



Departamento de Eng. Electrotécnica e de Computadores

FCTUC

# Relatório

---

## Computação Gráfica

---

Professor: José Carlos Teixeira ([teixeira@mat.uc.pt](mailto:teixeira@mat.uc.pt))

### Gestão Financeira Familiar

**Autores:**

João Barreiros

uc2014196880@student.uc.pt

João Ferreira

uc2013139657@student.uc.pt

**Data:**

15-04-2018

## ÍNDICE

<b>1. SUMÁRIO .....</b>	<b>3</b>
<b>2. INTRODUÇÃO .....</b>	<b>4</b>
2.1 VISÃO .....	4
2.2 ANÁLISE E ESPECIFICAÇÃO DE REQUISITOS .....	4
<b>3. DESENVOLVIMENTO .....</b>	<b>6</b>
3.1 CONCEÇÃO .....	6
3.2 ARQUITETURA DA SOLUÇÃO .....	6
3.3 INTERFACE COM O UTILIZADOR .....	7
3.4 ESTRUTURAS DE DADOS.....	8
3.5 PRINCIPAIS FUNÇÕES .....	9
<b>4. APRESENTAÇÃO DE UMA SESSÃO DE UTILIZAÇÃO.....</b>	<b>10</b>
4.1 ENTRADA .....	10
4.2 EXEMPLO DE UTILIZAÇÃO DA APLICAÇÃO .....	ERROR! BOOKMARK NOT DEFINED.
<b>5. CONCLUSÕES.....</b>	<b>13</b>
<b>6. BIBLIOGRAFIA.....</b>	<b>14</b>

## 1. SUMÁRIO

O objetivo do presente trabalho consiste no desenvolvimento de uma aplicação destinada a suportar a gestão financeira familiar, através da visualização dos gastos diários e acumulados mensais de gastos nas seguintes áreas: alimentação, vestuário, deslocações, lazer e outros. A aplicação deverá ser intuitiva e simples de forma a facilitar a sua utilização.

Inicialmente foi feita uma esquematização da aplicação com objetivo de perceber quais as funcionalidades que devemos implementar e a partir daí obter todos os inputs necessários. Dito isto, enumerámos as seguintes funcionalidades:

1. Gerir os valores dos gastos diários e/ou do orçamento anual na aplicação através da consola.
2. Usar ficheiros texto para armazenar toda a informação gerida pela aplicação.
3. Apresentar gráficos para os dados introduzidos, por dia, por mês e por ano.
4. Permitir que o utilizador possa analisar um ou mais gráficos em simultâneo.
5. Utilizar animações na apresentação de alguns gráficos.

Tendo em conta o que foi referido nestes 5 pontos, o presente trabalho tem uma parte dedicada à construção de uma interface gráfica que irá interagir com o utilizador, e uma parte de cálculos que suporta a interface e todos os métodos usados por esta. Por fim será necessário realizar os testes necessários à usabilidade, coerência e funcionalidade de todas as implementações elaboradas.

O presente relatório está então dividido nas seguintes secções:

- **Introdução:** Apresentação do tema e análise de requisitos funcionais e não-funcionais.
- **Desenvolvimento:** Proposta de solução e discussão sobre os métodos e estruturas de dados utilizados.
- **Apresentação de uma Sessão de Utilização:** Apresentação da forma de entrada e das diferentes vias de utilização da aplicação com exemplos.
- **Conclusão:** Resumo do que foi feito e de como foi feito, dos resultados, das eventuais dificuldades encontradas e das melhorias e/ou perspetivas futuras.

## 2. INTRODUÇÃO

### 2.1 Visão

O presente trabalho tem como tema a elaboração de um programa, em ambiente 2D, que tire proveito das várias primitivas do OpenGL. O objetivo principal consiste no desenvolvimento de uma aplicação destinada a suportar a gestão financeira familiar, sendo então necessário a monitorização dos gastos nas áreas da alimentação, vestuário, deslocações, lazer, entre outros. O programa foi desenvolvido na IDE *Microsoft Visual Studio 2017* e a linguagem de programação utilizada foi o C com bibliotecas de OpenGL.

### 2.2 Análise e Especificação de Requisitos

Define-se um requisito como um comportamento que é esperado do sistema. Durante a fase de análise de requisitos, interpretámos o enunciado para registá-los e categorizá-los em dois tipos distintos: Requisitos funcionais e requisitos não funcionais.

Os requisitos funcionais definem que serviços devem ser fornecidos pelo sistema, que operações podem ser realizadas, quais as reacções a certas entradas, e como deve o estado das entidades constituintes do sistema mudar devido à ocorrência de certas actividades. Apresentamo-los organizados em forma de tabela, e a cada requisito atribuímos uma prioridade de 1 a 3 (onde 1 corresponde a prioridade elevada e 3 prioridade baixa):

Requisitos funcionais	Prioridade
As áreas da alimentação, vestuário, deslocações, lazer e outros deverão ter um orçamento anual com controlo mensal.	1
A aplicação deverá permitir a entrada do orçamento anual bem como da informação diária de gastos de cada uma das áreas, individualmente e aglomerados através de um ficheiro.	1
Toda a informação gerida na aplicação deve ser armazenada em ficheiros	1
O utilizador poderá visualizar e comparar os valores dos gastos diários e dos gastos acumulados mensais e anuais, através dos gráficos mais adequados.	1
Na entrada da aplicação, o utilizador deve ter uma perfeita consciência da sua situação atual através de uma representação gráfica adequada e expressiva	2
Deverá ser utilizada animação em pelo menos numa das visualizações de informação.	3

Um requisito qualitativo ou não funcional descreve alguma característica qualitativa que a solução do problema deve ter. Estes requisitos estão relacionados com diversos aspectos, da performance do software à sua segurança, capacidade de utilização, manutenibilidade, precisão, exactidão, custo de desenvolvimento, meta temporal para apresentação do produto final, entre outros. Estes são os requisitos não funcionais que considerámos:

- A aplicação deverá ser simples e intuitiva
- O programa deverá ser robusto e eficiente na implementação

### 3. DESENVOLVIMENTO

#### 3.1 Conceção

A aplicação foi concebida com o objetivo de não ser só simples, robusta e intuitiva, mas também com o objetivo de permitir a expansão caso seja necessária. Assim, as estruturas e os métodos utilizados devem ser generalizados para que a introdução de novas áreas onde existam gastos sejam fáceis de implementar.

#### 3.2 Arquitetura da solução

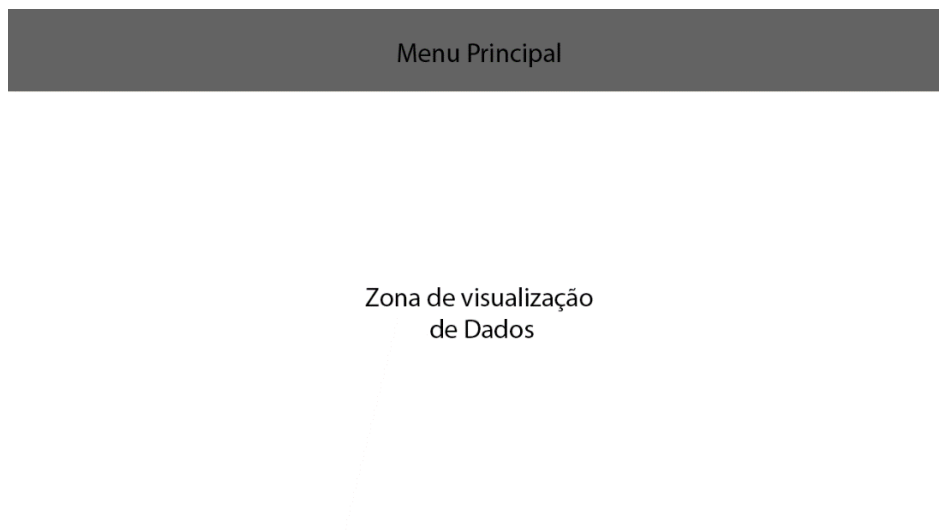
Tendo em conta os requisitos definidos e mencionados anteriormente, concebemos a solução para o problema constituída por: Criação de estruturas de dados que possam ser utilizadas por qualquer tipo de dados de finanças não contemplado neste projecto, sendo portanto generalizadas; Criação de estruturas que nos permitam armazenar em memória a posição  $(x,y)$  do rato, bem como os valores associados ao premir do mesmo. Foram também criadas estruturas para armazenar valores dos diferentes botões do nosso programa, que são usados para interagir com o programa.

Para facilitar uma utilização simples e intuitiva do utilizador, o menu principal está sempre visível, de forma a que o user consiga facilmente alterar o tipo de dados que pretende ver na área de análise de dados, diretamente a baixo do menu.

Para evitar perdas de dados e aumento do tempo de computação, usámos ficheiros para nos ajudar a armazenar todos os dados pertinentes às finanças da família.

### 3.3 Interface com o utilizador

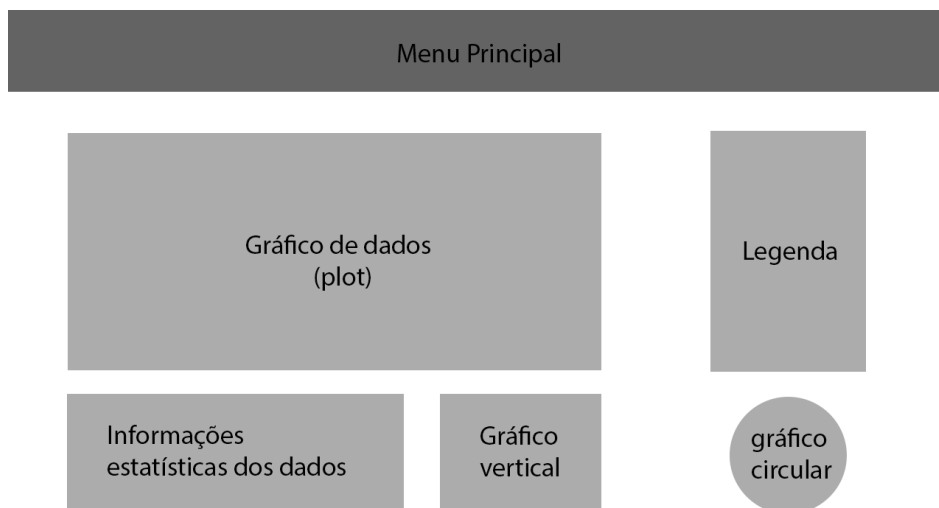
A interface programa-user foi pensada de forma a manter o uso da aplicação o mais intuitivo e simples possível, sem comprometer a análise de dados feita pelo mesmo. Logo, decidimos desde início ter um menu principal permanentemente visível pelo utilizador.



*Figura 1- Mockup da GUI*

No menu principal Incluímos botões que quando premidos levam o utilizador para a “tab” de observação do dado escrito no botão: Alimentação, Vestuário, Deslocações, Lazer e Outros.

Quando entramos numa aba de observação de dados, a interface é a seguinte:



No plot graphic temos também botões para mudar a vista do gráfico para mensal, anual ou em década. As informações estatísticas que o programa nos apresenta são os seguintes valores Mensais, anuais ou da década: Média, mínimo, máximo e desvio padrão.

A inserção de novos dados pode ser feita com um submenu que emerge quando clicamos com o botão direito do rato em qualquer sítio do programa.

### 3.4 Estruturas de dados

Além de estruturas fulcrais ao funcionamento de componentes do programa como rato e botões, usamos uma estrutura *Familia* para armazenar todo o tipo de dados acerca das finanças desta. Esta é constituída por um *ID*, um inteiro para a identificar, bem como um nome. Os elementos mais importantes desta struct são os ponteiros para arrays que nela são constituintes: *orcamentoAnual*, *gastoTotalDia*, *custoAlimentacaoDia*, *custoVestuarioDia*, *custoDeslocacoesDia*, *custoLazerDia* e *custoOutrosDia*. Estes ponteiros apontam todos para zonas de memória onde armazenamos, através da leitura dos ficheiros os dados diários dos gastos da família nos diversos tipos pedidos. A struct tem também um booleano *isEmpty* que vira FALSE quando é feita a leitura dos dados. Desta estrutura é inicializada 1 família, a qual acedemos aos dados durante toda a sessão de utilização.

Além desta struct utilizamos uma outra struct *Auxiliar* que nos auxilia a guardar os dados estatísticos calculados, e que é constituída por 8 ponteiros para arrays e 4 valores de vírgula flutuante que armazenam os valores máximos, mínimos, médios e o desvio padrão, cada um num intervalo de tempo mensal, anual ou em década: *MaxMensal*, *MaxAnual*, *MaxSempre*, *MinMensal*, *MinAnual*, *MinSempre*, *MedMensal*, *MedAnual*, *MedSempre*, *DevMensal*, *DevAnual*, *DevSempre*. Desta estrutura são inicializadas 6 tabelas auxiliares que guardam os valores estatísticos para cada tipo de gasto: alimentação, vestuário, deslocações, Lazer, outros e total.

São também utilizadas algumas variáveis globais para gestão de menus, cores e outras funcionalidades intrínsecas do programa.



### 3.5 Principais funções

***void MouseButton(int button, int state, int x, int y)*** – Associada a `glutMouseFunc()`. Esta função é chamada quando clicamos no botão do rato, de forma a permitir executar ações que sejam necessárias aquando o premir do botão. Assim, a função avalia onde foi feito o clique e guarda este ponto para executar certa função.

***void MouseMotion(int x, int y)*** – Esta função, associada a `glutMotionFunc()`, é chamada quando o rato é movido na janela ou quando é pressionado um dos botões do rato.

***void MousePassiveMotion(int x, int y)*** – Associada a `glutPassiveMotionFunc()`, esta função apenas é chamada quando o rato é movido e não se pressiona nenhum botão do mesmo.

***void buttonpress(botao \*b, int x, int y)*** , ***checkboxbutton(botao \*b, int x, int y)*** e ***buttonRelease(botao \*b, int x, int y)*** – Estas funções são chamadas quando pressionamos e largamos um determinado botão desenhado no ecrã, e executam as tarefas do mesmo. A segunda função verifica se temos o rato por cima do botão com auxílio da função `isHoveringButton`, para mudar a cor do mesmo, de forma a dar feedback ao utilizador.

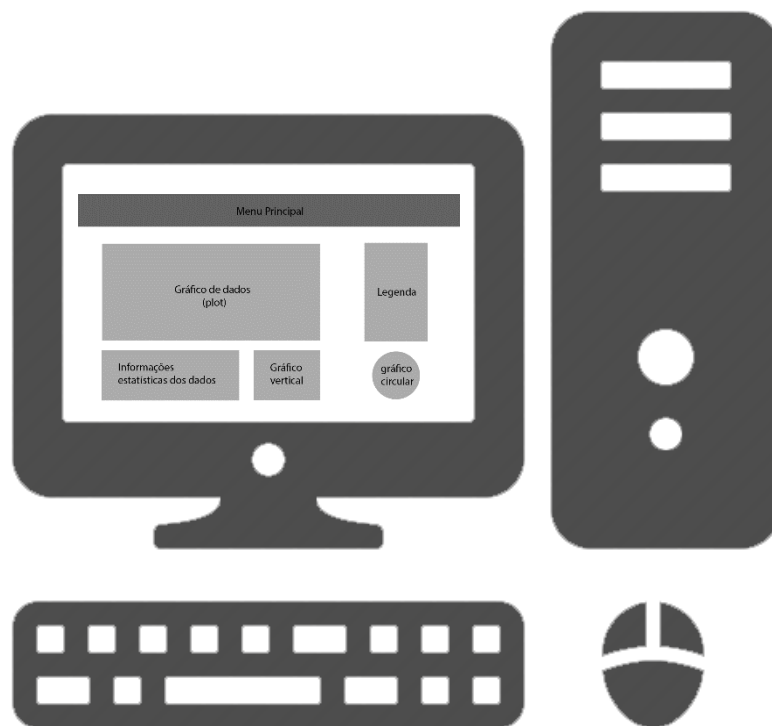
***void grafico\_barra(int Dia)*** e ***void grafico\_circulo(int Dia)*** – Estas funções desenham 2 gráficos no ecrã, um de barra vertical e um circular, respectivamente. Os dados utilizados para desenhar estes gráficos são acedidos no dia e é feita uma representação destes relativamente ao gasto total no mesmo espaço temporal.

***void plot(int x\_pos, int y\_pos, float \*TabelaDados, int min\_value, int max\_value, float cor[])*** – Esta função permite desenhar um gráfico de linhas no centro do ecrã. Os dados utilizados para desenhar este gráfico são obtidos no início da aplicação através da leitura dos ficheiros texto que criámos usando o Matlab. ***void desenharEixos(int x\_pos, int y\_pos)***- Esta função permite desenhar os eixos Ox e Oy, que vão ser usados para desenhar o gráfico de linhas.

## 4. APRESENTAÇÃO DE UMA SESSÃO DE UTILIZAÇÃO

### 4.1 Entrada

O programa tem como maneiras de introduzir dados o teclado via linha de comandos, e o rato como principal forma de navegar pelos menus. Para inserir dados ou mudar a cor dos gráficos existentes, o utilizador pode simplesmente carregar no botão direito do rato e fazer as alterações, que surgem nos submenus existentes.

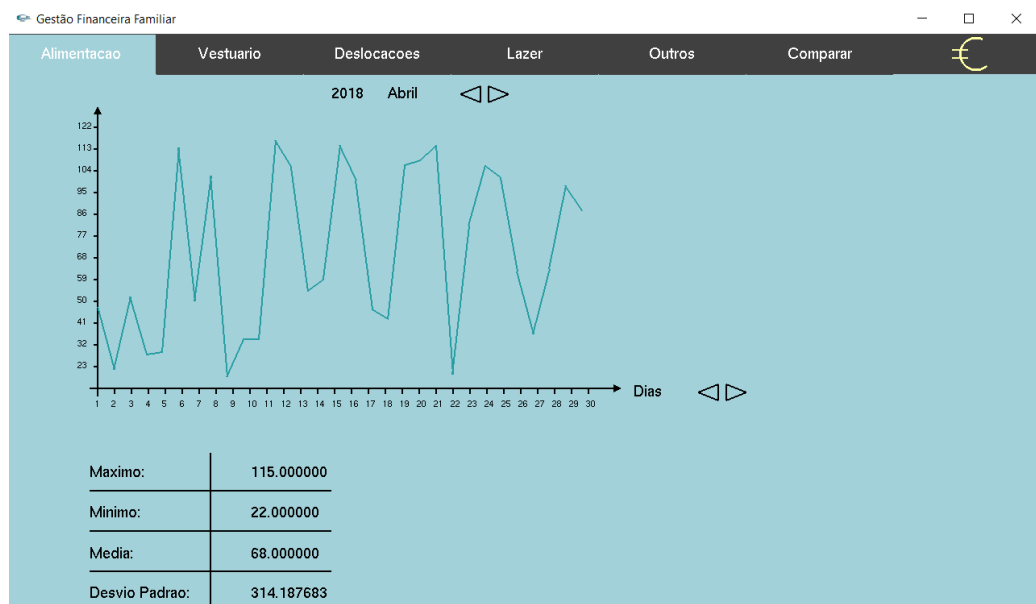


## 4.2 Exemplo de Utilização da Aplicação

Quando inicializada a aplicação, é apresentada ao utilizador a seguinte janela:

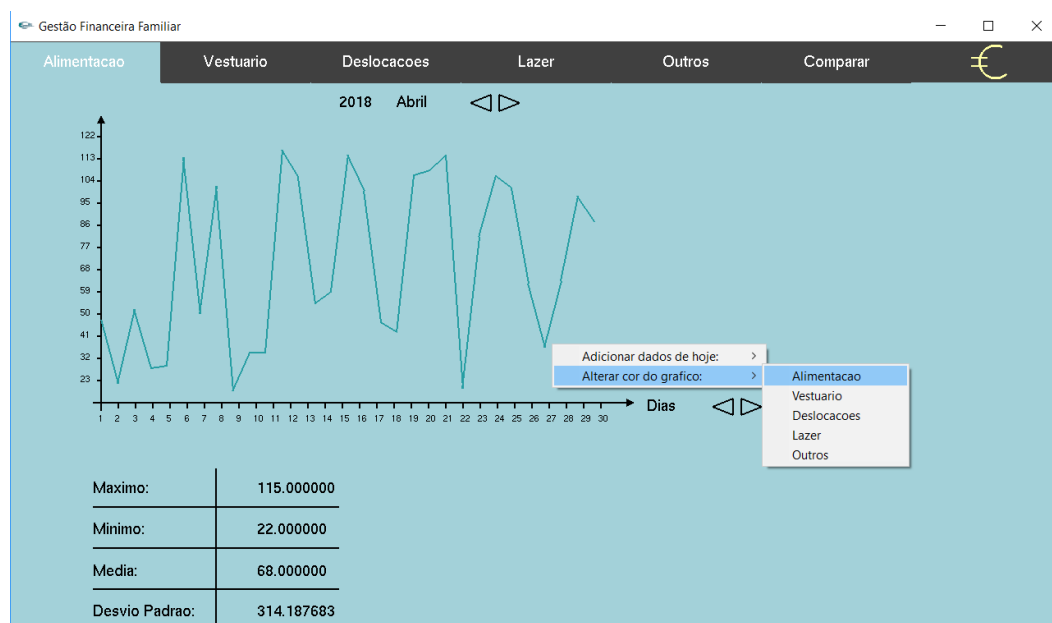


Aqui, o utilizador pode escolher qualquer uma das opções do menu principal, disponibilizado em cima. Em cada um dos menus consegue observar informações acerca desse tipo de dados, diariamente, mensalmente ou por década. Fica em seguida uma imagem que reflete uma das janelas-tipo do que é apresentado ao user quando este carrega num botão do menu:



Nesta janela podemos observar o gráfico dos dados acerca do tipo de dados escolhido, bem como mudar o tipo de visualização(dias,meses,anos) e o mês que pretendemos observar.

Podemos também aceder a dois submenus, como exemplifica a seguinte imagem:



Por fim, podemos carregar na tab “Comparar” onde podemos observar todos os valores e gráficos de todos os tipos de gastos, e evidenciar qual aquele que é mais predominante(onde se gasta mais), através de gráficos percentuais em relação ao gasto total somado por todos os tipos, gráficos estes que são em barras e circulares.



## 5. CONCLUSÕES

Consideramos que, embora o trabalho cumpra os requisitos definidos e sirva para o efeito desejado, muito pode ainda ser melhorado e adicionado de forma a aperfeiçoar a aplicação.

Foi um trabalho que levantou bastantes desafios e levou a diversas horas de debugging e pesquisa em temas desde OpenGL a algoritmos de cálculos. Foi uma boa forma de consolidar conhecimentos leccionados nas aulas práticas.

Admitimos também que o programa não é perfeito, e talvez por sobreposição de entregas (especialmente no ramo de computadores do Mestrado Integrado de Engenharia Electrotécnica e Computadores) semanais não tenhamos tido tempo de o entregar num estado em que achemos estar “a 100%”. Apesar disto, estamos satisfeitos com os resultados finais e achamos que a aplicação serve o propósito pedido no 1º trabalho prático da cadeira.

## 6. BIBLIOGRAFIA

José Carlos Teixeira, *Transformações Geométricas Afins*, Disciplina de Computação Gráfica, Departamento de Engenharia Eletrotécnica e de Computadores, Faculdade de Ciências e Tecnologia da Universidade de Coimbra, 2018

José Carlos Teixeira, *Redimensionamento Menus*, Disciplina de Computação Gráfica, Departamento de Engenharia Eletrotécnica e de Computadores, Faculdade de Ciências e Tecnologia da Universidade de Coimbra, 2018

José Carlos Teixeira, *Interação Teclado Rato*, Disciplina de Computação Gráfica, Departamento de Engenharia Eletrotécnica e de Computadores, Faculdade de Ciências e Tecnologia da Universidade de Coimbra, 2018

José Carlos Teixeira, *Utilização de visores*, Disciplina de Computação Gráfica, Departamento de Engenharia Eletrotécnica e de Computadores, Faculdade de Ciências e Tecnologia da Universidade de Coimbra, 2018

José Carlos Teixeira, *Geração de primitivas gráficas*, Disciplina de Computação Gráfica, Departamento de Engenharia Eletrotécnica e de Computadores, Faculdade de Ciências e Tecnologia da Universidade de Coimbra, 2018