

**RELATÓRIO DE PROJETO DE FUNDAMENTOS DE PROGRAMAÇÃO**

Curso Técnico Superior Profissional em Programação de Sistemas de Informação

<<nome >> nº …  
<<nome >> nº ..

Relatório realizado sob a orientação do Professor xxxx.

janeiro de 2024

**Resumo**

Durante este trabalho, iremos explicar como foi realizado o projeto requisitado pela disciplina de Fundamentos de Programação no ano letivo 2022/2023, utilizando os conhecimentos adquiridos ao longo do semestre. O objetivo principal é concluir todas as tarefas solicitadas de maneira simples, evitando repetição de código e seguindo as boas práticas. Abordaremos as estruturas de dados utilizadas, variáveis e constantes principais, funcionalidades implementadas e suas respetivas funções, funcionalidades pendentes, validação dos dados de entrada e realização de testes na aplicação. Por fim, apresentaremos os aspetos importantes a serem destacados e uma análise crítica sobre o projeto.

**Índice**

[1. Índice de Figuras IV](#_Toc124522915)

[2. Introdução 1](#_Toc124522916)

[1. Constituição da estrutura do programa 2](#_Toc124522917)

[1.1 Definições de pré-processamento 2](#_Toc124522918)

[1.2 Constantes 2](#_Toc124522919)

[1.3 Declaração de Estruturas 3](#_Toc124522920)

[1.4 Declaração de Funções 4](#_Toc124522921)

[1.5 Variáveis principais 5](#_Toc124522922)

[2. Funcionalidades implementadas e respetivas funções 6](#_Toc124522923)

[2.1 Registar e consultar os dados dos participantes 6](#_Toc124522924)

[2.2 Registar e consultar os dados das atividades 8](#_Toc124522925)

[2.4 Registar e consultar os dados das inscrições 13](#_Toc124522926)

[2.4 Estatísticas - Número de atividades realizadas por cada associação 15](#_Toc124522927)

[2.5 Estatísticas - Percentagem de inscrições por escola 16](#_Toc124522928)

[2.6 Estatísticas - Valor total das inscrições entre duas datas (horizonte temporal) por tipo de atividade 17](#_Toc124522929)

[2.7 Guardar e ler de ficheiros binários os dados da aplicação 18](#_Toc124522930)

[3. Funcionalidades por implementar 20](#_Toc124522931)

[4. Validação de dados de entrada 21](#_Toc124522932)

[5. Realização de testes na aplicação 26](#_Toc124522933)

[6. Análise crítica e proposta de melhorias 33](#_Toc124522934)

[7. Conclusão 34](#_Toc124522935)

[8. Referências bibliográficas 35](#_Toc124522936)

# Índice de Figuras

[Figura 1- Definições de pré-processamento 2](file:///C:\Users\rafa_\Documents\TeSP_PSI_22_23\1_Ano\F_Programação\Projeto\Relatorio_Projeto.docx#_Toc124522853)

[Figura 2-Constantes 2](file:///C:\Users\rafa_\Documents\TeSP_PSI_22_23\1_Ano\F_Programação\Projeto\Relatorio_Projeto.docx#_Toc124522854)

[Figura 3-Declaração de Estruturas 3](#_Toc124522855)

[Figura 4-Declaração de funções 4](#_Toc124522856)

[Figura 5-Variáveis principais 5](#_Toc124522857)

[Figura 6-Registo participantes 6](#_Toc124522858)

[Figura 7-menu escola 7](#_Toc124522859)

[Figura 8-Validar NIF 7](#_Toc124522860)

[Figura 9-Validar telefone 8](#_Toc124522861)

[Figura 10-Menu consulta participantes 8](#_Toc124522862)

[Figura 11-Registo atividades 9](#_Toc124522863)

[Figura 12-Validar ano 9](#_Toc124522864)

[Figura 13-Validar mês 10](#_Toc124522865)

[Figura 14-Validar dia 10](#_Toc124522866)

[Figura 15-Validar hora 11](#_Toc124522867)

[Figura 16-Validar minutos 11](#_Toc124522868)

[Figura 17-Menu atividade 12](#_Toc124522869)

[Figura 18-Validar valor 12](#_Toc124522870)

[Figura 19-Menu consulta atividades 13](#_Toc124522871)

[Figura 20-Registo inscrições 13](#_Toc124522872)

[Figura 21-Validar inscrição atividade 14](#_Toc124522873)

[Figura 22-Validar inscrição participante 14](#_Toc124522874)

[Figura 23-Data hora inscrição 15](#_Toc124522875)

[Figura 24-Consulta Inscrição 15](#_Toc124522876)

[Figura 25-Atividade por associação 16](#_Toc124522877)

[Figura 26-Inscrições por escola 17](#_Toc124522878)

[Figura 27-Inscrições entre duas datas 18](#_Toc124522879)

[Figura 28-Calculo valor da inscrição 18](#_Toc124522880)

[Figura 29- Gravar dados 18](#_Toc124522881)

[Figura 30- Ler dados 19](#_Toc124522882)

[Figura 31-switch registar consultar participante 21](#_Toc124522883)

[Figura 32-switch registar consultar atividade 21](#_Toc124522884)

[Figura 33-switch registar e consultar inscrições 22](#_Toc124522885)

[Figura 34-switch menu estatísticas 22](#_Toc124522886)

[Figura 35-switch gravar dados e sair 23](#_Toc124522887)

[Figura 36-ler opção 23](#_Toc124522888)

[Figura 37-validar datas 24](#_Toc124522889)

[Figura 38-validar nif 24](#_Toc124522890)

[Figura 39-validar numero 24](#_Toc124522891)

[Figura 40-validar valor 25](#_Toc124522892)

[Figura 41-Teste menu registar e consultar 26](#_Toc124522893)

[Figura 42-Teste registar e consultar atividades 26](#_Toc124522894)

[Figura 43-Registar e consultar inscrições 27](#_Toc124522895)

[Figura 44-Teste menu estatistica 27](#_Toc124522896)

[Figura 45-Teste guardar dados 27](#_Toc124522897)

[Figura 46-Teste menu sair 28](#_Toc124522898)

[Figura 47-Sem utilizadores registados 28](#_Toc124522899)

[Figura 48-Com utilizador registado 28](#_Toc124522900)

[Figura 49-Consulta NIF existente 29](#_Toc124522901)

[Figura 50-Consulta NIF não existente 29](#_Toc124522902)

[Figura 51-NIF errado 29](#_Toc124522903)

[Figura 52-Consulta de atividade inexistente 30](#_Toc124522904)

[Figura 53-Consulta atividade registada 30](#_Toc124522905)

[Figura 54-Quando atividade não existe 30](#_Toc124522906)

[Figura 55-Atividade por designação 30](#_Toc124522907)

[Figura 56-Consulta Sem inscrição registada 31](#_Toc124522908)

[Figura 57-Quando a inscrição existe 31](#_Toc124522909)

[Figura 58-Consulta atividades por associação 31](#_Toc124522910)

[Figura 59-Percentagem de inscrições por escola 31](#_Toc124522911)

[Figura 60-Valor total das incrições entre duas datas 32](#_Toc124522912)

[Figura 61-Guardar dados 32](#_Toc124522913)

[Figura 62-Ler dados automatico 32](#_Toc124522914)

# Introdução

Neste relatório, desenvolvido como parte da disciplina de Fundamentos de Programação (FP) no Curso Técnico Superior (CTesP) de Programação de Sistemas de Informação (PSI), o objetivo é aplicar os conhecimentos adquiridos durante as aulas de FP.

O projeto visa criar um protótipo de um aplicativo que permite gerenciar as inscrições em atividades realizadas pelas várias Associações de Estudantes do Politécnico de Leiria, utilizando as ferramentas fornecidas pela disciplina e aplicando os conhecimentos adquiridos em sala.

Este extenso documento aborda oito capítulos visando facilitar a compreensão completa, iniciando pela estrutura abrangente do projeto com os fundamentos essenciais para o seu desenvolvimento, subsequentemente a implementação de funcionalidades e respetivas funções, prosseguindo com um capítulo eventual sobre possíveis imprevistos, findando com a validação pertinente de dados e realização de testes para analisar criticamente e extrair conclusões lastreadas, relacionando igualmente referências bibliográficas pertinentes.

# Constituição da estrutura do programa

Nesta parte, apresentaremos uma visão geral da arquitetura do projeto. Abordaremos tópicos como pré-processamento, constantes utilizadas, estruturas de dados declaradas e funções implementadas, de modo a propiciar uma compreensão mais aprofundada da estrutura do programa. Buscamos explicar quais componentes foram incluídos e por que motivos, de forma a tornar transparente a lógica subjacente ao código. Esperamos que esta descrição auxilie na leitura e interpretação do código-fonte, permitindo ao leitor identificar a finalidade de cada seção e entender como todos os elementos se relacionam para compor a solução desenvolvida.

### Definições de pré-processamento

Uma imagem com texto

Descrição gerada automaticamenteAs declarações de pré-processamento, também conhecidas como bibliotecas, estão declaradas logo no início do programa e podem ser vistas na imagem abaixo.

Figura 1- Definições de pré-processamento

As primeiras duas definições de pré-processamento já vêm inseridas automaticamente quando um novo projeto é criado. Porém, outras três foram incluídas para permitir o uso de funcionalidades específicas, tais como:

* A <locale.h> garantirá a exibição adequada de caracteres especiais e acentuação em nosso código.
* A <string.h> ativará funções para manipulação de strings.
* A <time.h> permitirá a obtenção automática da data e hora no momento do cadastro.

### Constantes

Uma imagem com texto

Descrição gerada automaticamenteAs Constantes definidas durante o projeto são as visíveis na imagem abaixo:

Figura 2-Constantes

As constantes representam um valor que nunca é alterado ao longo da execução do programa, são acessíveis durante todo o programa e caso seja preciso alguma alteração é de fácil e rápida alteração e todo o programa fica atualizado, foram utilizadas para:

* Definir o mínimo e o máximo de cada menu, de forma que quando acrescentar ou retirar uma opção dos menus seja mais fácil a alteração.
* Definir o mínimo e o máximo para o NIF, que neste momento esta definido apenas para pessoa singular visto que as inscrições são para pessoas/participantes ou seja nove dígitos e podem começar por 1, 2 ou 3.
* Os números de telefone e telemóvel foram utilizados de acordo com o Plano Nacional de Numeração ou seja os números de telefone que iniciam por dois que são do serviço telefónico acessível ao público em local fixo e os que iniciam por nove que são dos serviços de comunicações moveis e ambos possuem nove dígitos.
* Por fim o número máximo de participantes, atividades e inscrições esta definido no enunciado e desta forma num futuro caso seja pedido a alteração é de fácil resolução.

### Declaração de Estruturas

De seguida podemos visualizar na imagem abaixo a declaração das estruturas.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Declaração de Estruturas

As estruturas são uma coleção de uma ou mais variáveis agrupadas sob o mesmo nome, cada variável dentro da estrutura é denominada por campo, estas são declaradas antes da função main() (antes dos protótipos das funções). Foi utilizado o typedef para a declaração das estruturas na realização deste programa de forma a facilitar a chamada as estruturas, visto que o mesmo apenas define um sinónimo para tipo de dados existentes.

Na imagem acima podemos visualizar a existência de estruturas simples e compostas, isto é, estruturas com uma ou mais estruturas dentro da mesma. Por exemplo as estruturas t\_Atividades e t\_Inscricoes são ambas uma estrutura composta, ou seja, dentro de cada uma dessas estruturas temos mais duas estruturas a t\_Data e a t\_horas, as duas últimas tiveram de ser declaradas primeiro para não dar erro na declaração da estrutura composta.

As estruturas foram declaradas para cumprir o objetivo do projeto sem repetição de código e sem utilizar várias variáveis para poder cumprir determinados objetivos.

### Declaração de Funções

Como solicitado no enunciado, de forma a facilitar a correta estruturação, implementação, entendimento e de forma a evitar a redundância e repetição de código, o programa foi dividido em várias funções, estas podem ser vistas na imagem abaixo.



Figura -Declaração de funções

Foram utilizadas um total de trinta e seis funções (sem contar com a main), e nenhuma delas é uma função sem parâmetros e que não devolvem qualquer valor, isto é, podem ser uma função com parâmetros, uma função que devolve algum valor ou ambos.

### Variáveis principais

As variáveis principais estão na função main() e podem ser vistas na imagem abaixo:

Uma imagem com texto

Descrição gerada automaticamente

Figura -Variáveis principais

Estas são as variáveis principais porque são elas que vão ser precisas e utilizadas em todo o programa. Quer seja para contabilizar os ids dos participantes, atividades e inscrições , gerir o o menu e submenus e as “estruturas” para os participantes, atividades e inscrições com as respetivas constantes a representar o máximo de cada uma.

# Funcionalidades implementadas e respetivas funções

As funcionalidades implementadas foram todas introduzidas no programa de forma a cumprir o solicitado de forma simples, sendo algumas introduzidas com algumas melhorias, foram cumpridas todas as funcionalidades solicitadas como podemos ver a seguir.

### Registar e consultar os dados dos participantes

Na imagem abaixo podemos visualizar a função para registar participantes, que recebe a variável participante[] da nossa já mencionada estrutura t\_Participantes e a variável \*id como ponteiro, que corresponde ao IDParticipante da nossa função main().

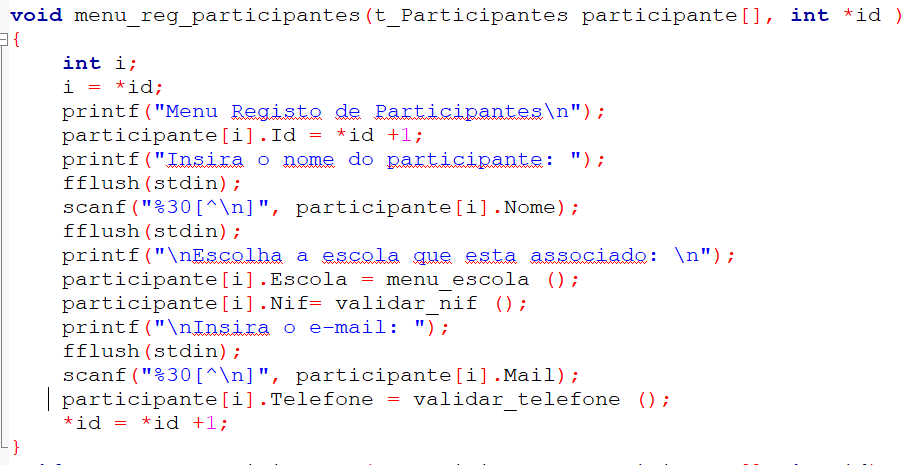


Figura -Registo participantes

No menu de registo de participante irá então:

* Receber o número de ID do participante.
* Receber o nome do participante, em que armazena todos os carateres até ser digitado o carater \n (que não fica armazenado), e limita o tamanho da string armazenada a 30 caracteres.
* Irá então guardar o número associado a escola através da função abaixo, antes de dar return ao número, o mesmo ainda irá ser validado como número de forma a não ser devolvido um carater qualquer.

Uma imagem com texto

Descrição gerada automaticamente

Figura -menu escola

* Irá validar o NIF do participante através da função na imagem abaixo, caso o NIF não seja de pessoa individual irá ser indicado e será validado como número assim como na função anterior de forma a não receber carateres.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Validar NIF

* Receber o mail do participante, em que armazena todos os carateres até ser digitado o carater \n (que não fica armazenado), e limita o tamanho da string armazenada a 30 caracteres.
* Irá validar o telefone/telemóvel do participante através da função abaixo, caso o telefone/telemóvel não seja correto irá ser indicado como proceder por fim será validado como número assim como na função anterior de forma a não receber carateres.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Validar telefone

* Por fim irá alterar o valor do \*id que esta na função main() como IDParticipante.

Para consultar os participantes:

Uma imagem com texto

Descrição gerada automaticamente

Figura -Menu consulta participantes

* Iremos receber novamente a variável participante[] do tipo t\_Participantes e a variável id que corresponde ao IDPartcipante da nossa função main().
* Através do for, iremos visualizar todos os participantes inseridos, pois o mesmo irá percorrer todos os participante[i] enquanto o i que começa em zero for menor que o id e é incrementado em +1 a cada volta.

### Registar e consultar os dados das atividades

Na imagem abaixo podemos visualizar a função para registar atividades, que recebe a variável atividade[] da nossa já mencionada estrutura t\_Atividades e a variável \*id como ponteiro, que corresponde ao IDAtividade da nossa função main().

Uma imagem com texto

Descrição gerada automaticamente

Figura -Registo atividades

* Receber o número de ID da atividade.
* Receber o nome da atividade, em que armazena todos os carateres até ser digitado o carater \n (que não fica armazenado), e limita o tamanho da string armazenada a 30 caracteres.
* Irá guardar o ano através da função abaixo, antes de dar return ao número, o mesmo ainda irá ser validado como número de forma a não ser devolvido um carater qualquer, e apenas aceitará atividades a partir de 2023 (visto ser o ano de entrega do trabalho) e até 2123.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Validar ano

* Irá guardar o mês através da função abaixo, antes de dar return ao número, o mesmo ainda irá ser validado como número de forma a não ser devolvido um carater qualquer, e aceitará apenas valores de 1 a 12 inclusive.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Validar mês

* Irá guardar o dia através da função abaixo, antes de dar return ao número, o mesmo ainda irá ser validado como número de forma a não ser devolvido um carater qualquer, e aceitará apenas valores de 1 a 31 inclusive.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Validar dia

* Irá guardar a hora através da função abaixo, antes de dar return ao número, o mesmo ainda irá ser validado como número de forma a não ser devolvido um carater qualquer, e aceitará apenas valores de 0 a 23 inclusive.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Validar hora

* Irá guardar os minutos através da função abaixo, antes de dar return ao número, o mesmo ainda irá ser validado como número de forma a não ser devolvido um carater qualquer, e aceitará apenas valores de 0 a 59 inclusive.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Validar minutos

* Irá guarda o nome do local da atividade, em que armazena todos os carateres até ser digitado o carater \n (que não fica armazenado), e limita o tamanho da string armazenada a 20 caracteres.
* Irá então guardar o número associado ao tipo de atividade através da função abaixo, antes de dar return ao número, o mesmo ainda irá ser validado como número de forma a não ser devolvido um carater qualquer.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Menu atividade

* Irá então guardar o número associado a associação através da função menu\_escola() já mostrada na Figura 6 anteriormente, antes de dar return ao número, o mesmo ainda irá ser validado como número de forma a não ser devolvido um carater qualquer.
* Irá então guardar o valor da inscrição através da função abaixo que devolve um float, pois o valor da inscrição não tem de ser inteiro. Aceita o valor de zero pois a inscrição pode ser grátis.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Validar valor

* Por fim irá alterar o valor do \*id que esta na função main() como IDAtividade e somar o +1 ao valor.

Para consultar as atividades:

* Iremos receber novamente a variável atividade[] do tipo t\_Atividades e a variável id que corresponde ao IDAtividade da nossa função main().
* Através do for, iremos visualizar todos as atividades inseridas, pois o mesmo irá percorrer todos as atividade[i] enquanto o i que começa em zero for menor que o id e é incrementado em +1 a cada volta como podemos visualizar na imagem abaixo.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Menu consulta atividades

### Registar e consultar os dados das inscrições

Na imagem abaixo podemos visualizar a função para registar inscrições, que recebe a variável inscricoes[] do tipo t\_inscricoes, a variável participante[] do tipo t\_Participantes, a variável atividade[] do tipo t\_Atividades, e as variáveis dos Ids, das atividades, participantes e inscrições.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Registo inscrições

* É adicionada as struct tm \*p; time\_t seconds; time(&seconds); e p= localtime(&seconds) para ir buscar a data e a hora de forma automática assim quando solicitado mais a frente vamos buscar a data e hora do momento que nos encontramos ao PC.
* Recebe o número de ID da inscrição.
* Guarda na variável X o valor do ID da atividade que desejam fazer a inscrição

Uma imagem com texto

Descrição gerada automaticamente

Figura -Validar inscrição atividade

* Passa esse valor na variável X para a variável inscricoes[i].IDAtividade, foi feito dessa maneira porque não conseguíamos passar esse valor diretamente e assim foi possível.
* Guarda na variável j o valor do ID do participante.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Validar inscrição participante

* Passa esse valor na variável j para a variável inscricoes[i].IDParticipante, pelo mesmo problema referido acima.
* (inscricoes[i].Escola) = (participante[j-1].Escola) ; utilizamos o -1 no j para que ele vá buscar o valor ao participante correto porque a estrutura começa em zero mas o id começa em um.
* inscricoes[i].tipoAtividade = atividade[x-1].Tipo\_Atividade; foi utilizado o -1 no x de forma a ir buscar o valor correto na atividade pela mesma razão descrita acima no caso do j.
* Irá guardar o valor pago através da função mostrada na Figura 17
* Quanto ao ano, mês, dia, hora, minutos e segundos da inscrição, serão guardados nas variáveis respetivas como demonstrado abaixo, através das funcionalidades inseridas anteriormente.

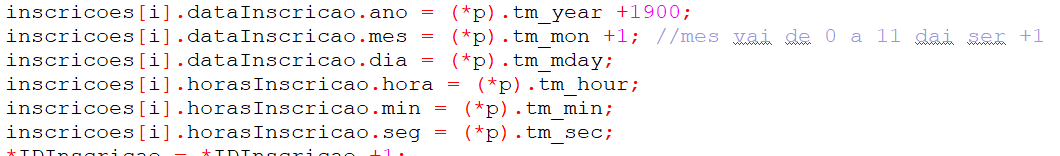


Figura -Data hora inscrição

* Por fim irá ser adicionado +1 ao \*IDInscrição da função main().

Para consulta das inscrições, podemos ver na imagem abaixo:

Uma imagem com texto

Descrição gerada automaticamente

Figura -Consulta Inscrição

* Iremos receber inscricoes[] do tipo t\_inscricoes e id que corresponde ao IDInscrições da main().
* Através do for, iremos visualizar todos as inscrições inseridas, pois o mesmo irá percorrer todos as inscricoes[i] enquanto o i que começa em zero for menor que o id e é incrementado em +1 a cada volta como podemos visualizar na imagem acima.
* Na parte da data e da hora é utilizado o %02d de forma a mostrar a data e a hora como solicitado no enunciado.

### Estatísticas - Número de atividades realizadas por cada associação

Para visualizar o número de atividades realizadas por cada associação, podemos observar na função abaixo(menu\_atividade\_por\_associacao) que a mesma recebe atividade[] e IDAtividade da main(), e depois quando chama a função (procura\_associacao) de forma a preencher quantas atividades foram criadas por cada associação envia atividade, IDAtividade e um número, este número esta associado ao numero da escola.

Na função procura\_associacao, a mesma possui um contador que começa em zero e apenas será incrementado caso o número que esta associado a escola seja igual ao número que está em atividade[i].Associacao. Caso não tenha nenhuma atividade criada seja devolvido zero caso tenha criado alguma atividade será devolvido o número de atividades criadas.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Atividade por associação

### Estatísticas - Percentagem de inscrições por escola

Para visualizar a percentagem de inscrições por escola, podemos observar na função abaixo (menu\_inscricao\_por\_escola) que recebe inscricoes[] e IDInscricao da main(), que a mesma irá mostrar o resultado em float, pois as percentagens não tem de ser inteiras, e que irá fazer o calculo da seguinte forma (QTD. De vezes que a escola realizou uma inscrição vezes 100) dividido pelo total de inscrições, e a apresentação desse valor é feita da seguinte forma “%.2f%%”, pois irá mostrar até duas casas decimais e tem o símbolo da percentagem logo a seguir.

A quantidade de vezes que a escola realizou uma inscrição é feita através da função que serve como contador na mesma imagem com o nome de (calculo\_por\_escola).

Uma imagem com texto

Descrição gerada automaticamente

Figura -Inscrições por escola

### Estatísticas - Valor total das inscrições entre duas datas (horizonte temporal) por tipo de atividade

De forma a visualizar o valor total das inscrições entre duas datas, por tipo de atividade, primeiro temos que pedir duas datas, a data de inicio com dia, mês e ano e a data de fim, com dia mês e ano, sendo que a data de fim tem de ser superior a data de inicio, caso não o seja temos o if para corrigir e solicitar a data de fim novamente. Após termos as datas, iremos então a semelhança do que foi feito na atividade por associação iremos chamar a função calculo\_valor\_inscricao com as seguintes variáveis inscricoes, IDInscricao, diaInicio, mesInicio, anoInicio, diaFim, mesFim, anoFim, número associado ao tipo de atividade, dentro desta função iremos então ter dois for em que o primeiro for será para percorrer as datas e o segundo for irá fazer a soma do valor pago da inscrição caso o tipo de atividade seja igual ao número associado ao tipo de atividade que foi enviado pela função anterior.

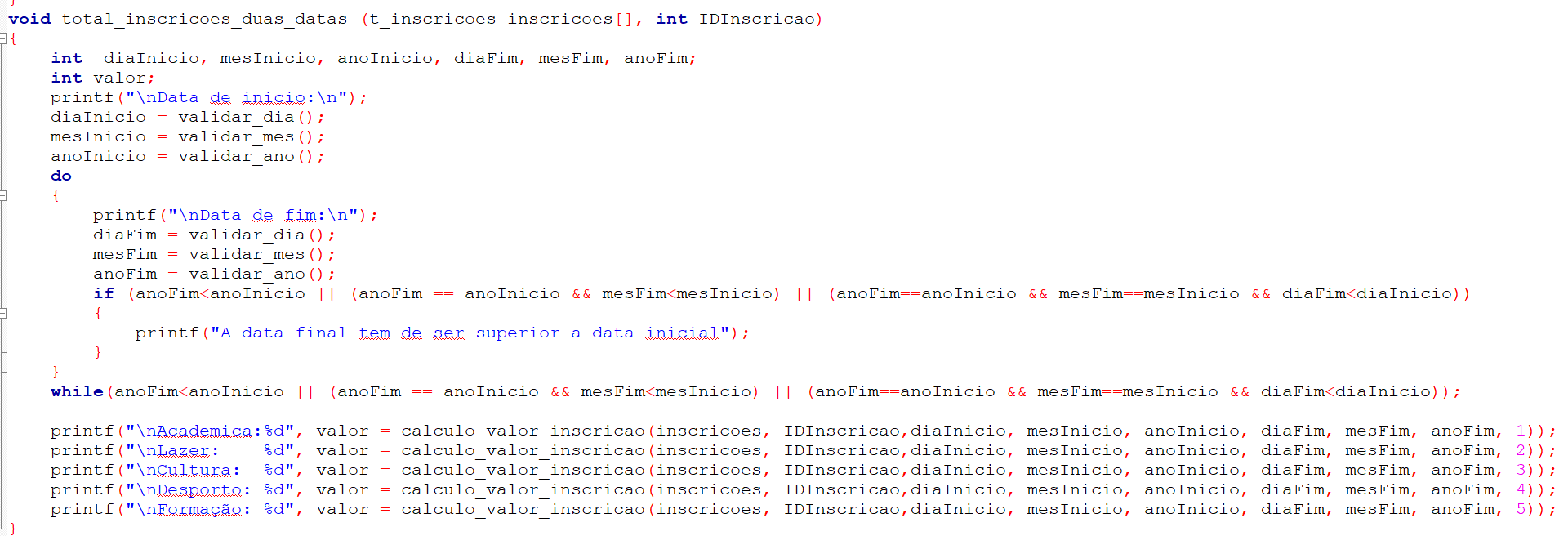


Figura -Inscrições entre duas datas

Uma imagem com texto

Descrição gerada automaticamente

Figura -Calculo valor da inscrição

### Guardar e ler de ficheiros binários os dados da aplicação

Nas imagens abaixo podem ver como foi feito a realização das funções para guardar e ler de ficheiros binários.

Figura - Gravar dados

Uma imagem com texto

Descrição gerada automaticamente

Para gravar está de forma a enviar os dados da estrututura e do respetivo id e guardar no mesmo ficheiro, no caso irá gerar 3 ficheiros.

Figura - Ler dados

Uma imagem com texto

Descrição gerada automaticamente

Para ler esta da mesma forma que ira ler uma estrutura e um id e ira retornar esses valores.

# Funcionalidades por implementar

Todas as funcionalidades solicitadas no enunciado foram implementadas.

# Validação de dados de entrada

Além de ser uma forma de restringir a entrada de dados em um campo o texto da validação permite que o programa forneça ao utilizador uma mensagem para ajudar o mesmo a inserir os dados que não são válidos.

Podemos visualizar através das próximas imagens que a validação de dados foi inserida em todas as funções durante o programa, e em alguns casos para otimizar o mesmo foi utilizado antes da chamada de outra função.

No switch abaixo como podemos ver só iremos entrar no case 1 caso o número de participantes não tenha sido ultrapassado e no case 2 só vamos conseguir entrar caso já tenha sido feito pelo menos um registo de participante.

Uma imagem com texto

Descrição gerada automaticamente

Figura -switch registar consultar participante

Podemos ver que o mesmo acontece na imagem abaixo, mas para o caso das atividades.

Uma imagem com texto

Descrição gerada automaticamente

Figura -switch registar consultar atividade

As validações usadas abaixo no menu das inscrições já são diferentes, podem visualizar na imagem abaixo que só consegue entrar no menu das inscrições caso tenha pelo menos um participante e uma atividade registadas anteriormente. E dentro do submenu das inscrições podemos ver a mensagem para ajudar o utilizador sobre o porque de não conseguir entrar no mesmo. O mesmo acontece para entrar no submenu das consultas das inscrições que caso não tenha sido registada nenhuma inscrição também não irá entrar dentro da função de consulta e irá mostrar a mensagem de erro.

Uma imagem com texto

Descrição gerada automaticamente

Figura -switch registar e consultar inscrições

Dentro do menu das estatísticas podem observar abaixo que só irá chamar a função para calcular as atividades por associação caso tenha sido criado pelo menos uma atividade, que só irá chamar a função para ver a percentagem de inscrições por escola e a do total de inscrições entre duas datas por tipo caso tenha sido pelo menos uma inscrição.

Uma imagem com texto

Descrição gerada automaticamente

Figura -switch menu estatísticas

Na gravação de dados podemos visualizar na imagem abaixo que só irá gravar dados se tiver pelo menos uma atividade, um participante, ou uma inscrição realizada. No caso de saída do programa temos duas opções, gravar e sair e sair sem gravar, a primeira só sendo possível se houver algo para gravar e a segunda é uma saída sem validação pois já foi validada a sair anteriormente.

Uma imagem com texto

Descrição gerada automaticamente

Figura -switch gravar dados e sair

Para transitar entre menus e submenus irá sempre ser chamada a função ler\_opcao de forma a evitar a repetição de código em todos os menus/submenus, a função ler\_opcao ainda irá a chmar a função validar\_numero de forma a ter a certeza que foi mesmo escolhido um número e não outro carater qualquer.



Uma imagem com texto

Descrição gerada automaticamente

Figura -ler opção

Para o valor total das inscrições entre duas datas temos que validar que a data final é superior a data inicial como mostra na imagem abaixo.

Uma imagem com texto

Descrição gerada automaticamente

Figura -validar datas

Validação do NIF, pode ser visualizada na imagem abaixo, o NIF neste programa está a ser validado para pessoa singular visto que as inscrições são para pessoas/participantes ou seja nove dígitos e podem começar por 1,2 ou 3.

Uma imagem com texto

Descrição gerada automaticamente

Figura -validar nif

Os números de telefone e telemóvel foram utilizados de acordo com o Plano Nacional de Numeração ou seja os números de telefone que iniciam por dois que são do serviço telefónico acessível ao público em local fixo e os que iniciam por nove que são dos serviços de comunicações moveis e devem possuem nove dígitos.

Uma imagem com texto

Descrição gerada automaticamente

Figura -validar numero

Para validação de número utilizamos algo aprendido em sala de aula como podem ver na imagem abaixo. Que é por val = scanf”…” isto faz com que o valor se for menor que um é um carater se for maior que um é um numero.

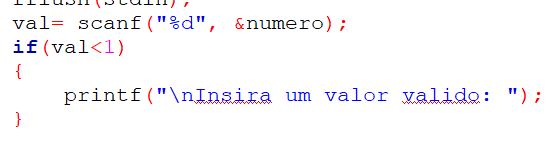


Figura -validar valor

A validação das horas, minutos, dias, meses e anos, já foram referidas acima no ponto 2.4.

# Realização de testes na aplicação

Os testes no desenvolvimento de aplicações são importantes porque eles ajudam a garantir a qualidade do software e a identificar e corrigir problemas de forma eficiente. Eles também podem ajudar a evitar regressões, ou seja, problemas que surgem após a implementação de novas funcionalidades ou mudanças no código. Além disso, os testes ajudam a aumentar a confiança dos desenvolvedores na qualidade do código e acelerar o processo de desenvolvimento.

Sem mais demoras vamos então aos testes.

Teste 1-Menu registar e consultar os dados dos participantes

Uma imagem com texto

Descrição gerada automaticamente

Figura -Teste menu registar e consultar

Teste 2- Em seguida o registar e consultar os dados das atividades, inserindo o valor 2

Uma imagem com texto

Descrição gerada automaticamente

Figura -Teste registar e consultar atividades

Teste 3- Menu registar e consultar os dados as inscrições, inserindo o valor 3, e como podemos observar como não realizada nenhuma inscrição o programa envia uma mensagem.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Registar e consultar inscrições

Teste 4- Menu Estatísticas inserindo o valor 4

Uma imagem com texto

Descrição gerada automaticamente

Figura -Teste menu estatistica

Teste 5- Guardar dados da aplicação inserindo o valor 5

Uma imagem com texto

Descrição gerada automaticamente

Figura -Teste guardar dados

Teste 6- Menu sair inserindo o valor 6

Uma imagem com texto

Descrição gerada automaticamente

Figura -Teste menu sair

Teste 7- Visualizar um utilizador, caso não haja utilizadores inseridos.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Sem utilizadores registados

Teste 8- Visualizar um utilizador com pelo menos um utilizador registado.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Com utilizador registado

Teste 9- Através de NIF.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Consulta NIF existente

Teste 10- Caso NIF não exista.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Consulta NIF não existente

Teste 11- Caso NIF não esteja correto

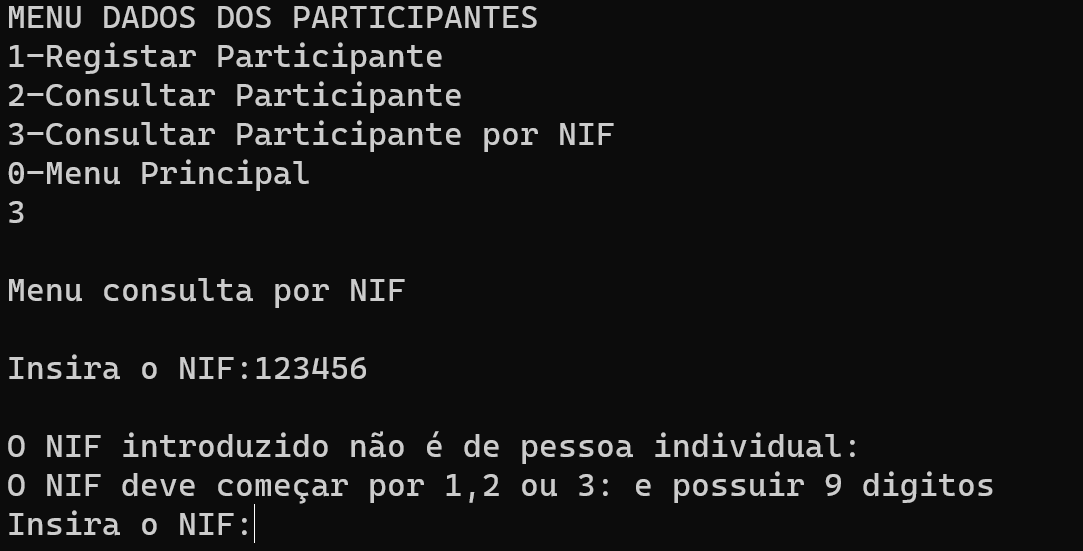


Figura -NIF errado

Teste 12- Consultar atividade que não existe

Uma imagem com texto

Descrição gerada automaticamente

Figura -Consulta de atividade inexistente

Teste 13- Consultar Atividade

Uma imagem com texto

Descrição gerada automaticamente

Figura -Consulta atividade registada

Teste 14- Consulta atividade por designação

Uma imagem com texto

Descrição gerada automaticamente

Figura -Quando atividade não existe

Teste 15- Atividade por designação

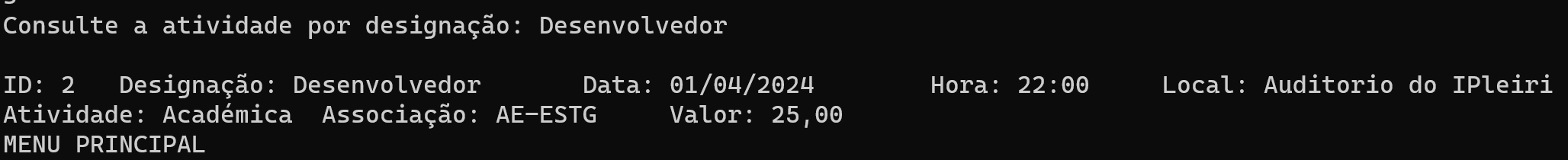


Figura -Atividade por designação

Teste 16- Consulta inscrições sem inscrição registada

Uma imagem com texto

Descrição gerada automaticamente

Figura -Consulta Sem inscrição registada

Teste 17- Consulta inscrição

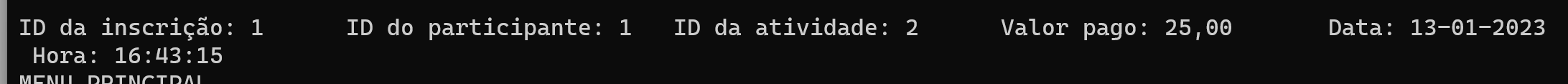


Figura -Quando a inscrição existe

Teste 18-Consulta do número de atividades por associação.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Consulta atividades por associação

Teste 19-Percentagem de inscrições por escola

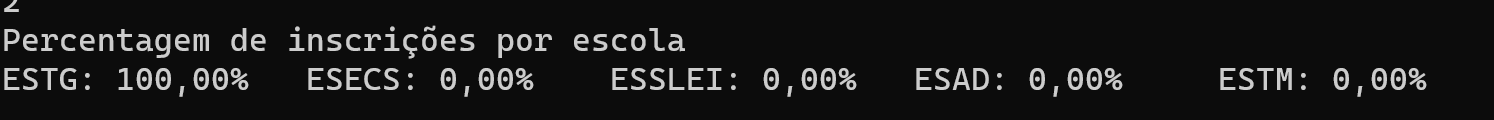


Figura -Percentagem de inscrições por escola

Teste 20-Valor total das inscrições entre duas datas

Uma imagem com texto

Descrição gerada automaticamente

Figura -Valor total das incrições entre duas datas

Teste 21-Guardar dados

Uma imagem com texto

Descrição gerada automaticamente

Figura -Guardar dados

Teste 22- Ler dados de forma automática ao entrar

Uma imagem com texto

Descrição gerada automaticamente

Figura -Ler dados automatico

# Análise crítica e proposta de melhorias

Após a realização do projeto e devido ao fato de termos realizado todos os objetivos do enunciado e de nos sentirmos mais a vontade com o projeto, sentimos que tem algum espaço para melhorias, tais como:

Rever as funções que validam o dia, mês e ano, de forma a validar o ano para bissexto, validar os dias consoante os meses, isto é, meses com 28, 30 ou 31 dias.

Validar o telemóvel pode ser feito de forma mais exaustiva, por exemplo os telemóveis estão apenas a ver se começam por nove e possuem nove dígitos, e poderiam validar mais especificamente pelos dois primeiros números.

Na função de registo de inscrições, utilizamos o “x” e o “j” como variáveis de forma a poupar espaço de escrita no código, mas acabamos por criar duas variáveis.

Na função de registo de participante, validar o mail de uma forma mais especifica de forma que veja se tem pelo menos um “@” e um “.”, e assim utilizar melhor a biblioteca string.h que foi adicionada.

É utilizado ponteiros nos Ids(participantes, atividades e inscrições) da função main() e não é preciso, poderia ser dado return do valor em vez de alterar diretamente através do uso de ponteiros.

Nos menus de consulta pode ser adicionado modos de pesquisa, por exemplo por ID ou outro qualquer.

No menu de registo de atividade, ao escolher a data da atividade esta deve ser sempre maior que a data do momento que nos encontramos (apesar que pode haver ocasiões que o utilizador queira registar atividades anteriores).

# Conclusão

Durante a realização do projeto, ficamos com os nossos conhecimentos sobre as matérias dadas na UC de FP mais consolidados pois foi solicitado de tudo um pouco.

Todos os objetivos solicitados no enunciado foram atingidos. O trabalho desenvolvido ao longo do projeto teve algumas alterações devido ao fato de irmos aprendendo matéria nova com o passar do tempo.

As principais aprendizagens sem ser o obvio, provavelmente foram a capacidade de pensar de forma objetiva para criar o programa e em ser capaz de separar o problema em pequenos problemas e a partir dai começar a construir e a encaixar as peças.

Em termos de aspetos positivos podemos falar sem dúvida que foi a consolidação dos conhecimentos, e de aspetos negativos foi o tempo consumido à procura de erros, que depois era apenas um ponto e virgula ou um fechar de chavetas ou mesmo uma falta de parenteses retos ao receber uma variável.

As consequências deste projeto para o nosso percurso académico, consideramos ser fundamental pois como o nome do curso indica é um curso de programação e esta é uma UC que irá ser a base para o resto do curso.

# Referências bibliográficas

PowerPoint fornecidos através do moodle:

<https://ead.ipleiria.pt/2022-23/course/view.php?id=782>

Data e hora automática:

<https://pt.stackoverflow.com/questions/159141/obter-a-data-atual-diretamente-pela-máquina>