



Trabalho Prático Final

Rodrigo Saraiva - 51738

Rafael Antunes - 51950

Trabalho Prático Final para
Ecosistema Web Movei e Nuvem
(2º Semestre de Informática Web, Movei e na Nuvem)

Professor: Prof. Tiago M. C. Simões

Junho de 2024

Agradecimentos

Gostaríamos de expressar a nossa sincera gratidão à Universidade da Beira Interior por fornecer o ambiente académico e os recursos necessários para a realização deste trabalho prático.

E em particular, agradecemos profundamente ao nosso professor, Tiago M. C. Simões, que sempre se disponibilizou para ajudar e cujas aulas foram cruciais para a nossa aprendizagem.

Resumo

Na Unidade Curricular (UC) de Ecossistema Web, Móvel e Nuvem foram-nos lecionados vários tópicos que aplicámos neste trabalho, entre eles sistemas numéricos, como fazer operações com estes e como desenvolver uma aplicação web, sendo o nosso objetivo integrar o que aprendemos numa aplicação web, dando-lhe uma estética diferente com o auxílio de HyperText Markup Language (HTML) e Cascading Style Sheets (CSS).

Tendo em conta que já tínhamos a lógica de cada funcionalidade feita, bastou integrar isso numa página web, usando uma biblioteca de *Python* lecionada em aula, o *Flask* como framework para backend, e uma interface interativa com o HTML e CSS como frontend, que já se supunha fazer parte do nosso conhecimento.

O relatório explora a composição da aplicação, de cada uma das funcionalidades e a implementação destas detalhadamente, usando excertos de código, imagens e diagramas, acompanhados por um breve resumo. Termina com um resumo do processo de desenvolvimento realizado ao longo do trabalho até chegarmos ao produto final.

Acrónimos

UC Unidade Curricular

HTML HyperText Markup Language

CSS Cascading Style Sheets

Índice

1. Introdução

- 1.1. Objetivos do Trabalho
- 1.2. Abordagem
- 1.3. Organização do Documento

2. Tecnologias e Ferramentas Utilizadas

- 2.1. Tecnologias
- 2.2. Ferramentas

3. Execução

- 3.1. Página Inicial
- 3.2. Criar Tabela de Gray
- 3.3. Conversão Entre Bases Numéricas
- 3.4. Operações entre Bases Numéricas

4. Detalhes da Implementação

- 4.1. Estrutura do Trabalho
- 4.2. Script Batch
- 4.3. Interface
- 4.4. Criar Tabela de Gray
- 4.5. Conversão Entre Bases Numéricas
- 4.6. Operações entre Bases Numéricas

5. Conclusões e Trabalho Futuro

5.1 Resumo dos Resultados

5.2 Conclusão Final

5.3 Trabalho Futuro

1 Introdução

Na UC de Ecossistema Web, Móvel e Nuvem, exploram-se temas vitais para a compreensão de todos os aspectos aplicados neste trabalho. O mais fundamental para este trabalho foi a framework em *Python* utilizada para o backend, o *Flask*, tendo os outros tópicos sido previamente lecionados nesta ou noutras UC da mesma temática.

O docente da UC atribuiu-nos a realização deste trabalho prático final visando colocar em prática o lecionado ao implementar as funcionalidades feitas no trabalho prático intermédio numa aplicação web, estas sendo: *I*) Criar tabelas de Gray de n bits (introduzidos pelo utilizador); *II*) Realizar operações de conversão entre sistemas numéricos (binário, decimal e hexadecimal); e *III*) Realizar operações aritméticas de adição e subtração, também nestes.

1.1 Objetivos do Trabalho

Os principais objetivos deste trabalho são conseguir desenvolver uma aplicação web capaz de, através de um

menu interativo, criar tabelas de Gray com n bits (introduzidos pelo utilizador), converter entre todos os sistemas numéricos lecionados em aula e fazer operações entre estes (adições e subtrações).

1.2 Abordagem

Antes de abordar o código diretamente, diagnosticámos o que precisávamos de fazer e, conforme o que nos foi lecionado, assimilámos resoluções para cada objetivo que teríamos de cumprir, pesquisando caso fosse preciso.

Após o diagnóstico, optámos por distribuir as responsabilidades do trabalho entre nós, ficando tanto na fase de desenvolvimento de software quanto na elaboração do relatório, cada um encarregue por metade das tarefas. Durante este processo, estabelecemos uma dinâmica colaborativa, na qual oferecemos assistência mútua e corrigimos eventuais inconsistências conforme necessário.

1.3 Organização do Documento

Conforme indicado no Índice, o documento está dividido em 5 partes principais. A Introdução apresenta uma visão geral do relatório, seus objetivos, nossa abordagem e organização. Em segundo lugar, abordamos as Tecnologias e Ferramentas Utilizadas no trabalho, explicando como as utilizamos e porquê. O terceiro capítulo trata da Execução do programa, onde demonstramos o funcionamento normal, apresentando as interfaces, exemplos de utilização de cada funcionalidade e os resultados obtidos.

No penúltimo capítulo, detalhamos a Implementação, explicando como o programa foi desenvolvido, oferecendo insights sobre o processo de desenvolvimento de cada funcionalidade. Por fim, nas Conclusões e Trabalho Futuro, reunimos os resultados, apresentamos uma conclusão final e discutimos possíveis trabalhos futuros.

2 Tecnologias e Ferramentas Utilizadas

Neste capítulo apresentamos quais ferramentas e tecnologias foram usadas ao longo do trabalho, mostrando as vantagens e os propósitos de termos usado cada uma.

2.1 Tecnologias

Como o nosso trabalho anterior baseava-se inteiramente em *Python*, decidimos adaptar tudo a essa linguagem, escolhendo tecnologias que se interligassem com ela e que oferecessem as melhores qualidades possíveis.

- ***Python e Flask*** -O *Python* é uma linguagem de programação de alto nível, amplamente reconhecida pela sua simplicidade, legibilidade e versatilidade. O *Flask* é uma biblioteca de *Python* minimalista projetada para facilitar o desenvolvimento de aplicações web. É leve e flexível, perfeito para o nosso trabalho que não requer algo muito complexo

e que está em *Python*, condizendo com o que já temos do trabalho prático intermediário.

- **HTML e CSS** - O HTML é uma linguagem de marcação usada para estruturar o conteúdo de uma página web, utilizada em conjunto com o CSS, este para controlar a aparência e o layout dos elementos definidos pelo HTML, responsável pela estética da aplicação web. Ambos são usados devido à sua presença destacada na produção de aplicações web.
- **Batch** - O Batch é uma linguagem de *scripting* usada em ambientes Windows, onde seu propósito é criar scripts com comandos internos que facilitam uma ou várias tarefas. Nós utilizamos o Batch para facilitar a instalação do ambiente virtual, que decidimos não incluir no GitHub devido ao espaço que ocupa, ajudando a executar a aplicação web mais facilmente após a instalação do ambiente virtual.

2.2 Ferramentas

Como ferramentas, utilizamos aquilo que ao longo do tempo percebemos que nos seria muito útil e facilitaria o

trabalho a longo prazo, aumentando assim a nossa eficiência e produtividade.

- **Git e Github** - O *Git* é um sistema de controlo de versões distribuído gratuito usado para lidar de forma rápida e eficiente com trabalhos pequenos ou grandes. Já o *Github* serve como uma nuvem para o *Git*, auxiliando no armazenamento online. No nosso trabalho, usamos ambos para conseguirmos estar a trabalhar na mesma versão do projeto, evitando problemas de colisões de versões que poderíamos ter, contribuindo para a cooperação na realização do trabalho.

Repositório

- **L^AT_EX e Overleaf** - O L^AT_EX é uma linguagem de marcação com uma sintaxe poderosa e extensível que permite a criação de documentos com formatação profissional, utilizada na preparação de documentos técnicos e científicos. O *Overleaf* é uma plataforma de edição de documentos L^AT_EX colaborativa baseada na web, que oferece uma interface intuitiva para criar e editar documentos L^AT_EX em tempo real e em grupo. Esta plataforma acabou por nos ajudar muito no trabalho devido à disponibilidade de cooperação de alta qualidade que oferece, simplificando a utilização do L^AT_EX na realização deste relatório, sendo uma escolha ideal para o trabalho.

- **Visual Studio Code** - O Visual Studio Code (VS Code) é um editor de código-fonte leve e poderoso, desenvolvido pela Microsoft, com grande versatilidade, oferecendo funcionalidades como destaque de sintaxe, autocompletar, depuração integrada, entre outros. Apesar de ser uma opção pessoal, ambos o usamos na criação do código para este trabalho pela sua versatilidade e funcionalidades úteis.
- **Figma** - O Figma é uma ferramenta de design baseada na web, amplamente utilizada na criação de interfaces de usuário e experiências de usuário. Permite a colaboração em tempo real e é bastante intuitiva, sendo útil para a prototipagem de aplicações na web. Escolhemos utilizá-la, apesar de não ser o foco principal do trabalho, devido à sua eficácia e facilidade de uso.

3 Execução

A estrutura da interface do utilizador foi concebida para ser clara e eficiente, garantindo que os utilizadores possam aceder rapidamente às funcionalidades desejadas. Utilizamos tecnologias web modernas para desenvolver o frontend, assegurando uma experiência de utilizador fluida e responsiva.

3.1 Página Inicial

É a primeira página com que o utilizador se encontra, simples e intuitiva, tem a identificação do trabalho, os créditos e instruções simples para se começar a utilizar, com um menu de navegação no topo presente em todas as páginas com as opções: *I) Home; II) Gray; III) Conversão e IV) Operações.*



Trabalho Prático Final

Ecossistema Web, Móvel e na Nuvem

Página Principal
Use os botões acima para começar

Rodrigo Saraiva
51738

Professor
Tiago M. C. Simões

Rafael Antunes
51950

Figura 1: Página Inicial

3.2 Criar Tabela de Gray

Ao seleccionar no menu o botão "Gray", vamos de encontro à segunda página, que contém a primeira funcionalidade, a de criar tabelas de Gray. Esta página serve de menu para o utilizador introduzir de quantos bits quer que seja criada uma Tabela de Gray, introduzindo na linha a tracejado.

Ao introduzirmos um número de 0 a 10, a página vai mudar para outra semelhante que indica de quantos bits é a tabela, um botão de voltar, que volta para a página anterior e em baixo a tabela de Gray pedida.

Home
Gray
Conversão
Operações

Tabelas de Gray

de n bits

Indique de quantos bits quer a tabela

Bits

Figura 2: Menu das Tabelas de Gray

Home
Gray
Conversão
Operações

Tabelas de Gray

de 4 bits

Numero	Codigo
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111
6	0101
7	0100
8	1100
9	1101
10	1111

Figura 3: Resultado da Criação da Tabela de Gray

Essa abordagem centrada no utilizador visa minimizar a possibilidade de erros e tornar o processo de criação de tabelas de Gray o mais fluido e acessível possível.

3.3 Conversão entre Bases Numéricas

A seguinte funcionalidade está disponível no botão "Conversão" e é a de conversão entre bases numéricas, uma funcionalidade que permite a conversão entre as bases binária, decimal e hexadecimal.

A interface solicita ao utilizador que insira a base de origem e a base de destino para a conversão, sendo estas opções rotuladas e dispostas de forma a minimizar a confusão. O utilizador deve introduzir o número na base de origem e, em seguida, a interface exibirá o resultado na base de destino após a interação com o botão "converter".

The screenshot shows a web application interface for converting between numerical bases. At the top, there is a navigation bar with four buttons: 'Home', 'Gray', 'Conversão' (which is highlighted in a darker gray), and 'Operações'. Below the navigation bar, the main heading is 'Conversão entre Bases Numéricas' in a large, bold, black font. Underneath this heading, in a smaller font, is the text 'Binária, Decimal e Hexadecimal'. The main content area contains the text 'Quer Converter de' followed by a dropdown menu, then 'para' followed by another dropdown menu. Below these dropdowns are two input fields: 'Número:' and 'Resultado:'. At the bottom of this section is a gray button labeled 'Converter'.

Figura 4: Conversão entre Bases Numéricas

Esta abordagem visa proporcionar uma experiência de

utilizador eficiente e sem complicações, permitindo que os utilizadores realizem conversões entre diferentes bases numéricas de forma rápida e precisa.

3.3.1 Operações

Ao seleccionar o último botão da barra de navegação, vê-se a terceira e última funcionalidade, a de operações aritméticas entre as mesmas bases numéricas da funcionalidade anterior, a binária, a decimal e hexadecimal.

Para a utilização normal da funcionalidade, o utilizador selecciona a base pela qual quer fazer a operação aritmética e de seguida insere os números conforme a base.

Home Gray Conversão **Operações**

Operações entre Bases Numéricas

Somas e Subtrações entre Binária, Decimal e Hexadecimal

Escolha a base: _____

_____ = _____

Calcular

Figura 5: Interface de Operações Aritméticas

Esta é a última funcionalidade disponível na nossa aplicação web, passando agora para um capítulo que fala mais detalhadamente sobre a parte lógica destas.

4 Detalhes da Implementação

A fundação do software por trás da aplicação web foi concebida de forma a ser o mais eficiente possível e sem problemas, para oferecer ao utilizador a melhor experiência possível. Para isso, utilizamos o *Flask*, referido nas tecnologias. Neste capítulo, iremos explicar detalhadamente como implementamos as funcionalidades na nossa aplicação.

4.1 Estrutura do Trabalho

Como apresentado a seguir, organizamos nossa diretoria de trabalho conforme as configurações do *Flask*, deixando as páginas web (HTML) dentro da diretoria *templates*, os arquivos de estilização (CSS) dentro de *css*, e os arquivos de scripts na diretoria principal.

Desses *scripts*, encontram-se dois deles escritos em *Python*, sendo fundamentais para a funcionalidade do trabalho. O *functions.py* contém apenas as funcionalidades, enquanto o *app.py* contém a sua interligação com o resto da aplicação web, podendo ser chamado de *backend*. O outro script presente nessa

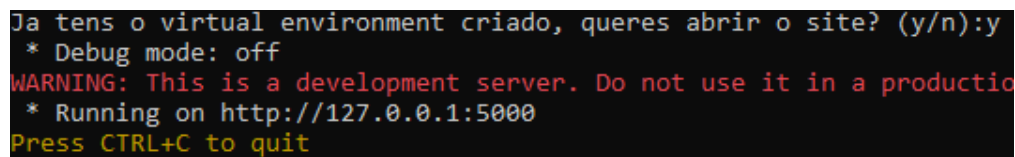
mesma diretoria é o script em *Batch*, *start.bat*.

```
ProjetoFinalEWMN
├── static
│   └── css
│       ├── reset.css
│       └── styles.css
├── templates
│   ├── homepage.html
│   ├── gray.html
│   ├── conversao.html
│   ├── operacoes.html
│   └── result.html
├── app.py
├── functions.py
└── start.bat
```

4.2 Script Batch

Apesar de não ser obrigatório para o nosso trabalho, decidimos compor este script *Batch* justo pela utilidade que nos oferece. Este serve para como ao importarmos atualizações do *Github*, o trabalho vinha sem o ambiente virtual porque escolhemos ausenta-lo lá pelo espaço que

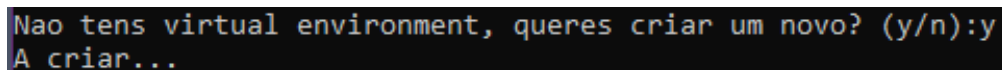
ocupa, preferindo que este seja criado localmente por este *script Batch*.



```
Ja tens o virtual environment criado, queres abrir o site? (y/n):y
* Debug mode: off
WARNING: This is a development server. Do not use it in a production
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Figura 6: Interface da Inicialização da Aplicação pelo Script Batch

Este *script Batch*, quando é executado, verifica se o ambiente virtual existe. Se não existir, ele pergunta se deseja instalá-lo. Caso a resposta seja negativa, o script termina. Se a resposta for positiva, ele procede com a instalação do ambiente virtual. Após a instalação, o script pergunta se deseja iniciar a aplicação web. Se todos os requisitos estiverem satisfeitos, a operabilidade é idêntica à anterior. Caso o ambiente virtual já esteja instalado, o script pula o primeiro passo mencionado e pergunta diretamente se deseja iniciar a aplicação web.



```
Nao tens virtual environment, queres criar um novo? (y/n):y
A criar...
```

Figura 7: Interface da Criação do Ambiente Virtual pelo Script Batch

4.3 Interface

A interface toda da aplicação foi toda desenvolvida em HTML com a estilização do CSS, com mínimas funções para auxiliar as funcionalidades principais.

4.4 Criar Tabela de Gray

Para a interligação da logica das tabelas de gray com a pagina web usamos argumentos da pagina, transmitidos pelo url, recebidos posteriormente pelo backend e transmitidos para a página para fazer as devidas alterações. De seguida está o código implementado nesta iniciativa.

```
@app.route("/gray_resultado")
def gray_resultado():
    bits = request.args.get("bits")
    if not bits:
        bits = 0
    tabela = criar_tabela_gray(int(bits))
    if tabela[0] == "erro":
        print("erro")
    else:
        return render_template("gray_resultado.html", resultado=tabela)
```

Listing 1: Código - Criar Tabela de Gray

4.5 Conversão entre Bases Numéricas

Nesta interligação usou-se um método chamado post, ou seja, ele no início da página esta desativado, pois não tem nenhum valor, mas quando o utilizador insere valores e submete este interage, recebendo os numeros, fazendo os calculos e retornando. Na interligação entre estas bases teve-se de ter cuidado com a validação, devolvendo erros caso houvesse algo mal escrito.

```
@app.route("/conversao", methods=["GET", "POST"])
def conversao():
    resultado = ''
    if request.method == 'POST':
        base_origem = request.form["base_origem"]
        base_destino = request.form["base_destino"]
        valor = request.form["valor"]

        if base_origem == 'bin':
            if not validar_binario(valor):
                resultado="erro"
            elif base_destino == 'dec':
                resultado = binario_para_decimal(valor)
            elif base_origem == 'bin' and base_destino == 'hex':
                resultado = binario_para_hexadecimal(valor)
            else:
                resultado = valor
        elif base_origem == 'dec':
            if not validar_decimal(valor):
                resultado="erro"
            elif base_destino == 'bin':
                resultado = decimal_para_binario(float(valor))
            elif base_destino == 'hex':
                resultado = decimal_para_hexadecimal(float(valor))
            else:
                resultado = valor
        elif base_origem == 'hex':
            if not validar_hexadecimal(valor):
                resultado="erro"
            elif base_destino == 'bin':
                resultado = hexadecimal_para_binario(valor)
            elif base_destino == 'dec':
```



```

        resultado = hexadecimal_para_decimal(valor)
    else:
        resultado = valor
    else:
        return render_template("conversao.html", resultado=
"erro")

    return render_template("conversao.html", resultado=
resultado)

```

Listing 2: Código - Conversão entre Bases Numéricas

4.6 Operações Aritméticas entre Bases Numéricas

Na funcionalidade das operações aritméticas, semelhantemente como nas conversões usou-se também o método post e fez-se a validação dos números.

```

@app.route("/operacoes", methods=["GET", "POST"])
def operacoes():
    resultado = ''
    if request.method == 'POST':
        base = request.form["base"]
        operacao = request.form["operacao"]
        valor1 = request.form["valor1"]
        valor2 = request.form["valor2"]

        if base == 'bin':
            if not validar_binario(valor1) or not
validar_binario(valor2):
                resultado="erro"
            elif operacao == '+':
                resultado = binario_para_decimal(valor1) +
binario_para_decimal(valor2)
                resultado = decimal_para_binario(resultado)
            elif operacao == '-':
                resultado = binario_para_decimal(valor1) -
binario_para_decimal(valor2)
                resultado = decimal_para_binario(resultado)

```

```

        elif base == 'dec':
            if not validar_decimal(valor1) or not
validar_decimal(valor2):
                resultado="erro"
            elif operacao == '+':
                resultado = float(valor1) + float(valor2)
            elif operacao == '-':
                resultado = float(valor1) - float(valor2)
        elif base == 'hex':
            if not validar_hexadecimal(valor1) or not
validar_hexadecimal(valor2):
                resultado="erro"
            elif operacao == '+':
                resultado = hexadecimal_para_decimal(valor1) +
hexadecimal_para_decimal(valor2)
                resultado = decimal_para_hexadecimal(resultado)
            elif operacao == '-':
                resultado = hexadecimal_para_decimal(valor1) -
hexadecimal_para_decimal(valor2)
                resultado = decimal_para_hexadecimal(resultado)

    return render_template("operacoes.html", resultado=
resultado)

```

Listing 3: Código - Operações Aritméticas entre Bases Numéricas

5 Conclusões e Trabalho Futuro

Neste último capítulo exploramos os resultados obtidos, tirando conclusões sobre eles e discutindo o que podia ser melhorado.

5.1 Resumo dos Resultados

Ao concluirmos o trabalho ficamos satisfeitos com o resultado final, tem uma boa interface, uma boa funcionalidade, é muito mais interativo que o trabalho intermédio e ficou ao nosso gosto, mas não queira dizer que haja espaço para melhoria.

5.2 Conclusão Final

Como referido anteriormente a nossa prestação e satisfação sobre este trabalho foi positiva, mas não queria dizer que houvesse espaço para melhoria, na nossa opinião, apesar de não ser muito impactante, ha pontos onde o nosso trabalho pode ser melhorado.

5.3 Trabalho Futuro

Como trabalho futuro ficam aqueles pontos que nos falharam, tal como a limpeza e clareza do nosso código, a melhor validação e forma como lidamos com os erros,

dando mais indicações no caso de estes acontecerem, e entre outros pontos que agora não se revelaram.

6 Bibliografia

Documentação de L^AT_EX :

<https://www.overleaf.com/learn>

Documentação de *Flask*:

<https://flask.palletsprojects.com/en/3.0.x/>