

Pedro Sobota

Software Engineer

Birth date: 07/01/1982 – São Paulo, SP, Brazil

pedro@sobo.cx | https://x.com/pedro_sobota | <http://github.com/pedroos> | <https://pedroos.github.io/> | <https://www.linkedin.com/in/pedrooss/>

Summary

A software engineer for more than 18 years, with experience delivering complex systems in production. Driven, self-motivated individual focused on delivering value through technology. Experienced in a range of backend and desktop technologies, adept at the full software lifecycle, from requirements to deployment. Clean code programmer, with attention to detail and code quality, readability, maintainability and documentation. Advocate for automation, with scripting experience, including Windows and Linux environments, and heavy user of AI tools for task acceleration. Experience with team leading and working in agile business settings. Demanding team player who pushes the team to perform better and supply precise feedback for stakeholders and business units. Self-driven, focused on value maximization and timely delivery.

Programming languages

C# - 10+ years of experience

C++ - 2 years of experience

Java – 3 years of experience

Scala – 2 years of experience

PowerShell

Python

Bash

SQL

Libraries

Windows Forms – 3 years of experience

QT – 2 years of experience

REST

SOAP

XML

JSON

Apache Spark

Apache Kafka

Environments

Windows, Linux Ubuntu, Linux CentOS, Docker

Cloud Environments

Azure, AWS

CI/CD

GIT

Jenkins, Azure

Testing
VSTest, Nunit, QT Test, pytest
Integration testing
Performance testing

Education

Universidade do Sul de Santa Catarina (UNISUL Online)

BsC, Mathematics - Incomplete

2017 – 2019

Certificates

- Azure Fundamentals, April 2024
- Azure AI Engineer, October 2024

Idioms

English, C1 Advanced

Experience

Software engineer

KIS Solutions

March 2022 to April 2023, Florianópolis, Brasil

Onboarded onto a Customer Data Platform system written in Scala consuming data from 16 Kafka topics and taking data through a Medallion Architecture pipeline. The system consisted of:

- 50.000+ lines of code Scala Spark codebase
- 100+ Databricks notebooks using Scala and Python
- 80+ Data Factory pipelines
- 4 Kafka instances
- 4 additional Java peripheral systems
- 1000+ table, tens of millions of data points Azure Synapse instance
- 100s of SQL Server stored procedures with 1000s of lines of SQL
- Tens of millions of data points MySql data warehouse
- 10s of MySql stored procedures
- Data lake with 100s of tables, some with more than a billion data points and deeply nested data (structs and XML)
- Sonatype Nexus instance
- REST API interfaces

A Site Reliability Engineering system was used for deployment management consisting of:

- 1000s of SQL scripts
- Jenkins instance
- 10s of hundred-line Groovy tasks for Jenkins, interfacing with GIT, Sonatype, Databricks, Azure, Nexus, DBFS
- 100s of Data Factory JSON files
- 100s of Databricks resource definitions

I was responsible for development of new integrations according to requirements. The integrations involved Kafka and SFTP (CSV) consumption, pipeline programming in Scala, and data warehouse programming in SQL, as well as management of Databricks and Azure infrastructure such as clusters, storage, pipelines, resources and logging and monitoring. Additional standalone systems in Java using Kafka were developed and ran on VMs.

Here are some milestones met in the project:

- Delivered new integration in Scala using Spark, developing performant Spark algorithms to clean and merge hundreds of millions of data points. The integration was delivered with a month in advance and with minimal maintenance work upon being deployed to production, working for 6 full months
- Reduced processing time from a full day to 3 hours using Delta partition tuning; right-sized clusters achieving optimal cost x performance balance
- Monitored Kafka topics and metrics such as offsets to estimate time to completion of data tasks
- Migrated full pipeline build system to sbt, resulting in 10x smaller and faster incremental deployments. Took full advantage of scripting to deliver productive and efficient operations
- Delivered a second integration from CSV performing CSV cleaning using a custom Spark RDD algorithm
- Executed precision and sensitive data maintenance tasks on large amounts of data in short time under high pressure
- Analyzed and managed tens-of-thousands of lines stored procedures and Scala source effectively
- Delivered precise reports from paged REST APIs using C#, LINQ and Newtonsoft
- Managed JSON Schema, XSD and Spark Schema using relevant tools
- Optimized performance and code across the board, eliminating bottlenecks and increasing maintainability
- Managed tricky databases with strict query requirements, utilizing indices efficiently

- Delivered clear and comprehensive documentation

List of technologies:

Java · Big Data · Azure Data Factory · SQL · Data Engineering · Maven · English · Scrum · Delta Lake · PowerShell · Nexus · Unit Testing · Big Data Analytics · Data Governance · Software Development · Transact-SQL (T-SQL) · Data Analytics · Apache Spark · Object-Oriented Programming (OOP) · SQL Server · Linux · Azure Synapse · Jenkins · Scala · Azure · Extract, Transform, Load (ETL) · Hive · Microsoft Azure · Data Modeling · Apache Kafka · SFTP · Stored Procedures · JSON · Groovy · Azure Databricks · Software Architectural Design · PySpark · Agile Methodologies · HDFS · Docker Products · C# · SBT · .NET · Synapse · CSV · Python (Programming Language) · Cloud Computing · Powershell · Databases · Language Integrated Query (LINQ)

Software engineer

Kantar Ibope Media

January 2021 to January 2022 (1 year), São Paulo, Brazil

I developed a new version of the TGI backend system for processing biannual results from the public opinion polling program. The system was rewritten from scratch from a legacy Visual Basic version, using C# and PowerShell, SQL Server as a data store, and Azure for CI pipeline. Developed in 3 months, it was deployed in a local server in production, running for two seasons and processing tens of millions of records without interferences.

I also modernized the SPMM application, processing results from an in-home consumer opinion polling program using Windows Forms. I created new functionality and UI to manage system entities, developing both the UI and codebehind code. The program was modernized, reducing line of code count, developing new UI, and adding consistency verifications and new features. A unit test suite was recovered and modernized. I also performed deep analysis of XML-structured data using Julia LibExpat.jl, which allowed quick and efficient analysis of XML for attestation.

I modernized and recompiled a Forms OCR application in Java and Glassfish. The application was recovered, code cleaned, dependencies updated, integration with C++ OpenCV reestablished, and dockerized.

- New iteration of TGI backend system using .NET, SQL Server, and Azure
- Modernized and refactored SPMM application in .NET
- Modernization of Forms OCR application using Java and Glassfish
- Dockerization of applications

List of technologies:

Java · Git · Azure DevOps · Maven · Scrum · PowerShell · Software Development · Transact-SQL (T-SQL) · Data Analytics · Object-Oriented Programming (OOP) · SQL Server · Linux · Azure · Microsoft Azure · ASP.NET MVC · Software Architectural Design · Agile Methodologies · Docker Products · C# · .NET · Glassfish · Cloud Computing · Software Design · Databases

Software engineer

3CON

September 2020 to November 2020 (3 months), São Paulo, Brazil

Project 1

- Participated in development of new Clearing system for the B3 Stock Exchange, upgrading a legacy .NET solution to .NET Core. Used libraries such as Dapper and Mapper and ASP.NET Web API components.

Project 2

- Development of sales and CRM software in ASP.NET MVC 5, using Entity Framework with T4 templates with a SQL Server database. Query and XML development using LINQ. AJAX programming with jQuery. Integration with Serasa external SOAP web service.

Summary:

- Financial exchange (B3) integration solution in .NET using REST APIs
- Port of .NET Framework to .NET Core solution, updating dependencies
- System reliability improvement, performance, and documentation
- Technical debt reduction (clean code)

List of technologies:

Git · Scrum · Software Development · Transact-SQL (T-SQL) · Object-Oriented Programming (OOP) · SQL Server · Azure · Microsoft Azure · ASP.NET MVC · Agile Methodologies · C# · .NET · Databases

Software engineer

ESSS

February 2020 to July 2020 (6 months), Florianópolis, Brazil

I was hired to develop in an area that was previously under-attended by the company (Ansys Workbench integration) in the Rocky DEM Application team. Ansys Workbench is an integration software for other simulation products and is written as a C# application and SDK containing objects for executing operations in Workbench be it from a script (IronPython), journaling commands, the user interface or an addin. An Addin is an independent .NET DLL loaded by Workbench to integrate an application, in this case, Rocky DEM, with other engineering applications. The addin is structured in commands and common objects (Project, Context, etc.) that provide a uniform execution environment.

My job was to keep developing the addin as well as integrating with the company's custom build system, written in Python, as well as its Continuous Integration service. I utilized conda, pytest, invoke, cog, as well as other Python tools. I developed custom integration tests for the Addin.

Some specific improvements I brought to Addin were:

- Identified the root cause of a connection mismatch between it and Mechanical Addin
- Stopped error messages about cached files appearing by taking the specific cached files out of Workbench's project structure, and developed a test for it
- Stopped error dialogs halting test job execution by sending error messages to Workbench's error area instead
- Identified a data sharing error with the SpaceClaim application by debugging a Python script inside of SpaceClaim
- Implemented, after meetings with testers, a version-matching structure between the Rocky Addin and Rocky, including a unit test suite
- Identified a wrong sequence of events in application cleanup where cleanup would not be performed before the application was closed
- Improved the logging structure in the Addin by implementing togglable categories

Besides working in the Addin, I interacted with the in-house custom build and CI system:

- Developed tests in pytest to test newly implemented functionality
- Conducted tests locally in source and packaged mode to reproduce errors
- Developed code-generating procedures in Python to, for example, generate an Assembly Info C# file containing proper version information; or generate custom Workbench configuration files to open custom versions of the Addin for development
- Changed artifact caching policies by including additional information in a cache hash
- Performed troubleshooting of failed jobs in Jenkins to identify causes of errors

Other tasks:

- Correcting errors in the dev machine and test server provisioning scripts in Bash and Batch
- Correcting a small error in Bash implementation of path functions in custom conda scripts
- Organizing code with mu git and conda workspaces
- Branching, merging, creating and reviewing Pull Requests

List of technologies:

Git · Scrum · Unit Testing · Software Development · Transact-SQL (T-SQL) · Object-Oriented Programming (OOP) · Jenkins · Software Architectural Design · Agile Methodologies · C# · .NET · Python (Programming Language) · pytest · Powershell · Databases

C++ developer and team lead

Condor Instruments

June 2018 to February 2020 (1 year, 9 months), São Paulo, Brazil

As a team leader and developer, I was responsible to guide development in a small team and lead development direction.

I defined a team workflow including customizing Jira with custom steps, issue fields, and custom rules.

Established guidelines to work with tasks in a streamlined way, taking developer's input into consideration regarding workflows. Aligned important points with Technology Director.

With time, I developed automation tools to streamline working with multiple concurrent tasks, so that the team could be ready to tackle urgent issues. This included downloading code associated with issues in specific branch combinations, facilitating working with GIT, and performing QT-related and other tasks, like deploying packages, with simple commands. We deployed more than 10 versions of the software following this procedure.

Mentored the team on improvements in C++ code, including more recent C++17 features like lambdas and tuples, as well as OO design. Gradually performed critical improvements to the C++ codebase.

Implemented together with the team tests in QtTests, improving test execution, and with new test creation guidelines to create more reliable and easily executed tests when human interaction was required. This resulted in a much more efficient integration testing routine which needed little to no supervision.

Identified and fixed a critical bug in low-level C API code where by repeated state reading, a device would lose synchronization and emit incorrect data over USB.

I modified the application code to enable reading multiple devices in parallel using threading, creating a custom version for that feature.

Developed a custom float hashing algorithm to tackle a security requirement in the application's database, with many numerical series of up to 200.000 data points stored needing to be verified for integrity.

Together with the team, planned a dataflow application for connecting scientific algorithms prototyped in JavaScript (using the no-flo library) and R, and then ported to C++ using the QT NodeEditor library.

I conducted technical interviews with developer candidates, and hired three new developers into the team.

- Led a team of developers developing desktop C++/QT software and hardware integration in the area of chronobiology
- Delivered parallelization of data ingress routine from hardware device
- Fixed buffer bugs in hardware communication in C language with HIDAPI library
- Ported tests platform to QTest and developed new unit and integration tests
- Identified possible sources of a bug in scientific algorithm (Cosinor) using GNU Octave
- Managed a team of developers, designing sprints, defining tasks and conducting reviews and assessments
- Implemented best practices such as code reviews, unit testing, automation, JIRA and Bitbucket implementation, and online documentation

List of technologies:

Git · Scrum · Qt · Unit Testing · Software Development · Object-Oriented Programming (OOP) · Linux · QTest · Jira · C++ · Software Architectural Design · Agile Methodologies · C · Bash · Databases

Java developer

Reamp

April 2016 to January 2018 (1 year, 10 months), São Paulo, Brazil

At Reamp, I was a Java back-end developer working on a internet traffic REST API data-collection and processing system from multiple simultaneous sources (Google Analytics, Criteo, etc.) using Java and Scala and the Java ecosystem (Java Spark Web, Maven, mockito, Eclipse).

I also worked with Node.js, ReactJS and Redux, to develop a new modern front-end system for dashboard applications; I developed custom ReactJS components and visualizations and developed workflows in Knime Analytics Platform.

Besides back-end systems, I also developed Elasticsearch queries in Java; maintained a Postgres database and packages a Ruby on Rails application in Docker.

- Internet audience data ingestion and transformation backends using Java and Scala
- Refactored and modernized existing backend services in Java
- Postgresql integration
- Dockerized applications

List of technologies:

Java · Git · MySQL · Maven · Scrum · Amazon Web Services (AWS) · Unit Testing · Software Development · Mockito · Transact-SQL (T-SQL) · Object-Oriented Programming (OOP) · Linux · Scala · Flyway · Redux.js · JavaScript · Software Architectural Design · Agile Methodologies · Docker Products · Cloud Computing · Software Design · Databases · ReactJS

Python/PHP developer

Seepix Digital

December 2014 to March 2016 (1 year, 4 months), São Paulo, Brazil

At SeePix, I developed back-end systems to a content management system for Out-of-Home displays in Python and PHP. I developed a Python backend task processing system from scratch, ingesting and encoding videos (integrating with ffmpeg) using JSON and uploading the files to Amazon S3. I developed a PHP SOAP webservice ingesting 20-50 messages per second from more than 400 displays deployed in the field; optimized content programming logic; developed a new version of the CMS in CodeIgniter and AngularJS; deployed applications to remote servers via SSH.

- Developed an extensible task processing service in Python to encode videos, upload files to cloud, among other scriptable tasks
- Developed a PHP SOAP web service using SOAP attending to a load of 20-50 incoming scheduling and content messages per second from field-deployed displays
- Improved data algorithms to optimize bandwidth usage, avoiding duplications and redundancies; helped migrate on-premise file storage into Amazon S3
- Developed a Content Management System in CodeIgniter and AngularJS
- Programmed automation in shell script and worked in command-line Linux environments (Ubuntu and CentOS) to develop and deploy applications over SSH, managing successful deployments.

List of technologies:

AngularJS · Git · MySQL · Amazon Web Services (AWS) · CodeIgniter · Software Development · Transact-SQL (T-SQL) · Linux · Web Services · PHP · JSON · Cascading Style Sheets (CSS) · JavaScript · Software Architectural Design · SOAP · Angular · Python (Programming Language) · Cloud Computing · Software Design · Databases · XML

.NET developer and team lead

Akayou

May 2013 to December 2013 (8 months), São Paulo, Brazil

I was engaged as a consultant to build a team for website creator product. I triaged candidates and performed technical interviews, hiring a team. Developed a backend in C# and led the team to develop landing pages and front-ends. I created infrastructure in AWS (S3, RDS, Route53, EC2) and managed deliveries and targets, maintaining regular communication with stakeholders and founders.

- Development of website platform in .NET/C# and MySql/SQL Server
- Tech leadership involving hiring, technical direction, team coordination
- Backend development using C#
- AWS integration (S3, RDS, EC2, Route53)

List of technologies:

Git · Amazon Web Services (AWS) · Unit Testing · Software Development · Transact-SQL (T-SQL) · Object-Oriented Programming (OOP) · SQL Server · Cascading Style Sheets (CSS) · JavaScript · Software Architectural Design · Team Leadership · C# · .NET · Cloud Computing · Databases

.NET developer

Tônica.ag

April 2011 to April 2013 (2 years, 1 month), São Paulo, Brazil

I created the website for HBuster, an electronics company, in ASP.NET MVC, featuring dynamic product and category displaying, deploying it to production. I also developed three websites in ASP.NET MVC: Bridgestone, Abbott Humanizar, and Relance. I developed a client database import and export routine from SQL Server for Iguatemi One - CRMALL in .NET (C#). I also developed a graphical game in JavaScript and HTML5 Canvas for deployment on Facebook.

- E-commerce development using ASP.NET MVC, SQL Server and JavaScript
- Data import/export routine development in SQL and C#
- Designed a totem and Facebook game for a brand using JavaScript and HTML5 Canvas

List of technologies:

Git · MySQL · CoffeeScript · Software Development · Transact-SQL (T-SQL) · SQL Server · ASP.NET MVC · JavaScript · Software Architectural Design · C# · .NET · Software Design · Databases · jQuery

.NET developer

Tech4B

March 2010 to March 2011 (1 year, 1 month), São Paulo, Brazil

Rewrote a backend application written in Visual Basic to C#. The application aggregated outputs of multiple code static analysis tools in multiple languages (FXCop, FindBugs, C and ABAP validators, etc.) and output reports. I performed the relational database modeling (SQL Server) and also developed in ASP.NET with concurrent AJAX a front-end website. The application was deployed in production and ran jobs for clients such as Embraer and Porto Seguro. I also developed a setup application in Windows Forms.

- Task processing system in .NET, integrating static analysis results from multiple tools
- Developed a fault-tolerant backend service for job processing, hosted as Windows Service
- Data modeling and scripting in SQL Server
- Concurrent AJAX web application development using ASP.NET and JavaScript

List of technologies:

Relational Data Modeling · Unit Testing · Software Development · Transact-SQL (T-SQL) · Object-Oriented Programming (OOP) · SQL Server · ASP.NET MVC · JavaScript · Software Architectural Design · C# · .NET · Software Design · Databases

.NET developer

LabOne Systems

March 2008 to February 2010 (2 years), São Paulo, Brazil

At LabOne, I initially developed a playlist editing functionality for the video player software, adapting it to improve its object-orientation after feedback from my leadership. I subsequently load-tested the player application using Apache JMeter, presenting a report of results to my leadership. I then integrated the MediaBox framework, in .NET, in web applications, which performed media ingestion, transformation and distribution tasks. I helped program the video portal Festival do Minuto in ASP.NET, including its Admin website; programmed a royalties

calculation system for UBC (Brazilian Union of Composers) in ASP.NET MVC; and developed end-user reports.

- Development of ASP.NET applications integrating in-house MedialBox framework for media management and distribution
- Development of performant reports using SQL Server
- Developed the video portal Festival do Minuto in ASP.NET MVC and administration website
- Development of royalty calculation system in .NET for composer's association

List of technologies:

Transact-SQL (T-SQL) · Object-Oriented Programming (OOP) · SQL Server · ASP.NET MVC · JavaScript · C# · .NET · Query Optimization · Databases · XML

.NET developer

SBS Consultores

September 2005 to February 2008 (2 years, 6 months), São Paulo, Brazil

Responsible for rewriting the Hospitalar Pharmacy application from Delphi to .NET. I worked from specifications and rewrote the application in Windows Forms, in three tiers (UI, Model and Data Access). I designed a configuration window using XML to persist generic configuration parameters. The solution had about 20.000 lines of code and an NUnit test suite was created with a hundred tests. The application utilized .NET dynamic loading to achieve database vendor-agnostic data storage (SQL Server, MySQL). It was also experimentally tested on Linux using Mono.

After finishing it, I worked on a data initiative interfacing with the company's Blood Bank software, by consuming its data using SQL Server Transformation Services and aggregating a cube in Analysis Services. Windows Workflow Foundation was also employed to design a workflow-oriented version of the application.

- Developed a new version of Hospitalar Dose application in .NET, Windows Forms, SQL Server and MySQL
- Utilized XML for modeling parameters, and XSLT/HTML for report rendering
- Developed a NUnit test suite
- Initiated development of a second application, Hemolog, following the platform created for the initial application
- Implemented a SQL Server Integration Services (SSIS) data transformation project consuming data from blood bank application

List of technologies:

MySQL · SQL Server Integration Services (SSIS) · Unit Testing · Software Development · Transact-SQL (T-SQL) · HTML · Object-Oriented Programming (OOP) · SQL Server · XSLT · PHP · Cascading Style Sheets (CSS) · JavaScript · Software Architectural Design · NUnit · Windows Forms · C# · .NET · Software Design · WinForms · Databases · XML

Two further experiences omitted.