protein, used a *graph traversal algorithm* to find the amino acid sequences (even with gaps) representing epitopes. The constraints to build a suitable graph of the surface amino acids were given by some bioinformatics researches.

Then I built a web portal in PHP and JavaScript, which, by interacting with my standalone tool, it allows any researcher to submit his 3D model of protein and search for epitopes on the submitted protein surface.

This thesis had 2 *scientific publications*:

1. [The mepsMAP Server. Mapping Epitopes on Protein Surface: Mining Annotated Proteins](#) - IEEE Transactions on Nanobioscience - June 2007
2. [The MEPS server for identifying protein conformational epitopes](#) - BMC Bioinformatics - March 2007

Bachelor's Degree in Computer Science, 2000-2003 - "Sapienza" University of Rome

The training for the bachelor thesis was done at the Department of Physics of "La Sapienza" and the project was commissioned by the INFN of Rome. It 'was developed in *C and C++* on the *ROOT* framework (<https://root.cern.ch>) for Linux and led to the *publication of a public notice* with title: '["Light" Event Display for CDF \(Collider Detector at Fermilab\)](#)'.

The software, built for the physical analysis of the events of protons and antiprotons collisions measured by the collider, produces two different representations of the events themselves. This view has been implemented thanks to the API library of 3D graphics made available to the framework itself.

HTML5 Course "HTML5 Mobile Mastery", 09/2012

A course in HTML5 taught by Richard Clark (Senior Curriculum Developer at Kaazing - California). During the course, Richard showed loads of bleeding edge technologies for the Web development with *HTML5, JavaScript and CSS3*. A substantial part concerned the illustration of *web sockets* and the interaction with the mobile devices.

High Performance Computing Course, 04/2008

An introduction course to HPC taught by Dr. Giorgio Amati at CASPUR.

In this course, Dr. Amati showed the most important performance optimization techniques for

Danilo Carrabino

email: danilo.carrabino@gmail.com | LinkedIn: www.linkedin.com/in/danilocarrabino

computing applications written in *C* or *FORTRAN*.