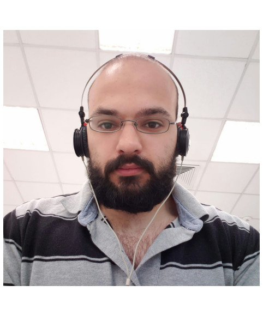

José Ricardo Carvalho Prado de Almeida

Software Architect, Engineer and Thinkerer

Rua João Rodolfo Schlenker 294 ap8, Curitiba, Água Verde, 80620-030, Brazil
January 16, 1992 **zeca_16@hotmail.com** **+55 (41) 988-234-826**

https://zeh-almeida.olamundo.org/
GitHub Personal Profile
LinkedIn Professional Profile

Hello there, my name is José Ricardo.
I am a Senior Developer with a career spanning across 13 years as of 2023 working on multiple different systems with a large number of languages and technologies, including .Net, java, Python on industries such as Banking, Oil and Gas, Automotive, Healthcare and general consulting, including others.
Since 2017 I have worked mainly as a Senior Developer. In this role I have analyzed requirements, drawn system architectures, written down tasks for other developers, mentored junior developers, defined processes, implemented the test pyramid paradigm and, of course, coded a lot of things as well.
Being in the fray for over a decade, I had the opportunity to work in many different companies and industries, connecting with people all over the world, making contact with many different cultures and life styles.
Currently, I have the personal objective to help other succeed in their development career as both a mentor and colleague. I strongly believe the best software is not written by a single person but by a very cohesive and tight team.

LANGUAGES

English	**Portuguese**
Proficient	Native
●●●●●	●●●●●

EDUCATION

Pontifícia Universidade Católica do Paraná (January 01, 2010 - March 27, 2014)
Bachelor, Information Technology
Information Systems

Pontifícia Universidade Católica do Paraná (January 01, 2016 - May 31, 2018)
Post Graduate, Software Engineering
Software Enginner

CERTIFICATIONS

Certificate of Proficiency in English (August 02, 2011)
University of Cambridge
Certificate Number 002994220

WORK EXPERIENCE

CI&T

(December 06, 2021 - July 30, 2023)
Senior Software Engineer

Projects:

- SAP integration with Node.js APIs using Azure for a major beverage company.
  - Responsible for analyzing requirements and breaking them down into technical documents which the team would work on.
  - Once the team started work, it was my responsibility to ensure the developers had the necessary assistance, tools and knowledge to complete their assignments using pair-programming and other techniques.
  - As soon as the code was written, I was one of the key people necessary to approve their work, revising their code and making suggestions.
  - Because of the analysis phase, I was also responsible for communicating with the client and other stakeholders in order to gather as much information on the requirements possible.
  - All this work was done with international teams from multiple geographical locations.
- Developing a solution to collect data from Brazilian Health Department (ANVISA) and make it available in a data lake using AWS for a major pharmaceutical company.
  - Responsible for designing the architecture as well as implementing it. I was responsible for communicating directly with the client and stakeholders in order to gather the requirements as well as having the architecture defined and approved.
  - Once approved, the team and I were responsible for implementing it, using Python, AWS Step Functions and AWS Lambda running pipelines on Jenkins with Terraform.
  - Techniques such as pair-programming proved to be very efficient and increased the team's productivity. Unit tests were also very important because of the data volume and data source availability.
- Architecture design of micro-services for a multi-corporation solution for a major technology conglomerate.
  - Responsible for gathering requirements, designing and validating solutions with clients and stakeholders.
  - One of my most curious assignments because most of what was asked to be designed already existed in the solution in a different form.
  - Therefore, most of my work was not in creating new things but showing the customer how to use the solution in a more effective way, increasing it's value.
- Development of solutions to gather data from a major financial institution in Brazil.
  - Implementation of programs to gather data from multiple sources, filter and process them in order to integrate with an external system.
  - Because of the age of some sources and data complexity, many meetings with different teams had to be arranged in order to validate sources, formats and other data characteristics.

Main technologies:

- .Net Core
- Node.js
- Python
- AWS
- Azure
- Micro-services

Main activities:

- Development
- Analysis
- Support for junior developers

Ahoy by Belago Group

(April 01, 2021 - December 01, 2021)
Full Stack Senior Developer

Project:

- Implementation of a complex solution for managing a Family Office.
  - Responsible for developing and supporting junior developers in the implementation of a Family Office system that would handle data input as well as many financial calculations.

Main technologies:

- .Net Core
- SQL Server
- Azure
- Micro-services

Main activities:

- Development
- Support

TIVIT

(October 01, 2020 - March 01, 2021)
Full Stack Senior Developer

Project:

- Architecture and implementation of a financial analysis workflow for a major industrial bank.
  - Responsible for designing and implementing a solution which would receive data from external systems in order to validate if a process would be approved or not.
  - This included data enrichment from internal company sources as well as external systems.
  - The implementation ran in a event-driven environment fully orchestrated in Azure using Azure Functions and ServiceBus.
  - Execution could be restarted at any point during the analysis flow in order to avoid unavailability of any external system.
  - Monitoring the execution flow in Production to evaluate possible blocks in the process.

Main technologies:

- .Net Core
- SQL Server
- Azure
- Azure Functions
- Azure Service Bus
- Micro-services

Main activities:

- Development
- Analysis
- Support

ExxonMobil Global Business Center

(November 01, 2017 - October 01, 2020)
Systems Analyst

Projects:

- Development and Maintenance of the Sales Order API.
  - One of the senior developers in a Agile team using Scrum. Responsible for Developing, testing and following the project pipeline in order to finish tasks.
  - All tasks were performed with pair-programming, rotating team members every sprint in order to get in sync with the team and improve out abilities.
  - Used Java and micro-service architecture in order to implement a RESTful API as well a event-driven API using Kafka.
- Design and Implementation of a company-wide calendar event scheduler.
  - Using then-new .Net Core 3.1 and then-recently released Azure Functions v2, I was tasked to design and implement a calendar event system akin to MS Outlook with a cloud-first operation.
  - The system improved Outlook's offering as it has a waiting queue when the participation quorum was full as well as e-mail notifications when event changed, queues moved, etc.
  - The system had to work on IE11 and Chrome using the company's Azure AD instance for authorization and authentication. The application was written with React on the front-end.

Main technologies:

- Java
- Jenkins
- Springboot
- OpenShift
- Azure DevOps
- .Net Core 3.1
- Azure Functions v2
- React
- Azure AD
- Service Bus
- Agile and Scrum

Main activities:

- Development
- Testing
- Design
- Team-building

Regazzo Soluções em Tecnologia

(October 01, 2016 - July 01, 2017)
Software Engineer

Project:

- Maintenance and evolution of a ASPX-based payroll deductible loan system for a regional bank in the Paraná state.
  - System written in a monolithic approach, with many controls in ASPX and many others in Javascript/jQuery.
  - Main job was to add the capability to scan paychecks, apply OCR to the documents and read as many data as possible from the document, as this would allow a much quicker filling of the loan form as well as some value, availability and rate validations.
  - System was previously written in C#6 and MVC6, I was tasked to move it to C#7 and MVC Core.
  - OCR system was available via an external API, written mostly in XML with embedded JSON.
    - Custom parser had to be developed for this weird combination of dialects.
    - Whole OCR service communication written in a event-driven approach inside the ASPX application, using services and workers.
  - System had to communicate with legacy on-premises banking system to fetch customer information as well as calculation parameters using SOAP.

Main technologies:

- .Net Core 3.1
- ASP.Net MVC Core
- Azure DevOps
- JSON
- XML
- SOAP
- OCR

Main activities:

- Development
- Testing
- Design

FH Consultoria

(August 01, 2016 - October 01, 2016)
Hybris Programmer

Projects:

- Maintenance of the Hybris store and implementation of analytics for a major clothing industry player.
  - Several improvements to the website structure, layout, styling and overall performance of the pages, as the layout was not using a structured HTML/CSS layout such as Bootstrap. Instead, it was pretty much free form.
  - Implementation of the analytic softwares such as Chaordic and wiring the configurations at the back-end.
- Rebuild of the pages for a major tire sales company.
  - The company had many different websites, all using the same layout base.
  - However, the base layout did not account for different form factors, responsiveness or any other quality-of-life.
  - All the website structure was migrated to Bootstrap CSS in order to leverage all the quality-of-life the library provides, as well as the pre-existing components.
- Development of a Git-based versioning system used in all Hybris-based projects.
  - Projects were still using the then-outdated SVN system for source control.
  - Used the basic guidelines to establish the culture of source control, commit messages and Pull Requests.

Main technologies:

- Java
- Hybris
- Javascript
- jQuery
- CSS
- Bootstrap
- Git

Main activities:

- Development
- Testing
- Design

Trackmob

(February 01, 2015 - July 01, 2016)
Software Engineer

Projects:

- Analysis and Development of the NGO CRM system
  - Written on Ruby on Rails in a monolithic approach, as it was an MVP.
  - Designed with a multi-client SaaS approach.
  - Used to manage information about all leads and donors of any NGO.
  - System must generate the files needed for banks to execute the donation payments.
    - Banks and formats used, non-exhaustive list:
      - Itaú, CNAB240
      - Santander, Febraban150, CNAB400
      - HSBC, Febraban150
      - Cielo - Proprietary format
      - Braspag - JSON API
      - PagarMe - JSON API
- Maintenance for the Trackmob Management System.
  - Existing system used by technicians to schedule appointments and coordinate their teams in their services;
  - Performance improvements to the loading times as well as queries and navigation;
  - Adding information to the service data such as parts used and requested, their price, equipment information, pictures, etc.
  - Maintenance of the API used to communicate with the Android app using JSON.
- Development of Service Desk guidelines for the company and all its projects.
  - Using ZenDesk system as base;
  - Creating automatic replies, classification and ordering of clients, their threads and responses;

Main technologies:

- Ruby
- Ruby on Rails
- Javascript
- CSS
- Bootstrap
- Git
- PostgreSQL

Main activities:

- Development
- Testing
- Design
- Quality Assurance

Regazzo Soluções em Tecnologia

(May 01, 2014 - February 01, 2015)
Programmer

Project:

- Maintenance of a Android app and it's web-service for vehicle inspection tailor-made for a vehicle factory park maintainer in a major factory in the Curitiba area.
  - Users took pictures of the vehicles in three distinct moments: when they left the assembly, when they were placed in the transports or in regular checks in the vehicle park where they resided.
  - The app allowed users to select specific chassis for different vehicle classes;
  - The app had to work in tablets and had to support for intermittent or simply no available networks at any time during execution.
  - Once the user finished the checklist, the data had to be uploaded to a SOAP/XML web-service with the form data as well as the pictures taken.
  - My job was to implement techniques for fault-tolerance with the web-server communication process in order not to lose any data as well as to retry if connection was lost.
  - Testing was done even in the building stairwells to ensure similar conditions before testing in the field. -The app was written by a third-party no longer involved with the project, with close to no documentation available.

Main technologies:

- .Net Framework 4.5
- Java
- SQL Server 2012

Main activities:

- Development
- Testing

PROJECTS

Level Generator

(March 27, 0202 - Present)
Documentation needed to impplement a Level generator for games

Definition of the requirements, architecture, user stories needed to implement the Level Generator

https://github.com/zeh-almeida/level-generator

requirements, architecture

6502-sharp

(November 18, 2021 - Present)
6502 Emulator written in C#
.Net

While there are a great amount of existing, supported and very well-maintained 6502 emulators, most of them are written in C/C++.
Not only that but most of them are very "machine friendly" but not really "human friendly", this means that code is very optimized to run as fast and as precise as possible.
The objective of this project is to present a alternative: while aiming at the original hardware, the goal is to provide an implementation which is as "human-friendly" as possible.
At the same time, this is a learning project: most people with high-level background tremble and shake when mentioning assembly, binary and any other low-level term, let's change that.
There are many techniques used to reach the objective:

- Object Oriented design
- Code documentation
- Unit tests
- Integration tests

https://github.com/zeh-almeida/6502-sharp

emulator, c#, dotnet