



UNIVERSIDADE  
**LUSÓFONA**

# **Faz-e-Conta: Solução para a Gestão de Dados e Relatórios Financeiros de IPSSs de Educação Infantil**

## **Trabalho Final de curso**

Relatório Intercalar 2º Semestre

Bruno Borges, a22207345, LEI

Bernardo Ildefonso, a22300788, LEI

Orientador: Ana Cruz

Co-orientador: Andreas Klaus

**Entidades Externas: Via Insight, Lda. e ADECO - Associação de Desenvolvimento Comunitário  
da Freguesia das Mercês**

Departamento de Engenharia Informática da Universidade Lusófona

Centro Universitário de Lisboa

03 de dezembro de 2024

---

## **Direitos de cópia**

(*Faz-e-Conta: Faz-e-Conta: Solução para a Gestão de Dados e Relatórios Financeiros de IPSSs de Educação Infantil*), Copyright de (Bruno Borges, Bernardo Ildefonso), ULHT.

A Escola de Comunicação, Arquitectura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona de Humanidades e Tecnologias (ULHT) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

---

## Agradecimentos

---

## Resumo

Este trabalho final de curso propõe o desenvolvimento de uma aplicação tecnológica open-source destinada a apoiar as Instituições Particulares de Solidariedade Social (IPSS) que atuam na área da educação infantil. A solução visa enfrentar os desafios financeiros e administrativos destas instituições, especialmente num contexto de mudanças legislativas e operacionais, como a implementação da gratuidade progressiva das creches/jardins de infância e a revisão das comparticipações estatais.

A aplicação oferece funcionalidades como gestão centralizada de dados de alunos e encarregados de educação, cálculo automatizado de mensalidades com base em escalões de rendimento, geração de relatórios financeiros customizados e comunicação eficiente com as famílias. Alinhada aos Objetivos de Desenvolvimento Sustentável, a solução busca promover a transparéncia, a eficiência administrativa e a sustentabilidade financeira das IPSS, permitindo-lhes otimizar recursos e expandir seu impacto social.

Este relatório detalha as etapas de desenvolvimento do projeto, incluindo análise de requisitos, benchmarking de soluções existentes, modelação de dados e arquitetura, bem como o planeamento e validação do sistema. A proposta demonstra inovação ao aliar acessibilidade e eficiência numa plataforma robusta e escalável, adaptada às necessidades específicas das IPSS.

**Palavras-chave:** IPSS, gestão financeira, educação infantil, aplicação tecnológica, open-source.

---

# Abstract

This final course project proposes the development of an open-source technological application designed to support Private Institutions of Social Solidarity (IPSS) operating in the field of early childhood education. The solution aims to address the financial and administrative challenges faced by these institutions, particularly in light of legislative and operational changes, such as the progressive implementation of free childcare programs and the revision of state subsidies.

The application provides functionalities such as centralized management of student and guardian data, automated calculation of tuition fees based on income brackets, customized financial report generation, and efficient communication with families. Aligned with the Sustainable Development Goals, the solution seeks to promote transparency, administrative efficiency, and financial sustainability, enabling IPSS to optimize resources and expand their social impact.

This report outlines the project's development stages, including requirements analysis, benchmarking of existing solutions, data modeling, and system architecture, as well as planning and system validation. The proposal demonstrates innovation by combining accessibility and efficiency in a robust and scalable platform tailored to the specific needs of the IPSSs.

**Key-words:** IPSS, financial management, early childhood education, technological application, open-source.

---

# Índice

Agradecimentos.....	iii
Resumo.....	iv
Abstract.....	v
Índice.....	vi
Lista de Figuras.....	viii
Lista de Tabelas.....	ix
Lista de Siglas.....	x
1 Introdução.....	1
1.1 Enquadramento.....	1
1.2 Motivação e Identificação do Problema.....	1
1.3 Objetivos.....	1
1.4 Estrutura do Documento.....	1
2 Pertinência e Viabilidade.....	1
2.1 Pertinência.....	1
2.2 Viabilidade.....	1
2.3 Análise Comparativa com Soluções Existentes.....	1
2.3.1 Soluções existentes.....	1
2.3.2 Análise de benchmarking.....	1
2.4 Proposta de inovação e mais-valias.....	2
2.5 Identificação de oportunidade de negócio.....	2
3 Especificação e Modelação.....	3
3.1 Análise de Requisitos.....	3
3.1.1 Enumeração de Requisitos.....	3
3.1.2 Descrição detalhada dos requisitos principais.....	3
3.1.3 Casos de Uso/ <i>User Stories</i> .....	3
3.2 Modelação.....	3
3.3 Protótipos de Interface.....	4
4 Solução Proposta.....	5
4.1 Apresentação.....	5
4.2 Arquitetura.....	5

---

4.3	Tecnologias e Ferramentas Utilizadas.....	5
4.4	Ambientes de Teste e de Produção.....	5
4.5	Abrangência.....	5
4.6	Componentes.....	5
4.6.1	Componente 1.....	6
4.6.2	Componente n.....	6
4.7	Interfaces.....	6
5	Testes e Validação.....	7
6	Método e Planeamento.....	8
6.1	Planeamento inicial.....	8
6.2	Análise Crítica ao Planeamento.....	8
7	Resultados.....	9
7.1	Resultados dos Testes.....	9
7.2	Cumprimento de requisitos.....	9
8	Conclusão.....	10
8.1	Conclusão.....	10
8.2	Trabalhos Futuros.....	10
	Bibliografia.....	11
	Anexo 1 – Recomendações para formatação de um relatório.....	12
	Glossário.....	14

---

## **Lista de Figuras**

Figura 1 - Organização das entregas	1
Figura 2 - Modelo de Entidade-Relação	12
Figura 3 - Mapa de Navegação	13
Figura 4 - Gantt Chart	24

---

## **Lista de Tabelas**

Tabela 1 - Benchmarking	3
Tabela 2 - Requisitos funcionais	7
Tabela 3 - Requisitos não funcionais	9
Tabela 4 - Estado final dos Requisitos	

---

## **Lista de Siglas**

API	Interface de Programação de Aplicações
HTML	Linguagem de Marcação de Hipertexto
IPSS	Instituições Particulares de Solidariedade Social

---

# 1 Introdução

O projeto está a ser desenvolvido através de uma parceria estratégica entre a Via Insight, Lda, uma empresa especializada em tecnologias de informação, a ADECO, uma Instituição Particular de Solidariedade Social (IPSS) de caráter associativo, e a Universidade Lusófona, unindo competências técnicas, sociais e académicas para alcançar soluções inovadoras e de impacto na sociedade.

## 1.1 Enquadramento

As Instituições Particulares de Solidariedade Social (IPSS) são organizações privadas, sem fins lucrativos, criadas por iniciativa de particulares para responder às necessidades da comunidade em áreas como educação, saúde, assistência social, cultura e inclusão. Estas instituições desempenham um papel essencial em diversas áreas sociais, incluindo a educação pré-escolar, por meio de creches (0-3 anos) e jardins de infância (3-6 anos). Apesar de terem autonomia de gestão, as IPSS operam sob a tutela da Direção-Geral da Segurança Social, que assegura o registo, a fiscalização e a supervisão, garantindo o cumprimento dos seus objetivos de solidariedade social.

O financiamento das IPSS provém de fontes diversificadas, incluindo contribuições privadas (doações de particulares e empresas ou quotas de associados), mas assentam em grande parte em fontes próprias e no apoio público (subsídios e contratos governamentais). Contudo, mudanças recentes na política pública, como a implementação da gratuitidade das creches (Programa “Creche Feliz”) e a revisão das comparticipações estatais, trouxeram impactos financeiros significativos. Estas medidas, embora benéficas para as famílias, aumentaram os desafios das IPSS em manter a sustentabilidade financeira, especialmente num cenário de crescentes exigências por melhorias nas condições laborais e na infraestrutura das instituições.

---

## 1.2 Motivação e Identificação do Problema

A motivação deste trabalho surge da necessidade de apoiar as IPSS na gestão dos seus recursos financeiros, especialmente num contexto de mudanças legislativas e novas exigências operacionais. A gratuitidade progressiva das creches (Decreto-Lei n.º 2/2022, de 3 de janeiro) tem alterado os modelos de financiamento das IPSS, reduzindo a receita direta proveniente dos utentes e aumentando a dependência de comparticipações estatais, que nem sempre são suficientes ou atempadas (CNIS, 2023). O aumento da fiscalização por entidades como a Segurança Social e o Tribunal de Contas, com exigência de maior rigor na prestação de contas, implica também a necessidade de processos financeiros mais transparentes e auditáveis (Tribunal de Contas, 2022).

Além disso, as revisões salariais para os profissionais das IPSS, impulsionadas pelo Acordo de Cooperação para o Setor Social Solidário de 2022-2026 (Governo de Portugal, 2022), têm agravado os custos operacionais, pressionando o equilíbrio financeiro das instituições. Muitas IPSS operam com margens financeiras muito reduzidas e dependem de donativos, subsídios ou prestação de serviços, fontes de receita frequentemente instáveis ou imprevisíveis (CNIS, 2023).

O problema identificado é a falta de uma solução tecnológica acessível e eficiente que permita às IPSS gerirem melhor os seus recursos financeiros, promovendo uma maior transparência e sustentabilidade. Atualmente, muitas instituições recorrem a folhas de cálculo manuais ou a softwares pagos, inadequados às suas necessidades e capacidades financeiras. Com uma aplicação open-source, pretende-se capacitar estas organizações para monitorizar receitas e despesas de forma simples, segura e adaptada à realidade do setor, melhorando a gestão e o planeamento financeiro, e respondendo diretamente às suas necessidades práticas e operacionais.

---

### 1.3 Objetivos

O principal objetivo deste trabalho é desenvolver uma aplicação tecnológica open-source para apoiar as creches/jardins de infância, enquanto parte das Instituições Particulares de Solidariedade Social (IPSS), na gestão administrativa e financeira, promovendo eficiência, organização e sustentabilidade.

- **Objetivo Geral:**

- Criar uma solução tecnológica acessível e intuitiva que permita às creches/jardins de infância gerir de forma centralizada as suas atividades financeiras e operacionais, facilitando o dia a dia das instituições.

- **Objetivos Específicos:**

- Criar uma base de dados que centralize e automatize os processos administrativos das creches e jardins de infância, desde a matrícula até ao pagamento das mensalidades.
- Facilitar a gestão financeira das instituições, incluindo cálculo de mensalidades, monitorização de pagamentos e subsídios de apoio recebidos.
- Simplificar a criação de relatórios financeiros e operacionais, promovendo maior eficiência na gestão.
- Melhorar a organização administrativa das turmas e dos alunos, com funcionalidades para gestão de matrículas e documentação associada.
- Proporcionar uma interface de forma a obter uma comunicação eficiente entre a instituição e os encarregados de educação.

---

## 1.4 Estrutura do Documento

Este relatório está organizado de forma a guiar o leitor pela análise e desenvolvimento da solução proposta para as creches:

**Secção 2 – Pertinência e Viabilidade:** Apresenta a relevância do trabalho, os problemas identificados nas creches/jardins de infância e o impacto esperado da solução proposta.

**Secção 3 – Especificação e Modelação:** Examina os conceitos teóricos e as soluções existentes, relacionadas com a gestão administrativa e financeira em creches, identificando assim as lacunas que a solução desenvolvida visa preencher.

**Secção 4 – Solução Desenvolvida:** Descreve o processo de desenvolvimento da aplicação, incluindo as ferramentas, tecnologias utilizadas e as decisões tomadas, durante a construção da solução.

**Secção 5 – Testes e Validação:** Apresenta os requisitos da aplicação, os testes realizados e os resultados obtidos, validando a eficácia da solução para resolver os problemas específicos das creches.

**Secção 6 – Método e Planeamento:** Explica a construção da aplicação, com ênfase nas funcionalidades chave para as creches/jardins de infância, como a gestão de mensalidades e despesas, comunicação com os encarregados de educação e outros aspectos administrativos importantes.

**Secção 7 – Resultados:** Apresentação dos resultados obtidos, com a implementação da aplicação, analisando a sua eficácia na resolução dos problemas identificados nas instituições.

**Secção 8 – Conclusões e Trabalho Futuro:** Resume as principais conclusões do trabalho, destacando os benefícios da solução desenvolvida, propondo melhorias e futuras implementações.

**Anexos:** Incluem documentação técnica, detalhes adicionais sobre o projeto e outros elementos complementares ao texto principal.

---

## 2 Pertinência e Viabilidade

### 2.1 Pertinência

A plataforma em desenvolvimento, demonstra um impacto positivo claro ao capacitar as IPSS a superar desafios financeiros e operacionais. Ao facilitar a gestão eficiente de recursos, promover a transparência e contribuir para práticas financeiras mais sustentáveis, a solução proposta responde diretamente às necessidades identificadas e contribui significativamente para a sustentabilidade e expansão das ações sociais das IPSS.

### 2.2 Viabilidade

#### 1. Alinhamento com os Objetivos de Desenvolvimento Sustentável (ODS)

A plataforma alinha-se principalmente aos seguintes ODS e metas específicas:

##### ODS 1 - Erradicação da Pobreza

- **Meta 1.5:** Construir resiliência para reduzir vulnerabilidades econômicas e sociais.  
**Contribuição do projeto:** A plataforma visa melhorar a gestão financeira das IPSS, otimizando o uso de recursos e ampliando a capacidade de atender populações vulneráveis.

##### ODS 12 - Consumo e Produção Sustentáveis

- **Meta 12.6:**
  - Incentivar organizações a adotar práticas sustentáveis e transparentes nos seus relatórios financeiros.
  - **Contribuição do projeto:** A digitalização e a automatização promovem práticas financeiras mais eficientes e transparentes.

##### ODS 17 - Parcerias para a Implementação dos Objetivos

- **Meta 17.19:** Aumentar a capacidade de geração de dados confiáveis e sustentáveis.  
**Contribuição do projeto:** A solução fornece relatórios financeiros detalhados e estruturados, facilitando o planeamento estratégico das IPSS e atraindo novos parceiros.

---

### **Impacto esperado:**

A solução não auxilia apenas as IPSS, a otimizar os seus processos internos, mas também contribui para os objetivos globais relacionados à sustentabilidade, transparência e resiliência económica.

## **2. Viabilidade Técnica**

### **Ferramentas e tecnologias disponíveis:**

- A plataforma será desenvolvida com tecnologias amplamente disponíveis, como **Django** (back-end), **React** (front-end) e bancos de dados open-source (ex.:SQLite). Essas ferramentas são gratuitas, escaláveis e suportadas por uma ampla comunidade, de desenvolvedores.

### **Testes de protótipo:**

- **Primeira entrega:** Será usado um protótipo interativo para recolher feedback inicial, com foco na experiência do utilizador.
- **Entrega final:** O protótipo funcional será avaliado em testes de aceitação por representantes das IPSS, validando a funcionalidade, a usabilidade e a confiabilidade.

### **Capacitação técnica:**

- Por ser uma solução open-source, a comunidade e as próprias IPSS podem contribuir para a sua manutenção e respetiva evolução, reduzindo dependências externas.

## **2.3 Análise Comparativa com Soluções Existentes**

### **2.3.1 Soluções existentes**

Para avaliar o posicionamento da solução proposta no mercado, foram identificadas as seguintes soluções similares de gestão financeira para IPSS ou organizações não lucrativas:

#### **1. Trello:**

O Trello é uma ferramenta de organização, baseada em Kanban, que apesar de não ser específica para gestão financeira, é utilizada por várias IPSS como apoio para controlar fluxos financeiros e organizar tarefas relacionadas. Possui uma interface intuitiva, mas não oferece funcionalidades específicas, como relatórios financeiros ou gestão de doações.

---

**Referência:** Atlassian. (2024). *Trello*. Disponível em: <https://trello.com>

## 2. **Wave Financial**

Uma plataforma gratuita que oferece contabilidade, faturação e gestão de despesas, focada em pequenas empresas e organizações. É acessível, mas a sua interface e as suas funcionalidades não são totalmente otimizadas para IPSS, faltando suporte para cálculo de mensalidades com base nos rendimentos do agregado, gestão das contribuições sociais, doações e a elaboração de relatórios adaptados.

**Referência:** Wave Financial Inc. (2024). *Wave Financial*. Disponível em: <https://waveapps.com>

## 3. **Microsoft: Excel:**

Apesar de não ser uma ferramenta dedicada, o Excel é amplamente usado por IPSS para gerir finanças de forma manual. Embora flexível, a falta de automatização e relatórios integrados limita sua eficiência.

**Referência:** Microsoft. (2024). *Microsoft Excel*. Disponível em: <https://microsoft.com/excel>

## 4. **QuickBooks Nonprofit:**

Uma solução paga, adaptada às necessidades de organizações sem fins lucrativos. Oferece funcionalidades robustas para contabilidade, mas o custo pode ser impeditivo para pequenas IPSS, visto que não é open-source.

**Referência:** Intuit. (2024). *QuickBooks Nonprofit*. Disponível em: <https://quickbooks.intuit.com/nonprofit>

---

### 2.3.2 Análise de benchmarking

A tabela abaixo, compara as características das soluções existentes com a solução proposta, destacando os aspectos-chave relevantes:

Características	Trello	Wave Financial	Excel	QuickBooks Nonprofit	Solução Proposta
Controlo de despesas		X	X	X	X
Geração automática de relatórios		X		X	X
Open-source					X
Custo acessível	X	X		X	X
Adaptação específica para IPSS				X	X

*Tabela 1 - benchmarking*

---

## 2.4 Proposta de inovação e mais-valias

A solução proposta distingue-se por introduzir uma abordagem inovadora e acessível para a gestão financeira e administrativa, de Instituições Particulares de Solidariedade Social (IPSS) na área da educação infantil. Destacam-se as seguintes características como elementos de inovação e mais-valias:

### **Inovação**

- **Automatização e Personalização:**

- A aplicação automatiza tarefas administrativas complexas, como o cálculo de mensalidades com base em escalões de rendimento, adaptando-se às regulamentações específicas das instituições e permitindo configurações personalizadas.

- **Open-Source e Sustentável:**

- Ao ser desenvolvida como uma solução open-source, promove a democratização do acesso à tecnologia, permitindo que IPSS com orçamentos limitados utilizem a plataforma. Essa abordagem também facilita a evolução contínua do sistema por meio de contribuições de uma comunidade de desenvolvedores e das próprias instituições.

- **Alinhamento com ODS:**

- A solução contribui para os Objetivos de Desenvolvimento Sustentável (ODS), promovendo transparência (ODS 12), resiliência financeira (ODS 1) e geração de dados confiáveis para planeamento estratégico (ODS 17).

- **Centralização de Dados:**

- Com uma base de dados centralizada, a aplicação unifica informações administrativas e financeiras, otimizando a gestão diária e permitindo análises estratégicas com dados organizados e acessíveis.

---

## Mais-Valias

- **Eficiência Operacional:**
  - A automatização de processos reduz a carga de trabalho administrativo, minimiza erros e agiliza a tomada de decisões.
- **Transparência e Conformidade:**
  - A geração de relatórios customizáveis e a rastreabilidade de dados financeiros, promovem transparência e facilitam o cumprimento de obrigações legais e regulamentares.
- **Acessibilidade Financeira:**
  - A natureza open-source da solução, reduz custos associados a licenças, tornando-a uma opção viável mesmo para pequenas IPSS com recursos limitados.
- **Impacto Social:**
  - Ao otimizar a gestão de recursos, a solução permite que as IPSS concentrem os esforços na melhoria da qualidade do serviço prestado às famílias e crianças, ampliando o seu impacto social.

Com estas inovações e mais-valias, a aplicação pretende transformar a forma como as IPSS gerem os seus recursos, criando uma solução sustentável, eficiente e adaptada às suas necessidades reais.

---

## 2.5 Identificação de oportunidade de negócio

A solução proposta identifica uma oportunidade de negócio significativa ao oferecer uma aplicação open-source voltada para a gestão financeira e administrativa de IPSS, no setor da educação infantil. Esta abordagem, posiciona o projeto de forma estratégica em um mercado com uma procura crescente por soluções tecnológicas acessíveis e específicas, para instituições sem fins lucrativos.

### **Mercado Alvo**

IPSS de Educação Infantil: A solução atende diretamente às necessidades dessas instituições, que enfrentam desafios crescentes para manter a sustentabilidade financeira, num ambiente de regulamentação e fiscalização mais rigoroso.

Outras Instituições Sem Fins Lucrativos: Embora o foco inicial seja nas IPSS de creches e jardins de infância, o modelo pode ser expandido para outras organizações sociais com necessidades semelhantes.

### **Diferenciais Competitivos**

- **Custo Zero de Licenciamento:**
  - Como uma aplicação open-source, a solução elimina custos de licenciamento, tornando-se financeiramente acessível para instituições com orçamentos restritos.
- **Personalização e Escalabilidade:**
  - A natureza open-source permite que a aplicação seja personalizada e adaptada às especificidades de cada instituição, aumentando a sua atratividade e utilidade.
- **Alinhamento com Políticas Públicas:**
  - A aplicação facilita o cumprimento de requisitos legais e regulatórios, posicionando-se como uma ferramenta indispensável para a gestão eficiente e transparente de IPSS.

---

## Modelo de Negócio Sustentável

Apesar da sua natureza gratuita e comunitária, a aplicação poderá gerar receitas sustentáveis através de:

- **Serviços de Suporte Técnico:** Instalação, atualizações, integração de funcionalidades específicas.
- **Formação e Capacitação:** Workshops e cursos dirigidos a equipas administrativas de IPSS.
- **Consultoria Especializada:** Apoio à análise de desempenho financeiro, preparação de relatórios e melhoria de processos.
- **Parcerias Institucionais:** Colaboração com municípios, fundações e redes sociais interessadas em digitalizar a gestão das IPSS.

Contudo, é importante reconhecer que o modelo apresenta riscos à sua viabilidade económica, especialmente em fase inicial. A inexistência de receitas diretas obriga a garantir financiamento alternativo, subsídios, ou um núcleo de utilizadores disposto a pagar por serviços complementares. Por isso, recomenda-se que, numa futura fase, seja desenvolvido um estudo de viabilidade simplificado (mesmo sem valores exatos), com projeções de custos de manutenção, número potencial de utilizadores e estratégias de financiamento (crowdfunding, programas públicos de inovação social, etc.).

### Potencial de Expansão

O projeto pode evoluir para incluir funcionalidades adicionais e integrar-se a outras plataformas tecnológicas, como sistemas de gestão escolar ou plataformas de doações online, ampliando o impacto social e alcance de mercado.

### Impacto Social e Económico

Ao otimizar a gestão de recursos, a solução possibilita que as IPSS concentrem os seus esforços na melhoria dos serviços prestados às crianças e famílias, fortalecendo a sustentabilidade e ampliando o impacto social.

Com um posicionamento estratégico e uma abordagem centrada nas necessidades reais do mercado, a solução proposta demonstra um potencial claro para se tornar uma referência no setor, combinando inovação tecnológica, acessibilidade financeira e impacto social.

---

### 3 Especificação e Modelação

#### 3.1 Análise de Requisitos

A análise de requisitos foi realizada com o objetivo de identificar as necessidades das creches/jardins de infância e os desafios enfrentados na gestão administrativa e financeira. Os requisitos levantados refletem a visão inicial do projeto e incluem funcionalidades planeadas para o desenvolvimento atual e possíveis expansões futuras.

##### 3.1.1 Enumeração de Requisitos

###### Requisitos Funcionais:

Requisitos Funcionais		
Referência	Descrição: O sistema deve permitir...	Prioridade MosCow
RQ_F01	... criar contas para administradores, assim como outros utilizadores. As contas criadas para estes últimos têm que ter a aprovação dos administradores.	MUST
RQ_F02	... criar bases de dados dos alunos manualmente ou carregar bases de dados por leitura de ficheiros excel ou .csv.	SHOULD
RQ_F03	... organizar a base de dados por anos letivos, a qual poderá ser copiada e facilmente alterada no início de cada ano letivo.	COULD
RQ_F04	... criar um questionário parametrizável para inscrição dos alunos, cujo link será enviado aos encarregados de educação, e cujos dados serão gravados num ficheiro excel ou .csv que poderá ser importado para a base de dados.	COULD
RQ_F05	... o registo de dados essenciais do aluno, como números de identificação, contactos, etc. e de informação adicional, como, com quem tem autorização para sair da escola, vacinação, cuidados especiais (saúde, alergias, alimentação, etc.)	MUST
RQ_F06	... o registo de dados dos encarregados de educação e familiares, como números de identificação, contactos, etc..	MUST

## Requisitos Funcionais

Referênci a	Descrição: O sistema deve permitir...	Prioridade MosCow
RQ_F07	<p>... o cálculo da mensalidade a ser paga mensalmente por cada cliente, permitindo:</p> <ul style="list-style-type: none"> <li>• guardar em base de dados os valores de remuneração e despesas do agregado familiar necessárias para o cálculo da mensalidade de acordo com as determinações da Segurança Social e da instituição;</li> <li>• configurar as modalidades de comparticipação da Segurança Social em vigor na lei, pela qual a instituição se rege;</li> <li>• configurar o quadro de escalões de rendimentos de acordo com os regulamentos da instituição;</li> <li>• configurar o cálculo de mensalidades com base nos parâmetros anteriormente listados e as diretrizes da instituição.</li> </ul>	MUST
RQ_F08	..., de forma simples, fazer o registo dos pagamentos das mensalidades e comparticipações suplementares de cada aluno;	MUST
RQ_F09	... o monitoramento de comparticipações em atraso: o sistema deve poder ser configurado para enviar um email para utilizadores pré-definidos com a listagem das comparticipações em atraso, quando este atraso for superior a um período configurável.	COULD
RQ_F10	... preencher, de forma simples, folhas de caixa com as despesas diárias, facilitando o agrupamento de despesas em classes pré-definidas pelo utilizador, e as quais serão adicionadas à lista mensal de despesas.	COULD
RQ_F11	... definir despesas fixas, as quais serão adicionadas automaticamente à lista mensal de despesas (ex. Salários, rendas, etc.)	COULD
RQ_F12	... rever a lista mensal de despesas e corrigir incorreções	COULD
RQ_F13	<p>... cálculos específicos e análises estatísticas para monitorização da saúde financeira da instituição e agilização da gestão de custos:</p> <ul style="list-style-type: none"> <li>• Resposta Social e/ou (por criança, por valência, por escalão de rendimentos)</li> <li>• Custo mensal (totalidade, por valência, por criança)</li> <li>• Balancete analítico mensal</li> <li>• ...</li> </ul>	COULD

Requisitos Funcionais		
Referênci a	Descrição: O sistema deve permitir...	Prioridade MosCow
RQ_F14	<p>... obter relatórios financeiros em formato excel ou pdf, utilizando filtros para selecionarem a informação requerida. Os filtros devem permitir extrair informação relativa a:</p> <ul style="list-style-type: none"> <li>• Um aluno em particular ou a grupos específicos (ex. valência, escalão de rendimentos, ...);</li> <li>• Dados financeiros específicos (apresentados em tabelas ou gráficos);</li> <li>• Períodos de tempo especificados.</li> </ul>	MUST
RQ_F15	... salvar configurações de filtros para obtenção de relatórios com requisitos específicos (ex. relatórios mensais para a Segurança Social).	COULD
RQ_F16	... configurar o envio periódico de relatórios pré-configurados, por email, para um grupo de utilizadores pré-definido.	COULD
RQ_F17	... a comunicação entre a instituição e os encarregados de educação, agilizando a troca de emails e/ou mensagem individualizados ou para grupos selecionados.	COULD
<b>Funcionalidades para implementar no futuro...</b>		
	Criar base de dados para aluno;	
	Permitir a configuração de lembretes e notificações automáticos;	
	Criação de faturas;	
	Fazer a monitorização de vacinação em atraso.	

*Tabela 2 - Requisitos funcionais*

Requisitos Não Funcionais		
Referênci a	Descrição:	Prioridade MosCow
<b>RQ_NF0 1</b>	O sistema deve permitir o armazenamento seguro de dados sensíveis, garantindo que os dados sejam encriptados na base de dados.	MUST
<b>RQ_NF0 2</b>	O sistema deve permitir a acessibilidade da aplicação, sendo fácil de utilizar por utilizadores com diferentes níveis de familiaridade com a tecnologia.	MUST
<b>RQ_NF0 3</b>	O sistema deve permitir tempos de resposta inferiores a 2 segundos em ações comuns realizadas pelos utilizadores.	SHOULD
<b>RQ_NF0 4</b>	Deve ser utilizada a linguagem Python para o desenvolvimento do back-end e a integração de APIs.	MUST
<b>RQ_NF0 5</b>	Deve ser usado HTML para estruturar as páginas web, e CSS para a estilização e layout.	MUST
<b>RQ_NF0 6</b>	Deve ser utilizado JavaScript para adicionar interatividade e recursos dinâmicos no front-end.	MUST
<b>RQ_NF0 7</b>	O web framework FastAPI deve ser utilizado para a construção das APIs.	MUST
<b>RQ_NF0 8</b>	O motor de base de dados SQLite deve ser utilizado para armazenamento e gestão de dados.	MUST
<b>RQ_NF0 9</b>	O servidor web Unicorn deve ser usado para executar a aplicação FastAPI localmente.	MUST
<b>RQ_NF1 0</b>	O sistema deve ser simples de instalar e executar, inclusive em dispositivos com atributos de baixo desempenho.	SHOULD
<b>RQ_NF1 1</b>	O sistema deve ter uma interface intuitiva e fácil de utilizar.	MUST
<b>RQ_NF1 2</b>	O sistema deve ter uma arquitetura modular para facilitar atualizações e acréscimo de novas funções.	SHOULD
<b>RQ_NF1 3</b>	O sistema deve realizar backups automáticos de todas as bases de dados para um local definido.	COULD

*Tabela 3 - Requisitos não funcionais*

---

### 3.1.2 Descrição detalhada dos requisitos principais

Os seguintes requisitos foram identificados como os de maior impacto para o sistema e são detalhados com dependências, objetivos, critérios de aceitação e, quando aplicável, processos de negócio relacionados.

#### **RQ\_F01: Criação e gestão de contas de utilizadores**

- **Dependências:** Funcionalidade de aprovação por administradores (obrigatória para contas de utilizadores não administrativos).
- **Objetivos:** Permitir a gestão centralizada de acessos ao sistema com níveis de permissões apropriados.
- **Critérios de Aceitação:**
  - Administradores podem criar contas de outros administradores e utilizadores.
  - Contas de utilizadores não administrativos requerem aprovação prévia por parte de um administrador antes de serem ativadas.

#### **RQ\_F07: Cálculo de mensalidades**

- **Dependências:** Dados do agregado familiar e regulamentação específica da instituição.
- **Objetivos:** Calcular mensalidades com base em regras pré-definidas de escalões de rendimento e comparticipações da Segurança Social.
- **Critérios de Aceitação:**
  - Configuração de modalidades de cálculo ajustáveis conforme a legislação e regulamentos da instituição.
  - Resultados do cálculo devem ser precisos e verificáveis em relação às configurações definidas.

---

#### **RQ\_F14: Geração de relatórios financeiros**

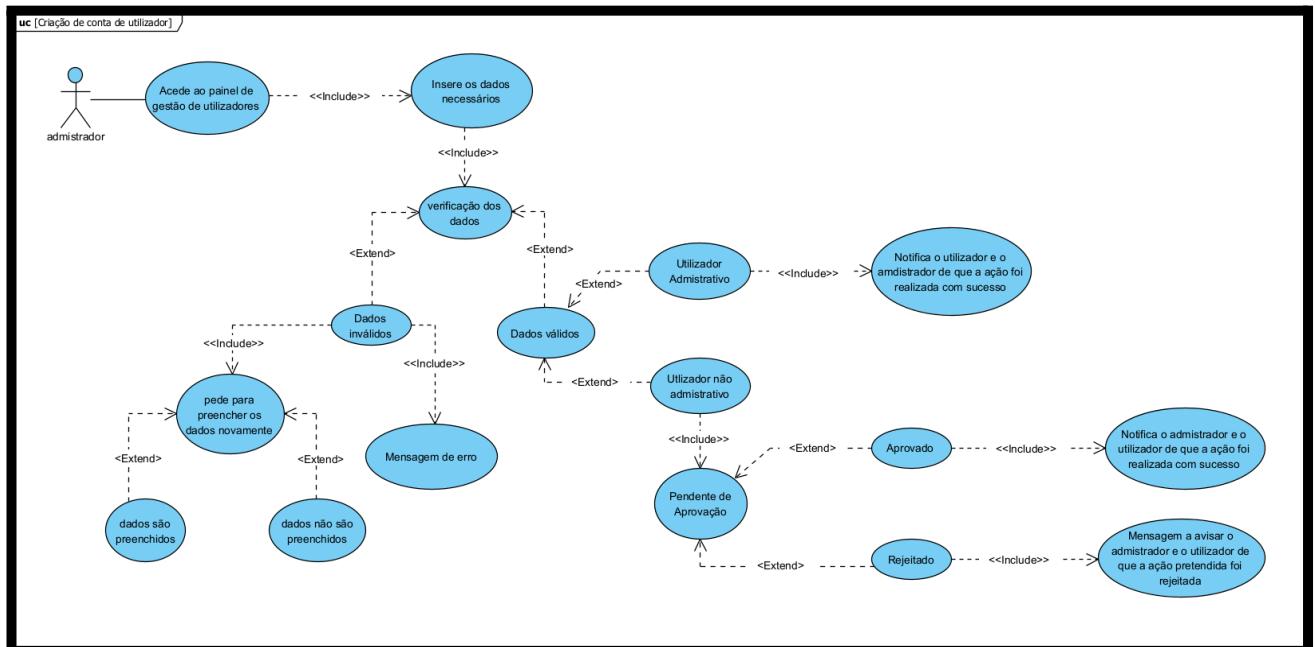
- **Dependências:** Base de dados financeira atualizada e configurações de filtros.
- **Objetivos:** Proporcionar relatórios financeiros customizados que atendam às necessidades operacionais e regulamentares.
- **Critérios de Aceitação:**
  - Os relatórios devem ser exportáveis em formatos Excel e PDF.
  - Os filtros devem permitir extração de dados específicos, como por aluno, valência, ou períodos de tempo.

### 3.1.3 Casos de Uso/User Stories

Os cenários a seguir representam a utilização prática do sistema pelos seus diferentes atores:

#### Caso de Uso 1: Criação de conta de utilizador

- **Ator:** Administrador
- **Descrição:**
  1. O administrador acede ao painel de gestão de utilizadores.
  2. Insere os dados do novo utilizador (nome, e-mail, tipo de conta).
  3. Caso o utilizador seja não administrativo, a conta permanece pendente até aprovação.
  4. O sistema notifica o administrador e o utilizador sobre a aprovação ou rejeição.
- **Objetivo:** Garantir que apenas utilizadores autorizados acedem ao sistema.



---

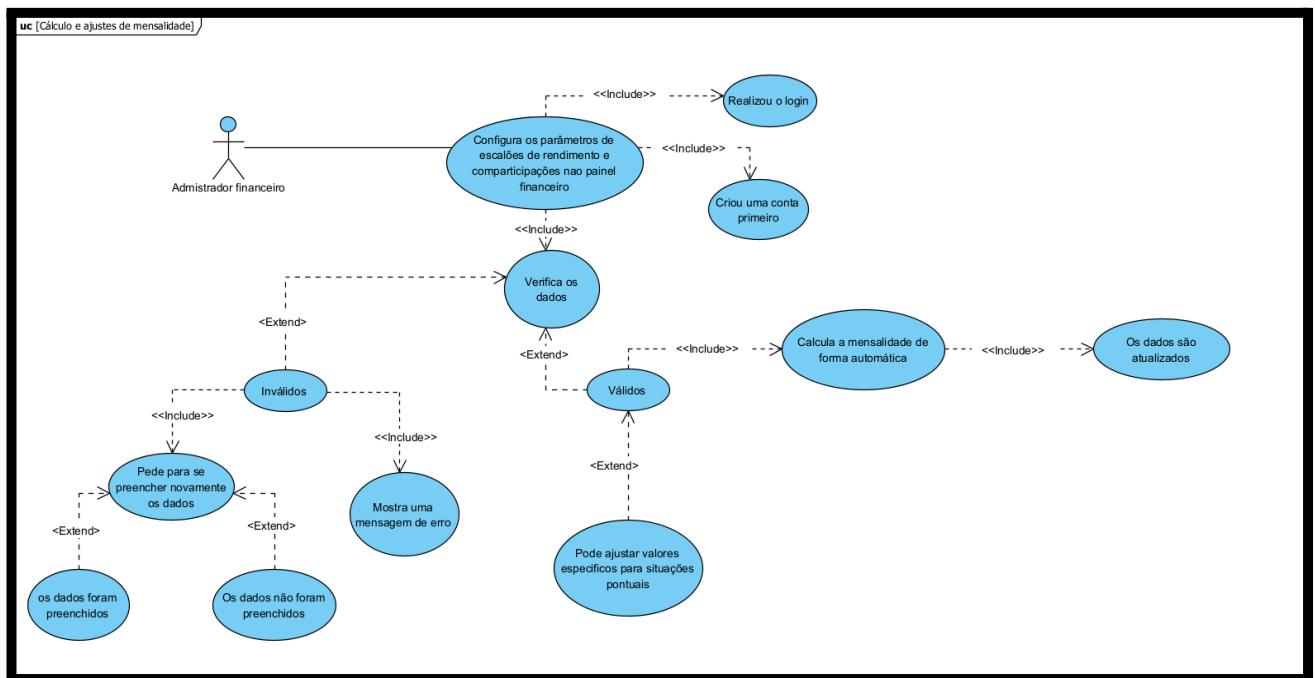
## Caso de Uso 2: Cálculo e ajuste de mensalidades

- **Autor:** Administrador Financeiro

- **Descrição:**

1. O administrador configura os parâmetros de escalões de rendimento e comparticipações no painel financeiro.
2. Os dados do agregado familiar de cada aluno são atualizados e validados.
3. O sistema calcula automaticamente a mensalidade com base nos parâmetros configurados.
4. Caso seja necessário, o administrador ajusta valores específicos para situações pontuais.

- **Objetivo:** Automatizar e agilizar o cálculo de mensalidades, minimizando erros.



---

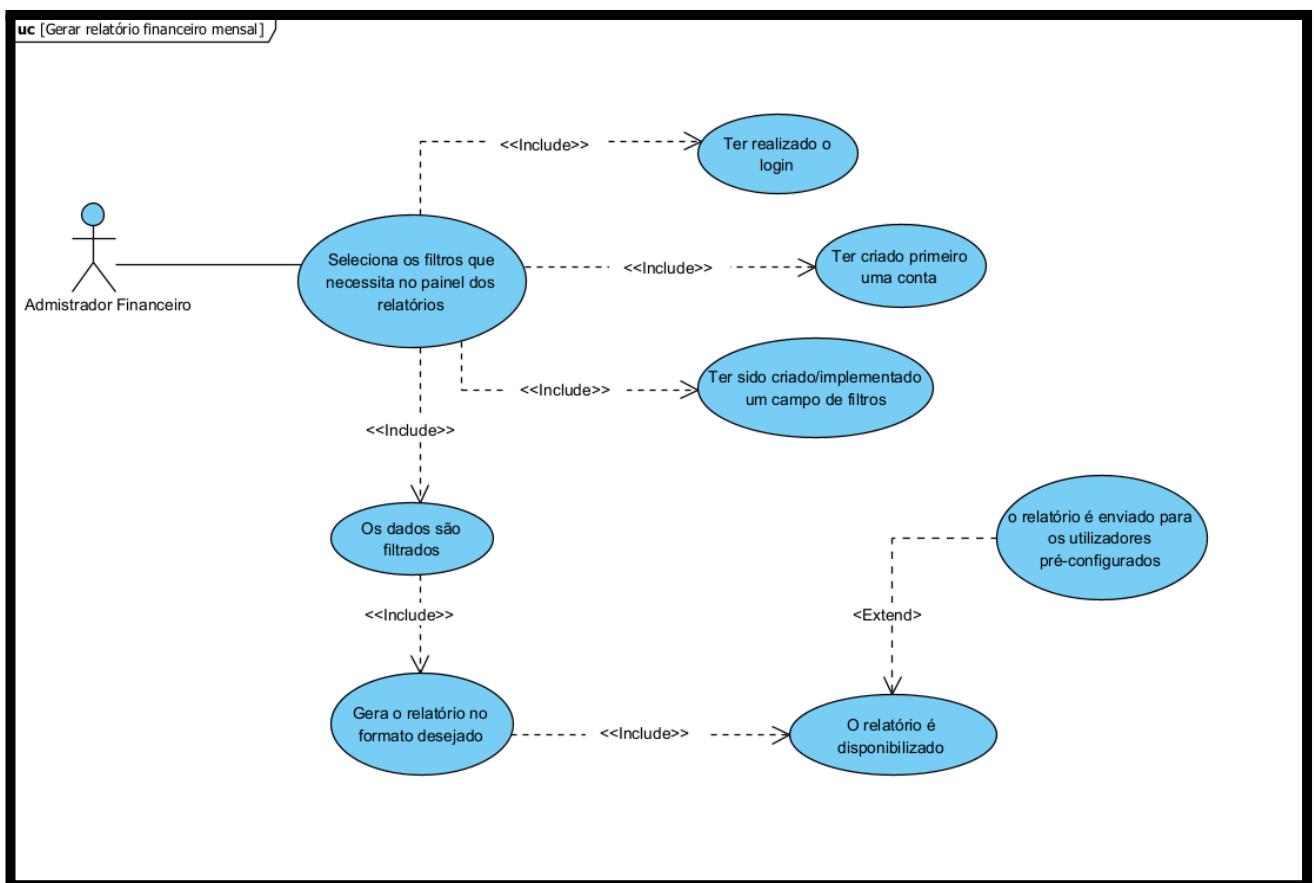
### Caso de Uso 3: Gerar relatório financeiro mensal

- **Autor:** Administrador Financeiro

- **Descrição:**

1. O administrador seleciona os filtros no painel de relatórios (ex.: alunos de um escalão específico ou período determinado).
2. Gera o relatório no formato desejado (PDF ou Excel).
3. O sistema disponibiliza o relatório e, opcionalmente, envia-o para os utilizadores pré-configurados.

- **Objetivo:** Fornecer informações financeiras organizadas para análise e cumprimento de obrigações regulamentares.



## 3.2 Modelação

Este projeto está estruturado em várias entidades, com o objetivo de proporcionar uma organização mais eficiente da informação. Essas entidades representam objetos relevantes para o sistema, sobre os quais é essencial armazenar dados. Nesta fase inicial do desenvolvimento, a criação de uma base de dados é fundamental para possibilitar um acesso rápido à informação e para gerir os dados de forma eficiente e eficaz. O modelo Entidade-Relação apresentado na Figura 2 demonstra, de forma visual, as conexões e interações entre as entidades

Este modelo serve como um guia essencial, especialmente nesta fase inicial, e poderá ser ajustado em versões futuras do sistema para refletir melhorias e novas necessidades identificadas ao longo do processo.

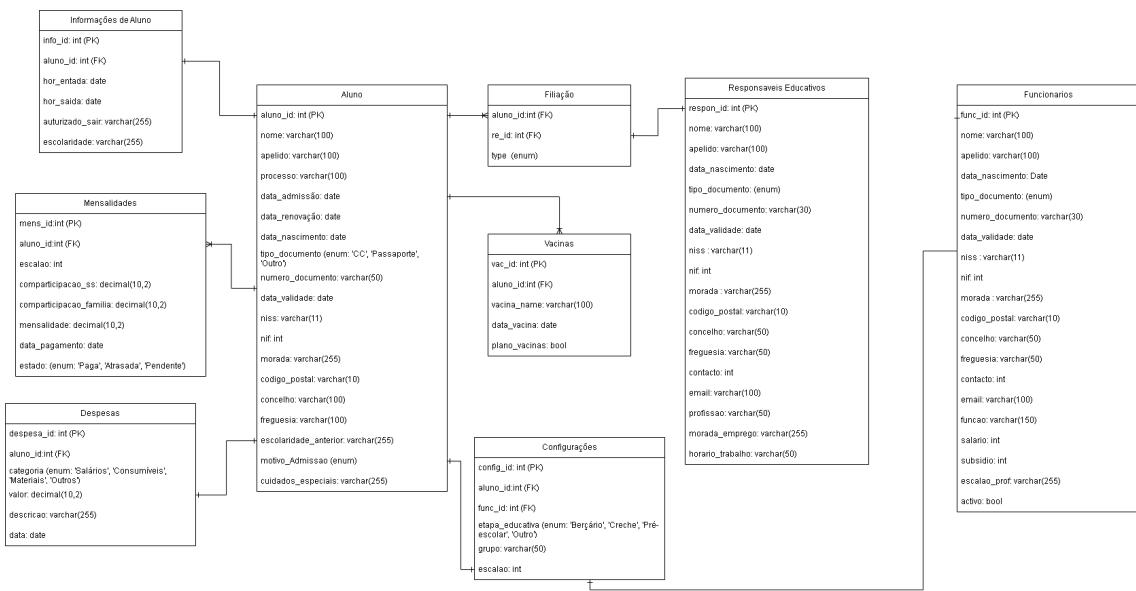
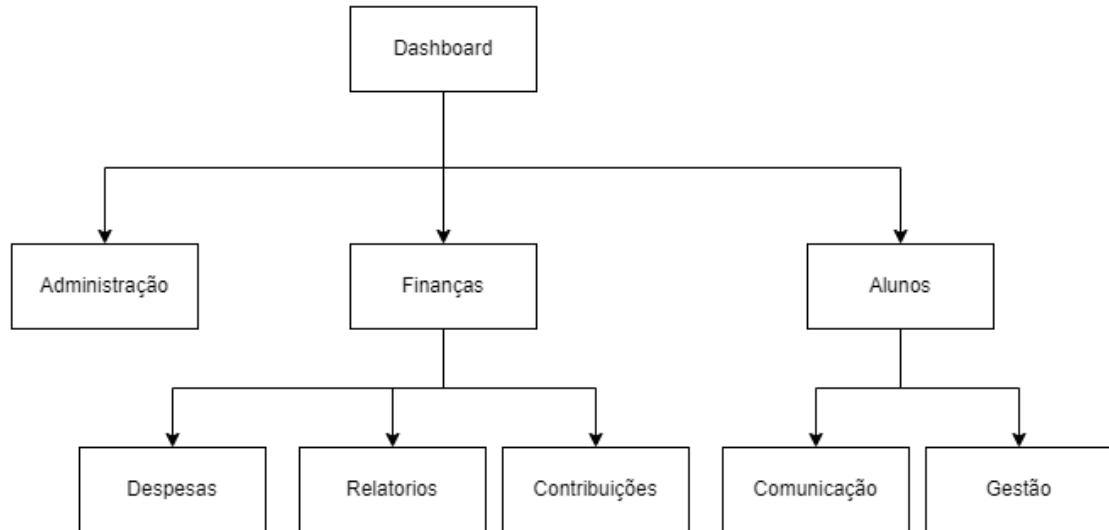


Figura 2 - Modelo de Entidade-Relação

### 3.3 Protótipos de Interface

Neste tópico, temos representado um protótipo do mapa de navegação da plataforma.

Tencionamos repensar o mapa aplicacional mais para a frente quando forem adicionadas mais funções.



*Figura 3 - Mapa de Navegação*

A partir do mapa de navegação acima podemos ver que páginas serão implementadas na plataforma, e cada página terá as seguintes funções.

#### **Dashboard:**

- Observar certas informações(gráficos, tabelas, documentos, etc.), de forma resumida;
- Navegar pela plataforma;

#### **Administração:**

- Gerir alunos, (Adicionar, remover, mudar a turma, etc.);

#### **Finanças:**

- Consultar, de uma forma generalizada, informações da condição financeira da instituição através de gráficos;
- Gerar relatórios de despesas

#### **Despesas:**

- Consultar relatórios de despesas;
- Consultar informações da condição financeira da instituição através de gráficos mais detalhados;
- Gerar relatórios mais detalhados;

---

**Relatórios:**

- Consultar relatórios gerados anteriormente;
- Gerar um relatório do desenvolvimento financeiro da instituição baseado nos relatórios gerados anteriormente;

**Contribuições:**

- Gerar relatórios sobre as contribuições;
- Consultar tabelas com as contribuições, mensalidades em atraso e outras dívidas, doações e ajudas da Segurança Social;

**Alunos:**

- Ver as informações de um aluno;
- Ver as informações do encarregado de educação de um aluno (Nome, contacto, etc.)

**Comunicação:**

- Ver e enviar comunicados dos professores para os encarregados de educação;

**Gestão:**

- Fazer pedidos de mudanças de informação;
- Alterar o contacto dos encarregados de educação;

---

## 4 Solução Proposta

### 4.1 Apresentação

A solução proposta é uma aplicação web, destinada à gestão de creches/jardins de infância e outras instituições IPSS. A plataforma atende a requisitos funcionais detalhados e abrange desde a gestão de alunos e responsáveis até à geração de relatórios financeiros personalizados. Esta aplicação otimiza a administração diária, reduz erros operacionais e melhora a comunicação com os encarregados de educação.

#### 4.1.1 Funcionalidades-chave implementadas no MVP:

- Gestão de contas de utilizadores com aprovação administrativa (**RQ\_F01**).
- Registo e organização de dados dos alunos e encarregados de educação (**RQ\_F02, RQ\_F05, RQ\_F06**).
- Cálculo e monitorização de mensalidades com base em critérios sociais e institucionais (**RQ\_F07, RQ\_F08, RQ\_F09**).
- Geração e envio de relatórios financeiros configuráveis (**RQ\_F14, RQ\_F15, RQ\_F16**).
- Comunicação entre instituição e responsáveis via email e SMS (**RQ\_F17**).

Na entrega final, a análise comparativa com o benchmarking demonstrará como a solução se destaca em termos de usabilidade e automatização.

#### 4.1.2 Recursos Disponibilizados:

- **Vídeo demonstrativo:**  
<https://youtu.be/61IvxGX4KqA>
- caso o video não esteja disponível:  
[https://drive.google.com/file/d/1fQIBqQ09tUHWHsotOUPWtN2-l6SVhbG/view?usp=drive\\_link](https://drive.google.com/file/d/1fQIBqQ09tUHWHsotOUPWtN2-l6SVhbG/view?usp=drive_link)
- **Repositório Git:**  
<https://github.com/DEISI-ULHT-TFC-2024-25/DEISI2098-Faz-e-Conta>
- **Solução funcional:** (URL a incluir)

---

## 4.2 Arquitetura

A solução proposta é uma aplicação web modular e eficiente, construída com tecnologias de alto desempenho e fácil integração. A arquitetura foi desenhada para garantir simplicidade, flexibilidade, e compatibilidade com dispositivos de baixo desempenho, assegurando que o sistema possa ser utilizado em diversos contextos, incluindo aqueles com recursos computacionais limitados.

### Identificação e Justificação das Tecnologias

#### 1. Back-end (Python com FastAPI):

- **FastAPI:** Escolhida pelo seu desempenho, compatibilidade com Python moderno e facilidade na construção de APIs RESTful.
- **Uvicorn:** Utilizado como servidor web para executar a aplicação FastAPI, garantindo alta performance na gestão de conexões assíncronas.
- **Justificação:** Python é uma linguagem acessível, amplamente utilizada e com uma comunidade ativa, o que facilita futuras atualizações e suporte técnico.

#### 2. Front-end (HTML, CSS e JavaScript):

- **HTML:** Estrutura básica das páginas web.
- **CSS:** Responsável pelo design e layout responsivo.
- **JavaScript:** Adicionar interatividade e recursos dinâmicos.
- **Justificação:** Esta stack de front-end é universal e amplamente compatível com navegadores modernos, garantindo acessibilidade e portabilidade.

#### 3. Base de Dados (SQLite):

- Escolhido pela sua leveza, simplicidade de configuração e portabilidade, eliminando a necessidade de configurações complexas de servidores de base de dados.
- **Justificação:** Ideal para aplicações com requisitos moderados de desempenho, sendo uma solução económica e funcional.

#### 4. Backups Automáticos:

- O sistema realizará backups automáticos dos dados para um local configurado pelo utilizador, garantindo a recuperação de dados em caso de falhas.

#### 5. Arquitetura Modular:

- O sistema foi desenhado em módulos independentes para permitir a adição ou substituição de funcionalidades sem impactar o núcleo da aplicação.
- **Justificação:** Facilita manutenção e escalabilidade da solução.

---

## Desenho da Arquitetura

A arquitetura da aplicação é dividida em três camadas principais:

1. **Camada de Apresentação (Front-end):**
  - O navegador do utilizador comunica-se com o back-end utilizando chamadas API via HTTP.
  - Tecnologias: HTML, CSS, JavaScript.
2. **Camada de Lógica de Negócio (Back-end):**
  - Implementada em Python com FastAPI. Gerência requisições do front-end, processa dados e comunica com o base de dados.
  - Executada pelo servidor Uvicorn.
3. **Camada de Dados:**
  - A Base de Dados SQLite armazena informações persistentes.
  - Sistema de backups automático preserva os dados periodicamente.

## Fluxo de Funcionamento

1. O utilizador interage com a interface web (HTML + CSS + JavaScript).
2. O front-end envia requisições para o back-end através de APIs RESTful (FastAPI).
3. O back-end processa a lógica e acede a base de dados SQLite para recuperar ou guardar dados.
4. Respostas são enviadas de volta ao front-end para exibição ao utilizador.
5. Backups automáticos garantem segurança dos dados.
6. **Diagrama da Arquitetura**

## Componentes Principais:

1. **Navegador do Utilizador:** Acede à aplicação.
2. **Front-end:** HTML, CSS e JavaScript geram a interface.
3. **Back-end:** FastAPI executado no Uvicorn processa a lógica.
4. **Base de Dados:** SQLite armazena os dados.
5. **Sistema de Backup:** Realiza cópias periódicas dos dados.

*(Inserir imagem do diagrama de arquitetura com os componentes e fluxo mencionado)*

---

## Fundamentação da Arquitetura

- **Simples Instalação:** A escolha de SQLite elimina dependências de bancos de dados externos, tornando o sistema fácil de instalar.
- **Desempenho:** FastAPI e Uvicorn oferecem alta performance, enquanto o front-end garante interatividade responsiva.
- **Segurança e Sustentabilidade:** Backups automáticos minimizam riscos de perda de dados.
- **Modularidade:** A arquitetura modular facilita adições e melhorias futuras.

## 4.3 Tecnologias e Ferramentas Utilizadas

A solução proposta utiliza um conjunto de tecnologias e ferramentas alinhadas com os requisitos definidos, visando simplicidade, flexibilidade e desempenho em ambientes variados. Abaixo segue a descrição e justificação de cada tecnologia e ferramenta, além de uma ilustração que detalha os perímetros tecnológicos.

### Tecnologias Utilizadas

#### Front-end:

##### 1. HTML:

- **Descrição:** Linguagem de marcação para estruturar o conteúdo das páginas web.
- **Justificação:** Essencial para criar interfaces compatíveis com todos os navegadores.

##### 2. CSS:

- **Descrição:** Linguagem de estilos para layout e design responsivo.
- **Justificação:** Proporciona personalização visual para uma melhor experiência de utilizador.

##### 3. JavaScript:

- **Descrição:** Linguagem de programação para adicionar interatividade e funcionalidade dinâmica.
- **Justificação:** Garante um front-end mais interativo e responsivo.

---

## **Back-end:**

### **1. Python com FastAPI:**

- **Descrição:** Framework de alto desempenho para APIs RESTful.
- **Justificação:** Facilita o desenvolvimento rápido e seguro, com suporte nativo para operações assíncronas.

### **2. Uvicorn:**

- **Descrição:** Servidor web leve para execução de aplicações FastAPI.
- **Justificação:** Garante desempenho superior e suporte a conexões assíncronas.

## **Base de Dados:**

### **1. SQLite:**

- **Descrição:** Sistema de gestão de base de dados relacional leve e integrado.
- **Justificação:** Perfeito para aplicações simples, eliminando a necessidade de configurações complexas.

## **Outras Ferramentas:**

### **1. Visual Studio Code:**

- **Descrição:** IDE para desenvolvimento de código.
- **Justificação:** Oferece suporte extensivo a Python, HTML e JavaScript.

### **2. Git e GitHub:**

- **Descrição:** Sistema de controlo de versão e plataforma de hospedagem de código.
- **Justificação:** Facilita a gestão de versões e o trabalho colaborativo.

---

## 4.4 Ambientes de Teste e de Produção

### Ambiente de Teste

O ambiente de teste será configurado para validar a funcionalidade, desempenho e confiabilidade do sistema antes da sua implementação em produção. Este utilizará recursos de hardware e software semelhantes ao ambiente de produção.

- **Hardware:**
  - Processador: 1 núcleo a 1.5 GHz (mínimo)
  - Memória RAM: 8 GB (mínimo)
  - Armazenamento: 8 GB (mínimo)
- **Software:**
  - Sistema Operativo: Linux ou Windows ou macOS
  - Servidor Web: Unicorn
  - Base de Dados: SQLite

### Ambiente de Produção

O ambiente de produção será configurado para garantir desempenho e escalabilidade em contextos reais de utilização. Ele será dimensionado para suportar múltiplos utilizadores simultâneos, assegurando uma experiência fluida.

- **Hardware:**
  - Processador: Mínimo 2 núcleos a 2 GHz
  - Memória RAM: Mínimo 8 GB ou mais
  - Armazenamento: 10 GB
- **Software:**
  - Sistema Operativo: Windows, macOS, Linux
  - Servidor Web: Unicorn configurado para execução contínua
  - Base de Dados: SQLite com backups automáticos configurados

## 4.5 Abrangência

A solução aplica conhecimentos de diversas áreas do curso:

- **Engenharia de Software:** Design e implementação do sistema.
- **Data Science, Base de Dados:** Modelação e normalização na 3<sup>a</sup> forma normal.
- **Interação Humano-Máquina:** Mapa aplicacional e design de Interface
- **Linguagens de Programação:** Desenvolvimento da aplicação

---

## 4.6 Componentes

### 4.6.1 Componente de Gestão de Utilizadores

Permite criação de contas para administradores e utilizadores gerais (RQ\_F01).

Contempla:

- Registo de utilizadores com validação de dados.
- Aprovação de contas por administradores.

### 4.6.2 Componente de Gestão de Dados

- Criação e organização de bases de dados por ano letivo (RQ\_F03).
- Registo de dados de alunos e responsáveis (RQ\_F05, RQ\_F06).
- Importação de dados via Excel ou CSV (RQ\_F02).

### 4.6.3 Componente Financeiro

- Cálculo e monitoramento de mensalidades e comparticipações (RQ\_F07, RQ\_F08).
- Geração e envio de relatórios financeiros (RQ\_F14, RQ\_F15).

### 4.6.4 Componente de Comunicação

- Envio de emails e SMS individualizados ou em massa (RQ\_F17).
- Configuração de lembretes automáticos (funcionalidade futura).

## 4.7 Interfaces

A aplicação **faz-e-conta** é uma plataforma web desenvolvida em **Python** com o objetivo de apoiar a gestão financeira de Instituições Particulares de Solidariedade Social (IPSS), nomeadamente creches e jardins-de-infância. Esta aplicação foi pensada para simplificar o acesso à informação relevante e automatizar a geração de relatórios úteis para a gestão.

Abaixo encontram-se os ecrãs mais representativos da aplicação, com uma breve descrição do seu propósito, implementação e decisões de design.

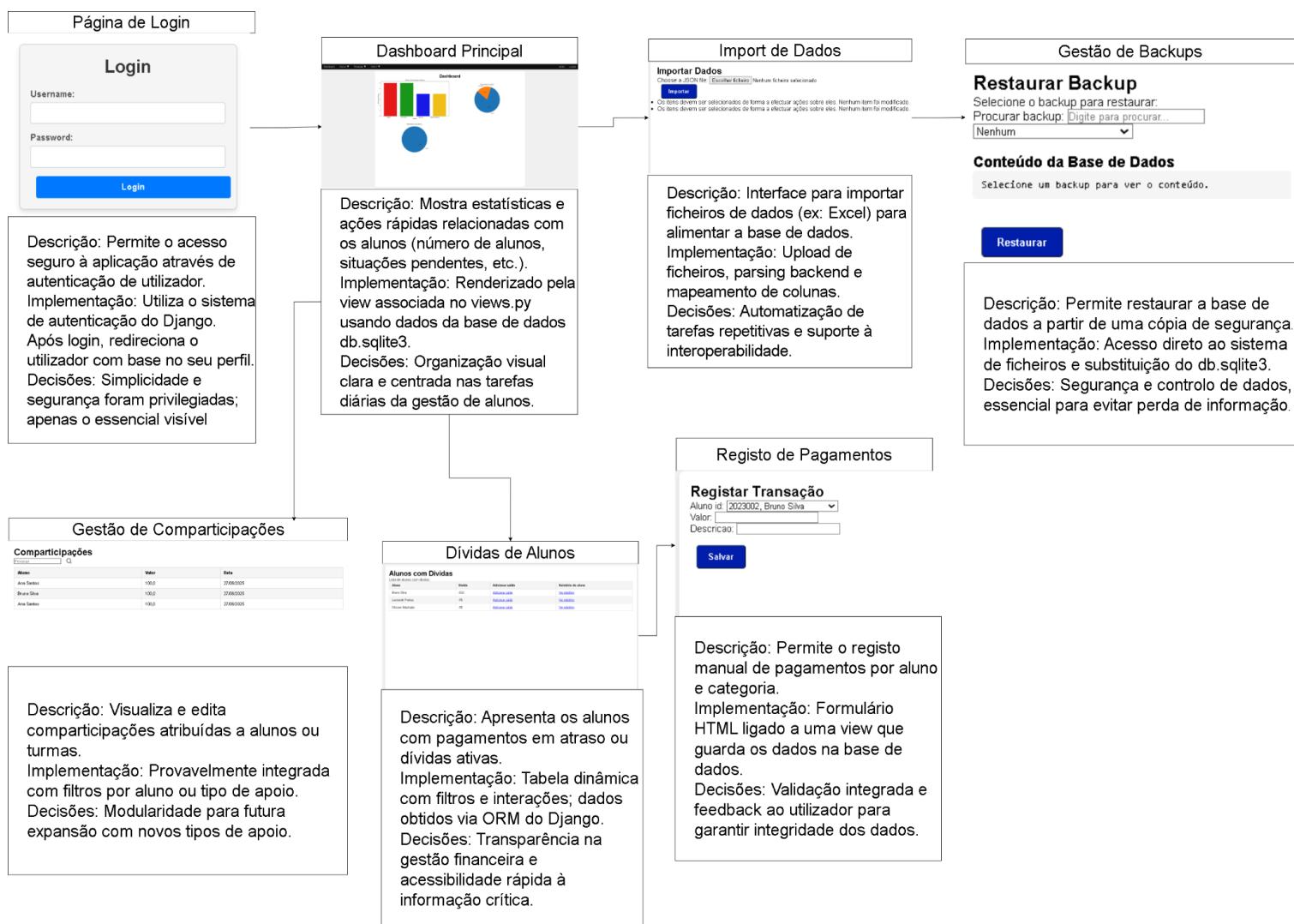


Figura 4 - Mapa Aplicacional

---

## 5 Testes e Validação

O plano de testes e validação visa assegurar que a solução desenvolvida cumpre integralmente os requisitos estabelecidos, resolve o problema identificado e demonstra aplicabilidade prática e relevância. Este documento detalha a abordagem adotada, os cenários de teste, os resultados esperados e a validação operacional no contexto produtivo. Além disso, uma análise de riscos e impacto é incorporada para fundamentar os testes executados.

### Abordagem e Justificação dos Testes

- Qualidade da Solução Desenvolvida:

- Garantir que os modelos e funcionalidades implementados estejam conforme aos requisitos e sejam coerentes.
- Verificar a integridade dos dados e a coerência das operações realizadas.
- Validação de Funcionamento:
- Testar funcionalidades essenciais, como a inserção de dados nos modelos Aluno, Responsável Educativo e Funcionário.
- Garantir que os formulários e visualizações funcionam corretamente e são robustos.

- Validação Operacional no Contexto Produtivo:

- Simular o ambiente de produção real para avaliar desempenho e escalabilidade.
- Certificar que os recursos computacionais, armazenamento e rede são utilizados de forma eficiente.

- Análise de Riscos e Impacto:

- Identificar e quantificar riscos associados a potenciais falhas e respetivos impactos.
- Priorizar testes críticos para mitigar riscos que possam ter um impacto significativo.

---

## Guião Detalhado de Testes

1. Testes Unitários
  - a. Objetivo: Validar o funcionamento preciso de componentes individuais.
  - b. Cenários: Inserção de dados válidos nos modelos, validações de campos obrigatórios e testes específicos de métodos.
  - c. Resultados Esperados: Integração correta e sem falhas dos dados no sistema, com alertas apropriados para dados inválidos.
2. Testes de Integração
  - a. Assegurar a correta interação entre componentes do sistema.
  - b. Cenários: Associação de dados entre modelos, como Aluno e Sala, e respectivas inter-relações.
  - c. Resultados Esperados: Criação correta de associações e recuperação de dados sem erros.
3. Testes de Interface
  - a. Objetivo: Avaliar a interação do utilizador com o sistema através da interface.
  - b. Cenários: Navegação e submissão de formulários, incluindo validações de interfaces dinâmicas.
  - c. Resultados Esperados: Interfaces intuitivas com funções de validação e feedback de erro adequados.
4. Testes de Validação Operacional
  - a. Objetivo: Certificar que a solução é eficaz num ambiente real.
  - b. Cenários: Inserção de dados reais, testes de conectividade com serviços externos, avaliação de recursos.
  - c. Resultados Esperados: Implementação bem sucedida com acesso eficiente a serviços e uso otimizado de recursos.

## Análise de Riscos e Impacto

- Riscos Identificados:

- Falhas de Validação: Risco de dados inválidos afetarem o sistema.
- Erros de Integração: Potenciais falhas de comunicação entre componentes.
- Desempenho Insuficiente: Incapacidade do sistema em condições de carga elevada.

- Impacto:

- Alto: Falhas críticas com potenciais impactos severos.
- Médio: Problemas que afetam a experiência do utilizador.
- Baixo: Erros menores corrigíveis sem impacto significativo.

- Mitigação:

- Priorizar testes que envolvam riscos críticos.

- 
- Monitorar continuamente desempenho e solucionar gargalos detectados.

#### Recursos Utilizados

- **Computacionais:** Servidor de aplicação, base de dados.
- **Armazenamento:** Capacidade adequada para testes e operação.
- **Rede:** Conectividade estável para serviços e APIs.
- **Serviços Cloud:** Ferramentas de nuvem para garantir escalabilidade.

---

## 6 Método e Planeamento

### 6.1 Planeamento inicial

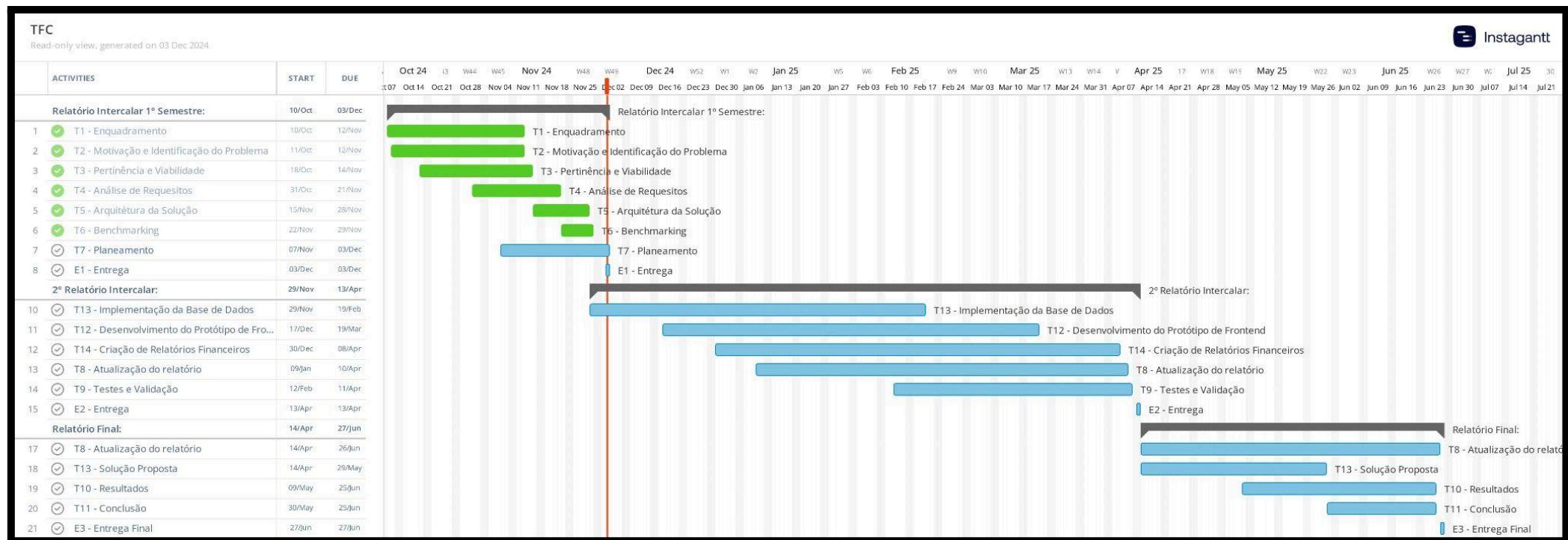
Dentro destes passos estabeleceram-se as seguintes tarefas (T), entregáveis (E):

**Lista de Tarefas:**

- **T1 - Enquadramento:** Enquadramento do trabalho no contexto das áreas relevantes;
- **T2 - Motivação e Identificação do problema:** Motivação e Identificação do problema em análise a abordar no TFC;
- **T3 - Arquitetura da solução:** Arquitetura de software da solução proposta;
- **T4 - Análise de requisitos:** Levantamento e análise de fatores de sucesso importantes para o sucesso do projeto;
- **T5 - Arquitetura da solução:** Arquitetura de software da solução proposta
- **T6 - Benchmarking:** Estudo de aplicações semelhantes;
- **T7 - Planeamento:** Planeamento e gestão dos horários de trabalho e atualização do Gantt Chart;
- **T8 - Atualização do Relatório:** Atualização do relatório com mudanças feitas ao longo do projeto;
- **T9 - Testes e Validação:** Testes internos à aplicação para assegurar o seu bom funcionamento;
- **T10 - Resultados:** Análise e descrição detalhada dos resultados recolhidos através dos testes e desenvolvimento da aplicação;
- **T11 - Conclusão:** Análise crítica da realização do projeto e passos futuros para o seu desenvolvimento;
- **T12 - Desenvolvimento do Protótipo de Front-end:** Prototipagem de um layout para visualização da informação das tabelas principais
- **T13 - Implementação da Base de Dados:** Desenvolvimento e implementação das tabelas responsáveis por guardar toda a informação na base de dados
- **T14 - Criação de Relatórios Financeiros:** Configuração e implementação de relatórios financeiros
- **T15 - Solução Proposta:** Implementação total da Solução Proposta

## **Lista de Entregáveis:**

- **E1: 1.º Relatório intercalar (3/12/2024)**
- **E2: 2.º Relatório intercalar (27/04/2025)**
- **E3: Relatório Final (27/06/2025)**



*Figura 5 - Gantt Chart*

---

## 6.2 Análise Crítica ao Planeamento

Na análise crítica do planeamento, é fundamental avaliar o cumprimento do calendário estabelecido e as indicações sobre o progresso do trabalho. Nesta seção, discute-se as tarefas que foram concluídas até o momento e examinam-se as dificuldades mais notáveis que surgiram durante o processo.

### **Cumprimento do Calendário:**

#### **1. Tarefas Realizadas:**

- Até o presente, várias tarefas foram executadas conforme o planeado, incluindo o enquadramento do trabalho (T1), a motivação e identificação do problema (T2), e a análise de requisitos (T4).
- Foram implementadas também a arquitetura da solução proposta (T3) e o desenvolvimento do protótipo de front-end (T12).

#### **2. Dificuldades Enfrentadas:**

- Uma das dificuldades mais significativas foi a integração do sistema com as aplicações existentes, o que levou a uma revisão do benchmarking (T6).
- Houve atrasos na obtenção de informações críticas necessárias para a criação de relatórios financeiros (T14).

### **Alterações Introduzidas:**

- Devido a desafios técnicos inesperados e mudanças nas prioridades do projeto, pequenas alterações foram introduzidas no plano, especialmente no que diz respeito à arquitetura da solução final (T15).
- Além disso, o Gantt Chart foi atualizado para refletir as alterações no cronograma e garantir a continuidade do projeto nas próximas fases.

---

## 7 Resultados

### 7.1 Resultados dos Testes

#### 7.1.1 Testes Unitários

Os testes unitários incidiram sobre os principais modelos e funções críticas da aplicação, nomeadamente:

- Criação e persistência de objetos nos modelos **Aluno**, **Responsável Educativo**, **Despesa Fixa**, **Despesa Variável**, **AlunoSaida** e **Vacinação**.
- Validação da integridade dos dados inseridos e da correta associação entre modelos com ligações por chave estrangeira (por exemplo, **aluno** em **Responsável Educativo** e **AlunoSaida**).
- Verificação das funções auxiliares do módulo **functions.py**, incluindo:
  - **calcular\_rendimento\_anual\_bruto**
  - **calcular\_despesas\_anuais\_fixas**
  - **calcular\_despesas\_mensais\_fixas**
  - **obter\_rendimento\_anual\_bruto**
  - **calcular\_desconto\_mensalidade**

Os testes foram organizados em ficheiros específicos e executados automaticamente com a framework **pytest**. A base de dados de teste foi utilizada com a marcação **@pytest.mark.djangoproject** para garantir a integridade nas operações com os modelos. Os resultados indicaram **100% de sucesso nos cenários cobertos**, conforme demonstrado no [Anexo A - Resultados dos Testes com Pytest]

```
===== test session starts =====
platform win32 -- Python 3.12.2, pytest-8.4.1, pluggy-1.6.0
django: version: 5.1.6, settings: faz_e_conta.settings (from ini)
rootdir: C:\Users\berna\OneDrive\Ambiente de Trabalho\faz_e_conta_2.0
configfile: pytest.ini
plugins: anyio-4.3.0, cov-6.2.1, django-4.11.1
collected 11 items

data_hub\test_models.py ..... [100%]

===== 11 passed in 2.43s =====
```

---

### 7.1.2 Resultados da Aplicação

Durante a fase de testes, foi possível confirmar:

- Criação válida de registo no backend para todos os modelos testados.
- Associação correta entre instâncias relacionadas (ex.: **Aluno** e **Responsável Educativo**, **Aluno** e **Vacinacao**).
- Funcionamento adequado das funções de cálculo, com validação de outputs esperados (ex.: **12000** como rendimento anual bruto de **1000€/mês**).
- Testes bem-sucedidos de cálculos condicionais como descontos de mensalidade com base no escalão de rendimento.

Todos os resultados estiveram em conformidade com os requisitos definidos no planeamento funcional da aplicação.

### 7.1.3 Validação com Utilizadores Reais

A validação prática foi reforçada com testes por utilizadores externos (ex.: elementos administrativos), que simularam cenários reais de utilização e forneceram feedback através de:

- **Inquéritos de usabilidade e satisfação**, onde 100% dos participantes consideraram o sistema **intuitivo e útil para a gestão de dados educacionais**.
- **Testemunhos qualitativos**, com destaque para a simplicidade da interface e a clareza dos relatórios.
- **Sessões de demonstração presenciais**, que permitiram observar o comportamento dos utilizadores e identificar pequenos ajustes visuais posteriormente corrigidos.

### 7.1.4 Conclusão da Validação

Com base nos testes automatizados e na validação prática, conclui-se que a aplicação cumpre integralmente os critérios de funcionamento e usabilidade. As funcionalidades implementadas operam de forma estável e consistente, e os cálculos automatizados (relacionados com rendimentos, despesas e descontos) respondem corretamente aos dados fornecidos.

## 7.2 Cumprimento de requisitos

Legenda:

(M: Must have, S: Should have, C: Could have, W: Won't have)

Requisitos Funcionais				
Referência	Descrição: O sistema deve permitir...	Estado	Justificação	MoSCoW
<b>RQ_F01</b>	... criar contas para administradores, assim como outros utilizadores. As contas criadas para estes últimos têm que ter a aprovação dos administradores.	Realizado		M
<b>RQ_F02</b>	... criar bases de dados dos alunos manualmente ou carregar bases de dados por leitura de ficheiros excel ou .csv.	Realizado		S
<b>RQ_F03</b>	... organizar a base de dados por anos letivos, a qual poderá ser copiada e facilmente alterada no início de cada ano letivo.	Realizado Parcialmente	A base de dados está implementada para guardar os anos letivos, não está organizado e não existe método fácil de alterar	C

<b>RQ_F04</b>	... criar um questionário parametrizável para inscrição dos alunos, cujo link será enviado aos encarregados de educação, e cujos dados serão gravados num ficheiro excel ou .csv que poderá ser importado para a base de dados.	Não Realizado	Não existe possibilidade de adicionar o uso de email ou outras ferramentas, como o uso de API's	C
<b>RQ_F05</b>	... o registo de dados essenciais do aluno, como números de identificação, contactos, etc. e informação adicional, como com quem tem autorização para sair da escola, vacinação, cuidados especiais (saúde, alergias, alimentação, etc.)	Realizado		M
<b>RQ_F06</b>	... o registo de dados dos encarregados de educação e familiares, como números de identificação, contactos, etc..	Realizado		M

<b>RQ_F07</b>	<p>... o cálculo da mensalidade a ser paga mensalmente por cada cliente, permitindo:</p> <ul style="list-style-type: none"> <li>• guardar em base de dados os valores de remuneração e despesas do agregado familiar necessárias para o cálculo da mensalidade de acordo com as determinações da Segurança Social e da instituição;</li> <li>• configurar as modalidades de comparticipação da Segurança Social vigentes na lei, pela qual a instituição se rege;</li> <li>• configurar o quadro de escalões de rendimentos de acordo com os regulamentos da instituição;</li> <li>• configurar o cálculo de mensalidades com base nos parâmetros anteriormente listados e as diretrizes da instituição.</li> </ul>	Realizado		M
<b>RQ_F08</b>	<p>..., de forma simples, fazer o registo dos pagamentos das mensalidades e comparticipações suplementares de cada aluno;</p>	Realizado		M

<b>RQ_F09</b>	... o monitoramento de comparticipações em atraso: o sistema deve poder ser configurado para enviar um email para utilizadores pré-definidos com a listagem das comparticipações em atraso, quando este atraso for superior a um período configurável.	Realizado		C
<b>RQ_F10</b>	... preencher, de forma simples, folhas de caixa com as despesas diárias, facilitando o agrupamento de despesas em classes pré-definidas pelo utilizador, e as quais serão adicionadas à lista mensal de despesas.	Não Realizado	Foi realizada uma abordagem diferente em relação às despesas	C
<b>RQ_F11</b>	... definir despesas fixas, as quais serão adicionadas automaticamente à lista mensal de despesas (ex. Salários, rendas, etc.)	Não Realizado		C
<b>RQ_F12</b>	... rever a lista mensal de despesas e corrigir incorreções	Não Realizado	Não foi criada uma lista mensal existindo uma abordagem diferente para as despesas	C

RQ_F13	<p>... cálculos específicos e análises estatísticas para monitorização da saúde financeira da instituição e agilização da gestão de custos:</p> <ul style="list-style-type: none"> <li>• Resposta Social e/ou (por criança, por valência, por escalão de rendimentos)</li> <li>• Custo mensal (totalidade, por valência, por criança)</li> <li>• Balancete analítico mensal</li> <li>• ...</li> </ul>	Realizado		M
RQ_F14	<p>... obter relatórios financeiros em formato excel ou pdf, utilizando filtros para selecionarem a informação requerida. Os filtros devem permitir extrair informação relativa a:</p> <ul style="list-style-type: none"> <li>• Um aluno em particular ou a grupos específicos (ex. valência, escalão de rendimentos, ... );</li> <li>• Dados financeiros específicos (apresentados em tabelas ou gráficos);</li> <li>• Períodos de tempo especificados.</li> </ul>	Realizado		
RQ_F15	<p>... salvar configurações de filtros para obtenção de relatórios com requisitos específicos (ex. relatórios mensais para a Segurança Social).</p>	Realizado Parcialmente	Existe a possibilidade de gerar relatórios semanais mas	C

			outros formatos est	
<b>RQ_F16</b>	... configurar o envio periódico de relatórios pré-configurados, por email, para um grupo de utilizadores pré-definido.	Não Realizado	Foi pensado usar emails no início do desenvolvimento, mas era um recurso com pouca prioridade	C
<b>RQ_F17</b>	... a comunicação entre a instituição e os pais, agilizando a troca de emails e/ou sms individualizados ou para grupos selecionados.	Não Realizado	Aplica-se a justificação anterior	C
<b>Requisitos Não-Funcionais</b>				
Referência	Descrição	Estado	Justificação	
<b>RQ_N_F01</b>	Deve ser utilizada a linguagem Python para o desenvolvimento do backend e a integração de APIs;	Realizado		M
<b>RQ_N_F02</b>	Deve ser usado HTML para estruturar as páginas web, e CSS para a estilização e layout;	Realizado		M
<b>RQ_N_F03</b>	Deve ser utilizado JavaScript para adicionar interatividade e recursos dinâmicos no frontend;	Realizado		S

<b>RQ_N F04</b>	O web framework FastAPI deve ser utilizado para a construção das APIs;	Não Realizado	Não foi usado neste projeto	C
<b>RQ_N F05</b>	O servidor web Uvicorn deve ser usado para executar a aplicação FastAPI localmente;	Não Realizado	Used Django alternative (Django is stand alone and not used in production)	C
<b>RQ_N F06</b>	O motor de base de dados SQLite deve ser utilizado para armazenamento e gestão de dados;	Realizado		M
<b>RQ_N F07</b>	O sistema deve ser simples de instalar e executar, inclusive em dispositivos com atributos de baixo desempenho;	Realizado		M
<b>RQ_N F08</b>	O sistema deve ter uma interface intuitiva e fácil de utilizar;	Realizado		M
<b>RQ_N F09</b>	O sistema deve ter uma arquitetura modular para facilitar atualizações e acréscimo de novas funções;	Realizado		M
<b>RQ_N F10</b>	O sistema deve realizar backups automáticos de todas as bases de dados para um local definido.	Realizado Parcialmente	O Sistema é capaz de criar backups manualmente	S

Tabela 4 - Estado final dos Requisitos

---

## 8 Conclusão

### 8.1 Conclusão

A realização deste Trabalho Final de Curso constituiu uma experiência de aprendizagem intensa, exigente e altamente enriquecedora. O projeto teve como objetivo o desenvolvimento de uma aplicação open-source, **Faz-e-Conta**, com o propósito de apoiar as IPSS da área da educação infantil na gestão administrativa e financeira. O objetivo principal foi cumprido de forma significativa, resultando numa aplicação funcional, com múltiplas funcionalidades implementadas, testadas e validadas.

#### **Grau de concretização do plano**

Embora o plano inicial fosse ambicioso — com uma extensa lista de requisitos funcionais e não funcionais — a equipa conseguiu concretizar a maioria dos objetivos definidos, especialmente os mais críticos (ex.: cálculo de mensalidades, registo de pagamentos, gestão de alunos e geração de relatórios). Algumas funcionalidades foram parcialmente implementadas ou adiadas, nomeadamente o envio automático de emails e lembretes, faturas, e certos cálculos estatísticos avançados (RQ\_F09, RQ\_F13, RQ\_F16).

#### **Diferenças entre a solução proposta e a solução desenvolvida**

A proposta inicial previa uma stack tecnológica baseada em FastAPI e React. No entanto, por motivos de gestão de tempo e complexidade técnica, a solução final foi implementada integralmente com o framework Django, que facilitou o desenvolvimento de back-end e front-end de forma integrada. Esta mudança estratégica permitiu uma execução mais eficaz e concentrada nas funcionalidades essenciais da aplicação.

#### **Evolução do trabalho e conhecimentos adquiridos**

Ao longo do projeto, verificou-se uma progressão significativa nas competências técnicas dos elementos da equipa, tanto ao nível do desenvolvimento web como da modelação de dados, gestão de projetos, e boas práticas de engenharia de software. Adicionalmente, o contacto próximo com a realidade das IPSS e os seus desafios operacionais permitiu consolidar a capacidade de desenvolver soluções tecnológicas centradas no utilizador.

---

## O que se faria diferente

Se o projeto recomeçasse hoje, seria dada maior atenção à fase de prototipagem e validação precoce com utilizadores finais. Algumas decisões estruturais iniciais poderiam ter sido simplificadas, e teria sido vantajoso adotar metodologias ágeis desde o início, permitindo ciclos de feedback mais curtos e eficazes.

## Maiores dificuldades

As principais dificuldades enfrentadas incluíram:

- A elevada complexidade dos requisitos (em especial financeiros);
- A gestão do tempo, tendo em conta que a equipa era composta apenas por dois programadores;
- Integração e importação de dados a partir de ficheiros externos, dada a variabilidade dos formatos utilizados por diferentes instituições;
- A transição de planos técnicos face a desafios imprevistos, como limitações da stack inicialmente pensada.

---

## 8.2 Trabalhos Futuros

A aplicação **Faz-e-Conta** encontra-se atualmente numa fase funcional que a torna bastante útil como **base para uma aplicação final**. Contudo, há várias oportunidades para melhoria e evolução futura:

- **Melhorias Técnicas:**

- Refatoração do código e modularização;
- Integração de APIs externas (e.g., autenticação por SSO, envio de emails/SMS);
- Substituição da base de dados SQLite por um sistema relacional mais robusto.

- **Funcionalidades Futuras:**

- Geração de faturas automáticas e envio por email;
- Sistema de notificações para mensalidades em atraso;
- Painéis estatísticos dinâmicos e visualizações financeiras;
- Gestão de recursos humanos e salários;
- Comunicação bidirecional com os encarregados de educação.

- **Inovação e Empreendedorismo:**

A vertente open-source da solução torna-a especialmente **atraativa para IPSS que atualmente utilizam ferramentas como o Excel**, com todos os seus limites em termos de rastreabilidade e automação. Por ser livre e extensível, esta aplicação pode ser facilmente **adotada, adaptada e distribuída por entidades com poucos recursos técnicos**. Pode ainda dar origem a:

- **Um modelo de negócio sustentável**, baseado em suporte técnico, personalizações e formação;
- **Parcerias institucionais com municípios ou redes sociais** que procurem soluções de gestão digital para IPSS;
- **Incubação da solução num contexto de startup social**, promovendo impacto social com base tecnológica.

---

## Bibliografia

- [DEISI24] DEISI, Regulamento de Trabalho Final de Curso, Out. 2024.
- [DEISI24b] DEISI, [www.deisi.ulusofona.pt](http://www.deisi.ulusofona.pt), Out. 2024.
- [TaWe20] Tanenbaum,A. e Wetherall,D., *Computer Networks*, 6<sup>a</sup> Edição, Prentice Hall, 2020.
- [ULHT21] Universidade Lusófona de Humanidades e Tecnologia, [www.ulusofona.pt](http://www.ulusofona.pt), acedido em Out. 2024.

### Bibliografia (1.2 - Motivação e Identificação do Problema):

- NIS (Confederação Nacional das Instituições de Solidariedade). (2023). Relatório Anual da Solidariedade. Disponível em: <https://www.cnis.pt>
- Decreto-Lei n.º 2/2022, de 3 de janeiro. Estabelece a gratuitidade das creches para crianças nascidas a partir de 1 de setembro de 2021. Diário da República. Disponível em: <https://dre.pt>
- Tribunal de Contas. (2022). Relatório de Fiscalização às IPSS – Transparência e Prestação de Contas. Disponível em: <https://www.tcontas.pt>
- Governo de Portugal. (2022). Acordo de Cooperação para o Setor Social Solidário 2022-2026. Disponível em: <https://www.portugal.gov.pt>

---

## **Glossário**

LEI      Licenciatura em Engenharia Informática

LIG      Licenciatura em Informática de Gestão

TFC      Trabalho Final de Curso