

Remodelação Projeto

Nesta remodelação do projeto foram melhorados os conceitos de modularização e encapsulamento. Para tal foi criada na diretoria do trabalho a pasta “REMODELACAO” onde se apresenta uma cópia exata das pastas “src” e “include” (ficheiros .h e .c) do projeto original. (Nesta também se encontra um “remodelacao.txt” que apresenta uma to-do list daquilo que foi feito de forma mais simples, caso necessário.)

Desta forma, durante o exercício de reescrita/remodelação do código, uma série de melhorias foram implementadas para **aumentar a modularização, encapsulamento e organização do código-fonte, visando torná-lo mais legível, manutenível e escalável.**

- Uma das principais melhorias foi a separação das bibliotecas dos arquivos de cabeçalho, (seguindo a sugestão dada pela equipa docente em apresentação). As funções foram implementadas nos arquivos “.c”, enquanto os protótipos e estruturas de dados foram definidos nos arquivos de cabeçalho correspondentes.
 - Para promover a **encapsulação e modularidade**, foram introduzidas **estruturas opacas** para ocultar detalhes de implementação e evitar dependências desnecessárias. Isso foi especialmente aplicado em áreas críticas, como o acesso a dados de usuários, voos, etc. (parsers) e também nos catalogos e restantes funções dependestes destes dados.
 - Além disso, foram criados **getters** e **setters**, proporcionando uma interface mais limpa e segura para acessar e manipular os dados e também melhorando o encapsulamento (**Setters com cópia de informação para uma maior fiabilidade**).
 - Os arquivos originais foram copiados e uma secção remodelada foi adicionada, identificada com a marcação `/** REMODELACAO */` em todos os ficheiros “.c” modificados facilitando a identificação das partes modificadas ou adicionadas. (**caso a equipa docente necessite pesquisar, podendo utilizar uma ferramenta de indexagem**) Os header files foram quase/maioritariamente todos modificados e como tal não foi necessária a sua marcação.
 - Para documentar as alterações realizadas, um arquivo `remodelacao.txt` foi criado, descrevendo detalhadamente as modificações feitas e suas justificativas.
 - Movemos “**defines**” e “**includes**” relevantes para os arquivos “.c”, como sugerido, para reduzir a poluição do espaço de nomes global e melhorar a modularidade do código (caso possível essa transição, caso não esses ficaram nos header files).
 - A estrutura de dados “appstatus” no modo iterativo também foi transformada em uma estrutura opaca, seguindo a mesma abordagem adotada para outras estruturas de dados sensíveis.
 - Finalmente, e como dito anteriormente, aplicamos getters e setters em todos os módulos e modos (iterativo), garantindo consistência e facilitando futuras alterações e manutenções no código.
- Com essas melhorias, o código foi refinado e está pronto para enfrentar os desafios futuros com mais robustez e flexibilidade.