

Project Artificial Intelligence Bank branches

José Emparanza, Alberto Sainz y Santiago Norzagaray

May 12, 2025

Contents

1	GitHub	2
2	Description of the project	2
3	Code structure	3
3.1	Main	3
3.2	Bank	3
3.3	Clients	3
3.4	Branches	3
3.5	Config	4
3.6	Fuzzy-risk	4
3.7	Steps-simulation	4
4	Technical Aspects	5
5	AI Employed	6
6	Execution analysis	7
6.1	Explanation	7
6.2	Screenshots	10

1 GitHub

<https://github.com/albertoosg/Practica-IA>

2 Description of the project

The main objective of this project is to simulate the functioning of a bank branch, providing a realistic representation of customer service dynamics, employee roles, and operational workflow. The simulation will include multiple branches (A, B, and C), each designed to reflect the structure and organization of a real-world banking environment.

Each branch will have a designated number of service posts available to attend to customers. These posts will be responsible for handling various essential banking operations as:

- **Deposit and withdrawal of money:** Customers can deposit funds into their accounts or withdraw cash as needed. These operations do not have any risk, as both are simple operations.

- **Investments:** Bank employees will provide investment advice, manage financial portfolios, and assist customers in making informed investment decisions. Each investment has a determined risk depending on the age of the customer, the earnings and the quantity variables. With this risk the clients will have to deal with when wanting to invest.

- **Bank loans and credits:** The bank will offer loan services, where customers can apply for personal or business loans, mortgages, and other credit-related products. Each loan has a determined risk depending on the age of the customer, the earnings and the quantity variables. With this risk the clients will have to deal with when asking for a loan.

Each service post will be managed by a dedicated employee who will be responsible for efficiently handling customer requests. In addition, each branch will have a branch manager, tasked with overseeing daily operations, ensuring workflow efficiency, and addressing any issues that arise.

Customers will arrive at the bank branch at a and will be organized into queues based on their order of arrival. They will then be directed to the appropriate service posts based on their specific banking needs. After performing operations in the branch, the customers leave.

Each action that happened during the simulation occurs at a specific moment in time, from the moment a customer enters the branch to the moment the customer leaves. Each action occurs at a specific step of the simulation. Moreover, the simulation shows how long a customer waits in the queue until its attended by an employee in a post of the branch.

3 Code structure

3.1 Main

Here we find the customer generator function, which creates customers during the simulation process. The random time between customer arrivals and transactions is specified, and customers are randomly assigned a counter and branch. Then the simulation environment is initialized, with the creation of as many branches as desired, with the creation of the bank, and with the generation of customers for the duration of the simulation.

In the end, we obtain a summary of the data of each branch with which we can make a comparison of which one is more efficient, which one has more clients, in which the average risk is higher or lower, or the ones that lend or receive more money for investments.

3.2 Bank

The Bank class serves as the central entity in the simulation, managing the overall structure and operations of the bank. It contains general information about the bank, such as its name, the total number of branches, and any additional attributes that may define its operations. This class is responsible for assigning customers to branches and ensuring a balanced distribution of clients across multiple locations.

3.3 Clients

The Clients class is designed to represent a customer who interacts with a bank branch in a simulated environment. This class stores and manages essential customer information while tracking their journey through the bank, from arrival to departure. Clients will be able to perform different operations, such as depositing or withdrawing money, investing, or borrowing with probability.

Each instance of this class will hold details about an individual customer, such as their unique identification, financial status, and the type of banking operations they may require.

3.4 Branches

The branch class provides information on the branches and the number of counters in each branch. It has counters for the total number of customers who come to the branch, the money lent by the bank, and the money invested by the customers.

3.5 Config

The config class defines the basic configuration parameters for the simulation, including the name of the bank, the number of simulation time steps, branch data (such as number of counters and interest rates for loans and investments), as well as possible ranges for the number of customers and their starting salaries.

3.6 Fuzzy-risk

The Fuzzy-risk class implements a fuzzy logic system with the skfuzzy library to calculate the risk associated with financial transactions based on the customer's age, balance and requested amount, using fuzzy sets and a set of rules that determine whether the risk is low, medium or high.

3.7 Steps-simulation

The Steps-simulation class contains the main logic for running the simulation in time steps using simpy, managing the operations performed by customers in each step, the concurrent execution of these operations and the generation of summaries at the end of the simulation, both at the level of each branch and of the entire bank.

4 Technical Aspects

The code simulates the activity of a multi-branch bank using the `simpy` library for event simulation. It is structured in seven main files: `main.py`, which organizes the simulation; `branch-client.py`, which defines the clients and their operations; `branch-bank.py`, which handles branch assignment and summary generation; `branch.py`, which stores the data for each branch; `config.py`, which defines the number of simulation time steps, branch data and the ranges for the number of customers and their starting salaries; `fuzzy-risk.py`, which calculates the risk associated with financial transactions based on the customer's age, balance and requested amount; and `steps-simulation.py`, which manages everything about the different steps of the simulation.

From the design point of view, the code seeks to follow very good encapsulation practices, trying to organize the functionality in specific classes such as `Client`, `Bank` and `Bank-Branch`. In practice, static typing is used, which improves code clarity, and an efficient structure is used to store transactions (`self.history` in the `Bank` class). In addition, the use of the `simpy` library allows modeling attention queues and waiting times in a realistic way.

Performing a more technical analysis of each file, it can be observed that:

- **main.py:** Presents a well-structured flow, since it configures branches, requests simulation parameters to the user, executes the process and generates a summary.
- **branch-client.py:** In this class, clients execute banking operations randomly, waiting for their turn in the assigned branch.
- **branch-bank.py:** This class manages branch randomization and generates eight CSV reports.
- **config.py:** The data structure used to define the branches allows easy parameterization of key characteristics such as the number of counters and interest rates, facilitating the scalability of the simulation to more branches or customized conditions.
- **fuzzy-risk.py:** The use of a fuzzy control system to calculate financial risk adds a more realistic and flexible approach than rigid rules, allowing complex situations to be evaluated based on continuous variables such as age, balance and amount.
- **steps-simulation.py:** The integration of `simpy.events.AllOf` to synchronize multiple concurrent client operations at each step accurately simulates the parallel execution of processes within a realistic banking environment.

5 AI Employed

The fuzzy risk algorithm implemented in the `fuzzy-risk.py` file uses fuzzy logic to evaluate the risk associated with a financial transaction (such as an investment or loan request) as a function of three input variables: the customer’s age, his available balance and the amount involved in the transaction. This approach allows uncertainty and subjectivity to be handled more realistically than deterministic techniques.

First, we define the fuzzy linguistic variables: age, balance and quantity, each divided into fuzzy subsets, for example “very young”, “medium”, “high”, by means of triangular membership functions using `skfuzzy`. These functions allow representing gradualness, so that a person can be “young” and “adult” at the same time, with different degrees of membership.

The output variable, risk, is also defined as fuzzy and has three categories: low, medium and high, representing the calculated risk level. Then, 24 fuzzy rules (of the if-then type) are created that combine the inputs to determine the output value. For example: if the age is very young, the balance is very low and the amount is very small, then the risk is high. These rules mimic human decisions in financial evaluation.

The inference engine combines the rules triggered by the input values using the `skfuzzy` control system, which generates a risk surface. For a specific set of inputs (age, balance, amount), the function is used to generate a risk surface. For a specific set of inputs (age, balance, amount), the `evaluate-risk()` function normalizes the values, inputs them to the simulation system (`ControlSystemSimulation`), and executes the computation.

The result is a numerical value between 0 and 100 representing the evaluated risk. This value can be interpreted or used to make automatic decisions on whether or not to approve an operation.

6 Execution analysis

6.1 Explanation

By executing the project code, the program asks the user how many customers the user wants to generate to enter and operate in a branch. In addition, the program asks the user to specify the minimum and maximum amount of money that clients will have available during the simulation. Once the user answers the three requirements, the simulation starts.

These customers enter a branch at a moment in time to try to withdraw or deposit money, invest, create a bank account, manage a bank account, or request a loan. Once they perform one or more of the operations that can be performed in the branch, the customers finish their appointment in the branch.

These customers can just withdraw money if their card balance allows them to withdraw the amount of money they want. Moreover, the investment operation, in the same way that occurs with the money withdrawal, can just be carried through if the card balance allows it. As explained in the project description, the card balance allows a customer to make a money withdrawal or investment if the card balance is higher or equal to the amount of money the customer is trying to operate with.

The program shows, when running, everything that happens during the simulation. The messages that could be shown during the simulation are:

- **"Client x arrives to Bank Branch X at x time units"**: This message is shown when a customer enters a branch. Specifies the number of the client, the branch it is entering, and the time at which the client has entered.
- **"Client x is being attended to at Bank Branch X at x time units. (Time waiting x time units)"**: This message is shown when a customer is attended at a post by an employee. Specifies the number of the client, the branch it is in being attended to, the time at which the client has been attended to, and how long the client waited in the queue.
- **"Client x apply for a loan of x€. Current balance: x€"**: This message is shown when a customer applies for a loan at a branch post. Specifies the number of the client, the amount the client applied, and the balance after performing the operation.
- **"Client x withdraw x€. Current balance: x€"**: This message is shown when a customer withdraws an amount of money. Specifies the number of the client, the amount of money the customer wants to withdraw, and the balance after performing the operation (the balance must be higher than or equal to the amount invested).

- **"Client x tries to withdraw x€, but only has x€"**: This message is shown when a customer tries to withdraw money, but the amount the customer wants to withdraw is not available because the card balance is not enough. Specifies the number of the client, the amount of money the customer wants to withdraw, and the balance before performing the operation (the balance must be less than the amount tried to withdraw).
- **"Client x invest x€. Current balance: x€"**: This message is shown when a customer invests in a financial portfolio. Specifies the number of the client, the amount of money the customer wants to invest, and the balance after performing the operation (the balance must be higher than or equal to the amount invested).
- **"Client x cant invest x€, insufficient balance."**: This message is shown when a customer tries to invest money, but the amount the customer wants to withdraw is not available because the card balance is not enough. Specifies the number of the client, the amount of money the customer wants to invest, and the balance before performing the operation (the balance must be less than the amount tried to invest).
- **"Client x deposit x€. Current balance: x€"**: This message is shown when a customer deposits an amount of money. Specifies the number of the client, the amount the customer is depositing, and the balance after performing the operation.
- **"Client x finishes his appointment at Bank Branch X at x time units"**: This message is shown when a customer leaves the branch after performing one or more operations. Specifies the number of the client, the branch at which the customer has been operating, and the time at which the client has finished his appointment and leaves.

As explained in the description of the project, each operation is performed in a moment of time during the simulation, depending on when each customer has entered the branch and it is organized in the queue to be attended.

Once the simulation is finished, the program generates a summary of each branch. This summary collects all the information and every event that happened during the simulation. It shows the user the number of clients that attended each branch, the total amount of money lent by the bank in each branch, and the total amount of money invested by the clients in each branch during the simulation.

All the information is stored at the end of the simulation in eight files .csv. These files name and content are different because of the number of clients that are in the simulation of each file, and because of the minimum and the maximum amount of money the clients can have at the beginning, in the simulation of each file. These files are called: "simulation-x-y-z.csv", where x represents the number of clients, y represents the minimum amount of money the

clients can have at the beginning of the simulation, and z represents the maximum amount of money the clients can have at the beginning of the simulation.

Moreover, all the data are analyzed in "bank-analisis.ipynb", where six graphics are created to represent some simulation data and the AI employed. This six graphics represent: the risk evolution per branch over simulation's time (in each step); the risk distribution per operation (only invest and loan); the risk average and the quantity of money operated with per branch; the distribution of risks levels across all operations in all simulations; the correlations between numerical variables like "quantity", "risk", "balance" and "step"; and the average risk and final balance for each simulation scenario.

6.2 Screenshots

```
Simulation with 10 clients with salary between 100 and 1000 and Steps: 5

--- STEP 1 of 5 ---
Performing 21 operations this step (Client count: 10).
Client 3 is being attended to at Bank Branch C at 0.00 time units. (Time waiting 0.00 time units).
Client 8 is being attended to at Bank Branch C at 0.00 time units. (Time waiting 0.00 time units).
Client 2 is being attended to at Bank Branch A at 0.00 time units. (Time waiting 0.00 time units).
Client 7 is being attended to at Bank Branch A at 0.00 time units. (Time waiting 0.00 time units).
Client 4 is being attended to at Bank Branch B at 0.00 time units. (Time waiting 0.00 time units).
Client 7 withdraw 108€. Current balance: 600€
Client 7 finishes operation retire at 1.06 time units in step 1. Current balance: 600.00€
Client 10 is being attended to at Bank Branch A at 1.06 time units. (Time waiting 1.06 time units)
Client 3 deposit 774€. Current balance: 1218€
Client 3 finishes operation deposit at 1.08 time units in step 1. Current balance: 1218.00€
Client 1 is being attended to at Bank Branch C at 1.08 time units. (Time waiting 1.08 time units).
Client 8 withdraw 325€. Current balance: 217€
Client 8 finishes operation retire at 1.18 time units in step 1. Current balance: 217.00€
Client 8 is being attended to at Bank Branch C at 1.18 time units. (Time waiting 1.18 time units).
Client 4 has a risk value of 83.02% for an investment of 846€
Client 4 invest 846€. The amount they will get back is 913.68€
Will receive 182.74€ per step for 5 steps. Current balance: 94.00€
Client 4 finishes operation invest at 1.18 time units in step 1. Current balance: 94.00€
Client 2 has a risk value of 82.67% for an investment of 980€
Client 2 cant invest 980€, insufficient balance.
Client 2 finishes operation invest at 1.78 time units in step 1. Current balance: 493.00€
Client 2 is being attended to at Bank Branch A at 1.78 time units. (Time waiting 1.78 time units).
Client 1 deposit 123€. Current balance: 869€
Client 1 finishes operation deposit at 2.10 time units in step 1. Current balance: 869.00€
Client 5 is being attended to at Bank Branch C at 2.10 time units. (Time waiting 2.10 time units).
Client 10 has a risk value of 84.13 for a loan of 138€
Client 10 apply for a loan of 138€. The amount they will have to pay back 144.90
Will pay 28.98€ per step for 5 steps. Current balance: 945€
Client 10 finishes operation loan at 2.19 time units in step 1. Current balance: 945.00€
Client 10 is being attended to at Bank Branch A at 2.19 time units. (Time waiting 2.19 time units)
Client 8 has a risk value of 85.71 for a loan of 413€
Client 8 has a high risk value. Loan request denied.
Client 8 finishes operation loan at 2.47 time units in step 1. Current balance: 217.00€
Client 5 is being attended to at Bank Branch C at 2.47 time units. (Time waiting 2.47 time units).
Client 2 deposit 831€. Current balance: 1324€
Client 2 finishes operation deposit at 2.85 time units in step 1. Current balance: 1324.00€
```

Figure 1: Example Simulation 1

```

--- STEP 2 of 5 ---
Performing 30 operations this step (Client count: 20).
Client 3 pays 370.39€ for active loans this step.
Client 7 pays 606.53€ for active loans this step.
Client 9 pays 655.62€ for active loans this step.
Client 10 receives 760.97€ from active investments this step.
Client 14 pays 765.08€ for active loans this step.
Client 17 pays 765.03€ for active loans this step.
Client 18 pays 744.64€ for active loans this step.
Client 18 receives 894.89€ from active investments this step.
Client 19 receives 819.29€ from active investments this step.
No se activo ninguna regla para edad = {val_age}, saldo = {val_saldo}, cantidad = {val_quantity}
No se activo ninguna regla para edad = {val_age}, saldo = {val_saldo}, cantidad = {val_quantity}
No se activo ninguna regla para edad = {val_age}, saldo = {val_saldo}, cantidad = {val_quantity}
No se activo ninguna regla para edad = {val_age}, saldo = {val_saldo}, cantidad = {val_quantity}
Client 4 is being attended to at Bank Branch A at 12.87 time units. (Time waiting 0.00 time units).
Client 13 is being attended to at Bank Branch A at 12.87 time units. (Time waiting 0.00 time units).
Client 19 is being attended to at Bank Branch B at 12.87 time units. (Time waiting 0.00 time units).
Client 1 is being attended to at Bank Branch C at 12.87 time units. (Time waiting 0.00 time units).
Client 3 is being attended to at Bank Branch C at 12.87 time units. (Time waiting 0.00 time units).
Client 10 is being attended to at Bank Branch B at 12.87 time units. (Time waiting 0.00 time units).
Client 19 has a risk value of 55.00 for a loan of 2224€
Client 19 apply for a loan of 2224€. The amount they will have to pay back 2312.96
Will pay 578.24€ per step for 4 steps. Current balance: 3818.288€
Client 19 finishes operation loan at 13.97 time units in step 2. Current balance: 3818.29€
Client 2 is being attended to at Bank Branch B at 13.97 time units. (Time waiting 1.10 time units).
Client 10 tries to withdraw 2665€, but only has 1796.968€
Client 10 finishes operation retire at 14.06 time units in step 2. Current balance: 1796.97€
Client 18 is being attended to at Bank Branch B at 14.06 time units. (Time waiting 1.19 time units).
No se activo ninguna regla para edad = {val_age}, saldo = {val_saldo}, cantidad = {val_quantity}
Client 3 has a risk value of 100.00 for a loan of 4206€
Client 3 has a high risk value. Loan request denied.
Client 3 finishes operation loan at 14.22 time units in step 2. Current balance: 5863.61€
Client 14 is being attended to at Bank Branch C at 14.22 time units. (Time waiting 1.35 time units).
Client 13 deposit 748€. Current balance: 7173€
Client 13 finishes operation deposit at 14.29 time units in step 2. Current balance: 7173.00€
Client 11 is being attended to at Bank Branch A at 14.29 time units. (Time waiting 1.42 time units).
Client 4 deposit 3287€. Current balance: 5070€
Client 4 finishes operation deposit at 14.69 time units in step 2. Current balance: 5070.00€
Client 4 is being attended to at Bank Branch A at 14.69 time units. (Time waiting 1.82 time units).
Client 1 has a risk value of 18.18 for a loan of 3124€

```

Figure 2: Example Simulation 2


```

Client 4 is being attended to at Bank Branch A at 14.69 time units. (Time waiting 1.82 time units).
Client 1 has a risk value of 18.18 for a loan of 3124€
Client 1 apply for a loan of 3124€. The amount they will have to pay back 3217.72
Will pay 804.43€ per step for 4 steps.Current balance: 5029€
Client 1 finishes operation loan at 14.75 time units in step 2. Current balance: 5029.00€
Client 1 is being attended to at Bank Branch C at 14.75 time units. (Time waiting 1.88 time units).
Client 18 has a risk value of 55.00 for a loan of 1838€
Client 18 apply for a loan of 1838€. The amount they will have to pay back 1911.52
Will pay 477.88€ per step for 4 steps.Current balance: 4046.248€
Client 18 finishes operation loan at 15.40 time units in step 2. Current balance: 4046.25€
Client 20 is being attended to at Bank Branch B at 15.40 time units. (Time waiting 2.53 time units).
Client 11 deposit 4328€. Current balance: 11442€
Client 11 finishes operation deposit at 15.65 time units in step 2. Current balance: 11442.00€
Client 9 is being attended to at Bank Branch A at 15.65 time units. (Time waiting 2.78 time units).
Client 2 withdraw 3286€. Current balance: 2813€
Client 2 finishes operation retire at 15.67 time units in step 2. Current balance: 2813.00€
Client 10 is being attended to at Bank Branch B at 15.67 time units. (Time waiting 2.81 time units).
Client 14 deposit 801€. Current balance: 4110.916€
Client 14 finishes operation deposit at 15.97 time units in step 2. Current balance: 4110.92€
Client 3 is being attended to at Bank Branch C at 15.97 time units. (Time waiting 3.10 time units).
No se activo ninguna regla para edad = {val_age}, saldo = {val_saldo}, cantidad = {val_quantity}
Client 1 has a risk value of 100.00% for an investment of 3126€
Client 1 has a high risk value. Investment request denied.
Client 1 finishes operation invest at 16.09 time units in step 2. Current balance: 5029.00€
Client 6 is being attended to at Bank Branch C at 16.09 time units. (Time waiting 3.23 time units).
No se activo ninguna regla para edad = {val_age}, saldo = {val_saldo}, cantidad = {val_quantity}
Client 4 has a risk value of 100.00% for an investment of 4153€
Client 4 has a high risk value. Investment request denied.
Client 4 finishes operation invest at 16.46 time units in step 2. Current balance: 5070.00€
Client 11 is being attended to at Bank Branch A at 16.46 time units. (Time waiting 3.59 time units).
Client 9 has a risk value of 55.00 for a loan of 2590€
Client 9 apply for a loan of 2590€. The amount they will have to pay back 2719.50
Will pay 679.88€ per step for 4 steps.Current balance: 5541.38€
Client 9 finishes operation loan at 17.01 time units in step 2. Current balance: 5541.38€
Client 8 is being attended to at Bank Branch A at 17.01 time units. (Time waiting 4.14 time units).
Client 20 deposit 4180€. Current balance: 6662€
Client 20 finishes operation deposit at 17.31 time units in step 2. Current balance: 6662.00€
Client 20 is being attended to at Bank Branch B at 17.31 time units. (Time waiting 4.44 time units).
Client 10 tries to withdraw 4927€, but only has 1796.968€
Client 10 finishes operation retire at 17.32 time units in step 2. Current balance: 1796.97€
Client 10 is being attended to at Bank Branch B at 17.32 time units. (Time waiting 4.45 time units).

```

Figure 3: Example Simulation 3

```

--- STEP 4 of 5 ---
Performing 60 operations this step (Client count: 20).
Client 1 pays 804.43€ for active loans this step.
Client 3 pays 370.39€ for active loans this step.
Client 3 receives 1185.43€ from active investments this step.
Client 4 receives 1242.52€ from active investments this step.
Client 5 receives 1013.04€ from active investments this step.
Client 6 pays 1359.94€ for active loans this step.
Client 7 pays 1157.21€ for active loans this step.
Client 8 pays 2814.35€ for active loans this step.
Client 9 pays 1335.50€ for active loans this step.
Client 10 pays 666.64€ for active loans this step.
Client 10 receives 760.97€ from active investments this step.
Client 11 pays 231.53€ for active loans this step.
Client 12 pays 92.01€ for active loans this step.
Client 14 pays 765.08€ for active loans this step.
Client 16 pays 1093.75€ for active loans this step.
Client 17 pays 765.03€ for active loans this step.
Client 18 pays 1977.21€ for active loans this step.
Client 18 receives 894.89€ from active investments this step.
Client 19 pays 578.24€ for active loans this step.
Client 19 receives 819.29€ from active investments this step.

```

Figure 4: Example Simulation 4

```

Summary of Bank Branch A:
Attended clients: 89
Total amount lent by the branch: 47867.00€
Total amount invested by the clients: 8014.00€
Interest for loans: 5.00%
Interest for investments: 3.00%

Summary of Bank Branch B:
Attended clients: 72
Total amount lent by the branch: 28676.00€
Total amount invested by the clients: 17913.00€
Interest for loans: 4.00%
Interest for investments: 8.00%

Summary of Bank Branch C:
Attended clients: 59
Total amount lent by the branch: 30384.00€
Total amount invested by the clients: 7705.00€
Interest for loans: 3.00%
Interest for investments: 6.00%

GENERAL BANK SUMMARY:
Total clients attended: 220
Total amount lent: 106927.00€
Total amount invested: 33632.00€

Best Branch Conditions Summary:
Best loan conditions: Bank Branch C with 3.00%
Best invest conditions: Bank Branch B with 8.00%
Simulation data exported to /home/cunef/entorno1/prueba_banco2/data_analisis/simulation_20_200_5000.csv

```

Figure 5: Summary Simulation