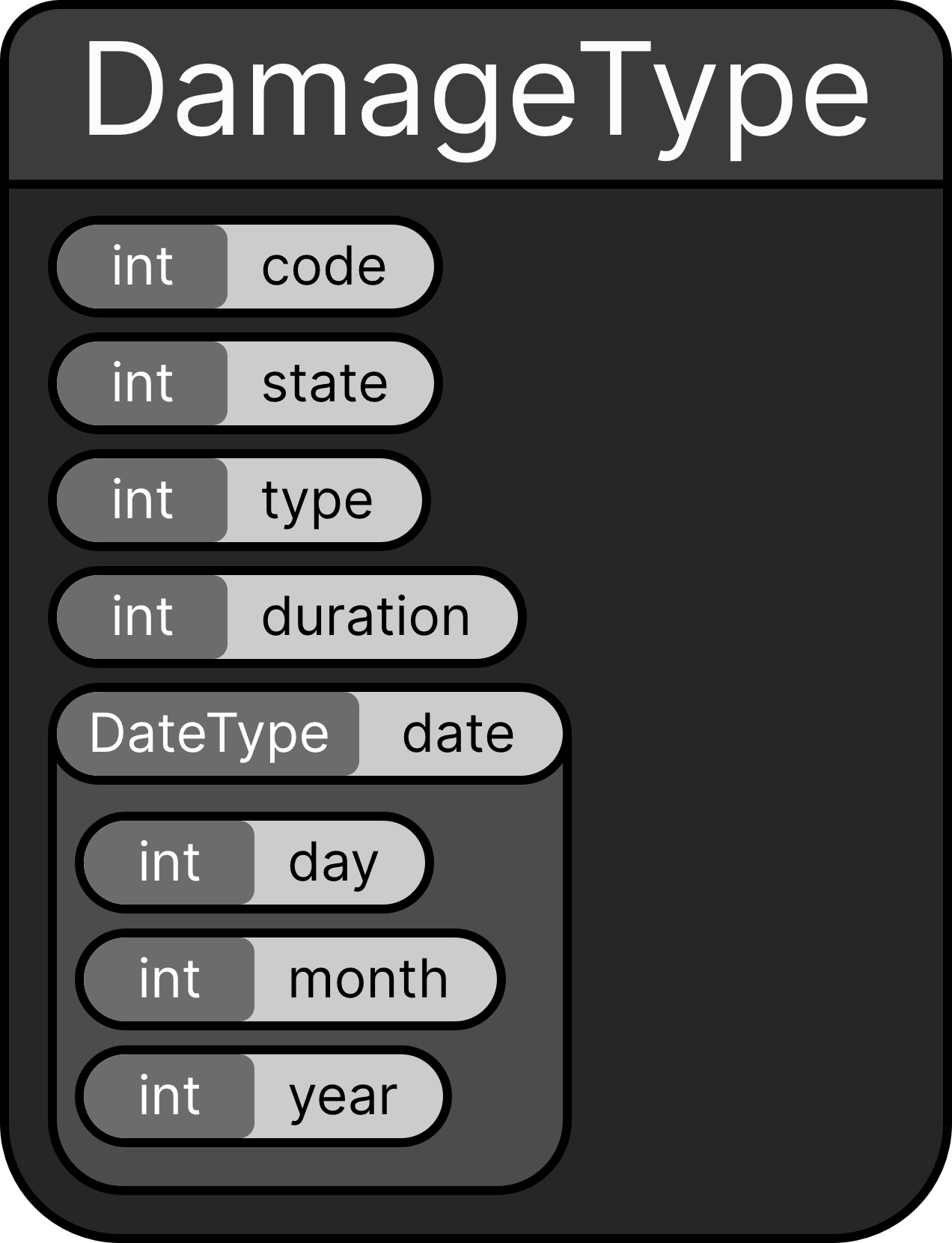
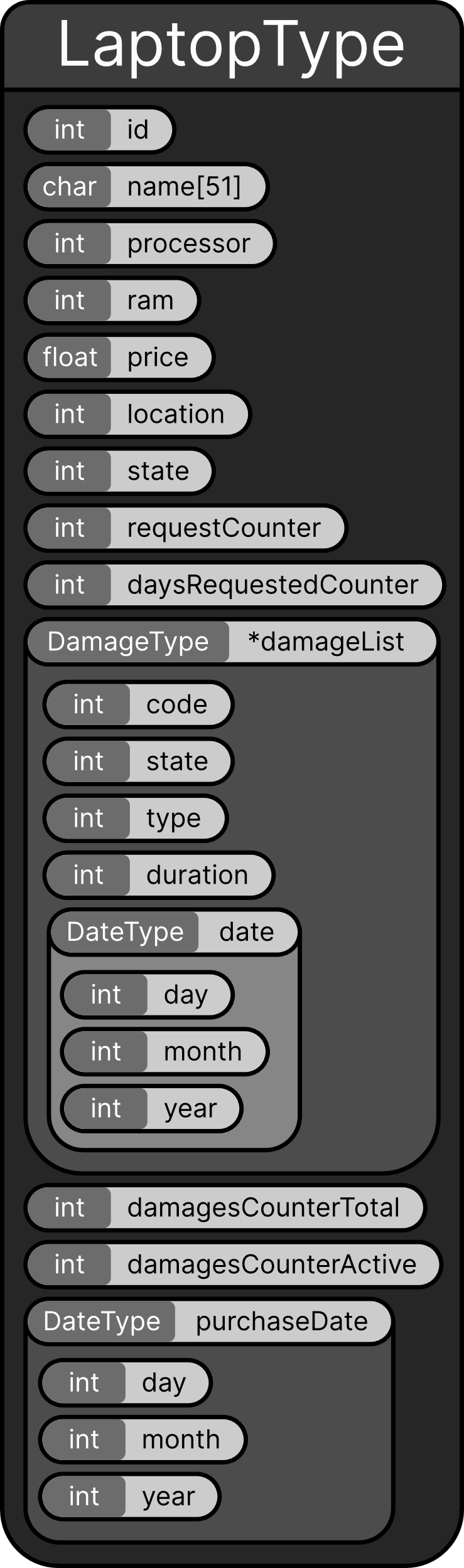
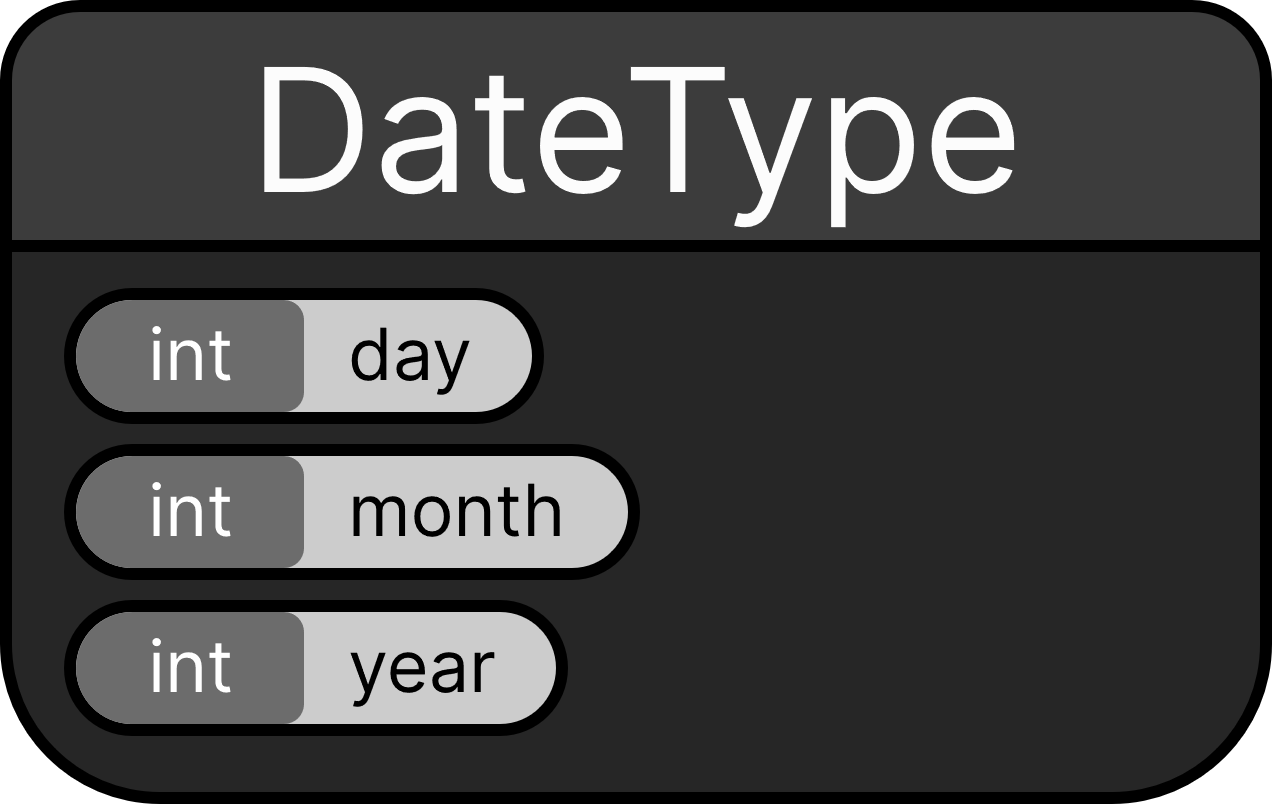
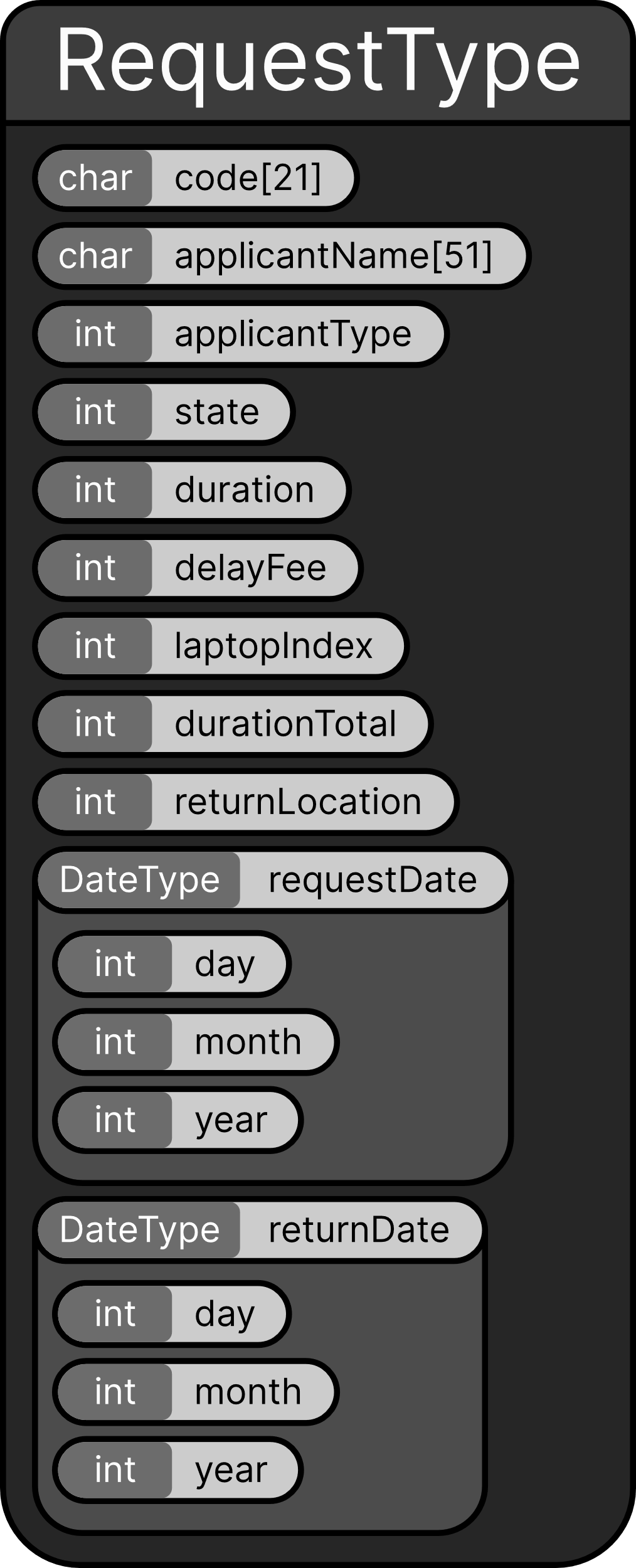
Programação 1

**LAPTOP REQUEST**Projeto Engenharia Informática - 2022/23

2221839 – André Rosa Monteiro  
2221866 – Caio Maxwel Ribeiro e Barbosa



|  |  |  |
| --- | --- | --- |
| Funcionalidade | Nome da Função | |
| Apresentar no menu principal as informações:  • Quantidade de portateis existentes;  • Quantidade de portateis disponíveis;  • Quantidade de requisições efetuadas;  • Quantidade de requisições ativas. | drawMainMenu; | ✓ |
| Registar a adição de novos portateis. | registerNewLaptop; | ✓ |
| Registar a alteração de localização de um portatil. | changeLaptopLocation; | ✓ |
| Registar avaria de um portatil. | registerNewDamage; | ✓ |
| Registar reparação de um portatil. | registerNewRepair; | ✓ |
| Listar a informação referente a todos os portateis e o histórico de requisições. | showLaptopInfo; | ✓ |
| Listar a informação referente a todos os portateis e o histórico de avarias. | showDamageHistoric; | ✓ |
| Registar a requisição de um portatil. | registerNewRequest; | ✓ |
| Registar a devolução de um portatil. | registerReturnByLaptopId;  registerReturnByRequestCode; | ✓ |
| Registar a renovação de uma requisição existente. | registerRenewByLaptopId;  registerRenewByRequestCode; | ✓ |
| Listar a informação de uma ou mais requisições. | showRequestInfo; | ✓ |
| Apresentar os seguintes dados estatísticos:  • Percentagem de portateis com cada tipo de processador;  • O custo médio de cada multa paga;  • A menor multa paga;  • A maior multa paga;  • Percentagem de requisições de cada tipo de utente;  • Tipo(s) de utente(s) com a menor quantidade de requisições efetuadas;  • Até 3 das devolução(ões) mais recente(s). | showStatistics; | ✓ |
| Gravar a informação dos portateis e das requisições em ficheiro binário. | storeInfoToFile; | ✓ |
| Ler do ficheiro binário a informação dos portateis e das requisições. | loadFileToInfo; | ✓ |
| Gravar a informação das devoluções em ficheiro de texto. | storeLogData; | ✓ |

Para a criação do projeto adotamos alguns princípios:

* Criar funções para simplificação do código e facilidade de leitura;
* Criar funções sem retorno, a priorizar o(s) retorno(s) por ponteiro(s);
* Utilizar variáveis do tipo inteiro, mesmo em situações como o “tipo do utente”;
* Criar simulações de janela para melhor aspeto e entendimento do utilizador;
* Manter dinâmicas as janelas criadas;
* Permitir o utilizador sempre voltar atrás em suas ações.

A partir destes princípios subdividimos as funções nos ficheiros:

* funcoesDeDesign.h e funcoesDeDesign.c;
* funcoesDePortateis.h e funcoesDePortateis.c;
* funcoesDeRequisicoes.h e funcoesDeRequisicoes.c;
* funcoesDeData.h e funcoesDeData.c;
* funcoesDeDados.h e funcoesDeDados.c;
* constantes.h.

As variáveis de localização(int location do LaptopType e int returnLocation do RequestType) podem receber os seguintes valores:

* 1 - Residencias;
* 2 - Campus1;
* 3 - Campu2;
* 4 - Campu3;
* 5 - Campu5.

A variável de processador do portátil(int processor do LaptopType) pode receber os seguintes valores:

* 3 - Intel Core i3;
* 5 - Intel Core i5;
* 7 - Intel Core i7.

Obs.: Para facilitar a apresentação dos processadores, foi escolhido o próprio número de cada componente para representá-lo (3, 5 ou 7);

A varíavel de estado do portátil(int state do LaptopType) pode receber os seguintes valores:

* 1 - Disponível;
* 2 - Requisitado;
* 3 - Avariado temporariamente;
* 4 - Avariado permanentemente.

A variável de estado das requisições(int state do RequestType) e dos danos(int state do DamageType) pode receber os seguintes valores:

* 1 - Ativa;
* 2 - Completa/Arranjada.

A variável de identificação do componente avariado(int code do DamageType) pode receber os seguintes valores:

* 1 - Ecrã;
* 2 - Teclado;
* 3 - Telado numérico;
* 4 - Bateria;
* 5 - Porta USB;
* 6 - Porta de Audio;
* 7 - Internet sem fio;
* 8 - Internet a cabo;
* 9 - Placa mãe;
* 10 - Placa gráfica;
* 11 - Processador;
* 12 - Memória RAM.

Existem variáveis do tipo inteiro que simulam o tipo lógico e podem receber os seguintes valores:

* 0 - Para falso;
* 1 - Para verdadeiro.

Ao decorrer da realização do projeto, percebemos que ao tratar todas as informações como inteiros, a melhor decisão seria transformar estes inteiros em contantes no ficheiro “constantes.h”.

Também consideramos que cada portátil só poderia ter uma avaria permanete por cdrawMainMenuomponente e que a data desta avaria deveria ser igual ou superior a data da avaria temporaria mais recente do componente em questão.

E já que uma avaria permanente é impossível de arranjar, concluímos que o portátil torna-se inutilizável, o que altera seu estado para indisponível por tempo indeterminado.