

Guilherme Patrão

My Portfolio | guilherme.a.patrazo@gmail.com

Projects

IVM Management – Data Bases class - <https://github.com/gui-alm/ivm-management>

This class was all about the conception and analysis of data bases.

The project consisted of developing SQL queries, developing complex integrity constraints, creating a web app prototype and writing OLAP queries as well as using SQL indexes. For this, PostgreSQL and Python were used.

By accessing the link provided, you should have access to every file. Each file corresponds to a certain topic that was an object of evaluation.

The context for the project is that it's a fictitious web application for managing vending machines from different manufacturers. This was a group project with a group composed of 3 people. Although not all the code was entirely made by me, I had a hand in almost every single file. For the content of the following files, all its code is of my authorship: web/core.cgi ; web/templates (every file in this folder) ; loading-script.py ; populate.sql; setup_db.py. I also had a hand in the making of the ICs.sql and queries.sql files, however they are not 100% of my authorship, unlike the previously mentioned files.

| | |
|-------------------|---|
| loading-script.py | Script used to create the populate.sql file. |
| populate.sql | Contains the SQL code to create and populate the database tables. |
| ICs.sql | Contains the code for the complex integrity constraints. |
| queries.sql | Contains the code for some SQL queries. |
| view.sql | Contains the code for a view. |
| web folder | Contains the code for the web application. |
| analytics.sql | Contains the code for the OLAP queries. |

The project was graded with 19.60 out of 20 possible points.

Subsequence Finder – Analysis and Synthesis of Algorithms class -

<https://github.com/gui-alm/subsequence-finder>

This small project consisted of solving the longest increasing subsequence problem (LIS) and the longest common increasing subsequence problem (LCIS). Given a sequence of numbers, the goal was to output the size of the LIS as well as the number of possible subsequences of such size for the first part and to output the size of the LCIS for the second part.

This was a group project with a group composed of 2 people.

Besides the code, a document with a description of the problem, a description of the solution for the problem and an analysis of experimental tests to determine the algorithm's complexity was also handed over for evaluation. This report is written in portuguese and can also be found in the linked GitHub repository.

The project was graded with 19.75 out of 20 possible points (17/17 for the code and 2.75/3 for the project report).

Warehouse Manager – Programming with Objects class -

<https://github.com/gui-alm/warehouse-manager-application>

This project consisted of developing an application that could manage a warehouse. Among many features, the main ones are buying and selling products, registering product breakdowns, consulting past transactions, consulting product inventories, a partner system where each partner could participate in a points system granting discounts and more as well as being able to be notified of inventory changes, load and save files, lookup data, etc. The goal for the project was to teach students how to work with objects and how to efficiently plan a project.

The project also had a planning component where some of its structure was planned in UML beforehand. However, the initial UML that was created ended up not being used for the most part.

This was a group project and therefore both the code and UML were made by 2 people. I can say I had a contribution of about 80% for the code and 30% for the UML.

Everything in project/po-stdlib was provided by the professor and is not of mine or my colleague's authorship.

The project was graded with 16.30 out of 20 possible points.

File System – Operating Systems class - <https://github.com/gui-alm/ist-file-system>

The project consists of a custom basic file system with the ability to execute various client processes concurrently.

The project started with making the file system's basic operations (opening, closing, writing files, reading files) thread-safe and only then moved on to adding support to serve multiple client processes in a concurrent manner, rather than sequential.

The functionalities of the file system are offered in a server process. Other processes can call the file system's functions with system calls to the operating system's kernel which then redirects those requests to the server process through an open pipe used to communicate. Once the function has finished its job, the content returned by the function is sent to the client via another pipe.

This was a group project with a group composed of 2 people, both contributing the same amount towards it.

The project was graded with 19.16 out of 20 possible points.

Takuzu – Artificial Intelligence class - <https://github.com/gui-alm/takuzu-ai>

The project consists of a Python application that solves the Takuzu puzzle problem using AI search algorithms. The application receives an input (puzzle) and prints the solution to the given puzzle.

This was a group project with a group composed of 2 people, both contributing the same amount towards it. The code in src/search.py and src/utls.py is not of mine or my colleague's authorship, it was provided by the professor.

Besides the code, a document with a description of the problem, a description of the solution for the problem and an analysis of experimental tests to compare the different searches (DFS, BFS, A* search and Greedy search) complexities as well as efficiency, execution time and number of generated and expanded nodes was also handed over for evaluation. This report is written in portuguese and can also be found in the linked GitHub repository.

The project was graded with 18.93 out of 20 possible points (15/15 for the code and 3.93/5 for the report).

Hangman Game Client and Server - Computer Networks class -

<https://github.com/gui-alm/hangman-client-and-server>

The goal of this project was to develop a simple word game, inspired by the classical Hangman Game. Its development required implementing a Game Server (server) and a Player Application (client).

The implementation uses the application layer protocols operating according to the client-server paradigm, using the transport layer services made available by the socket interface. The project sought to evaluate the development of applications and the implementation of the underlying communication protocols (UDP and TCP), requiring an understanding of computer networking concepts.

This was a group project with a group composed of 2 people, with both of us being fully involved in all of the project's implementation.

The project was graded with 18.50 out of 20 possible points.

Hotel Reviews Classification - Natural Language Processing class -

<https://github.com/gui-alm/nlp-hotel-reviews-classification>

In this NLP project, I leveraged diverse classifiers, encompassing Logistic Regression, Support Vector Classifier, and Naïve Bayes algorithms, to tackle the nuanced task of classifying hotel reviews into four categories: positive-deceptive, negative-deceptive, positive-truthful, and negative-truthful.

This was a group project with a group composed of 2 people, both contributing the same amount towards it.

Working with a balanced dataset containing 1400 reviews, I delved into the intricacies of Natural Language Processing. Throughout the analysis, I encountered challenges inherent to NLP tasks, notably addressing context misinterpretation errors and navigating dataset ambiguities.

The project concluded with a detailed report covering various topics such as the models used, an experimental setup and results section, a discussion section where I analyzed the dataset and the obtained results in order to identify common errors in the classification and a future work section. The full report, available as a PDF file, can be found in the GitHub repository.

The project was graded with 19.00 out of 20 possible points, attributed to a score of 6/5 for the code, where a bonus point was awarded for achieving an accuracy bigger than 85%, and 13/15 for the report.

Internal Healthcare System - Network and Computer Security class -

<https://github.com/gui-alm/healthcare-internal-system>

This project is set in a medical environment, where both doctors and patients must be able to communicate with the system to store or view medical records, keeping the confidentiality of sensible patient information, as well as the authenticity of the records registered by the doctors.

In order to achieve this, a security library that implements 3 major functionalities was created (including encrypting, decrypting and verifying documents), which can be used on the command-line, as well as a multitude of methods that can be imported to code projects.

The project was graded with 19.00 out of 20 possible points.

