

```

#ifndef DADOS_E ESTRUTURAS
#define DADOS_E ESTRUTURAS

/*****
/*Samuel Anjos nº 13305 */
/*Luis Camilo nº13987 */
*****/

/*****/
/*declaração de enumeradores*/
/*****/

/* tipo de funcionario */
enum tipoFuncionario {
    NORMAL,
    ADMINISTRATIVO
};
/* boolean*/
enum boolean {
    NAO,
    SIM
};
/* funcionalidade do Empréstimo*/
enum finalidadeEmprst {
    COMPRA,
    CONSTRUCAO,
    OBRAS
};
/* destino da habitação, para que uso*/
enum destHabitacao {
    HAB_PROP_PERMANENTE,
    HAB_SECUNDARIA,
    HAB_PARA_ARRENDAMENTO
};
/* tipo de imovel*/
enum tipoImovel {
    APARTAMENTO,
    MORADIA,
    TERRENO,
    ARMAZEM,
    GARAGEM
};
/* local do imovel */
enum localImovel {
    CONTINENTE,
    REGIAO_AUTONOMA,
};
/* tipo de taxa de juro*/
enum taxaJuro{
    INDX_EURIBOR_3MES,
    INDX_EURIBOR_6MES,
    TAXA_FIXA
};

/*****/
/* declaração de estruturas */
/*****/

/* estrutura para os dados referente ao empréstimo*/
struct dadosEmprestimo
{
    enum finalidadeEmprst finalidEmprestimo;
    enum destHabitacao destinoHabit;
    enum tipoImovel tipoImovl;
    enum boolean garagem;
    enum localImovel localDoImovel;
    enum taxaJuro taxaDeJuro;
    enum boolean creditoDeficiente;
    enum boolean imovelBanco;
    enum boolean clienteProtocol;
    int montanteSolicitado;
    int valorEscritura;
};

```

```

    int prazoEmprest;
};

/* estrutura para os produtos e serviços do cliente*/
struct produtosServicos
{
    enum boolean cartaoCredito;
    enum boolean cartaoDebito;
    enum boolean bancoOnline;
    enum boolean domiciPagmtPerid;
};

/* estrutura para os dados da morada */
struct morada
{
    char rua [30];
    char codigoPostal [25];
    char localidade [20];
};
/* estrutura para os dados da data */
struct data
{
    int dia;
    int mes;
    int ano;
};
/* estrutura para os dados pessoais */
struct dadosPessoais
{
    char nomeProprio [20];
    char apelido [20];
    int nif;
    int telefone;
};

/* estrutura para os serviços dos promponentes pretendem aderir*/
struct servicosProponentes
{
    int quantSegurVida;
    enum boolean seguroMultirrisc;
    enum boolean domicilRendimentos;
    struct produtosServicos produtosServicos;
};

/* dados para o segundo proponente */
typedef struct segundoProponente
{
    enum boolean existe;
    struct dadosPessoais dadsPessoaisSegPropont;
    struct data dataNascimento;
    float rendAnualBruto;
};

/* estrutura para os dados do funcionario */
typedef struct funcionario
{
    struct dadosPessoais dadosPessoais;
    enum tipoFuncionario cargo ;
    char username [20];
    char password [10];
};
/* estrutura para os dados do cliente */
typedef struct cliente
{
    struct dadosPessoais dadosClt;
    struct morada moradaClt;
    struct data dataNascimento;
};

/* estrutura para os dados pessoais necessarios para a simulacao de credito */
typedef struct dadosPessoaisSimulacao
{

```

```

    float enrgMenslOutrCreditos;
    int aggrFamiliar;
    float rendAnualBruto;
    int contaPupancaHabit;
    struct segundoProponente segundoProponente;
    struct cliente cliente;
    struct dadosEmprestimo dadosEmprestimo;
    struct servicosProponentes servicosProponentes;
};

typedef struct simulacao
{
    int nProcesso;
    struct dadosPessoais dadosCliente;
    struct segundoProponente dadosProponente;
    struct data dataSimulacao;
    float spread;
    float prestacaoMensal;
    int mesesEmprestimo;
};

/* estrutura para a lista ligada dos clientes */
typedef struct listaClientes
{
    struct cliente cliente;
    struct listaClientes *next;
};

/* estrutura para a lista ligada dos funcionarios */
typedef struct listaFuncionarios
{
    struct funcionario funcionarios;
    struct listaFuncionarios *next;
};

/* estrutura para a lista ligada das simulacoes */
typedef struct listaSimulacoes
{
    struct simulacao simulacao;
    struct listaSimulacoes *next;
};

/* estrutura para a função setEcra */
typedef struct small_rect {
    short Left;
    short Top;
    short Right;
    short Bottom;
}small_rect;

/*
    lista funções que pertencem a menus.h
*/
struct listaFuncionarios * menuControloFuncionarios(struct listaFuncionarios * head);
struct listaFuncionarios * menuCriaNovoFuncionario();
void menuEditarFuncionario( struct listaFuncionarios * current );
int perguntaUtilizador();
void menuLogin(char * username, char * password);
void alteraCorTexto(int cor);
void menuAdministrativos(struct listaFuncionarios * utilizador,
                        struct listaFuncionarios * funcionariosHead,
                        struct listaClientes * clientesHead,
                        struct listaSimulacoes *simulacoesHead);
struct listaClientes * menuControloCliente(struct listaClientes * head);
struct listaClientes * menuCriaNovoCliente();
void menuEditarCliente(struct listaClientes *current);
void setEcra();
void menuSimularCredito(char user[50], struct listaClientes * head, struct listaSimulacoes * simulacoesHead) ✓
;
void menuFuncionario(struct listaFuncionarios * utilizador, struct listaClientes * clientesHead, struct ✓
    listaSimulacoes * simulacoesHead);
void menuOutputSimulacao(char user[50], struct dadosPessoaisSimulacao * dadosSimulacao, struct ✓
    dadosPessoaisSimulacao * simulacaoHead);

/*

```

```

    lista funções que pertencem a funcionarios.c
*/
struct listaFuncionarios * adicionarFuncionario ( struct listaFuncionarios * head);
struct listaFuncionarios * imprimeFuncionarios ( struct listaFuncionarios * head);
void printFuncionario (struct listaFuncionarios * head);
void editarFuncionario(struct listaFuncionarios * head);
struct listaFuncionarios * procuraNodeFuncionario( struct listaFuncionarios * head , int *nif );
struct listaFuncionarios * apagaFuncionario(struct listaFuncionarios * head);
struct listaFuncionarios* deleteHeadFuncionario(struct listaFuncionarios*head);
struct listaFuncionarios * deleteUltimoFuncionario(struct listaFuncionarios * head);
struct listaFuncionarios * removerFuncionario(struct listaFuncionarios *head, struct listaFuncionarios *
    current);
void escreveFuncionarios ( struct listaFuncionarios * head);
struct listaFuncionarios * carregaFuncionarios (struct listaFuncionarios * head);
struct listaFuncionarios * procuraFuncionarioLogin(struct listaFuncionarios * head, char * username, char *
    password);
struct listaFuncionarios * loginFuncionarios(struct listaFuncionarios * head);
//void menuControloCliente();

/*
* lista funções que pertencem a cliente.c
*/
struct listaClientes * adicionarCliente(struct listaClientes * head);
struct listaClientes * imprimeClientes(struct listaClientes * head);
void printCliente(struct listaClientes * current);
void escreveClientes(struct listaClientes * head);
void libertaListaClientes(struct listaClientes *head);
struct listaClientes * carregaClientes(struct listaClientes * head);
void editarCliente(struct listaClientes *head);
struct listaClientes * procuraNodeCliente(struct listaClientes * head, int *nif);
struct listaClientes * apagaCliente(struct listaClientes *head);
struct listaClientes * removerCliente(struct listaClientes *head, struct listaClientes *current);
struct listaClientes* deleteHeadClientes(struct listaClientes*head);
struct listaClientes * deleteUltimoClientes(struct listaClientes * head);
struct listaClientes * procuraNodeAplellido(struct listaClientes * head, char * apelido);

/*
    lista funções que pertencem a pre_fichieros.c
*/
FILE * abreFicheiroBinario (char * nomeFicheiro , char *permissao);
void buscaData(char * data);
void libertaLista(struct listaFuncionarios *head);
int validaNumInteiros(int x);
int validaLetras(char *palavra);
int validaData(int dia, int mes, int ano);

/*
lista funções que pertencem a simulacao.c
*/
struct listaClientes * existeClienteNif(struct listaClientes * head);
void pedeDadosPessoais(char user[50], struct listaClientes * head, struct listaSimulacoes * simulacoesHead);
struct dadosPessoaisSimulacao * dadosPCliente(struct listaClientes * clienteCorrente);
struct dadosPessoaisSimulacao * dados2PCliente(struct dadosPessoaisSimulacao * dadosSimulacao);
int existeSegundoProponente();
struct dadosPessoaisSimulacao * dadosReferentesEmprestimo(struct dadosPessoaisSimulacao * dadosSimulacao);
struct dadosPessoaisSimulacao * dadosReferentesServicosProp(struct dadosPessoaisSimulacao * dadosSimulacao)
    ;
void encargosMensais(struct dadosPessoaisSimulacao * dadosSimulacao);

//void guardarSimulacaoFicheiro(struct dadosPessoaisSimulacao * dadosSimulacao, struct listaSimulacoes *
    simulacaoHead);

#endif

```