



# **RELATÓRIO DE PROJETO DE DESENVOLVIMENTO DE APLICAÇÕES**

Curso Técnico Superior Profissional em Programação de Sistemas de  
Informação

Gonçalo Leitão N° 2230238

Tiago Neto N° 2230239

Pedro Sousa N° 2231433

Janeiro 2024



# **RELATÓRIO DE PROJETO DE DESENVOLVIMENTO DE APLICAÇÕES**

Curso Técnico Superior Profissional em Programação de Sistemas de  
Informação

Gonçalo Leitão N° 2230238

Tiago Neto N° 2230239

Pedro Sousa N° 2231433

Relatório de projeto realizado sob a orientação do Professor Alexandre Frazão Rosário

Janeiro 2024





# Índice

<b>Lista de Figuras.....</b>	<b>v</b>
<b>Lista de Siglas e Acrónimos .....</b>	<b>vii</b>
<b>Introdução .....</b>	<b>1</b>
<b>1. Apresentação de Formulários e Funcionalidades.....</b>	<b>2</b>
<b>1.1. Startup e Login .....</b>	<b>2</b>
<b>1.2. Criação de User.....</b>	<b>3</b>
<b>1.3. Main Form – Formulário Principal.....</b>	<b>4</b>
<b>1.4. Barra Superior de Opções/User .....</b>	<b>5</b>
1.4.1. Edição de Users .....	5
1.4.2. Logout/Sair .....	6
<b>1.5. Extras.....</b>	<b>6</b>
1.5.1. Criação de Extras.....	6
1.5.2. Edição de Extras .....	7
<b>1.6. Pratos .....</b>	<b>8</b>
1.6.1. Criação de Pratos .....	8
1.6.2. Edição de Pratos .....	9
<b>1.7. Menus.....</b>	<b>10</b>
1.7.1. Criação de Menu.....	10
1.7.2. Edição de Menu .....	11
1.7.3. Visualização de Menu .....	12
<b>1.8. Clientes .....</b>	<b>13</b>
1.8.1. Criação de Clientes: Aluno/Professor .....	13
1.8.2. Edição de Clientes .....	14
1.8.3. Carregamento de Saldo.....	15
<b>1.9. Reservas.....</b>	<b>16</b>

<b>1.10. Multas .....</b>	<b>17</b>
1.10.1. Criar Multas.....	17
1.10.2. Editar Multas .....	18
<b>1.11. Registrar Refeição.....</b>	<b>19</b>
<b>2. Estrutura Usada .....</b>	<b>20</b>
<b>Conclusão .....</b>	<b>29</b>

# Lista de Figuras

Figura 1- Login Form .....	2
Figura 2- Registo de novo User .....	3
Figura 3- Main Form .....	4
Figura 4- Barra Superior de Opções (1).....	5
Figura 5- Gestão de Users .....	5
Figura 6- Barra Superior de Opções (2).....	6
Figura 7- Confirmação de fecho da Aplicação .....	6
Figura 8- Criação de Extras .....	6
Figura 9- Edição de Extras .....	7
Figura 10- Criação de Pratos .....	8
Figura 11- Edição de Pratos.....	9
Figura 12- Criação de Menu .....	10
Figura 13- Editar Menu .....	11
Figura 14- Display de Menu criado .....	12
Figura 15- Criação de Cliente Aluno .....	13
Figura 16- Criação de Cliente Professor .....	13
Figura 17- Edição de Cliente .....	14
Figura 18- Carregamento de Saldo .....	15
Figura 19- Realização de Reserva .....	16
Figura 20- Impressão de uma Reserva.....	17
Figura 21- Criação de Multas .....	17
Figura 22- Editar Multas.....	18
Figura 23- Registar Refeição .....	19
Figura 24- Disposição das pastas do aplicativo .....	20
Figura 25- Models usados.....	20
Figura 26- Modelo de Cliente.....	21
Figura 27- Modelo de Prato.....	22
Figura 28- Modelo de Extras .....	23

Figura 29- Modelo de Multa .....	24
Figura 30- Modelo de Menu (1).....	24
Figura 31- Modelo de Menu (2).....	25
Figura 32- Modelo de Professor.....	25
Figura 33- Modelo de Reserva (1) .....	26
Figura 34- Modelo de Reserva (2) .....	26
Figura 35- Modelo de Funcionário.....	27
Figura 36- Modelo de Estudante .....	27
Figura 37- Modelo de User .....	28



# Lista de Siglas e Acrónimos

Elemento a figurar, **quando aplicável**.

CTesP	Curso Técnico Superior Profissional
DA	Desenvolvimento de Aplicações
ESTG	Escola Superior de Tecnologia e Gestão
PSI	Programação de sistemas de Informação



# Introdução

O presente relatório foi elaborado no âmbito da unidade curricular de Desenvolvimento de Aplicações (DA), no Curso Técnico Superior Profissional (CTesP) de Programação de sistemas de Informação (PSI) da Escola Superior de Tecnologia e Gestão (ESTG). O trabalho foi realizado com a finalidade colocar em contexto prático os conhecimentos adquiridos durante as aulas de FP.

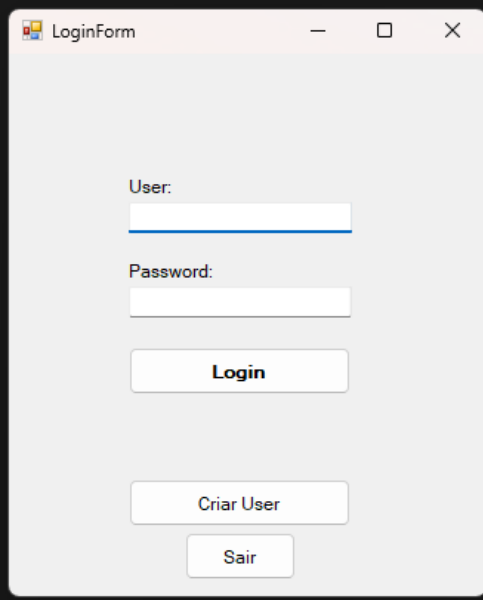
Foi proposta a elaboração de um sistema de gestão de cantina para uma escola, que permita gerir os menus assim como as funcionalidades necessárias para a gestão dos seus pratos e extras, clientes na qualidade de alunos ou docentes, reservas e multas.

O relatório, realizado com o objetivo de facilitar a compreensão do aplicativo, encontra-se organizado por capítulos onde iremos seccionar e apresentar as diferentes interações do programa, por último iremos apresentar uma breve conclusão, onde abordaremos o trabalho realizado e realizaremos reflexões de como a realização do mesmo decorreu.

# 1. Apresentação de Formulários e Funcionalidades

## 1.1.Startup e Login

Figura 1- Login Form



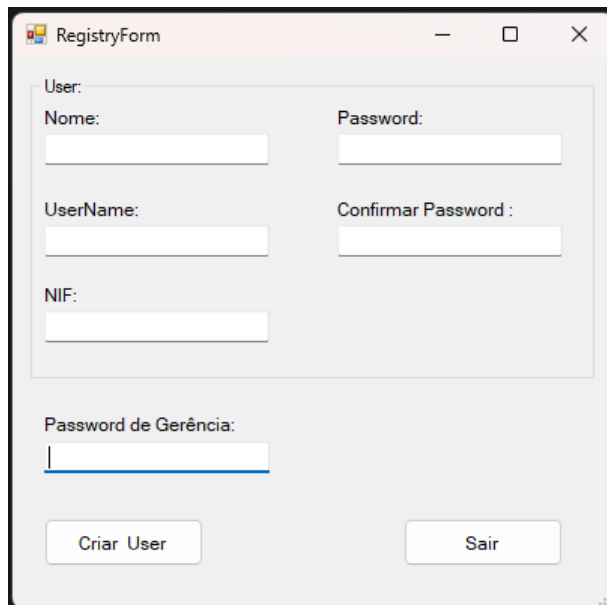
The image shows a window titled "LoginForm" with a standard Windows-style title bar (minimize, maximize, close buttons). The window has a light gray background. It contains two text input fields: the first is labeled "User:" and the second is labeled "Password:". Below the "Password:" field is a "Login" button. Further down are two more buttons: "Criar User" and "Sair".

A primeira interface com a qual o utilizador terá contacto será a de Login. Aqui deverão ser introduzidas as credenciais do funcionário da cantina, enquanto utilizador do programa.

Caso não tenha perfil criado para efetuar um login, um poderá ser criado através do botão “Criar User”. O Botão “sair” irá fechar imediatamente a aplicação.

## 1.2. Criação de User

Figura 2- Registo de novo User



The image shows a Windows-style application window titled "RegistryForm". Inside the window, there is a registration form with the following fields and labels:

- User:** (group label)
  - Nome:** (text input field)
  - Password:** (password input field)
  - UserName:** (text input field)
  - Confirmar Password :** (password input field)
  - NIF:** (text input field)
- Password de Gerência:** (password input field)

At the bottom of the form, there are two buttons: "Criar User" and "Sair".

No formulário de registo de User, o utilizador poderá preencher os campos com os dados necessários para a criação de um novo perfil. Estes terão de ser obrigatoriamente preenchidos, e de forma correta, como requerido pelo programa. O programa não aceitará dados preenchidos de forma incorreta.

A criação de um perfil não pode estar desprotegida a um membro externo que aceda ao programa. Deste modo é **obrigatório** que a criação seja acompanhada de uma Password de Gerência.

A Password de Gerência utilizada no projeto foi: "Admin2024".

### 1.3. Main Form – Formulário Principal

Figura 3- Main Form

The screenshot displays the 'MainForm' application window. At the top, there is a title bar with the text 'MainForm' and standard window controls. Below the title bar, a menu bar contains 'Opções' and 'User'. The main content area is divided into several sections. On the left, a 'Semanal' (Weekly) section shows a clock icon, the time '22:09:35', and the date '19 de junho de 2024'. Below this, a grid of buttons represents the weekly menu planning. The grid has five columns for the days of the week: 'Segunda-Feira', 'Terça-Feira', 'Quarta-Feira', 'Quinta-Feira', and 'Sexta-Feira'. Each column contains two buttons: 'Almoço' (Lunch) and 'Jantar' (Dinner). At the bottom of the grid, there are navigation arrows '<-' and '->' and a text label 'Semana: 17 de junho de 2024 a 21 de junho de 2024'. On the right side of the window, there are three distinct groups of buttons. The first group, labeled 'Cliente' (Client), includes 'Adicionar' (Add) and 'Editar' (Edit) buttons, followed by a 'Carregamento' (Loading) button. The second group, labeled 'Multas' (Fines), includes 'Criar Multa' (Create Fine) and 'Editar' (Edit) buttons. The third group, labeled 'Refeição' (Meal), includes 'Criar Menu' (Create Menu), 'Criar Prato' (Create Dish), 'Extras', and 'Fazer Reserva' (Make Reservation) buttons, each with an 'Editar' (Edit) button. At the bottom of this right-hand section is a prominent green button labeled 'Registrar Refeição' (Register Meal).

O painel ou formulário principal de operação da aplicação.

Permite ao utilizador acesso a todos os controlos necessários para gerir a cantina. A barra de opções superior permite o acesso à informação de utilizadores. Os grupos de botões do lado direito permitem controlar a gestão dos elementos necessários ao funcionamento da cantina como os clientes, pratos, extras, menus e multas. O

O grupo central do formulário apresenta um planeamento semanal, permite a visualização dos menus planeados para o correspondente dia da semana e horário. É possível ver o conjunto de dias correspondentes e escolher uma semana seguinte ou anterior ao usar as setas no fundo da janela.

## 1.4. Barra Superior de Opções/User

### 1.4.1. Edição de Users

Figura 4- Barra Superior de Opções (1)

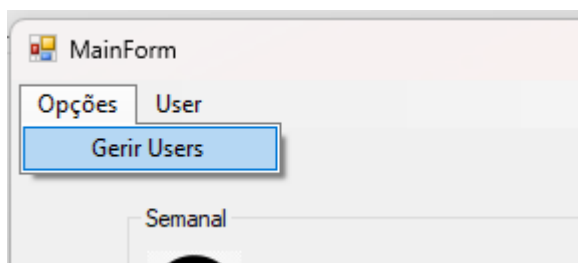


Figura 5- Gestão de Users

Nome:

UserName:

NIF:

Password:

ID: 1

Password de Gerência:

	Uusername	ID	Name	NIF
▶	pedro	1	Pedro	887766554
	tiago	7	tiago	987654321

Delete      Modificar      Sair

O formulário de Gestão de Users permite a edição ou a remoção dos dados de um perfil de utilizador. As alterações estão sujeitas às mesmas condições das de criação. Apenas funciona se for introduzida uma Password de Gerência.

A Password de Gerência utilizada no projeto foi: “Admin2024”.

## 1.4.2. Logout/Sair

Figura 6- Barra Superior de Opções (2)

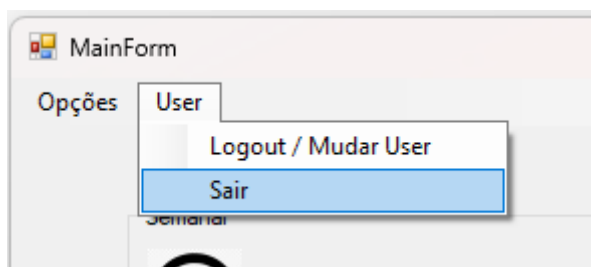
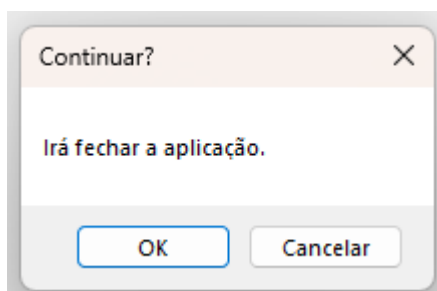


Figura 7- Confirmação de fecho da Aplicação



As seguintes opções da Barra de Opções permitem:

- Efetuar o Logout do utilizador de forma imediata, regressando ao Login;
- Fechar a aplicação, este passo passa por uma confirmação para evitar fechar acidentalmente.

## 1.5. Extras

### 1.5.1. Criação de Extras

Figura 8- Criação de Extras



No formulário de Criação de Extras será possível criar um produto de consumo do tipo “Extra”, isto será referente aos possíveis acompanhamentos dos pratos principais, incluindo bebidas, sobremesas, pão, entre outros, assim como o respectivo preço.

### 1.5.2. Edição de Extras

Figura 9- Edição de Extras

The screenshot shows a window titled "ExtrasEditForm" with standard Windows window controls. The form is divided into two main sections. The top section is for editing a specific extra item, and the bottom section is a table listing all extra items.

**Top Section (Form):**

- Extra:** A label indicating the current item being edited.
- Nome:** A text field containing "Água".
- ID:** A text field containing "1".
- Tipo:** A group of radio buttons with labels: "Sopa", "Sobremesa", "Bebida" (selected), and "Outro".
- Preço estudante:** A numeric input field containing "0,50".

**Bottom Section (Table):**

	ID	Description	Type	Price	Active	Reservations
▶	1	Água	Drink	0,5	✓	
	2	Sumo	Drink	1	✓	
	3	Canja	Soup	0,6	✓	
	4	Caldo Verde	Other	0,6	✓	

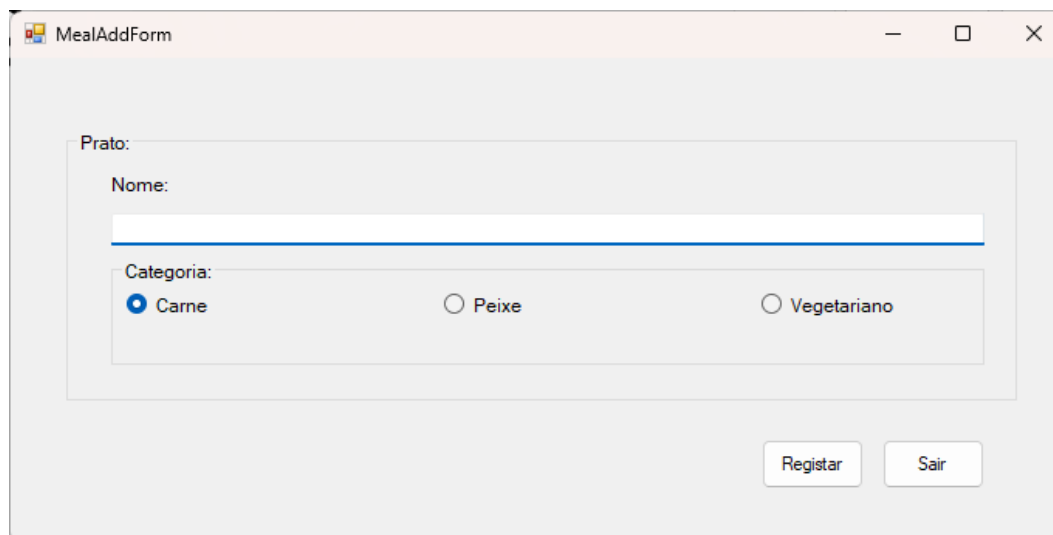
At the bottom of the form, there are three buttons: "Delete" (on the left), "Registrar" (on the right), and "Sair" (on the right).

O formulário de Edição permite ver, alterar e remover dados referentes aos bens de consumo do tipo “Extras”, utilizando um layout baseado no da criação.

## 1.6. Pratos

### 1.6.1. Criação de Pratos

Figura 10- Criação de Pratos



The screenshot shows a Windows-style window titled "MealAddForm". Inside the window is a form with the following elements:

- A label "Prato:" followed by a sub-label "Nome:" and a text input field.
- A label "Categoria:" followed by three radio button options: "Carne" (selected), "Peixe", and "Vegetariano".
- At the bottom right, there are two buttons: "Registrar" and "Sair".

O formulário de Criação de Pratos apresenta o mesmo layout do de extras. Será possível criar um prato, dentro de uma categoria pré-determinada (Carne, Peixe ou Vegetariano) para utilização nos menus. Não possui preço individual pois o preço de Menu é fixo.

### 1.6.2. Edição de Pratos

Figura 11- Edição de Pratos

MealEditForm

Prato:

Nome: ID: 1 ☒ Active

Francesinha

Categoria:

☒ Carne ☐ Peixe ☐ Vegetariano

	ID	Description	Type	Active	Reservations
▶	1	Francesinha	Meat	<input checked="" type="checkbox"/>	
	2	Lombo	Meat	<input checked="" type="checkbox"/>	
	3	Sardines	Fish	<input checked="" type="checkbox"/>	
	4	Pizza Veggie	Veggie	<input checked="" type="checkbox"/>	
	5	Salada	Veggie	<input checked="" type="checkbox"/>	

Delete Editar Sair

O formulário de Edição de pratos, semelhante ao de Extras, permite ver, alterar e remover dados dos pratos disponíveis para os menus, assim como o estatuto de “active”.

Se um prato em uso num Menu for apagado, o sistema pede confirmação e avisa o utilizador que os menus em que se encontra serão apagados, sendo necessário substituí-los.

## 1.7. Menus

### 1.7.1. Criação de Menu

Figura 12- Criação de Menu

The screenshot shows a Windows-style application window titled "MenuAddForm". The interface is organized into several sections:

- Data:** A date picker set to "19 de junho de 2024" and two radio buttons for meal type: "Almoço" (selected) and "Jantar".
- Menu:** A section containing three dropdown menus for "Prato":
  - Carne:** "francesinha"
  - Peixe:** "Sardines"
  - Vegetariano:** "Pizza vegetariana"
- Extras:** A section with a "Filtros:" row of radio buttons: "Todos" (selected), "Sopas", "Sobremesas", "Bebidas", and "Outros".
- Extras Disponíveis:** A list box containing "Pão" and "Água".
- Extras Selecionados:** A list box containing "canja".
- Buttons:** Between the two list boxes are "Selecionar ->" and "<- Remover" buttons.
- Outros:** At the bottom, two numeric input fields: "Quantidade:" with the value "100" and "Preço base:" with the value "4,50".
- Footer:** Two buttons at the bottom right: "Registrar" and "Sair".

O formulário de Criação de Menu permite a atribuição de uma combinação de tipos de Pratos (criados previamente) e de Extras, num Menu, assim como a quantidade disponível e respetivo preço. O Menu é atribuído a uma data e horário (almoço ou jantar).

### 1.7.2. Edição de Menu

Figura 13- Editar Menu

The screenshot shows the 'MenuEdit' application window. On the left, a list of menus is displayed, with 'Menu: 19-06-2024 - Jantar' selected. Below this list, there are input fields for 'Outros: Quantidade' (set to 7) and 'Preço base' (set to 4,50). At the bottom left are buttons for 'Editar', 'Eliminar', and 'Sair'. The main area on the right is titled 'Menu' and contains several sections: 'Data' with a date picker set to '19 de junho de 2024' and radio buttons for 'Almoço' and 'Jantar' (selected); 'Prato' with dropdown menus for 'Carne' (Francesinha), 'Peixe' (Sardines), and 'Vegetariano' (Pizza Veggie); 'Extras' with radio buttons for 'Filtros' (Todos, Sopas, Sobremesas, Bebidas, Outros) where 'Todos' is selected; and two lists of extras. 'Extras Disponíveis' contains 'Sumo' and 'Caldo Verde', while 'Extras Seleccionados' contains 'Água' and 'Canja'. Between these lists are buttons for 'Selecionar ->' and '<- Remover'.

Ao carregar no botão editar o utilizador é levado para o formulário de edição de Menu, onde pode rever, atualizar ou eliminar a informação do Menu criado previamente

### 1.7.3. Visualização de Menu

Figura 14- Display de Menu criado

The screenshot displays the 'MenuDisplay' application window. At the top, there is a date selector set to '19 de junho de 2024' and radio buttons for 'Almoço' (selected) and 'Jantar'. Below this, the 'Pratos' section contains three dropdown menus: 'Came' (Francesinha), 'Peixe' (Sardines), and 'Vegetariano' (Pizza Veggie). The 'Extras' section features a list box with 'Canja' selected. The 'Outros' section includes input fields for 'Quantidade' (7) and 'Preço base' (4,50). At the bottom of this window are 'Registrar' and 'Sair' buttons. In the background, another window is partially visible, showing sections for 'Cliente' (with 'Adicionar' and 'Editar' buttons), 'Multas' (with 'Criar Multa' and 'Editar' buttons), and 'Refeição' (with 'Criar Menu', 'Criar Prato', 'Extras', and 'Fazer Reserva' buttons, plus a green 'Registrar Refeição' button). The bottom of the main window shows a weekly calendar grid with 'Almoço' and 'Jantar' buttons for each day, and a status bar indicating 'Semana: 17 de junho de 2024 a 21 de junho de 2024'.

Ao carregar num dos horários do calendário semanal apresentado no formulário principal é possível aceder ao Menu criado para o respetivo dia/horário.

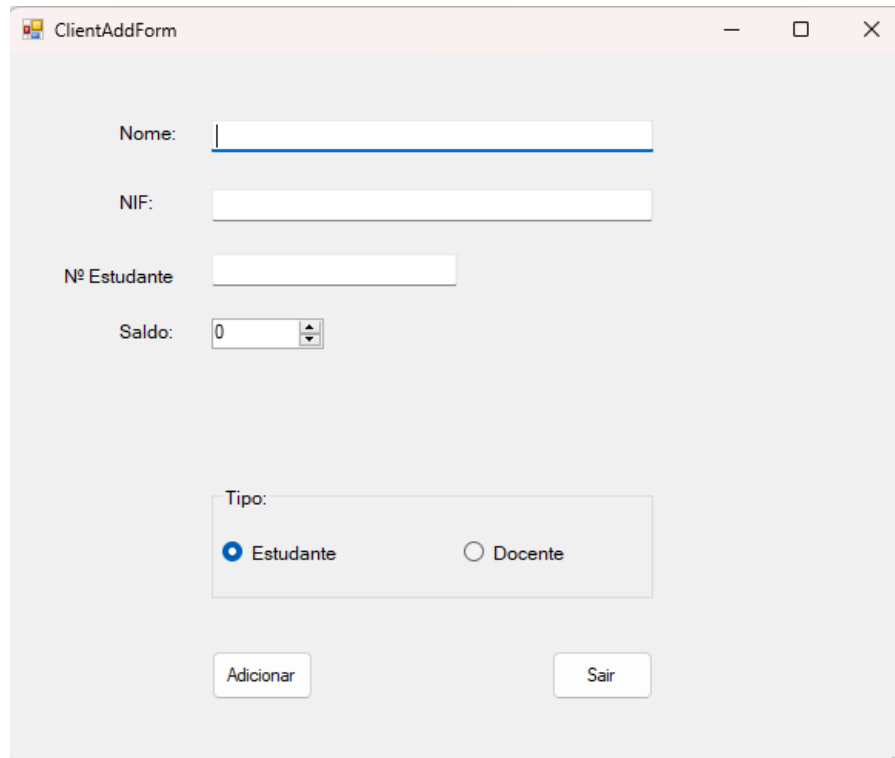
A janela apresenta todas as informações guardadas aquando da criação do Menu, assim como a quantidade disponível atualmente.

Ao premir o botão “Registrar” o utilizador é levado de volta para o formulário de edição.

## 1.8. Clientes

### 1.8.1. Criação de Clientes: Aluno/Professor

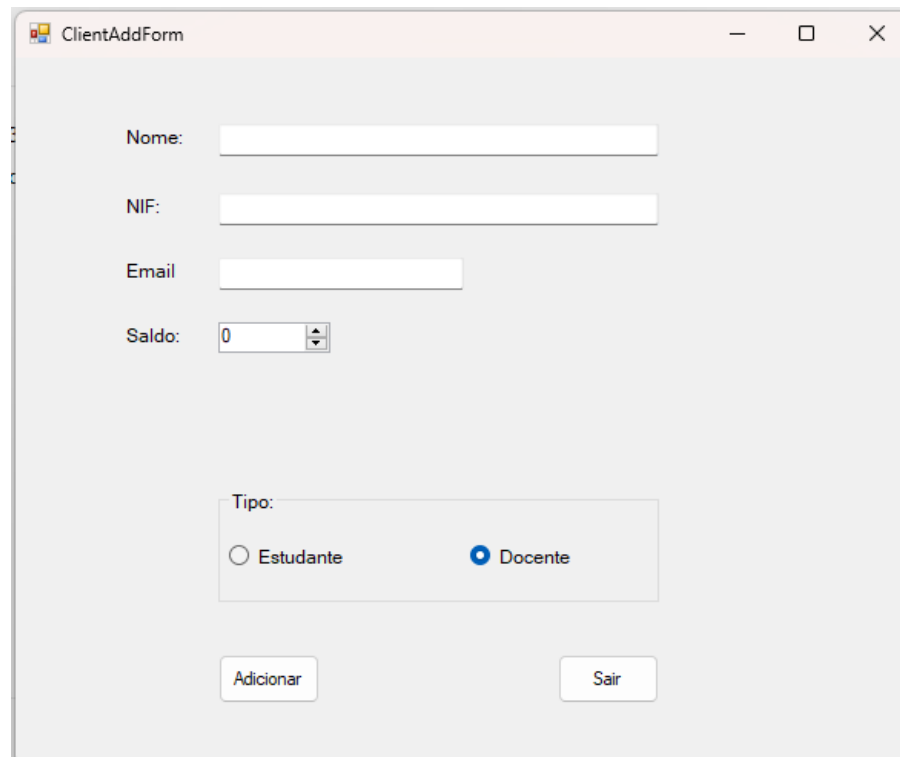
Figura 15- Criação de Cliente Aluno



The screenshot shows a Windows-style window titled "ClientAddForm". It contains the following fields and controls:

- Nome:** A text input field.
- NIF:** A text input field.
- Nº Estudante:** A text input field.
- Saldo:** A numeric input field with a spinner, currently showing "0".
- Tipo:** A group box containing two radio buttons: "Estudante" (which is selected) and "Docente".
- Buttons:** "Adicionar" and "Sair" at the bottom.

Figura 16- Criação de Cliente Professor



The screenshot shows the same "ClientAddForm" window, but with the "Docente" radio button selected under the "Tipo:" group box. The other fields (Nome, NIF, Email, Saldo) and buttons (Adicionar, Sair) are identical to the previous figure.

A criação de Clientes é feita no mesmo formulário, os campos a preencher são apropriadamente alterados de acordo com o tipo de Cliente selecionado.

### 1.8.2. Edição de Clientes

Figura 17- Edição de Cliente

BalanceForm

Pesquisa:

NIF:  Procurar

Resultado:

Nome:

NIF:

Tipo: ☒ Aluno ☐ Docente

Nº Estudante

Saldo:

Modificar Apagar Sair

Clientes:

- Aluno: pedro miguel - 123
- Professor: Zé Carlos - prof@prof.pt
- Aluno: Tiago - 39
- Aluno: Gonçalo - 38

Mostrar Todos

O formulário de edição de dados de Cliente permite procurar clientes por NIF ou selecionar por lista, os dados são depois exibidos, incluindo o seu saldo disponível, no entanto apenas a informação pessoal será editável, o saldo não será acessível neste formulário.



### 1.8.3. Carregamento de Saldo

Figura 18- Carregamento de Saldo

The screenshot shows a window titled "BalanceForm" with a light gray background. It contains several sections:

- Pesquisa:** A search section with a label "NIF:" followed by a text input field and a "Procurar" button.
- Resultado:** A section displaying the search results for a specific NIF. It shows:
  - Nome:** pedro miguel
  - NIF:** 123456789
  - Tipo:** Aluno
  - Saldo:** 2,00€
- Saldo:** A section for managing the balance, featuring a numeric input field with "0,00", an "Adicionar" button, and a "Sair" button.
- Cientes:** A list box on the right side containing the following entries:
  - Aluno: pedro miguel - 123
  - Professor: Zé Carlos - prof@prof.pt
  - Aluno: Tiago - 39
  - Aluno: Gonçalo - 38
- Mostrar Todos:** A button located at the bottom right of the list box.

O formulário de carregamento permite procurar por NIF ou selecionar por lista um Cliente, exibir os seus dados, incluindo o seu saldo disponível, e de seguida indicar o valor a adicionar ao seu saldo já existente.

## 1.9. Reservas

Figura 19- Realização de Reserva

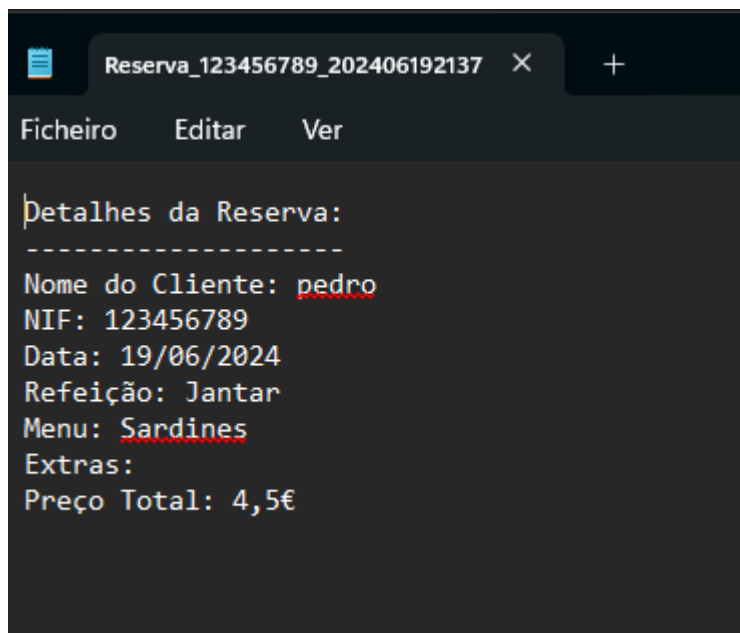
The screenshot shows a web application window titled "ReservationForm". It contains several sections for user input and display:

- Pesquisa:** A search section with a "NIF:" label, a text input field containing "123456789", and a "Procurar" button.
- Calendar:** A calendar for "junho de 2024". The date "19" is selected and highlighted. Below the calendar, it says "Hoje: 19/06/2024".
- Horário:** A section with two radio buttons: "Almoço" and "Jantar". The "Jantar" option is selected.
- Menu:** A section with three radio buttons: "Francesinha", "Sardines", and "Pizza Veggie". The "Francesinha" option is selected.
- Resultado:** A section displaying the reservation details: "Nome: pedro", "NIF: 123456789", "Data:", "Menu: Francesinha", and "Price: 5,1€".
- Extras:** A section with a "Filtros:" row containing five radio buttons: "Todos", "Sopas", "Sobremesas", "Bebidas", and "Outros". The "Todos" option is selected.
- Extras Disponíveis:** A large empty rectangular box for displaying available extras.
- Extras Seleccionados:** A rectangular box containing the text "Canja".
- Buttons:** Between the "Extras Disponíveis" and "Extras Seleccionados" boxes are two buttons: "Selecionar ->" and "<- Remover".
- Footer:** At the bottom of the window are three buttons: "Registar", "Imprimir", and "Sair".

O Formulário permite identificar um Cliente através do NIF e realizar uma Reserva de um Menu e Extras Disponíveis, adapta os menus apresentados conforme a altura do dia seleccionada (Almoço, Jantar) e regista o prato escolhido, os extras, o cliente e os valores.

Após a realização de uma Reserva é impresso pelo sistema um documento com os detalhes em formato .txt.

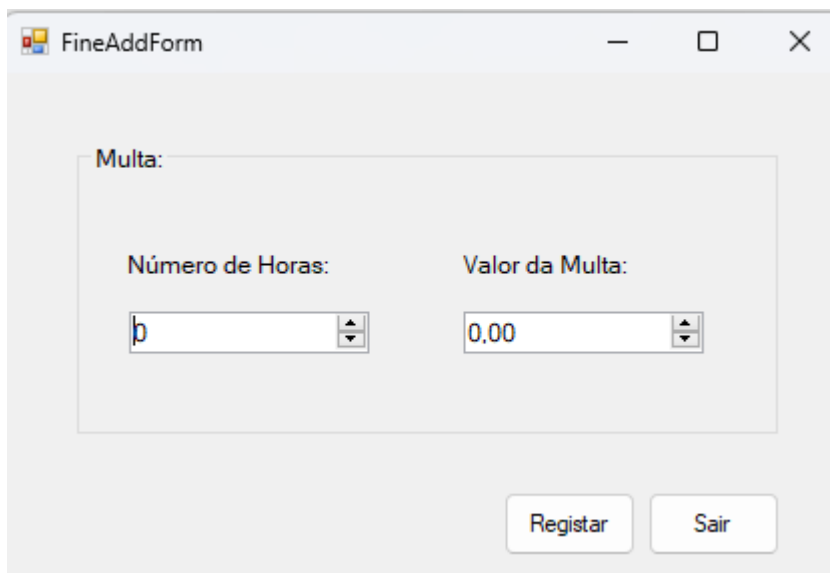
Figura 20- Impressão de uma Reserva



## 1.10. Multas

### 1.10.1. Criar Multas

Figura 21- Criação de Multas

A screenshot of a Windows-style application window titled 'FineAddForm'. Inside the window, there is a label 'Multa:' followed by a rectangular container. Within this container, there are two labels: 'Número de Horas:' and 'Valor da Multa:'. Below 'Número de Horas:' is a text box containing the letter 'p' and a small up/down arrow icon. Below 'Valor da Multa:' is a text box containing '0,00' and a similar up/down arrow icon. At the bottom right of the window, there are two buttons: 'Registar' and 'Sair'.

A criação de multas funciona através da atribuição de quantas horas faltam até à hora da refeição e seu consequente valor

Se um número de horas for 3, então, se for feita uma reserva 3 horas antes da hora refeição, o valor da multa será o associado a esse valor.

### 1.10.2. Editar Multas

Figura 22- Editar Multas

The screenshot shows a window titled 'FineEditForm'. Inside, there is a form labeled 'Multa:' with the following fields:

- ID: 1
- Número de Horas: 1
- Valor da Multa: 1.00

Below the form is a table with the following data:

	ID	NumHours	Value
▶	1	3	1
	2	2	2
	3	1	4

At the bottom of the window, there are three buttons: 'Delete', 'Registar', and 'Sair'.

Este Formulário permite a visualização, o controlo e a remoção das multas criadas.

Na Figura é possível perceber que quanto mais próximo da hora de refeição maior é o valor da multa a plicar.

## 1.11. Registrar Refeição

Figura 23- Registrar Refeição

The screenshot shows a Windows application window titled "MealRegisterForm". The interface is divided into several sections:

- Pesquisa:** A search section with a text input field labeled "NIF:" and a button labeled "Procurar".
- Cliente:** A section displaying client information:
  - Nome:** pedro
  - NIF:** 123456789
  - Tipo:** Aluno
  - Saldo:** -1,50
- Menu:** A section for selecting menu items:
  - Prato:** A dropdown menu currently showing "Sardines".
  - Preço:** A text input field containing "4,5".
  - Horário:** A dropdown menu currently showing "Jantar".
  - Extras:** An empty rectangular box for additional items.
- Reservas por marcar:** A list box on the right side of the window. It currently contains one entry: "ID Reserva: 3 | NIF: 123456789".
- Buttons:** At the bottom right, there are two buttons: "Mostrar Todas" (white) and "Registrar Reserva" (green).

O formulário de registo de Refeição apresenta uma lista com as reservas feitas com respetivos pedidos e clientes para aquele dia com o Horário apropriado, assim como o valor a pagar.

Apenas são mostradas as reservas do próprio dia. Sendo que as reservas de dias futuros não são exibidas.

Após aceite o registo, a Reserva é retirada da lista e o saldo do Cliente é atualizado, assim como as quantidades de refeições disponíveis.

## 2. Estrutura Usada

O projeto utiliza uma estrutura MVC, dividindo as lógicas de modelo, controlo e visualização em pastas separadas e apropriadas á sua função.

Figura 24- Disposição das pastas do aplicativo

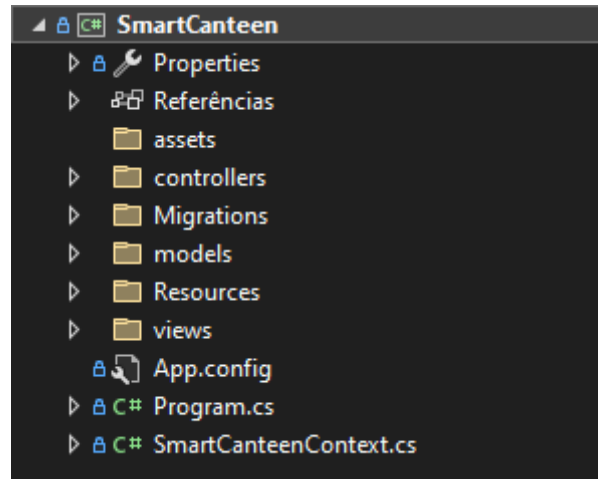


Figura 25- Models usados

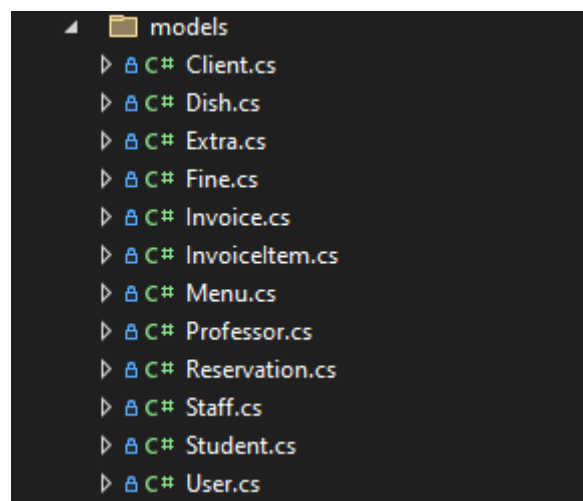


Figura 26- Modelo de Cliente

```
namespace SmartCanteen.models
{
    30 referências
    abstract class Client : User
    {
        7 referências
        public decimal Balance { get; set; }

        2 referências
        public Client(string name, string nif, decimal balance) : base(name, nif)
        {
            string Name = name;
            string Nif = nif;
            decimal Balance = balance;
        }

        0 referências
        public Client()
        {
        }

        2 referências
        public override string ToString()
        {
            return base.ToString();
        }
    }
}
```

Figura 27- Modelo de Prato

```
namespace SmartCanteen.models
{
    32 referências
    public enum DishType
    {
        Meat,
        Fish,
        Veggie
    }

    47 referências
    internal class Dish
    {
        [Key]
        16 referências
        public int ID { get; set; }
        11 referências
        public string Description { get; set; }
        11 referências
        public DishType Type { get; set; }
        6 referências
        public bool Active { get; set; }

        3 referências
        public virtual ICollection<Menu> Menus { get; set; }
        2 referências
        public virtual ICollection<Reservation> Reservations { get; set; }
        1 referência
        public Dish()
        {
            Menus = new HashSet<Menu>();
            Reservations = new HashSet<Reservation>();
        }

        1 referência
        public Dish(string description, DishType dishType)
        {
            Description = description;
            Type = dishType;
            Active = true;
        }
    }
}
```



Figura 28- Modelo de Extras

```
public enum ExtraType
{
    Soup,
    Dessert,
    Drink,
    Other
}
46 referências
internal class Extra
{
    [Key]
    17 referências
    public int ID { get; set; }
    7 referências
    public string Description { get; set; }
    10 referências
    public ExtraType Type { get; set; }
    6 referências
    public double Price { get; set; }
    3 referências
    public bool Active { get; set; }

    2 referências
    public virtual ICollection<Menu> Menus { get; set; }
    2 referências
    public virtual ICollection<Reservation> Reservations { get; set; }
    1 referência
    public Extra()
    {
        Menus = new HashSet<Menu>();
        Reservations = new HashSet<Reservation>();
    }

    1 referência
    public Extra(string description, ExtraType extraType, double price)
    {
        Description = description;
        Type = extraType;
        Price = price;
        Active = true;
    }
}
```

Figura 29- Modelo de Multa

```

namespace SmartCanteen.models
{
    9 referências
    internal class Fine
    {
        [Key]
        5 referências
        public int ID { get; set; }
        7 referências
        public int NumHours { get; set; }
        7 referências
        public double Value { get; set; }

        1 referência
        public Fine()
        {
        }

        1 referência
        public Fine(int numHours, double value)
        {
            NumHours = numHours;
            Value = value;
        }
    }
}

```

Figura 30- Modelo de Menu (1)

```

internal class Menu
{
    [Key]
    8 referências
    public int ID { get; set; }
    7 referências
    public DateTime Date { get; set; }
    7 referências
    public int Quantity { get; set; }
    5 referências
    public double StudentPrice { get; set; }
    3 referências
    public double TeacherPrice { get; set; }
    8 referências
    public MealTime Time { get; set; }

    12 referências
    public virtual ICollection<Dish> Dishes { get; set; }
    16 referências
    public virtual ICollection<Extra> Extras { get; set; }
    2 referências
    public virtual ICollection<Reservation> Reservations { get; set; }

    0 referências
    public Menu()
    {
        this.Reservations = new HashSet<Reservation>();
    }
}

```

Figura 31- Modelo de Menu (2)

```

1 referência
public Menu(DateTime date, int quantity, double price, List<Dish> dishes, List<Extra> extras, MealTime time)
{
    Date = date;
    Quantity = quantity;
    StudentPrice = price;
    TeacherPrice = price + 0.6;
    Dishes = dishes;
    Extras = extras;
    Time = time;
}

0 referências
public override string ToString()
{
    string meatDishes = string.Join(", ", Dishes
        .Where(dish => dish.Type == DishType.Meat)
        .Select(dish => dish.Description));
    if (Time == MealTime.Dinner)
        return $"Menu: {Date.ToString("dd-MM-yyyy")} - Jantar";
    else
        return $"Menu: {Date.ToString("dd-MM-yyyy")} - Almoço";
}
}

```

Figura 32- Modelo de Professor

```

namespace SmartCanteen.models
{
    6 referências
    internal class Professor : Client
    {
        4 referências
        public string Email { get; set; }
        1 referência
        public Professor(string name, string nif, decimal balance, string email) : base(name, nif, balance)
        {
            Name = name;
            NIF = nif;
            Balance = balance;
            Email = email;
        }

        0 referências
        public Professor()
        {
        }

        1 referência
        public override string ToString()
        {
            return "Professor: " + Name + " - " + Email;
        }
    }
}

```

Figura 33- Modelo de Reserva (1)

```

namespace SmartCanteen.models
{
    28 referências
    internal class Reservation
    {
        [Key]
        3 referências
        public int ID { get; set; }
        3 referências
        public DateTime Date { get; set; }
        2 referências
        public MealTime MealTime { get; set; }
        2 referências
        public double TotalPrice { get; set; }
        5 referências
        public bool Used { get; set; }
        //Menu
        2 referências
        public int MenuID { get; set; }
        3 referências
        public virtual Menu Menu { get; set; }
        //User
        2 referências
        public int UserID { get; set; }
        6 referências
        public virtual User User { get; set; }
        //Dish
        2 referências
        public int DishID { get; set; }
        4 referências
        public virtual Dish Dish { get; set; }
        //Extra
        6 referências
        public virtual ICollection<Extra> Extras { get; set; }

        0 referências
        public Reservation() { Extras = new HashSet<Extra>(); }
    }
}

```

Figura 34- Modelo de Reserva (2)

```

1 referência
public Reservation(DateTime date, MealTime mealTime, double totalPrice, int menuId, int clientId, int dishId, List<Extra> extras)
{
    Date = date;
    MealTime = mealTime;
    TotalPrice = totalPrice;
    MenuID = menuId;
    UserID = clientId;
    DishID = dishId;
    Extras = extras;
    Used = false;
}

0 referências
public override string ToString()
{
    return "ID Reserva: " + ID + " | NIF: " + User.NIF;
}
}

```

Figura 35- Modelo de Funcionário

```
namespace SmartCanteen.models
{
    12 referências
    internal class Staff : User
    {
        6 referências
        public string Username { get; set; }
        6 referências
        public string Password { get; set; }

        1 referência
        public Staff(string name, string nif, string username, string password) : base(name, nif)
        {
            Name = name;
            NIF = nif;
            Username = username;
            Password = password;
        }

        1 referência
        public Staff()
        {
        }
    }
}
```

Figura 36- Modelo de Estudante

```
namespace SmartCanteen.models
{
    9 referências
    internal class Student : Client
    {
        4 referências
        public string StudentID { get; set; }

        1 referência
        public Student(string name, string nif, decimal balance, string studentID) : base(name, nif, balance)
        {
            Name = name;
            NIF = nif;
            Balance = balance;
            StudentID = studentID;
        }

        0 referências
        public Student()
        {
        }

        1 referência
        public override string ToString()
        {
            return "Aluno: " + Name + " - " + StudentID;
        }
    }
}
```

Figura 37- Modelo de User

```
namespace SmartCanteen.models
{
    11 referências
    abstract class User
    {
        [Key]
        7 referências
        public int ID { get; set; }
        16 referências
        public string Name { get; set; }
        24 referências
        public string NIF { get; set; }
        2 referências
        public virtual ICollection<Reservation> Reservations { get; set; }

        2 referências
        public User(string name, string nif)
        {
            string Name = name;
            string Nif = nif;
        }

        0 referências
        public User()
        {
            this.Reservations = new HashSet<Reservation>();
        }
    }
}
```

## Conclusão

Consideramos o projeto proposto para Desenvolvimento de Aplicações como um projeto muito completo, que requeria elementos e colocação em prática de diversos tipos de conhecimento, incluindo o de outras cadeiras.

Foi um bom exercício de construção de um projeto de raiz, focado em apresentar um produto final o mais próximo possível das necessidades do cliente. Consideramos ter sido um projeto desafiante capaz de colocar à prova o conhecimento dos alunos menos experientes e também dos membros com mais experiência.

Estamos satisfeitos com o trabalho realizado relativamente à funcionalidade e também à aparência simples mas funcional da aplicação que consegue satisfazer as necessidades de um utilizador sem se tornar confusa ou visualmente desagradável. Lamentamos no entanto a não inclusão da geração de um documento em formato PDF.