

# Programação Não Procedural

Evaristo Domingos Paulo

## Índice

1 – Introdução -----	03
2 – Objectiva -----	04
4 – Fluxo/Funcionalidade -----	05
4 – Conclusão -----	07
5 – Usuários autorizados -----	08
6 – Bibliografia -----	09

## Introdução

Neste trabalho pretende-se desenvolver um sistema em haskell (Linguagem de Programação Funcional) que simula os serviços do Banco P.J.

## Objectiva

O objectivo inicial é desenvolver um sistema para o banco PJ, que destaca a qualidade, eficiência e produtividade dos seus funcionários, desacelerando assim, os processos burocráticos nas suas diversas facetas.

## Fluxo/Funcionalidade

Os sistemas bancários reais armazenam grandes quantidades de informações, permitindo assim que as informações sejam persistidas de forma permanente, eficiente e a todos os níveis, e para isso, é necessário recorrermos a uma base de dados. Neste sistema em particular, usaremos os conceitos de ficheiros para persistir as informações que advêm do fluxo da aplicação.

O sistema foi dividido em dois (2) módulos:

1. Fila
2. Atendimento

### 1. Fila

Ao aceder as instalações do banco, o cliente deverá escolher um serviço, receber uma senha (gerada automaticamente) e aguardar a sua chamada para o atendimento de acordo ao serviço pretendido.

Os serviços são:

- A. Criação de conta
- B. Depósito
- C. Levantamento
- D. Transferência
- E. Extracto
- F. Consulta

### 2. Atendimento

Apenas pessoas autorizadas podem ter acesso a este módulo, é necessário que se informa o usuário e senha para posterior validação (ou não) da credencial.

Supondo que, a credencial foi validada, então o usuário autorizado é encaminhado ao menu principal deste módulo, contendo a seguinte descrição:

1. Atendimento geral
2. Relatório diário
3. Relatório geral

### 1. Atendimento geral

Nesta secção, encontra duas opções de escolha:

1. Chamar
2. Sair

## 1. Chamar

Está e a secção responsável pelo atendimento geral. Com base as senhas recebidas anteriormente, os clientes são chamados no balcão, de forma aleatória, e sua situação resolvida.

## 2. Sair

Voltar ao módulo atendimento

## 2. Relatório diário

Nesta secção, é possível ver relatórios de contas, a partir de uma data fornecida.

Os relatórios diários que podem ser visualizados são:

1. Criação de conta
2. Depósito
3. Levantamento
4. Transferência
5. Estatística

## 3. Relatório geral

Nesta secção, também é possível ver relatórios de conta dos clientes, mas diferente da secção relatório diário, nesta secção não há filtro de data.

Os relatórios gerais que podem ser visualizados são:

1. Criação de conta
2. Depósito
3. Levantamento
4. Transferência
5. Estatística

## Conclusão

As dificuldades foram de vários ordens, mas principalmente ligadas a conclusão e a entrega do projecto no prazo estipulado. Por outra, as manipulações de lista de tuplas, tivemos bastante dificuldade em encontrar ou implementar funções que melhores adaptam-se a este tipo de dado.

Foi um projecto bastante enriquecedor, pois para além de nos ajudar a compreender melhor a linguagem Haskell, nos ajudou também a entender melhor como funciona o paradigma funcional.

Usuários autorizados

Usuário	Senha
Evaristo	11111
Etelvina	22222
Maria	222272



## Bibliografia

Todos os slides da cadeira de Programação não Procedural do 3º Ano

<https://hoogle.haskell.org/>

<http://zvon.org/>

<http://haskell.tailorfontela.com.br/starting-out>

[https://www.youtube.com/results?search\\_query=read+tuple+in+file+in+haskell](https://www.youtube.com/results?search_query=read+tuple+in+file+in+haskell)

[https://www.youtube.com/results?search\\_query=how+to+install+system.random+in+haskell](https://www.youtube.com/results?search_query=how+to+install+system.random+in+haskell)