Curriculum Vitae | Nacer DERGAL

Personal

Contact & Social

E-Mail dergal.nacer@outlook.fr | Github ndergal | Linkedin Profile

Whoami	Location	Phone
Nacer DERGAL - 08/26/1993	Quebec,QC	+33(0)7 81 25 18 44

Professional Experience

Cloud / Microservices Developer

From: 02/2020 until Present

EXFO - Onsite at Quebec, Canada

Description

- Development of cloud native services using AWS, Google Cloud, Node.js, Java and Python
- Automating infrastructure deployments using Azure Pipelines and Gitlab
- Managing infrastructure as a code (IaC) using Terraform/Serverless/Pulumi
- User authentication using Auth0, IAM or Google Cloud Identity Platform
- Storing data using MongoDB Atlas
- Working in an agile environment using BDD
- Validating services contract using Contract Testing
- Document services/API using mkdocs or FastAPI

Software Developer

From: 09/2018 until 02/2020

EXFO - Onsite at Quebec, Canada

Description

- Work in a SCRUM environment
- Software development in C# on a Linux based embedded system using the Mono Framework
- Software development in .NET Compact Framework.
- Conception of a Test API using Graph Theory to be easily able to go through each path/scenario of an application with automated tests.
- Conception of an Image Recognition framework combined with the Test API to test our GUI.
- Conception of a driver for an Optical Channel Monitor.
- Design and conception of an Android-like Notification service in C#.

Software/Cloud Engineer Co-worker

From: 09/2015 until 09/2018

IGN - Onsite at Champs-Sur-Marne, France 🇫🇷

Description

- Implementation and administration of a virtualized software forge on Docker
- Automation of project creation
- Implementation of web application
- Website creation
- Deployment of a secured Docker Registry
- Creation of a Restful API from a database describing a road network with its traffic topology

🎓 Education

Master of Science in Computer Science and Network (Ing., M. Sc.)

- 🎓 Diplomed in 09/2018
- Started in 09/2015
- ESIPE

Bachelor of Science in Computer Science

- 🎓 Diplomed in 12/2017

- Started in 08/2017
- University of Sherbrooke

Bachelor of Science in Electrical and Industrial Computing

- Diplomed in 09/2015
- Started in 09/2011
- UPEC

Skills

Programming skills

Object Oriented Programming, IOT, Embedded System, Cloud Computing, Operating System, GUI, Concurrent programming, Distributed Computing, Network, Data Mining, Functional Programming, Management, TDD, Computer Vision, REST/SOAP/gRPC/GraphQL/Layr

Programming Languages

Java 15/Android/Spring/Swing, C/++/#, Assembler, Bash, JS/TS/Angular, Python, Ruby (Rails), Go, Racket

Embedded/IOT

Arduino, Intelligent Agent programming (Swarming), ZigBee, LR WPAN, Z-Wave, EnOcean, 6LowPan, MQTT, Wi-Fi, Bluetooth, RFID, Yocto

CI/CD

Jenkins, Azure, Gitlab, TFS, Maven, Ant, Gradle, WhiteSource, SonarQube, JUnit, Mockito, Cucumber, Mocha, Sinon

Database

Postgresql,Atlas/MongoDB, MariaDB, Cassandra, Redis, SQLite, MySQL

Network

Protocols, LANs/WANs/WLANs/WPAN, Wireless, Cellular, Cyber security, Routing, Cryptography/Certificate

Virtualization/Cloud/IaC

Docker compose/swarm, Virtual machine (Virtualbox/VMware), Kubernetes, AWS, Google Cloud Platform, Terraform, Pulumi, Serverless

Web server

Nginx, Apache, Tomcat, Vertx, Unicorn, Fastapi

Load balancing / Proxy

Haproxy, Squid

Projects

Ordogene

- </> Source

Description

Ordogene is a software that can be used to determine, using a genetic algorithm, the actions of a given process to maximize or minimize some resources in a minimum amount of time. These processes can be very different, such as getting ready in the morning, organizing an event or manufacturing industrial parts.

PapayaDB

- </> Source

Description

A self-optimized document-oriented database storing a set of value in JSON format. The project was divided in several parts implanting the database itself (with atomic document insertion), a Web server (REST) to make queries (the queries are themselves JSON documents) and a client API in Java to make requests to the REST server (a request result had to be a Java Stream). Each module must be separate and independent. It had to be delivered as modular JARs (compatible with Java9 modules).