

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Gestão de fluxo de dados para processos de injunção

Mário Macedo



Mestrado em Engenharia Informática e Computação

Orientador: Prof. João Correia Lopes

Co-orientador: Dr. Tiago Lopes de Azevedo

4 de Outubro de 2022

Gestão de fluxo de dados para processos de injunção

Mário Macedo

Mestrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: Prof. Rui Camacho

Arguente: Prof. Maria Benedita Malheiro

4 de Outubro de 2022

Resumo

O conceito de dívida é mais antigo que o conceito de moeda ou transação comercial e ao longo dos tempos têm sido desenvolvidas novas estruturas e metodologias para melhorar e simplificar o ato de cobrança de dívida. Contudo, apesar de estas se terem tornado mais humanas com o evoluir da civilização, através da utilização de vias judiciais, tornaram-se também mais complexas, morosas e dispendiosas para todos os envolvidos: credores, devedores e agentes de justiça.

No sentido de combater a complexidade inerente a estes processos, os diferentes países foram implementando diferentes tipos de medidas. Em 2008, o Ministério da Justiça português criou o processo de injunção que consiste num procedimento judicial que tem como objetivo o cumprimento de obrigações pecuniárias, decorrentes de contratos, com valor não superior a 15 000,00€ que é gerido pelo BNI, também instaurado na mesma altura. De forma a suportar esta nova entidade judicial, foi desenvolvida uma plataforma *Web*, denominada Citius, que permitiu a desmaterialização de vários tipos de processos (incluindo o processo de cobrança de dívida) anteriormente geridos presencialmente e organizados fisicamente, em papel.

Apesar da melhoria ter sido significativa, como provado por estudos internacionais, a abertura de um processo de injunção unitária através do Citius continua a ser algo moroso e altamente sujeito a erros humanos, como é o exemplo do cálculo de juros. Esta plataforma permite também a submissão de lotes de injunções, o que reduz o custo de abertura dos processos; no entanto requer que o agente de justiça tenha conhecimento de XML ou tenha algum tipo de *software* proprietário o que é reservado tradicionalmente a empresas com grande volume de negócios.

Assim, no âmbito desta dissertação foi desenvolvido um MVP de uma aplicação *Web*, que se pretende instaurar no mercado nacional como um serviço à base de subscrição, acessível a empresas de menor dimensão. Esta aplicação, *Fluxio*, permite a gestão de todos os devedores e processos de dívida do credor, destacando-se o cálculo de juros automático e a exportação de um ficheiro XML, que possui um número indefinido de processos, que pode ser depois submetido na secção dos lotes de injunções, no Citius.

O objetivo desta dissertação foi obtido com sucesso e, do ponto de vista das regras de negócio, validado através de sessões de demonstração. Já do ponto de vista técnico, o MVP foi validado através da implementação de um validador de XML e através de testes de desempenho onde se identificaram melhorias na ordem dos 40% no caso de leituras à base de dados.

Palavras-chave: injunção, taxas, prazos, XML, dívida, cobrança de dívida, MVP

Abstract

The concept of debt is older than the concept of money or commercial transactions and over time new structures and methodologies have been developed to improve and simplify the act of debt collection. Even though these have become more humane with the evolution of civilisation, through the use of judicial means, they have also become more complex, time consuming and costly for all involved: creditors, debtors and bailiffs.

In order to combat the complexity inherent to these processes, different countries have implemented different types of measures. In 2008, the Portuguese Ministry of Justice created the injunction process which consists in a judicial procedure aimed at the compliance with pecuniary obligations, arising from contracts, with a value not exceeding EUR 15,000.00 which is managed by the BNI, also instated at the same time. In order to support this new judicial entity, a Web platform, called Citius, was developed, which allowed the dematerialisation of several types of processes (including the debt collection process) previously managed in person and organised physically, using paper.

Despite the significant improvement, as proven by international studies, the opening of a unitary injunction process through Citius is still somewhat time consuming and highly subject to human error, as is the case of the calculation of interest rates. This platform also allows the submission of batches of injunctions, which reduces the cost of opening proceedings; however it requires that the bailiff has knowledge of XML or has some type of proprietary software which is traditionally reserved for companies with high turnover.

Thus, in the scope of this dissertation, an MVP was developed for a Web application, which is intended to be established in the national market as a service on a subscription basis, accessible to smaller companies. This application, *Fluxio*, allows the management of all debtors and creditor's debt processes, highlighting the automatic interest rates calculation and the exportation of an XML file, which has an indefinite number of processes in it, that can then be submitted in the injunction batches section, in Citius.

The goal of this dissertation was successfully achieved and, from the point of view of the business rules, validated through demonstration sessions. From the technical point of view, the MVP was validated through the implementation of an XML validator and through performance tests where improvements of about 40% were identified in the case of database readings.

Keywords: injunction, taxes, deadlines, XML, debt, debt collection, MVP

Agradecimentos

Em primeiro lugar gostaria de agradecer ao Prof. João Correia Lopes, por me ter aceite como aluno auto-proposto, por compreender a minha situação familiar e profissional, resultando em reuniões fora do horário laboral e por continuamente me manter motivado ao longo do último ano.

De seguida, agradeço ao Dr. Tiago Lopes de Azevedo, com o qual tenho tido a sorte de vir a trabalhar esporadicamente ao longo dos últimos anos e do qual partiu a ideia para esta dissertação e para o produto desenvolvido. É de realçar o extenso conhecimento sobre todas as questões judiciais mencionadas ao longo da dissertação e em particular pela curiosidade tecnológica e mentalidade empreendedora.

Ao Tiago Tavares por me introduzir ao mundo da *Web* e ensinar-me Laravel e PHP enquanto ainda estava no segundo ano do curso. A ele devo muito do meu sucesso profissional. Obrigado por seres o amigo que és e por ainda hoje continuares a “puxar” por mim na Cyber-Duck.

Aos meus colegas na LudusCristaltec, por terem marcado o início da minha carreira enquanto programador e gestor de projetos. Em especial ao Manuel, Samuel e João por me terem dado todas as oportunidades e mais algumas para me destacar enquanto profissional e por compreenderem a minha situação de trabalhador-estudante durante quase 6 anos.

Ao Império dos Jabardos, o meu grupo de amigos do ensino secundário e que ainda hoje prevalece, por terem partilhado a jornada académica comigo, por vezes um pouco atribulada.

À minha Mãe, um agradecimento especial por ter feito um enorme esforço pessoal e financeiro para que eu e o meu irmão pudéssemos ter mais esta oportunidade e por ser a minha inspiração como pessoa.

Ao meu irmão, com o qual tive o prazer de partilhar alguns anos de Universidade, para não seguir o meu exemplo e terminar o curso em menos de 10 anos.

Faço um agradecimento especial à minha esposa Sara, com o qual tenho o enorme prazer de partilhar 12 anos da minha vida e por não me ter deixado desistir do Mestrado.

Por fim, ao meu filho Afonso, que à data da defesa deste trabalho pode ainda não ter nascido ou ter apenas alguns dias de vida. A tua chegada foi a motivação que precisava para terminar o curso este ano.

Mário Macedo

*“Whether you think you can, or you think you can’t,
you’re right.”*

Henry Ford

Conteúdo

1	Introdução	1
1.1	Contexto e motivação	1
1.2	Objetivos e resultados esperados	2
1.3	Estrutura da dissertação	2
2	Estado da arte	3
2.1	Introdução	3
2.2	Citius	4
2.3	Procedimento para cobrança de dívidas	5
2.4	O processo de injunção	5
2.4.1	Modalidades de instauração do procedimento de injunção	6
2.4.2	Injunção unitária	9
2.4.3	Injunções em lote	13
2.5	Revisão da tecnologia	14
2.6	Sumário	18
3	Identificação do problema e definição dos requisitos	19
3.1	Introdução	19
3.2	Especificação dos requisitos	20
3.2.1	Atores	20
3.2.2	<i>User stories</i>	23
3.2.3	Requisitos técnicos	26
3.2.4	Requisitos do sistema	26
3.3	Modelo do domínio	27
3.4	Sumário	28
4	Fluxio	30
4.1	Introdução	30
4.2	Arquitetura da aplicação	32
4.3	Cenários de utilização	33
4.3.1	Autenticação e autorização	33
4.3.2	Domínio do ator Administrador	36
4.3.3	Domínio do ator Credor	36
4.4	Sumário	54
5	Validação	56
5.1	Introdução	56
5.2	Validação dos requisitos	56

5.3	Validação do lote de injunções	57
5.4	Testes de desempenho	59
5.5	Sumário e conclusões	60
6	Conclusões e trabalho futuro	61
6.1	Resultados	61
6.2	Trabalho futuro	62
	Referências	63
A	XSD de validação do ficheiro XML das injunções em lote (V1.5)	64
B	Ficheiro XML gerado pelo Fluxio com um lote the injunções	77
C	Validador do ficheiro XML que contém o lote de injunções a ser submetido no Citius	84

Lista de Figuras

2.1	Gráfico comparativo da pontuação e <i>rating</i> no processo de cobrança de dívida entre vários países do mundo (Fonte: Estudo da Euler Hermes[5])	7
2.2	Tabela comparativa da complexidade de diferentes etapas no processo de cobrança de dívida entre vários países do mundo (Fonte: Estudo da Euler Hermes[5])	8
2.3	Primeiro passo do formulário do Citius para a submissão de um processo de Injunção Unitária	9
2.4	Segundo passo do formulário do Citius para a submissão de um processo de Injunção Unitária	10
2.5	Campos de seleção sem pesquisa no segundo passo do formulário do Citius	11
2.6	Terceiro passo do formulário do Citius para a submissão de um processo de Injunção Unitária	12
2.7	Quarto passo do formulário do Citius para a submissão de um processo de Injunção Unitária	12
2.8	Quinto e sexto passos do formulário do Citius para a submissão de um processo de Injunção Unitária	13
2.9	Último passo do formulário do Citius para a submissão de um processo de Injunção Unitária	13
2.10	Formulário de submissão do ficheiro XML na plataforma Citius	14
2.11	Representação gráfica do modelo MVC	16
3.1	Diagrama de atores da aplicação	21
3.2	Diagrama de atores e sub-atores da aplicação	23
3.3	Modelo de domínio da aplicação Fluxio	28
4.1	Fluxograma representativo dos processos de cobrança de dívida e de injunção	31
4.2	Diagrama de componentes do Fluxio	32
4.3	Fluxio - Página de autenticação	34
4.4	Fluxio - Painéis de controlo apresentados depois de o utilizador se autenticar	35
4.5	Fluxio - Registo e detalhes de um Credor por parte do Administrador	37
4.6	Fluxio - Registo de um novo Advogado	38
4.7	Fluxio - Listagem e detalhes da entidade Advogado	39
4.8	Fluxio - Formulário de atualização dos dados de um Advogado existente	40
4.9	Fluxio - Registo de um novo Devedor	40
4.10	Fluxio - Páginas de listagem e detalhes da entidade Devedor	41
4.11	Fluxio - Formulário de atualização dos dados de um Devedor existente	42
4.12	Fluxio - Registo de um novo Processo	43
4.13	Fluxio - Listagem dos Processos previamente registados	44
4.14	Fluxio - Detalhes de um Processo	45

4.15 Fluxio - Formulário de atualização dos dados de um Processo existente	46
4.16 Fluxio - Registo de uma nova Fatura	48
4.17 Fluxio - Filtros para filtragem dos processos mostrados	50
4.18 Fluxio - Criação do ficheiro XML a partir dos processos seleccionados	51
4.19 Fluxio - Notificação de ficheiro disponível para transferir dentro da aplicação . . .	53
4.20 Fluxio - Notificação de ficheiro disponível para transferir através de e-mail	54
4.21 Fluxio - Listagem de todos os ficheiros XML gerados pelo sistema	55
5.1 Submissão do ficheiro XML gerado pelo Fluxio no Citius com sucesso	57
5.2 Quantidade de <i>queries</i> efetuadas e tempo total do pedido quando é utilizado <i>Eager Loading</i>	59
5.3 Quantidade de <i>queries</i> efetuadas e tempo total do pedido quando é utilizado <i>Lazy Loading</i>	59

Lista de Tabelas

3.1	Descrição e exemplos para cada ator do sistema	21
3.2	Descrição e exemplos para cada ator e sub-ator do sistema	22
3.3	<i>User stories</i> do ator Não Autenticado	24
3.4	<i>User stories</i> do ator Autenticado	24
3.5	<i>User stories</i> do Credor	24
3.6	<i>User stories</i> do Administrador	26
3.7	Requisitos técnicos da aplicação	27
3.8	Requisitos de sistema da aplicação	27

Listings

4.1	Permissão de visualização da listagem de Taxas de Juro	36
4.2	Implementação da estatística “Processos ativos”	42
4.3	Registo dos campos da entidade Processo	47
4.4	<i>Accessor</i> da entidade Fatura responsável for calcular o valor dos juros	49
4.5	<i>Accessor</i> da entidade Processo responsável for somar o valor dos juros de todas as faturas associadas	49
4.6	<i>Class</i> responsável pelo despacho das notificações de ficheiro XML disponível para transferir	52
5.1	Criação e validação do ficheiro XML antes do mesmo ser comprimido para ser disponibilizado ao utilizador.	58
5.2	<i>Eager Loading</i> de todas as relações da entidade Processo necessárias para gerar o ficheiro XML.	59

Abreviaturas e Símbolos

ACID	Atomicidade, Consistência, Isolamento e Durabilidade
AJAX	<i>Asynchronous JavaScript And XML</i>
API	<i>Application Programming Interface</i>
BNI	Balcão Nacional de Injunções
CRUD	<i>Create, Read, Update, Delete</i>
CSS	<i>Cascading Style Sheets</i>
HDD	<i>Hard Disk Drive</i>
HTML	<i>HyperText Markup Language</i>
JS	<i>JavaScript</i>
MVC	<i>Model-View-Controller</i>
MVP	<i>Minimum Value Product</i>
NVMe	<i>Non-Volatile Memory Express</i>
ORM	<i>Object-Relational Mapper</i>
RAID	<i>Redundant Array of Independent Disks</i>
REST	<i>Representational State Transfer</i>
RGPD	Regulamento Geral da Proteção de Dados
SMTP	<i>Simple Mail Transfer Protocol</i>
SPA	<i>Single Page Application</i>
SSD	<i>Solid State Drive</i>
XML	<i>Extensible Markup Language</i>
XSD	<i>Extensible Markup Language Schema Definition</i>

Capítulo 1

Introdução

1.1 Contexto e motivação

De acordo com autores como David Graeber, o conceito de dívida é mais antigo que o conceito de moeda [4] e desde então que é acompanhado por mecanismos para resolução de dívida, mais vulgarmente conhecidos como cobrança de dívida. Estes mecanismos passaram de escravatura, na Idade Média [8], a métodos mais humanos, mas mais complexos para todos os envolvidos, através da utilização de vias judiciais.

Simultaneamente, a evolução da civilização, em especial nas últimas décadas, com acesso rápido e fácil a diferentes serviços e produtos e onde a moeda é cada vez mais invisível, traduziu-se numa escalada do volume de processos de cobrança de dívida. Estes traduzem-se então em processos judiciais que tradicionalmente envolvem múltiplos agentes de justiça, advogados e juízes, múltiplas deslocações por parte de todos os intervenientes (credores, devedores, advogados, juízes e testemunhas) a tribunais. Tudo isto, resulta em processos dispendiosos para credores e devedores e períodos extensos para a resolução dos mesmos, para além de sobrecarregar instituições como tribunais.

De forma a combater esta complexidade, os diferentes países do Mundo, nomeadamente os países economicamente mais desenvolvidos, têm vindo a desenvolver estruturas e metodologias para simplificar e acelerar o ato de cobrar uma dívida. O Ministério da Justiça português seguiu a mesma tendência e em 2008 criou o Balcão Nacional de Injunções, o Citius (aplicação *Web*) e o processo de injunção. Este trio, veio reduzir significativamente o tempo e o custo da resolução dos processos de cobrança de dívida no panorama nacional.

Claro está, que apesar de a melhoria ter sido expressiva, a criação de um processo de injunção através do Citius, não é de todo a melhor experiência para o utilizador final (que neste caso será um agente de execução, advogado ou solicitador) dado que por cada processo tem que ser preenchido um formulário extenso e altamente sujeito a erro humano (especialmente o cálculo de juros). Quando se tem de preencher este formulário para múltiplos devedores, possivelmente com diversos

processos cada um, percebe-se a limitação do Citius, já para não considerar a interface obsoleta, não responsiva e com graves falhas ao nível da usabilidade. O Citius permite também a submissão de um lote de injunções, o que significa, a submissão de um ficheiro XML com uma quantidade indefinida de devedores e processos. Contudo, este não é um formato de representação de dados que os envolvidos nos processos de injunção estejam tipicamente familiarizados e para o qual não existem ferramentas de criação no mercado nacional.

1.2 Objetivos e resultados esperados

O trabalho apresentado neste documento visa colmatar a lacuna mencionada na secção anterior ao tentar apresentar-se como o primeiro serviço disponível no mercado judicial português para a gestão de processos de dívida e, principalmente, criação automática de lotes de injunções no formato suportado pelo Citius.

A aplicação proposta deverá ser acessível a partir de qualquer lugar, incluindo dispositivos móveis, permitir a gestão de advogados, devedores e respetivos processos e automatizar o máximo de tarefas possíveis. Nomeadamente, a criação de lotes de injunções e o cálculo de juros automático com base nos dias de atraso do pagamento e com o valor dos juros de mora tabelados pelo Ministério da Justiça.

Por último, apesar do produto pretendido no final desta dissertação ser apenas um *Minimum Viable Product* (MVP), devem ser tidos em conta requisitos como a escalabilidade, persistência de dados, dos recursos do servidor onde estiver alojado, velocidade e otimização das *queries* de leitura e escrita e a usabilidade da aplicação por parte de utilizadores com algum tipo de incapacidade.

1.3 Estrutura da dissertação

Para além da introdução, esta dissertação contém mais cinco capítulos. No Capítulo 2 é descrito o estado da arte, clarificando os conceitos judiciais relevantes para os capítulos seguintes e a falta de soluções no mercado nacional. Por sua vez, no Capítulo 3 é feita a especificação dos requisitos da aplicação, recorrendo à identificação dos atores do sistema, *User stories* e requisitos técnicos e de sistema. Já no Capítulo 4 é apresentada a implementação do MVP, evidenciando os principais casos de utilização, arquitetura do sistema e explicação de alguns pormenores técnicos relevantes para a solução. De seguida, no Capítulo 5 é explicada a abordagem no momento de validação dos requisitos da aplicação e descritos os testes de desempenho que foram realizados. Por fim, no Capítulo 6 faz-se uma comparação dos requisitos pretendidos e do que foi efetivamente concretizado. Neste capítulo faz-se também uma análise do trabalho futuro para que a solução proposta possa ter sucesso no mercado.

Capítulo 2

Estado da arte

2.1 Introdução

Nos termos do artigo 1.º do Decreto-Lei 269/98¹, o processo de injunção é um dos procedimentos judiciais que tem por finalidade o cumprimento de obrigações pecuniárias, decorrentes de contratos, de valor não superior a 15 000,00 €². Tendo por finalidade genérica o cumprimento de obrigações pecuniárias, este procedimento de injunção materializa-se no reconhecimento do vínculo pecuniário, do crédito, entre o respetivo credor e devedor.

Assim, do lado do credor, o mesmo pode utilizar o procedimento de injunção para ver reconhecido o seu crédito através de um mecanismo processual simplificado, célere e mais económico, que é tramitado num balcão único nacional, denominado Balcão Nacional de Injunções (BNI).

Em termos breves, este procedimento inicia-se com um pedido, no qual o credor (requerente) fundamenta a origem do seu crédito. Se o devedor (requerido) colocar em causa esse reconhecimento ou crédito, o procedimento de injunção transforma-se numa ação judicial, necessariamente menos ligeira, já que já tem intervenção de um juiz e pode implicar a realização de um julgamento. Porém, se o devedor não colocar em causa o crédito, o procedimento de injunção termina com a aposição de uma fórmula executória³.

Esta fórmula executória permite que o pedido do credor tenha “força de sentença”, podendo o credor iniciar um processo de execução com vista à penhora e venda de bens ou direitos do devedor. Nestes termos, de acordo com o artigo 7.º do Decreto-Lei 32/2003⁴, o procedimento de

¹<https://dre.pt/dre/detalhe/decreto-lei/269-1998-566629> - Publicado a 01/09/1998

²Não se vai definir, por não parecer ser relevante, se no caso se está perante um procedimento ou face a um processo. Utilizar-se-ão os dois conceitos indistintamente, bem se sabendo que juridicamente podem ser distintos, consoante se entenda que este processo tramita junto de uma entidade administrativa, sendo um procedimento, ou junto de uma entidade judicial ou parajudicial, sendo neste caso um processo. Tudo depende da natureza administrativa ou judicial do Balcão Nacional de Injunções.

³Esta fórmula executória não é mais do que um carimbo automático no requerimento do pedido de procedimento de injunção com o seguinte conteúdo: “Este documento tem força executiva.” Desta forma, o credor passa a ter um título executivo, que tem o mesmo valor que uma sentença judicial.

⁴<https://dre.pt/dre/detalhe/decreto-lei/32-2003-189484> - Publicado a 17/02/2003

injunção “(...) tem por fim conferir força executiva a requerimento destinado a exigir o cumprimento das obrigações a que se refere o artigo 1.º do diploma preambular (...)”.

Em suma, o procedimento de injunção permite que os credores tenham uma ferramenta jurídica simples, célere e com menores custos, com vista ao pagamento de créditos (obrigações pecuniárias) resultantes do não pagamento ou atrasos no pagamento, tenham ou não origem comercial.

Este processo caracteriza-se pela simplicidade, já que o seu preenchimento pode ser feito por qualquer credor, sem necessidade de recurso a advogado e através de um formulário amigável e apreensível a qualquer pessoa. Por outro lado é mais célere que o método tradicional, com recurso a uma ação judicial, na qual os factos e respetivo pedido têm de ser descritos em “texto corrido”, com recurso a termos jurídicos e sem utilização de campos de formulário, o que impõe o seu recurso através de técnicos qualificados, como advogados ou solicitadores.

Finalmente, como se está perante um procedimento padronizado, a sua instauração por parte do credor apresenta-se como uma solução mais económica para todos os intervenientes, com a taxa de justiça a ser mais baixa, uma vez que evita, ou tenta evitar, o recurso a uma ação judicial e a tudo o que a mesma pode significar: audiências de partes, julgamentos, sentença e respetivos recursos.

2.2 Citius

O Citius é uma plataforma *Web*⁵ criada em 2008 pelo Ministério da Justiça que procurou, a desmaterialização de vários tipos de processos que tradicionalmente tramitavam nos tribunais judiciais de forma física, em papel, organizados em maços e cosidos à linha ou juntos em capas.

Esta plataforma traduz-se num aglomerado de serviços informáticos especializados nos vários tipos de processos, nomeadamente: publicação da distribuição de processos nos diferentes tribunais, citação e notificação edital, instauração de procedimento especial de despejo, venda de bens penhorados, venda de bens em processos declarativos, publicidade de Processos Especiais de Revitalização, de Processos Especiais de Acordo de Pagamento, de Processos Especiais de Acordo de Pagamento, de Processos Especiais de Revitalização e de processos de insolvência, publicidade da composição da massa envolvente, lista pública de execuções, divisão judicial e administrativa, diligências marcadas, contactos dos tribunais, transição de processos para a nova estrutura judiciária, publicidade de lista de sociedades com mais de 200 ações, intervenções em diligência judiciais e confirmação da autenticidade da comunicação. Finalmente, para o que por ora interessa, a plataforma Citius permite igualmente a instauração de procedimentos de injunção.

Complementarmente ao conjunto de serviços acima referidos, o Citius tem definido um conjunto de permissões ou autorizações para os diversos agentes de justiça em Portugal, tais como, magistrados, funcionários e mandatários judiciais, que permite a utilização parcial ou total dos serviços jurídicos, fazendo com que estes não tenham de fazer tantas deslocações aos tribunais e consigam dar seguimento e acompanhar diferentes processos de forma mais célere, amigável e

⁵<https://www.citius.mj.pt/>

eficaz. Antes da plataforma Citius, as peças processuais tinham de ser enviadas pelos advogados através de correio postal ou entregues em mão, o que implicava necessariamente o respeito dos horários dos correios ou do próprio tribunal; após a plataforma Citius, as peças podem ser enviadas digitalmente, sem qualquer limitação de horário, já que neste caso os tribunais estão “abertos 24 horas por dia”. Por outro lado, esta plataforma permite a consulta dos processos em qualquer lugar, sem necessidade de deslocações aos tribunais.

2.3 Procedimento para cobrança de dívidas

Até 2008, data da criação do BNI, a forma que um prestador de serviços (credor) tinha para recuperar o valor resultante de uma ou mais faturas não pagas ou pagas fora do prazo válido, era através de uma ação judicial junto do Tribunal competente.

Esta ação traduzia-se em despesas elevadas para ambas as partes (credores e devedores) uma vez que ao montante da dívida acresciam custas judiciais elevadas, por força das taxas de justiça com a instauração da ação judicial, e uma grande morosidade, com a realização de julgamentos e sentenças, a que juntavam os honorários dos advogados do credor e do devedor.

Para além dos custos avultados, antes da instalação do Balcão Nacional de Injunções, os tribunais ficavam sobrelotados com estas ações judiciais traduzindo-se em resoluções lentas e que exigiam uma fase de apresentação de articulados que atrasava a cobrança, a par de pelo menos uma audiência de discussão e julgamento, o que implicava mais deslocações dos envolvidos de forma a acompanharem o processo.

Esta necessidade de celeridade revelou-se mais premente quando ficou melhor estabelecido o conceito de consumidor, enquanto utilizador de bens ou serviços públicos essenciais (como o fornecimento de água, gás, eletricidade, *Internet*, telefone ou telemóvel). Estes serviços eram (e são) prestados e cobrados em massa [2]. Os curtos prazos de prescrição com vista à cobrança judicial daqueles serviços (6 meses), levavam a que as entidades prestadoras instaurassem ações judiciais em quantidades in comportáveis para o sistema judicial, transformando os tribunais em “cobradores de dívidas” em vez de solucionadores de conflitos⁶.

2.4 O processo de injunção

Acompanhando a instauração do Balcão Nacional de Injunções em 2008, foi definido um novo conceito jurídico denominado de **Procedimento ou Processo de Injunção**. Grosso modo, o processo de injunção permite a emissão por uma secretaria judicial, de um título executivo que serve de base à instauração de um processo de cobrança coerciva ou processo de execução.

Não sendo a obrigação voluntariamente cumprida, tem o credor o direito de exigir judicialmente o seu cumprimento, através de penhora e venda de património do devedor. Ora, para a instauração desse processo de execução de património, o credor necessita de um título judicial,

⁶Estas ações judiciais, já se disse, implicavam a realização de julgamentos e sentenças em massa, ocupando recursos judiciais que naturalmente têm uma natureza escassa e cara ao erário público.

quando criado por um tribunal, ou extrajudicial, se não tem origem numa decisão de um tribunal. O processo de injunção permite então a criação desse título executivo, sem necessidade de intervenção de um juiz.

Assim, em geral, sendo o credor possuidor de uma ou mais faturas e não sendo as mesmas pagas dentro do prazo, pode iniciar este processo de injunção. Não havendo oposição da parte do devedor (negando a dívida, por exemplo), a secretaria do Balcão Nacional de Injunções limita-se a conferir força executória ao requerimento de injunção através da aposição de um selo. Esta força executória é assim conferida através da participação de uma secretaria judicial, sem necessidade de marcação de julgamento e sem necessidade de elaboração de uma sentença por um juiz.

Assim, estamos perante um contencioso simples para processar um elevado número de devedores. Veja-se os casos da cobrança de faturas de serviços públicos essenciais, como água, gás, eletricidade ou telecomunicações.

Em suma, **a celeridade é o principal motivo de utilização do processo de injunção, a par de menores custos com taxas de justiça.**

A celeridade deste processo pode ser verificada através dos dados presentes no *Website* do Citius onde se pode ler que em 2011 o tempo médio para a resolução de injunções era de menos de 3 meses⁷. Mais tarde, em 2018, segundo um estudo realizado pela Euler Hermes[5], Portugal era o oitavo país do mundo onde o processo de recuperação de dívidas, por parte de empresas, era menos complexo. Como se pode ver nas Figuras 2.1 e 2.2, numa escala de 1 a 4 onde 1 representa menor complexidade e 4 maior complexidade, no que diz respeito aos custos e relação judicial no processo de cobrança de dívida, Portugal apresenta-se com o nível 2 em ambos os casos. Este mesmo estudo também aponta que a ajudar a menor complexidade está o facto de Portugal ter contratos de retenção de título onde os devedores podem garantir a propriedade dos bens até o credor pagar a totalidade do valor previamente acordado.

2.4.1 Modalidades de instauração do procedimento de injunção

Como já se afirmou, foi a necessidade de enquadrar a cobrança de dívidas em massa num sistema judicial moderno, padronizado e célere, que levou à criação do procedimento de injunção.

Ora, mesmo no domínio das cobranças de obrigações pecuniárias, rapidamente se verificaram dois tipos de credores: os credores ordinários, com algumas faturas vencidas, e os grandes credores, que emitiam e pretendiam cobrar judicialmente faturas em massa. O procedimento de injunção tinha de ser adaptado a estes dois tipos de credores.

Assim, adaptou-se a plataforma Citius, através da possibilidade de instauração de procedimentos de injunção de forma individual, através do preenchimento individualizado de um formulário, e em massa, através da possibilidade de instauração em lote de um número indeterminado de injunções.

⁷<https://www.citius.mj.pt/portal/article.aspx?ArticleId=1531> - Publicado a 13/12/2019

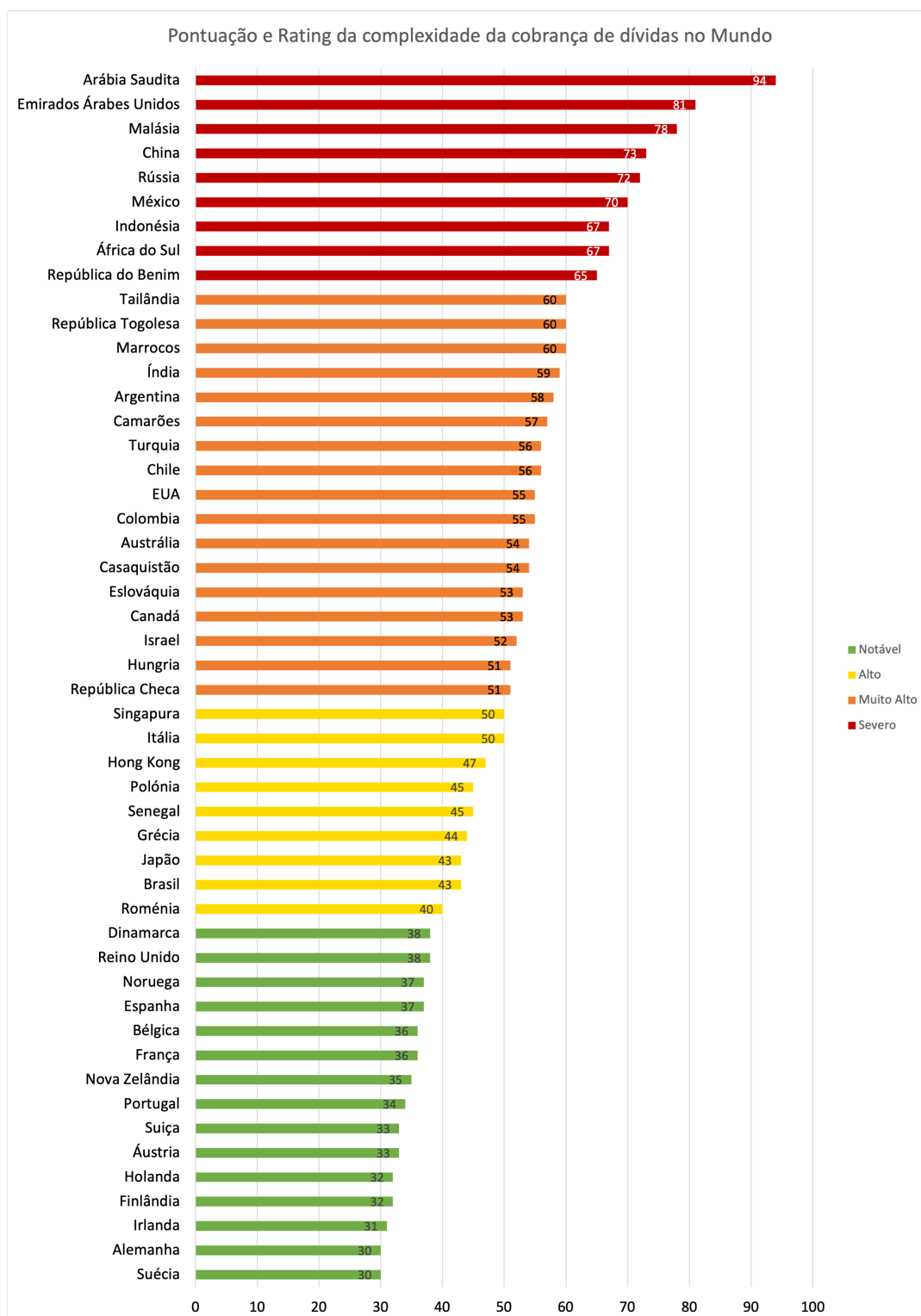


Figura 2.1: Gráfico comparativo da pontuação e *rating* no processo de cobrança de dívida entre vários países do mundo (Fonte: Estudo da Euler Hermes[5])

Complexidade de várias etapas no processo de cobrança de dívida no Mundo			
	Complexidade do processo de pagamento	Complexidade dos processos judiciais	Complexidade do processo de insolvência
Suécia	1	1	1
Alemanha	1	1	2
Irlanda	2	1	1
Finlândia	1	2	1
Holanda	1	1	2
Áustria	1	1	2
Suiça	2	1	2
Portugal	2	2	1
Nova Zelândia	3	2	1
França	1	2	2
Bélgica	2	1	2
Espanha	3	2	1
Noruega	1	2	2
Reino Unido	2	1	2
Dinamarca	1	2	2
Roménia	1	2	2
Brasil	3	2	2
Japão	1	3	2
Grécia	3	2	2
Senegal	4	2	2
Polónia	1	1	4
Hong Kong	4	2	2
Itália	2	3	2
Singapura	4	3	2
República Checa	1	3	3
Hungria	1	2	3
Israel	2	2	4
Canadá	3	2	3
Eslováquia	1	3	3
Casaquistão	4	3	2
Austrália	4	4	2
Colômbia	3	3	3
EUA	4	3	3
Chile	3	3	3
Turquia	4	4	2
Camarões	4	2	4
Argentina	4	3	3
Índia	4	4	2
Marrocos	4	3	3
República Togolesa	3	3	4
Tailândia	3	4	2
República do Benim	3	2	4
África do Sul	4	4	3
Indonésia	4	4	2
México	4	4	4
Rússia	4	4	4
China	4	4	4
Malásia	3	4	4
Emirados Árabes Unidos	3	4	4
Arábia Saudita	4	4	4

Figura 2.2: Tabela comparativa da complexidade de diferentes etapas no processo de cobrança de dívida entre vários países do mundo (Fonte: Estudo da Euler Hermes[5])



(a) Abertura de um novo processo de injunção



(b) Ecrã com mensagem sobre a informação de pagamento depois de terminado o preenchimento do formulário

Figura 2.3: Primeiro passo do formulário do Citius para a submissão de um processo de Injunção Unitária (Fonte: <https://www.citius.mj.pt/portal/default.aspx>)

2.4.2 Injunção unitária

O procedimento de injunção unitária permite a instauração de um, e apenas um, procedimento de injunção. Esta injunção é submetida no Citius de forma totalmente manual através de um formulário extenso, cuja interface não é responsiva (não se adapta aos diferentes tamanhos de ecrãs) nem a utilizadores com necessidades especiais.

A título de exemplo, se um credor possuir múltiplos processos com o mesmo devedor, se proceder à submissão de uma injunção unitária por cada um desses processos, irá ter de preencher múltiplas vezes os seus dados e os dados do credor (que, à partida, serão os mesmos entre os vários processos), para além de ter de preencher de forma manual os dados de cada processo e cada fatura associada. Para além disto, a submissão de injunções unitárias requer que o cálculo dos juros seja feito pelo credor.

As imagens abaixo mostram os diferentes passos que o utilizador tem de realizar aquando da submissão de uma injunção unitária na plataforma Citius.

O primeiro passo (Figura 2.3) marca a abertura de um novo requerimento e subsequente nota a informar que será gerada uma notificação com uma referência para pagamento e uma segunda nota a lembrar que deve ser o utilizador a gravar o progresso de forma manual de forma a não perder os dados já preenchidos.

(a) Formulário principal para o preenchimento dos dados referentes à liquidação e factos

(b) Modal para preenchimento dos dados referentes a cada fatura, incluindo valor das taxas de juro aplicadas à mesma

Figura 2.4: Segundo passo do formulário do Citius para a submissão de um processo de Injunção Unitária (Fonte: <https://www.citius.mj.pt/portal/default.aspx>)

No passo seguinte (Figura 2.4) efetua o preenchimento dos dados, em particular, os dados do processo referentes ao capital em dívida, juros, dados do contrato que está na origem da dívida, detalhes de cada fatura associada ao processo, escolha do tribunal que será responsável pelo processo e a argumentação formal do processo (fundamentação do pedido e da causa de pedir). Neste passo, destaca-se o facto de a interface abrir múltiplos modais que não se adaptam a ecrãs mais pequenos e a caixas de seleção de opções (por exemplo, a seleção do tribunal) que contém dezenas de opções e não possuem qualquer tipo de pesquisa para agilizar o preenchimento do formulário, como se pode ver na Figura 2.5.

No terceiro passo (Figura 2.6), procede-se ao preenchimento dos dados de identificação do requerente (denominado credor ao longo desta dissertação).

À semelhança do passo anterior, no quarto passo (Figura 2.7) procede-se ao preenchimento dos dados de identificação do requerido (denominado de devedor ao longo desta dissertação). Para além disto, escolhe-se também quem será o autor da notificação gerada no final: o BNI, o mandatário do credor ou um agente de execução.

(a) Campo de seleção do tipo de contrato, sem qualquer tipo de pesquisa

(b) Campo de seleção do tribunal, sem qualquer tipo de pesquisa

Figura 2.5: Campos de seleção sem pesquisa no segundo passo do formulário do Citius sem pesquisa para melhorar a usabilidade (Fonte: <https://www.citius.mj.pt/portal/default.aspx>)

Citius | Entrada | Citius | Util | Portal Citius | Certidões Eletrónicas - sem novas atualizações | Não tem peças por subscrever | Não tem notificações eletrónicas novas | Bem-vindo(a) Tiago Lopes de Azevedo | Último login: 04-06-2021 14:38 | Perguntas mais frequentes | Ajuda | Sair | 19:54:38 6 de junho de 2021

Requerimento de Injunção nº: 30090535 | Voltar à Lista de Requerimentos de Injunção

Requerimento
Liquidação e Factos
Requerente
Novo Requerente
Requerido
Novo Requerido
Notificação Requerido(s)
Intervenientes
Intervenientes Registrados
Mandatário
Morada de Notificação
Documento
Pré-visualizar Requerimento
Enviar Requerimento

Ficha do Interviente

Identificação do interveniente

Tipo de Interviente: Requerente

NIF/NIPC: ☒ Nacional ☐ Estrangeiro

BI:

Nome / Designação:

Morada

Morada / Sede:

Localidade:

Cód. Postal*: ☒ Nacional ☐ Estrangeiro

Endereço Eletrónico:

A indicação de endereço de correio eletrónico permite-lhe receber por esta via comunicações e notificações, bem como o título executivo electrónico que eventualmente venha a ser formado

Telefone: Fax:

Identificação bancária

IBAN:

Gravar

Figura 2.6: Terceiro passo do formulário do Citius para a submissão de um processo de Injunção Unitária (Fonte: <https://www.citius.mj.pt/portal/default.aspx>)

Citius | Entrada | Citius | Util | Portal Citius | Certidões Eletrónicas - sem novas atualizações | Não tem peças por subscrever | Não tem notificações eletrónicas novas | Bem-vindo(a) Tiago Lopes de Azevedo | Último login: 04-06-2021 14:38 | Perguntas mais frequentes | Ajuda | Sair | 19:55:40 6 de junho de 2021

Requerimento de Injunção nº: 30090535 | Voltar à Lista de Requerimentos de Injunção

Requerimento
Liquidação e Factos
Requerente
Novo Requerente
Requerido
Novo Requerido
Notificação Requerido(s)
Intervenientes
Intervenientes Registrados
Mandatário
Morada de Notificação
Documento
Pré-visualizar Requerimento
Enviar Requerimento

Ficha do Interviente

Identificação do interveniente

Tipo de Interviente: Requerido

NIF/NIPC: ☒ Nacional ☐ Estrangeiro

BI:

Nome / Designação:

Domicílio Convencionado? ☐ Sim ☒ Não

Morada

Morada / Sede:

Localidade:

Cód. Postal*: ☒ Nacional ☐ Estrangeiro

Telefone: Fax:

Gravar

(a) Preenchimento dos dados do devedor

Citius | Entrada | Citius | Util | Portal Citius | Certidões Eletrónicas - sem novas atualizações | Não tem peças por subscrever | Não tem notificações eletrónicas novas | Bem-vindo(a) Tiago Lopes de Azevedo | Último login: 04-06-2021 14:38 | Perguntas mais frequentes | Ajuda | Sair | 19:56:03 6 de junho de 2021

Requerimento de Injunção nº: 30090535 | Voltar à Lista de Requerimentos de Injunção

Requerimento
Liquidação e Factos
Requerente
Novo Requerente
Requerido
Novo Requerido
Notificação Requerido(s)
Intervenientes
Intervenientes Registrados
Mandatário
Morada de Notificação
Documento
Pré-visualizar Requerimento
Enviar Requerimento

Notificação dos Requeridos

Notificação do(s) requerido(s) a efectuar por: **Por favor registre pelo menos um requerido**

☐ Balcão Nacional de Injunções ☐ Mandatário do Requerente ☐ Agente de Execução

Gravar

(b) Seleção da entidade que irá notificar o devedor do processo de injunção

Figura 2.7: Quarto passo do formulário do Citius para a submissão de um processo de Injunção Unitária (Fonte: <https://www.citius.mj.pt/portal/default.aspx>)

(a) Resumo dos devedores associados ao processo de injunção a ser aberto

(b) Seleção da morada do mandatário para envio da notificação de abertura do processo

Figura 2.8: Quinto e sexto passos do formulário do Citius para a submissão de um processo de Injunção Unitária (Fonte: <https://www.citius.mj.pt/portal/default.aspx>)

Nos passos seguintes (Figura 2.8) é apresentada a lista de intervenientes registados no processo e é escolhida a morada para onde será enviada a notificação para pagamento da dívida.

Por fim, no último passo (Figura 2.9) é apresentado o documento com a notificação a ser enviada ou uma lista de erros no preenchimento dos passos anteriores que impossibilitam a apresentação da injunção.

2.4.3 Injunções em lote

A instauração de injunções em lote permite o envio, de uma só vez, de um número indeterminado de faturas, em relação a um número indeterminado de devedores. Este envio em grupo possibilita a junção e o pagamento único das taxas de justiça de todas as injunções.

Figura 2.9: Último passo do formulário do Citius para a submissão de um processo de Injunção Unitária (Fonte: <https://www.citius.mj.pt/portal/default.aspx>)



Figura 2.10: Formulário de submissão do ficheiro XML na plataforma Citius (Fonte: <https://www.citius.mj.pt/portal/default.aspx>)

Operativamente, evita igualmente que se insira repetidamente os dados do credor, do devedor (se estiverem em causa diversas faturas), se faça o cálculo manual dos juros de mora e de outras quantias em dívida e se fundamente o pedido, igualmente de forma automática.

A submissão de um lote de injunções é feita através de um ficheiro XML (*Extensible Markup Language*) que tem que ser gerado pelo credor de forma manual ou através de *software* que produza este ficheiro com base numa base de dados.

Na Figura 2.10 pode ver-se que a submissão deste ficheiro consiste no carregamento e validação no Citius através de um XSD (*XML Schema Definition*), criando os processos de injunção internamente.

À data da elaboração desta dissertação, o Citius encontra-se na **versão 1.5** do ficheiro XSD, que pode ser encontrado no Anexo A.

2.5 Revisão da tecnologia

Desde o aparecimento ao público em 1991⁸ da *World Wide Web*, o desenvolvimento de aplicações *Web* tem evoluído rapidamente. Nos dias de hoje, desenvolvimento *Web* é mais do que simples páginas de apresentação de conteúdos. Hoje, as aplicações *Web* constituem software tão ou mais complexo que aplicações de *desktop* com a vantagem acrescida de poderem ser acedidas a partir de qualquer dispositivo e a partir de qualquer parte do mundo [3]. Aplicações como redes sociais, sistemas de gestão bancários, lojas online, etc., utilizam inteligência artificial, *machine learning*, mecanismos de automatização inteligentes e muitas outras tecnologias avançadas.

De forma bastante genérica, uma aplicação *Web* divide-se essencialmente em dois tipos de tecnologias: *backend* e *frontend* que comunicam entre si através de um protocolo de comunicação normalmente denominado de API (*Application Programming Interface*).

As tecnologias de *frontend*, camada do cliente, são responsáveis por mostrar ao utilizador final de forma apelativa e acessível o que foi processado no *backend*. Os três pilares desta tecnologias são o HTML (*HyperText Markup Language*), o CSS (*Cascading Style Sheets*) e o JavaScript (JS). A primeira é responsável pela definição da estrutura e do conteúdo do documento *Web*. Já a

⁸<https://www.w3.org/Help/#webinternet> - Consultado a 03/05/2022

segunda, tem como função estilizar o conteúdo da página e redefinir o *layout* de cada página de forma a tornar a aplicação mais agradável, apelativa e acessível ao público alvo. Por fim, o JS responsabiliza-se por processar e agilizar os diferentes passos necessários para o utilizador utilizar a aplicação *Web*.

Nos dias que correm são raras as situações em que utilizar qualquer uma destas três tecnologias na sua forma nativa é mais benéfico adotar *frameworks* ou abstrações destas tecnologias atômicas. Isto deve-se, na generalidade dos casos, a questões de tempo de desenvolvimento, facilidade de desenvolvimento e manutenção, redução da quantidade de código fonte, entre outras. Em particular, no caso do JS a escolha de uma *framework* é ainda mais frequente especialmente quando a aplicação se trata de uma *Single Page Application* (SPA), que se caracteriza por uma experiência de utilizador superior e mais natural já que não existem atualizações de página entre cada pedido e todo o conteúdo é obtido e carregado em segundo plano através de pedidos AJAX (*Asynchronous JavaScript And XML*)⁹. Além do mais, neste tipo de aplicações, o JS é também responsável por fazer a gestão da *cache* e do armazenamento local. Neste contexto existem *frameworks* como *AngularJS*¹⁰, *ReactJS*¹¹, *Meteor*¹² ou *VueJS*¹³.

Por sua vez, do outro lado da balança, encontra-se o *backend* que tem como função suportar a lógica do negócio, fazendo a interação com a camada de persistência de dados e processamento mais pesado da aplicação *Web*. É também este o responsável por preparar os dados e definir a API que é consumida pelo *frontend*. À semelhança do que acontece no *frontend*, atualmente o mais comum é a utilização de *frameworks* para agilizar o desenvolvimento de uma aplicação.

Das inúmeras *frameworks* existentes, há duas que se destacam em popularidade: *Laravel*¹⁴ que tem como base *PHP*¹⁵ e *NodeJS*¹⁶ que se alicerça no *JavaScript*¹⁷. Ambas assentam no modelo de desenvolvimento de software *Model-View-Controller* (MVC). Neste trio, como se pode ver na Figura 2.11, o *Modelo* é a entidade que define a estrutura dos dados, a *Vista* é a que apresenta os dados formatados ao utilizador final, permitindo que este interaja com os mesmos, e, por fim, o *Controlador* é a ponte entre os dois primeiros e garante que os pedidos feitos pelo utilizador são válidos, requer os dados ou a alteração dos dados ao Modelo, remetendo ao utilizador os dados devidamente formatados e/ou atualizados.

Como mencionado atrás, uma aplicação *Web* possui algum tipo de persistência. A escolha do tipo de persistência é, em grande parte, dos casos mais importante que a escolha das tecnologias de *frontend* ou *backend*, uma vez que a escolha desta irá influenciar a velocidade de leitura dos dados, velocidade de manipulação de dados, velocidade de pesquisa, escalabilidade da aplicação, facilidade de implementação do *backend*, entre outras.

⁹https://www.w3schools.com/Xml/ajax_xmlhttprequest_send.asp - Consultado a 21/08/2022

¹⁰<https://angularjs.org/>

¹¹<https://reactjs.org/>

¹²<https://www.meteor.com/>

¹³<https://vuejs.org/>

¹⁴<https://laravel.com/>

¹⁵<https://www.php.net/>

¹⁶<https://nodejs.org/en/>

¹⁷<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

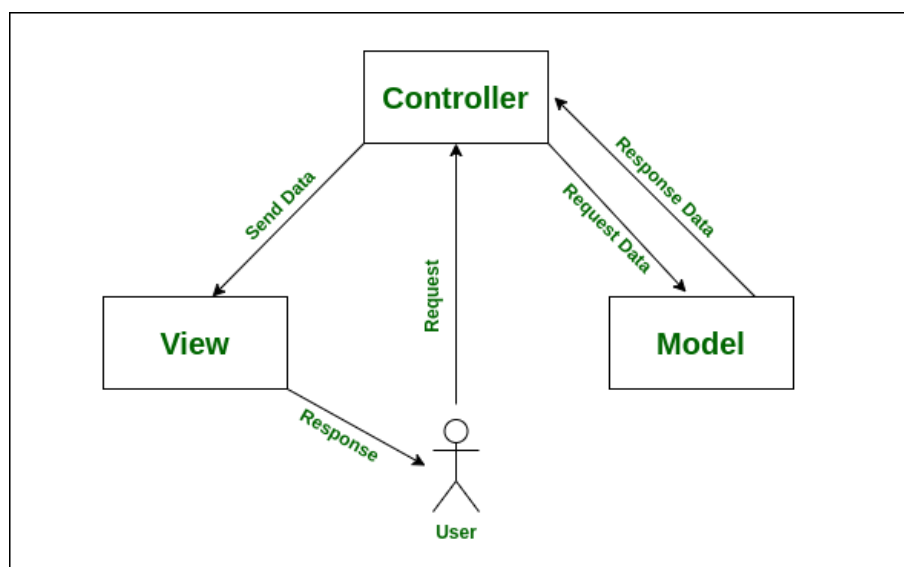


Figura 2.11: Representação gráfica do modelo MVC (Fonte: <https://www.geeksforgeeks.org/benefit-of-using-mvc/>)

Enquanto existem múltiplos tipos de persistência, é comum diferenciar dois tipos em particular: relacional e não relacional. A primeira caracteriza-se por ser um tipo de persistência estruturada, em que, como o próprio nome indica, os diferentes tipos de dados estão relacionados entre si assegurando, de forma fácil, a integridade dos dados, mesmo que por vezes a manipulação e leitura de dados não seja tão veloz quanto uma persistência não relacional. Bases de dados relacionais como *PostgreSQL*¹⁸, *MySQL*¹⁹ e *MariaDB*²⁰ são talvez as mais comuns, principalmente para aplicações com um volume de dados baixo ou média, uma vez que nestes casos a menor velocidade no processamento dos dados não é tão significativa e os princípios de persistência de dados ACID (Atomicidade, Consistência, Isolamento e Durabilidade) não são tão relevantes.

Por outro lado, bases de dados não relacionais como *MongoDB*²¹, *ElasticSearch*²², *Redis*²³ ou *DynamoDB*²⁴ são velozes no que diz respeito à manipulação e leitura de dados uma vez que fazem uso de outros métodos para a gestão de dados, nomeadamente, persistência de dados em memória ao invés de armazenamento em disco, indexação dos dados de forma a permitir réplicas de dados e leituras mais rápidas. Por estes motivos, este tipo de persistências são populares nas aplicações com grandes quantidades de dados, a pesquisa rápida de dados é a prioridade da aplicação e/ou a integridade dos dados não é relevante, entre outras.

De realçar que uma aplicação *Web* não tem necessariamente que usar apenas um tipo de persistência. De facto, as aplicações modernas combinam diferentes tipos de persistências para atingir

¹⁸<https://www.postgresql.org/>

¹⁹<https://dev.mysql.com/>

²⁰<https://mariadb.org/>

²¹<https://www.mongodb.com/>

²²<https://www.elastic.co/>

²³<https://redis.io/>

²⁴<https://aws.amazon.com/pt/dynamodb/>

diferentes objetivos. A título de exemplo, se analisarmos uma loja *online*, é normal esta usar pelo menos três tipos de persistências: uma base de dados relacional para guardar de forma estruturada os seus clientes, encomendas, produtos, categorias, etc.; uma base de dados em memória (*cache*), por exemplo, *Redis* ou *Memcached*²⁵ para guardar em memória o diferente conteúdo das páginas (ou até mesmo as páginas completas) de forma a que o carregamento das mesmas seja muito mais rápido; uma base de dados como *ElasticSearch* para a criação de motores de busca robustos e rápidos (nomeadamente, pesquisa de produtos).

Contudo, é importante realçar que ao longo dos anos foram encontradas soluções para tentar combater os principais pontos negativos das bases de dados relacionais, a velocidade de acesso aos dados e a escalabilidade. Uma dessas soluções passa pela utilização de sistemas de indexação como uma camada em cima da base de dados relacional, como é o exemplo do *SingleStore*²⁶, permitindo obter o melhor dos dois mundos: integridade dos dados da persistência relacional e velocidade de acesso e manipulação de persistências em memória.

Outro fator que veio a ajudar em muito este tipo de persistências foi a evolução do *hardware*, em particular a utilização de SSD (*Solid State Drive*) ou NVMe (*Non-Volatile Memory Express*) em detrimento de HDD (*Hard Disk Drive*). Versões mais recentes da tecnologia SSD conseguem atingir valores de leitura na ordem dos 3500 MB/s enquanto o segundo apenas atinge 150 MB/s. Isto, em conjunto com infraestruturas de servidores mais potentes e avançadas, permite que cada aplicação *Web* tenha a sua própria receita para o sucesso.

Relativamente a tipos de infraestruturas de servidores, existem dois que se destacam pela sua popularidade: servidores *baremetal* (dedicados) e servidores em *cloud*. O primeiro é mais antigo e mais rudimentar uma vez que se trata de uma máquina ou um conjunto de máquinas onde têm que ser instalados todos os requisitos para uma aplicação *Web* funcionar: sistema operativo, bases de dados, servidores SMTP (*Simple Mail Transfer Protocol*), etc. Este tipo de servidores normalmente estão numa determinada localização geográfica e continuam a ser preferidos por grandes empresas para manterem os dados no mesmo local; dependendo do tamanho das aplicações, os custos podem ser menores quando comparados com servidores em *cloud*. No entanto, a escalabilidade de um sistema torna-se mais complexa e exige mais tempo de configuração e manutenção.

Por outro lado, servidores em *cloud* tornaram-se mais populares nos últimos anos, principalmente porque a manutenção, escalabilidade e redundância dos dados é mais fácil de conseguir e acontece de forma automática sem ser necessário colocar as aplicações em modo de manutenção. Outras vantagens deste tipo de servidores são a maior rapidez na recuperação em caso de falha, monitorização dos diferentes serviços, pagamento apenas dos recursos que se estão a utilizar em cada momento e as diferentes otimizações que acontecem automaticamente em cada serviço que fornecem. Dois dos provedores com maior número de utilizadores são a AWS²⁷ e a Microsoft Azure²⁸.

²⁵<https://memcached.org/>

²⁶<https://www.singlestore.com/>

²⁷<https://aws.amazon.com/>

²⁸<https://azure.microsoft.com/pt-pt/>

2.6 Sumário

Os processos de recuperação de dívidas por via judicial caracterizavam-se por ser morosos e dispendiosos, uma vez que exigiam múltiplas deslocações a tribunais por parte dos credores, devedores e respetivos advogados, realização de julgamentos o que se traduzia em custos avultados.

Face a isto, em 2008 o Ministério da Justiça português procedeu à criação do BNI e do Citius e com estes nasceu o Processo de Injunção que se distingue pela sua simplicidade, celeridade e consequente redução de custos para todos os envolvidos.

Contudo, enquanto *software*, o Citius apresenta as suas limitações nomeadamente no que respeita a usabilidade e acessibilidade, especialmente quando se fala da submissão de uma Injunção Unitária. É certo que, também é possível a submissão de um Lote de Injunções, mas para tal é necessário que o credor (ou o seu representante) possua um ficheiro XML devidamente estruturado com todos os processos, o que tipicamente é uma limitação para estes, uma vez que no mercado não existem ferramentas que permitam a geração deste ficheiro.

Capítulo 3

Identificação do problema e definição dos requisitos

3.1 Introdução

Tal como foi explorado no Capítulo 2, os processos de cobrança de dívida são tradicionalmente dispendiosos e demorados, daí a criação do BNI e da plataforma Citius. Contudo, esta solução tem ainda falhas visíveis no que diz respeito à forma como os dados das injunções são inseridos e processados, uma vez que, como visto na Secção 2.4.2, todos os dados devem ser preenchidos manualmente. A título de exemplo, se um devedor possuir duas dívidas com o mesmo credor, as informações de ambos têm que ser preenchidas duas vezes (uma para cada dívida). A outra hipótese seria exportar um lote de injunções num ficheiro XML com um formato específico para ser aceite pelo Citius. No entanto, não existe nenhum serviço que permita gerar este XML de forma simples, tendo em consideração que usualmente quem é responsável pelo preenchimento e processamento destes processos, pouco ou nenhum conhecimento técnico tem sobre processamento e tratamento automático de dados.

Por exemplo, um provedor de serviços de telecomunicações em Portugal detém uma carteira de milhões de clientes e consequentemente milhares de processos de dívida para resolver, o que requer equipas de pessoas a tratar destes processos traduzindo-se em muito tempo e consequentemente custos avultados. Claro está, que empresas com este volume de negócios (seguradoras, provedores de serviços essenciais, etc) normalmente possuem *software* próprio, desenhado à medida das suas necessidades. No entanto, o volume de dívidas de pequenas e médias empresas não justifica este tipo de *software* desenhado à medida, fazendo com que se cinjam ao processo manual.

Estes processos possuem lacunas claras em termos de eficiência e custos que podem ser melhorados pelo uso de serviços digitais que automatizam tarefas repetitivas, retirando da equação o

erro humano nessas mesmas tarefas, por exemplo, cálculo de juros e emissão de notificações para pagamento de uma dívida.

Com base nesta premissa, este trabalho propõe um serviço *Web*, de subscrição responsável por automatizar parte dos processos legais que se iniciam com a contração de uma dívida por parte de um devedor até que seja paga ou que seja iniciado um processo de injunção junto do BNI através do Citius.

Ademais, o *Minimum Value Product* (MVP) resultante deste trabalho assume-se como ponto de partida de uma nova sociedade comercial que numa fase inicial direcionará o seu serviço para pequenas e médias empresas que não detém o capital financeiro para suportar equipas de pessoas ou *software* desenhado à medida para processos de recuperação de dívida, como acontece em empresas com milhões de clientes e milhares de processos de dívidas (por exemplo Seguradoras ou empresas de telecomunicações).

3.2 Especificação dos requisitos

Na secção anterior mencionou-se que o principal problema tem que ver com a falta de software de acesso público (serviço ou subscrição) que seja capaz de automatizar parte do processo de preenchimento de dados de injunções, emissão de avisos e acima de tudo, criação dos ficheiros XML de injunções em lote suportados pelo Citius.

Para explorar mais o problema e em simultâneo definir os requisitos do sistema proposto nesta dissertação, abaixo serão descritos os atores do sistema e as *User stories* para cada um deles.

3.2.1 Atores

Atores, em Engenharia de Software, definem papéis interpretados por pessoas ou outras entidades que interagem diretamente com o sistema¹. No caso de sistemas de informação, esta interação normalmente traduz-se por inserção, leitura, edição ou remoção de dados.

Assim sendo, foram identificados três atores principais, que se encontram representados na Figura 3.1: **Credor, Administrador e Balcão Nacional de Injunções**. Por sua vez, na Tabela 3.1 pode-se ler com mais detalhe o que representa cada ator.

Os atores acima descritos são cruciais para o funcionamento do sistema proposto nos capítulos seguintes, logo serão explorados em mais detalhe. No entanto, esta mesma proposta permite a expansão para outros atores e/ou desdobramentos dos atores em sub-atores. A título de exemplo, o diagrama da Figura 3.2 e a Tabela 3.2 poderão representar o futuro do sistema proposto, onde o ator Credor é desdobrado em três sub-atores: Administrador do Credor, Funcionário do Credor, Advogado do Credor. A diferença entre estes sub-atores poderá ser simplesmente uma diferenciação do tipo de acesso aos detalhes dos processos ou funcionalidades completamente distintas.

¹<https://www.uml-diagrams.org/use-case-actor.html> - Consultado a 13/07/2022

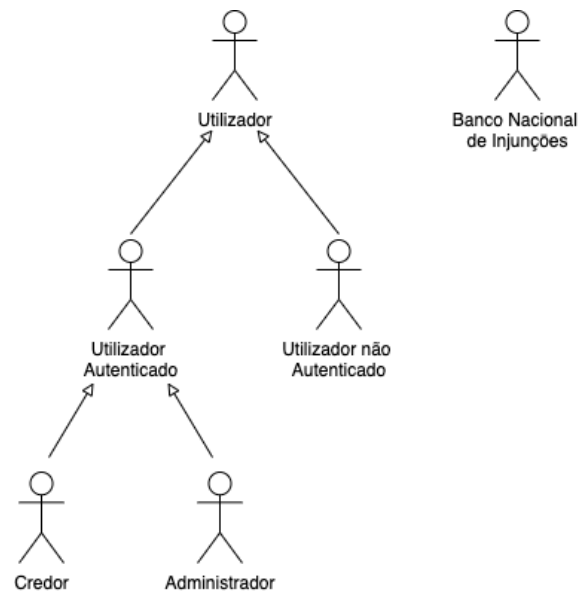


Figura 3.1: Diagrama de atores da aplicação

Tabela 3.1: Descrição e exemplos para cada ator do sistema

Id	Descrição	Exemplos
Credor	entidade individual ou coletiva que presta um determinado serviço ou comercializa um produto e emite uma ou mais faturas sobre o mesmo	MEO, Allianz, EDP, Hospitais, etc.
Administrador	responsável pela manutenção geral do sistema e com todas as permissões necessárias para a administração de outros atores, tais como, os credores; responsável também pela administração da subscrição ao serviço por parte dos credores.	admin
Balcão Nacional de Injunções	Secretaria do Ministério da Justiça responsável por processar e dar seguimento judicial aos processos de injunção emitidos a nível nacional.	BNI

Tabela 3.2: Descrição e exemplos para cada ator e sub-ator do sistema

Id	Sub-ator	Descrição	Exemplos
Credor	Administrador do Credor	Administrador de uma empresa provedora de serviços ou vendedora de um produto que emite uma ou mais faturas sobre os mesmos.	Administrador(a) de um Hospital
	Funcionário do Credor	Funcionário responsável pela gestão de um certo tipo/conjunto de injunções de uma empresa.	Secretário(a) de um Hospital
	Advogado do Credor	Advogado de uma empresa responsável por seguir os processos de injunção e gerir o conteúdo das notificações emitidas aos devedores.	Advogado contratado por um Hospital
Administrador		Responsável pela manutenção geral do sistema e com todas as permissões necessárias para a administração de outros atores, tais como, os credores. Responsável também pela administração de da subscrição ao serviço por parte dos credores.	admin
Balcão Nacional de Injunções		Secretaria do Ministério da Justiça responsável por processar e dar seguimento judicial aos processos de injunção emitidos a nível nacional.	n/a

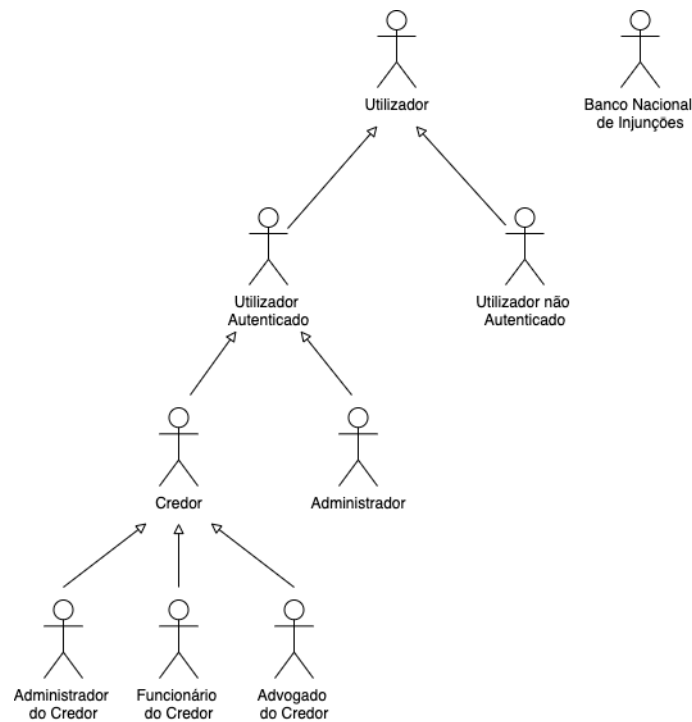


Figura 3.2: Diagrama de atores e sub-atores da aplicação

3.2.2 User stories

User stories são uma metodologia popular em Engenharia de Software no que diz respeito ao levantamento e representação de requisitos[1] e são frequentemente escritas seguindo o modelo “*Como (papel), eu quero (objetivo), de forma a (benefício)*”. A sua utilização foi sendo popularizada e utilizada cada vez mais[6], em particular, nas metodologias de desenvolvimento ágil[9] em Engenharia de Software.

No contexto dos processos de injunção, o ator *Credor* apresenta-se como o ator principal, uma vez que, primeiro, é o mais interessado em utilizar o mesmo pois este visa facilitar o processo de recuperação de dívida e, em segundo lugar, é o que tem mais interações com a plataforma, o que se traduz em mais e mais importantes *User stories*.

Nas Tabelas 3.3, 3.4, 3.5 e 3.6 encontram-se descritas todas as *User stories* identificadas para os quatro atores, sendo que cada *User story* possui um identificador único, um nome para mais fácil identificação, uma prioridade que pode ser baixa, média ou alta e uma descrição que representa a *User story* em si e segue o modelo acima referido.

Tabela 3.3: *User stories* do ator Não Autenticado

Id	Nome	Prio.	Descrição
US001	Autenticação	Alta	Como utilizador Não Autenticado, preciso de me autenticar para poder gerir os meus processos de injunção ou gerir os utilizadores da aplicação.
US002	Recuperar palavra-passe	Alta	Como utilizador Não Autenticado, preciso de recuperar a minha palavra-passe de forma a poder alterá-la em caso de esquecimento.

Tabela 3.4: *User stories* do ator Autenticado

Id	Nome	Prio.	Descrição
US101	Autenticação de 2 fatores	Baixa	Como ator Autenticado, quero ser capaz de ativar a autenticação de dois fatores de forma a aumentar a proteção dos meus dados.
US102	Alterar a palavra-passe	Baixa	Como utilizador Não Autenticado, quero ser capaz de alterar a minha palavra-passe de forma a poder aumentar a segurança da minha conta.

Tabela 3.5: *User stories* do Credor

Id	Nome	Prio.	Descrição
US201	Abertura de Injunção	Alta	Como credor, quero começar um novo processo de injunção.
US202	Adicionar fatura	Alta	Como credor, quero adicionar uma ou mais faturas a um processo de injunção.
US203	Adicionar Devedor	Alta	Como credor, quero adicionar os dados de um novo devedor de forma a poder associá-lo a um processo de injunção.
US204	Adicionar Advogado	Alta	Como credor, quero adicionar os dados de um novo advogado de forma a poder associá-lo a um processo de injunção.
US205	Editar fatura	Alta	Como credor, quero alterar os dados de uma fatura.
US206	Editar Devedor	Alta	Como credor, quero alterar os dados de um devedor de forma a corrigir dados errados ou incompletos.

Continua na página seguinte...

Tabela 3.5 – Continuação da página anterior

Id	Nome	Prio.	Descrição
US207	Editar Advogado	Alta	Como credor, quero alterar os dados de um advogado de forma a corrigir dados errados ou incompletos.
US208	Remover fatura	Média	Como credor, quero apagar uma fatura associada a uma injunção.
US209	Remover Devedor	Média	Como credor, quero apagar um devedor e todas as injunções associadas.
US210	Remover Advogado	Média	Como credor, quero apagar um advogado e remover a associação a todos os processos de injunção.
US211	Alterar número de avisos	Média	Como credor, quero definir quantos avisos devem ser enviados por processo.
US212	Alterar intervalo entre avisos	Média	Como credor, quero definir o intervalo entre avisos.
US213	Criar modelo de aviso	Baixa	Como credor, quero criar um ou mais modelos de avisos para depois serem utilizados nos processos de injunção.
US214	Geração automática de avisos	Alta	Como credor, quero que os avisos sejam automaticamente gerados para todos os processos de injunção.
US215	Exportar injunções em lote	Alta	Como credor, quero exportar lotes de injunções em formato XML para depois ser importado na plataforma Citius.
US216	Terminar processo de injunção	Média	Como credor, quero terminar um ou mais processos de injunção
US217	Montante liquidado	Média	Como credor, quero ver qual o montante em dívida recuperado, por processo ou a nível global.
US218	Definir plano de pagamento	Baixa	Como credor, quero poder definir planos de pagamento de capital em dívida para facilitar o processo de recuperação da dívida.
US219	Definir métodos de pagamento	Média	Como credor, quero poder configurar métodos de pagamento para depois serem utilizados nos avisos de pagamento.
US220	Receber notificações	Baixa	Como credor, quero receber notificações via email de processos com avisos a chegar ao limite de pagamento para mais facilmente dar continuidade ao processo.

Continua na página seguinte...

Tabela 3.5 – Continuação da página anterior

Id	Nome	Prio.	Descrição
US221	Caixa de entrada interna	Baixa	Como credor, quero que uma caixa de entrada interna seja criada de forma a receber as atualizações dos processos de injunção emitidas pelo BNI.

Tabela 3.6: *User stories* do Administrador

Id	Nome	Prio.	Descrição
US301	Criação de Credor	Média	Como administrador, quero criar um novo credor para depois o mesmo ter acesso ao sistema.
US302	Bloqueio de Credor	Baixa	Como administrador, quero bloquear o acesso do credor à plataforma.
US303	Emissão de faturas	Baixa	Como administrador, quero emitir de forma manual ou automática faturas aquando o pagamento da subscrição por parte do credor.
US304	Bloquear meios de pagamento	Baixa	Como administrador, quero poder bloquear meios de pagamento de forma a não serem utilizados pelos credores na criação de injunções.

3.2.3 Requisitos técnicos

À semelhança dos pontos anteriores, os requisitos técnicos permitem-nos definir uma determinada solução, em particular, definem que aspetos técnicos que a plataforma deve garantir, tais como: acessibilidade, proteção de dados, disponibilidade, entre outros.

Na Tabela 3.7 encontram-se os requisitos técnicos da solução proposta tendo como principal alicerces o facto de ser um serviço *Web* de subscrição e que gere dados pessoais de diversas entidades, que devem ser protegidos ao abrigo do Regulamento Geral da Proteção de Dados (RGPD)².

3.2.4 Requisitos do sistema

Enquanto que as *User stories* e os Requisitos Técnicos definidos e descritos nas subsecções 3.2.2 e 3.2.3 definem a solução aqui proposta, existe ainda um requisito de sistema, definido

²REGULAMENTO (UE) 2016/ 679 DO PARLAMENTO EUROPEU E DO CONSELHO relativo à proteção das pessoas singulares no que diz respeito ao tratamento de dados pessoais e à livre circulação desses dados - <https://eur-lex.europa.eu/legal-content/PT/TXT/?uri=celex%3A32016R0679> - Publicado a 27/04/2016

Tabela 3.7: Requisitos técnicos da aplicação

Id	Nome	Descrição
TR001	Disponibilidade	A plataforma deve estar disponível pelo menos 99% do tempo por ano.
TR002	RGPD	A plataforma deve cumprir com os requisitos legais impostos pelo Regulamento Geral de Proteção de Dados.
TR003	Usabilidade	A plataforma deve ser fácil de usar sendo por isso obrigatório uma interface gráfica simples e completa.
TR004	Acessibilidade	A plataforma deve ser utilizada na sua plenitude e de forma fácil por qualquer utilizador, incluindo utilizadores com algum tipo de incapacidade.
TR005	Redundância	A infraestrutura da plataforma deve garantir que toda a persistência (base de dados, ficheiros, etc) apresenta algum tipo de redundância de forma a ser possível a recuperação dos dados em caso de falha, por exemplo, armazenamento de ficheiros e base de dados em sistema RAID (<i>Redundant Array of Independent Disks</i>).
TR006	Segurança dos dados	A plataforma deve garantir que um credor não consiga aceder a processos de outro credor.
TR007	Métodos de pagamento	A plataforma deve integrar com métodos de pagamento, por exemplo, Ifthenpay (https://ifthenpay.com/), Paypal (https://www.paypal.com/pt/home), Stripe (https://stripe.com/en-pt), etc.

na Tabela 3.8, que completa ambos e cuja prioridade é baixa e a sua implementação dependerá do *feedback* recebido pelos utilizadores aquando a validação do MVP.

O requisito SR001 pretende que a cada credor/cliente da aplicação *Web* a ser desenvolvida neste trabalho, de nome Fluxio, no momento do registo na plataforma, seja atribuído um endereço de email único que será utilizado no momento de criação de injunções em lote. Desta forma, todas as vezes que o Citius emitir uma notificação (para esse mesmo email) com uma atualização do estado de um processo, a plataforma pode ler essa mesma notificação automaticamente e atualizar o sistema sem ser necessária a intervenção por parte do credor.

3.3 Modelo do domínio

De forma a cumprir com os objetivos acima propostos foi definido o modelo de domínio, que se encontra na Figura 3.3. Neste modelo destaca-se a entidade *Process* como entidade central do

Tabela 3.8: Requisitos de sistema da aplicação

Id	Nome	Prio.	Descrição
SR001	Atualização de processo de injunção	Baixa	Como BNI, quero emitir atualizações dos processos de injunção para uma caixa de email interna da aplicação.

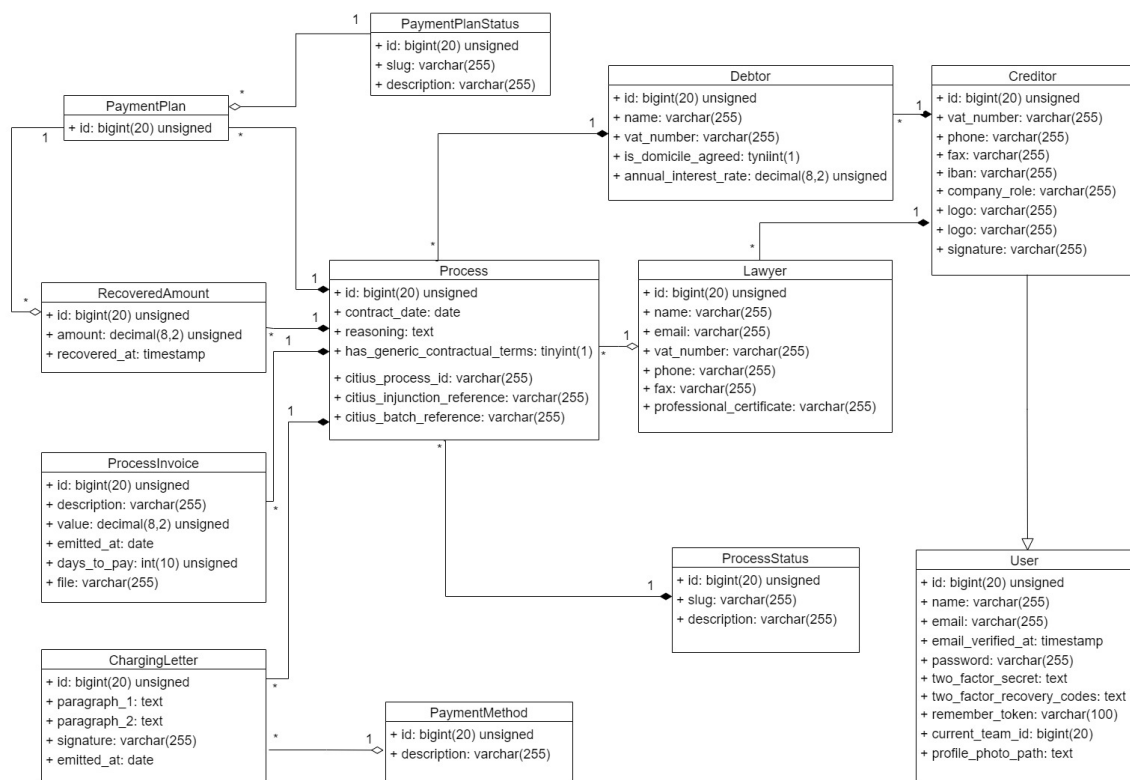


Figura 3.3: Modelo de domínio da aplicação Fluxio

modelo, uma vez que esta representa a Injunção e relaciona-se direta ou indiretamente com as restantes entidades.

Ao analisar o formulário de submissão de uma injunção unitária através do Citius (Secção 2.4.2) juntamente com o modelo de domínio proposto, pode-se concluir que o conjunto das entidades *Process*, *ProcessInvoice*, *Debtor* e *Creditor* contém grande parte da informação necessária no que diz respeito ao processo de injunção.

As restantes entidades aqui apresentadas têm como principal objetivo permitir implementar requisitos secundários do sistema, tais como: obter dados estatísticos do capital em dívida recuperado, criação automática ou semi-automática de notificações escritas aos devedores e gestão do estado de múltiplos processos.

Para além das entidades do domínio irão fazer parte do sistema outras entidades de suporte, para uma implementação mais fácil, mais escalável e mais eficiente, tais como *Court*, *Address*, entre outras.

3.4 Sumário

Existe uma lacuna no panorama de *software* como serviço para gestão de processos de injunção no mercado Português. Grandes empresas, com milhões de clientes e milhares de processos de

dívida, optam pela criação de software privado e customizado, o que não é comportável financeiramente para pequenas e médias empresas.

Com o objetivo que colmatar esta mesma lacuna (ou pelo menos parte da mesma), foi feito o levantamento de requisitos para a criação de uma aplicação que visa funcionar como um serviço, talvez à base de subscrição.

Como parte desse levantamento, conclui-se que a funcionalidade fulcral da aplicação é a geração de injunções em lote num formato XML para ser depois importada na plataforma do Ministério da Justiça Português, Citius. De forma a poder concretizar esta funcionalidade é necessário que a aplicação seja também uma plataforma de gestão de processos de injunção.

Para a primeira iteração do MVP, foram identificados três Atores, foram definidas *User stories* para cada um deles, assim como requisitos de sistema e requisitos técnicos. Mais importante ainda, foi estruturado um modelo de domínio capaz de identificar e caracterizar as principais entidades do sistema e como elas se relacionam.

Capítulo 4

Fluxio

4.1 Introdução

Como referido nos capítulos anteriores, o processo de injunção caracteriza-se pela sua simplicidade e celeridade. Como tal, como pode ser visto no fluxograma da Figura 4.1, o processo de injunção pressupõem a existência de um crédito relacionado com um bem ou serviço, que não foi pago.

Face à existência de uma ou mais dívidas, e numa primeira instância, é emitido o primeiro aviso de pagamento, que deve ser feito via carta registada para ter validade jurídica. Esta primeira notificação consiste num aviso emitido pelo credor com o devedor como destinatário, com os dados: valor original da dívida e respetiva fundamentação, data da contração da dívida, data da emissão do presente aviso, juros de mora calculados entre a data da contração da dívida até à data de emissão do aviso, valor total em dívida, métodos de pagamento disponíveis e data limite para pagamento do valor total, contada a partir da data de emissão deste aviso.

Após a receção da primeira notificação, pode ocorrer o seguinte, alternativamente: (i) o devedor pode proceder ao pagamento do valor total presente no aviso, dando-se como terminado o processo por pagamento; (ii) o devedor não procede ao pagamento dentro do prazo indicado. No segundo cenário, procede-se à emissão de um segundo aviso em tudo semelhante ao primeiro, nomeadamente, é dado um novo prazo a partir da emissão deste aviso e os juros são atualizados uma vez que o devedor passa a pagar juros desde a data de contração da dívida até à data da emissão do segundo aviso. Neste caso, pode haver a soma de um valor relativo a custos administrativos.

Mais uma vez, à semelhança do primeiro aviso, caso o devedor proceda ao pagamento do valor em causa dentro do prazo atribuído, o processo de cobrança de dívida termina, sem nunca se ter tornado numa injunção, uma vez que nunca deu entrada no tribunal. Caso o devedor não o faça, dá-se início a uma ação judicial tradicional, procedendo-se à submissão da injunção através do portal Citius, com destino ao BNI.

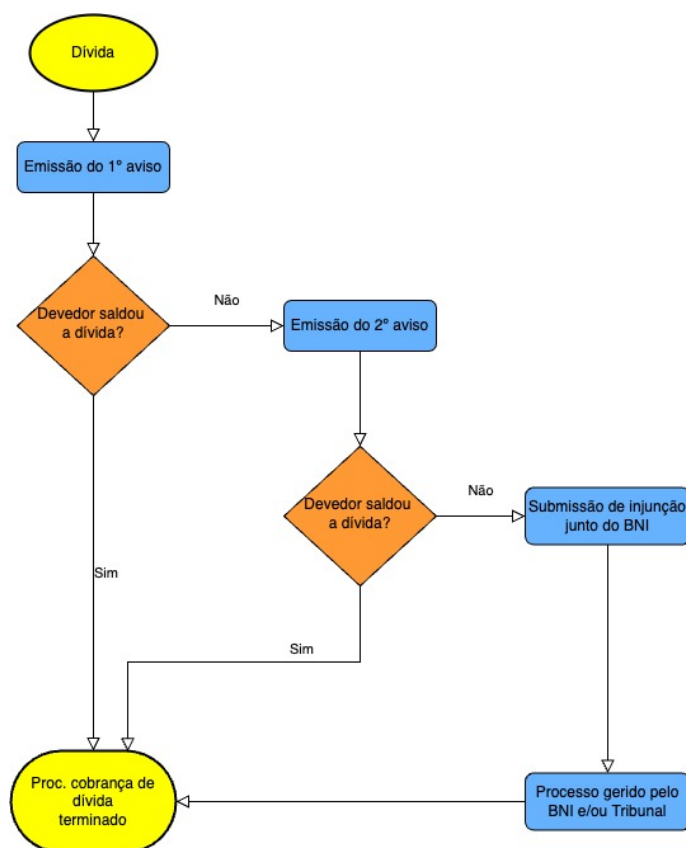


Figura 4.1: Fluxograma representativo dos processos de cobrança de dívida e de injunção

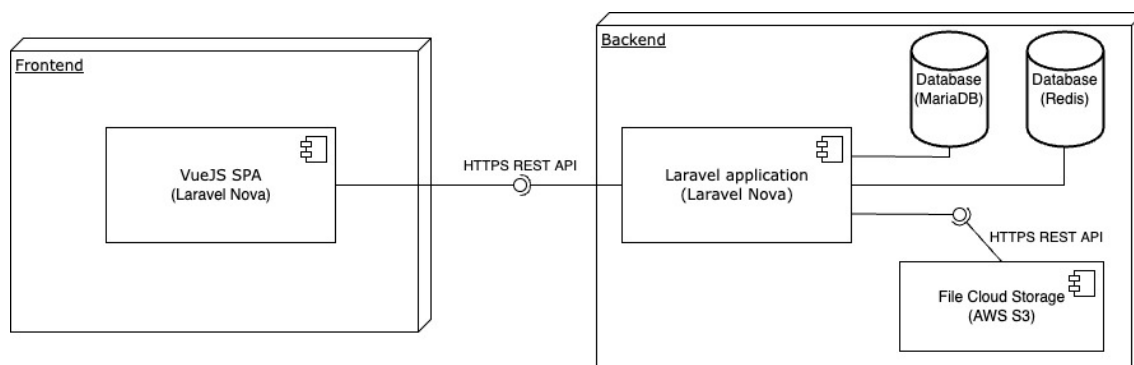


Figura 4.2: Diagrama de componentes do Fluxio

4.2 Arquitetura da aplicação

No que respeita ao desenho da aplicação, como se pode ver no diagrama representado na Figura 4.2, a mesma divide-se em dois componentes principais: *frontend* e *backend*.

O *frontend* diz respeito à camada com a qual o utilizador irá interagir diretamente e é constituída por uma *Single Page Application* (SPA), parte integrante do Laravel Nova¹ e que tem como base de desenvolvimento *VueJS*, uma *framework* de *JavaScript*, e *TailwindCss*², uma *framework* de CSS.

O segundo componente, *backend*, engloba toda a camada lógica da aplicação, nomeadamente, a lógica de negócio programada em PHP com duas camadas de abstração: *Laravel* como *framework* de PHP e *Laravel Nova* como uma extensão de Laravel para criar painéis de administração.

No *backend* para além da lógica de negócio são utilizadas duas instâncias de base de dados com objetivos distintos: uma instância de base de dados relacional (*MariaDB*) responsável pela gestão e organização de todos os dados relevantes para a aplicação e para os utilizadores e uma instância de *Redis* que é uma base de dados em memória, mas que no contexto do *Fluxio*, será usada como *cache* dada a sua grande velocidade de leitura/escrita.

Por fim, dado que uma das funcionalidades da aplicação proposta nesta dissertação prende-se com o armazenamento de faturas, é utilizado armazenamento na *cloud*, através do serviço S3³ da Amazon AWS, tornando a aplicação mais flexível e rentável ao nível de custo do armazenamento, segurança, escalabilidade, redundância dos ficheiros guardados e velocidade de disponibilização dos mesmos.

Uma aplicação *Web* moderna tem normalmente na sua infraestrutura uma ou mais API com o principal objetivo de transferir algum tipo de informação entre dois ou mais componentes da infraestrutura. Para além disto, uma API, quando bem desenhada e implementada, traz mais flexibilidade e escalabilidade a uma solução *Web*. Contudo, uma API é normalmente exposta ao exterior,

¹<https://nova.laravel.com/>

²<https://tailwindcss.com/>

³<https://aws.amazon.com/s3/>

para ser acedida por outros componentes, passando a segurança da mesma a ser um fator importante, o que resulta na inclusão de sistemas de autenticação e autorização nestas mesmas API⁴. A solução proposta nesta dissertação contará também com uma API, mais especificamente uma API com estrutura REST (*Representational State Transfer*), que faz parte da ferramenta utilizada para construir a interface gráfica da aplicação, Laravel Nova. Como foi explicado na Secção 2.5, o Laravel Nova é uma ferramenta que tem por base uma SPA programada em VueJS e uma API para estabelecer a comunicação com a camada lógica da aplicação.

Face a isto, e tal como referido na Secção 3.1, o produto desta dissertação será a base de uma nova sociedade comercial; como tal não há necessidade imediata de construir uma API diretamente relacionada com o modelo de negócio proposto.

Não obstante, depois de o *Minimum Viable Product* (MVP) ser validado junto dos utilizadores finais, existe a possibilidade de estender a API do Laravel Nova ou criar um conjunto de *endpoints* de raiz para, por exemplo, desenvolver um conjunto de aplicações nativas para *Android* e *iOS* ou permitir outros serviços de terceiros comunicar com o sistema aqui proposto.

4.3 Cenários de utilização

Como referido anteriormente, o objetivo desta dissertação é a obtenção de um MVP e para tal foram utilizados os requisitos funcionais e não funcionais, descritos no Capítulo 3 e respetivas prioridades aquando o desenvolvimento da aplicação *Fluxio*. As funcionalidades com prioridade alta foram abordadas em primeiro lugar, seguidos dos requisitos com prioridade média. Nas secções seguintes serão exploradas as funcionalidades mais relevantes, acompanhadas de algumas imagens ilustrativas da aplicação desenvolvida.

4.3.1 Autenticação e autorização

As primeiras funcionalidades implementadas foram as relacionadas com autenticação e autorização, uma vez que a jornada do utilizador irá começar sempre com a autenticação na plataforma. Para efeito do MVP foi implementada uma autenticação tradicional através de e-mail e palavra-passe como pode ser visto na Figura 4.3.

Dependendo do tipo de utilizador com o qual é feita a autenticação, o menu de navegação apresentado é diferente (Figura 4.4). Se a autenticação for feita com um utilizador do tipo **Administrador** apenas dois menus são apresentados enquanto que se o utilizador se autenticar como **Credor** o menu apresenta quatro entradas e o painel de controlo apresenta um conjunto de dados estatísticos que apenas dizem respeito ao credor autenticado.

A gestão das entradas a apresentar no menu de navegação bem como qualquer tipo de autorização na aplicação recorre à utilização de Laravel Policies⁵. A título de exemplo, no excerto de

⁴<https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces> - Publicado a 02/06/2022

⁵<https://laravel.com/docs/9.x/authorization#policy-methods> - Consultado a 03/08/2022



Bem-Vindo De Volta!

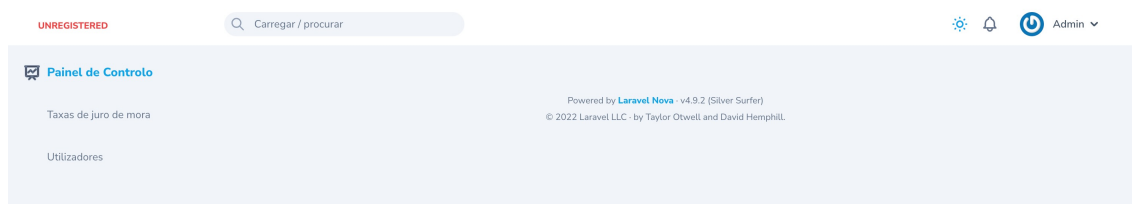
Endereço Email

Palavra-passe

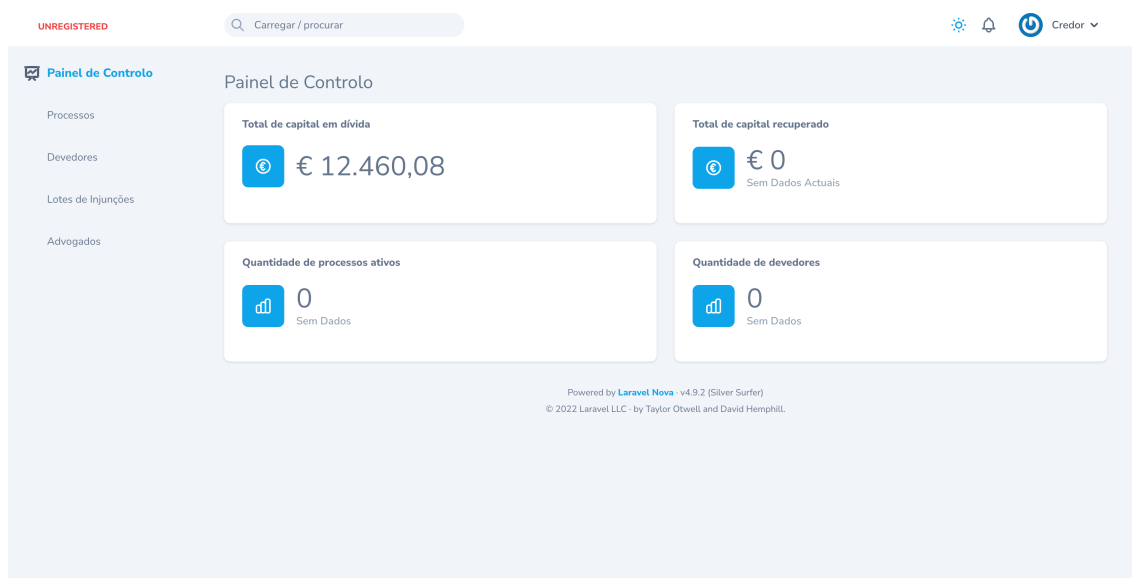
☐ Lembrar-me [Esqueceu a Palavra-passe?](#)

Log In

Figura 4.3: Fluxio - Página de autenticação



(a) Painel de controlo do Administrador



(b) Painel de controlo do Credor

Figura 4.4: Fluxio - Painéis de controlo apresentados depois de o utilizador se autenticar

Código 4.1 é possível ver como é definida a autorização de forma a que a entrada “Taxas de juro de mora” apenas seja apresentada no menu dos administradores da aplicação.

```
1 class LatePaymentInterestRatePolicy
2 {
3     use HandlesAuthorization;
4
5     /**
6      * Determine whether the user can view any models.
7      *
8      * @param User $user
9      * @return bool
10     */
11     public function viewAny(User $user): bool
12     {
13         return $user;
14     }
15 }
```

Código 4.1: Permissão de visualização da listagem de Taxas de Juro

A implementação em causa faz uso da *policy class* `app/Policies/LatePaymentInterestRatePolicy.php` em conjunto com permissões (cuja lógica é implementada pelo pacote de Laravel *spatie/laravel-permission*⁶).

4.3.2 Domínio do ator Administrador

No domínio do administrador, o cenário mais relevante é o **registo de novos utilizadores**. Apesar de não ser o formato final da funcionalidade enquanto serviço a ser comercializado, para o efeito desta dissertação optou-se pelo registo do credor ser feito exclusivamente pelo administrador em vez de este se registar autonomamente. Assim sendo, no momento do registo de um novo utilizador (Figura 4.5), o Administrador deve escolher o tipo de utilizador que este deverá ter e automaticamente um conjunto de permissões serão associados.

4.3.3 Domínio do ator Credor

Passando para o domínio do credor, os principais cenários de utilização são: **criação e gestão de advogados, criação e gestão de devedores, criação e gestão de processos de cobrança de dívida e criação de lotes de injunções**. Como mencionado no início da presente secção, todos estes cenários pressupõem uma autenticação prévia por parte do Credor e a garantia que cada credor apenas terá acesso aos dados registados por si mesmo e nunca a dados de outros Credores presentes no sistema.

⁶<https://github.com/spatie/laravel-permission>

The screenshot shows the 'Create Utilizador' form in the Fluxio admin panel. The form is titled 'Create Utilizador' and is located in the 'Utilizadores' section of the 'Painel de Controlo'. The form contains the following fields:

- Tipo de utilizador: dropdown menu with 'creditor' selected.
- Nome *: text input with 'Mário Macedo' entered.
- E-mail *: text input with 'mario.macedo@mail.com' entered.
- Palavra-passe *: password input field with masked characters.

At the bottom of the form, there are three buttons: 'Cancelar', 'Criar & Adicionar Outro', and 'Create Utilizador'. The page footer indicates it is powered by 'Laravel Nova - v4.9.2 (Silver Surfer)' and includes a copyright notice for 2022.

(a) Formulário de registo de um novo Credor

The screenshot shows the 'Utilizador Details: Mário Macedo' page in the Fluxio admin panel. The page displays the details of the user 'Mário Macedo' in a table format:

ID	5
Avatar	
Tipo de utilizador	creditor
Nome	Mário Macedo
E-mail	mario.macedo@mail.com

At the top right of the details page, there are three icons: a menu icon, a refresh icon, and a delete icon. The page footer indicates it is powered by 'Laravel Nova - v4.9.2 (Silver Surfer)' and includes a copyright notice for 2022.

(b) Detalhes de um Credor após ter sido registado

Figura 4.5: Fluxio - Registo e detalhes de um Credor por parte do Administrador

The screenshot displays the 'Create Advogado' form within the Fluxio application. The form is located in the main content area, with a sidebar menu on the left and a top navigation bar. The sidebar menu includes 'Processos', 'Devedores', 'Lotes de Injunções', and 'Advogados'. The top navigation bar shows 'UNREGISTERED', a search bar, and user information 'Credor'. The form itself has the following fields: 'Nome' (required), 'E-mail' (required), 'NIF', 'Phone', 'Fax', 'Cédula Profissional', and 'Morada' (required, dropdown menu). The form is styled with a light blue background and white input fields. At the bottom of the form, there are three buttons: 'Cancelar', 'Criar & Adicionar Outro', and 'Create Advogado'. The footer of the page mentions 'Powered by Laravel Nova - v4.9.2 (Silver Surfer)' and '© 2022 Laravel LLC - by Taylor Otwell and David Hemphill'.

Figura 4.6: Fluxio - Registo de um novo Advogado

4.3.3.1 Criação e gestão de advogados

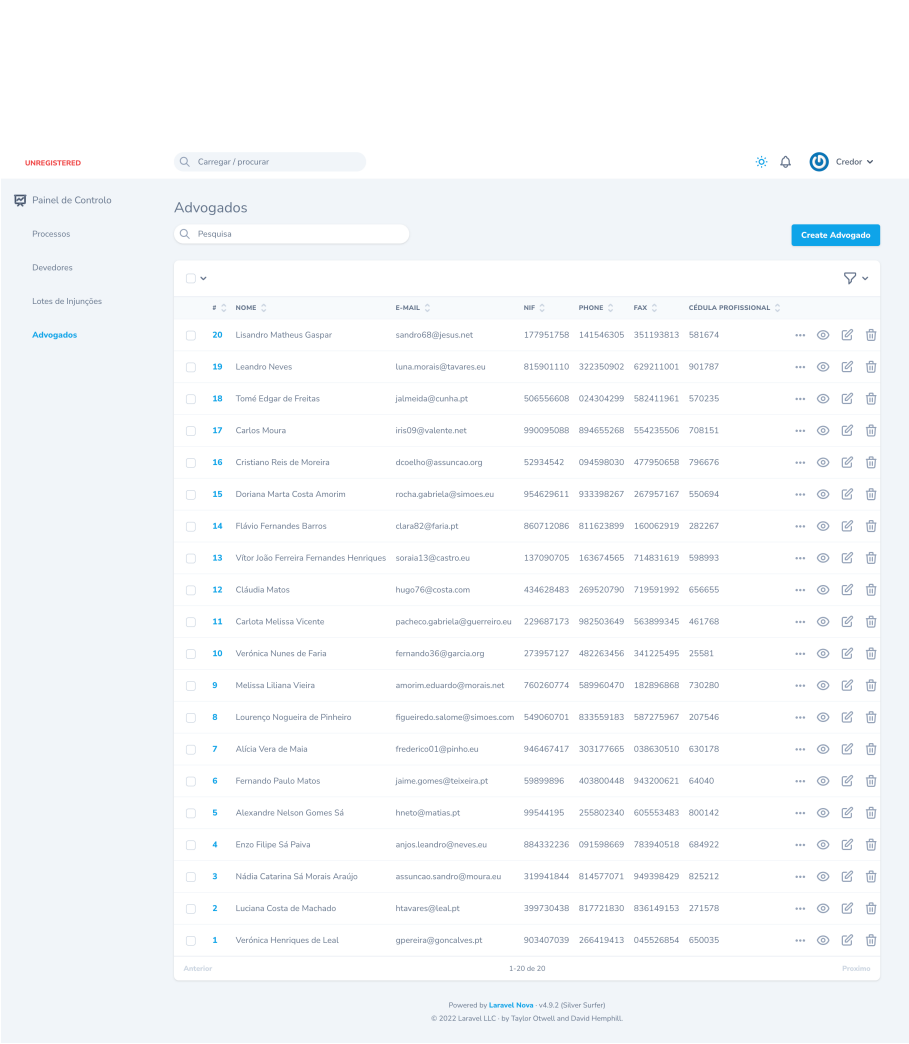
No que respeita ao primeiro cenário, **criação e gestão de advogados**, trata-se de um típico CRUD⁷ (*Create, Read, Update, Delete*), isto é, o credor tem a possibilidade de criar novos advogados (Figura 4.6), visualizar detalhes de advogados previamente inseridos na plataforma (Figura 4.7), atualizar (Figura 4.8) e apagar os dados dos mesmos, sendo que, quando se apaga um advogado a relação com todos os processos que tenham esse mesmo advogado associado também será apagada.

4.3.3.2 Criação e gestão de devedores

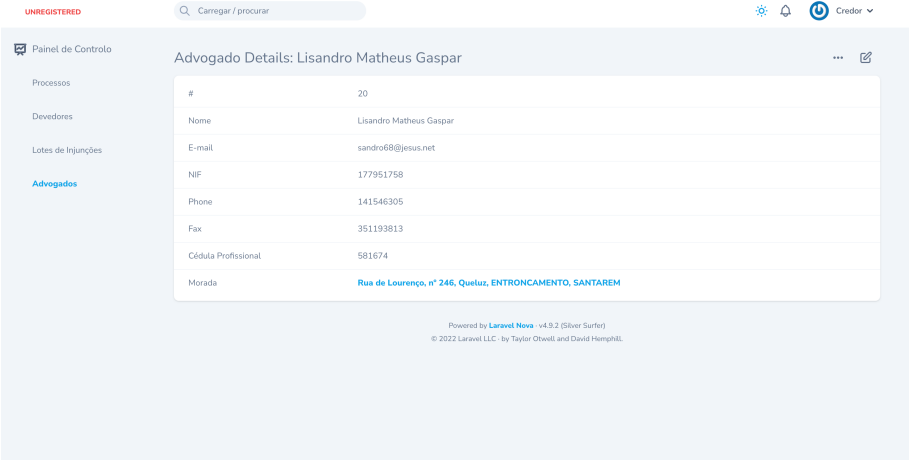
À semelhança do cenário anterior, a **criação e gestão de devedores** é também um típico CRUD, onde o credor tem a possibilidade de criar um novo Devedor (Figura 4.9), visualizar detalhes de devedores previamente inseridos na plataforma (Figura 4.10), atualizar os dados do mesmo (Figura 4.11) e apagar um devedor, sendo que, neste último caso todos os processos associados ao devedor também serão apagados ou anonimizados.

As principais diferenças entre o CRUD dos Devedores e o CRUD dos Advogados são a presença de três dados estatísticos na página de detalhes de um Devedor, como se pode ver na Figura 4.10. Estes dados estatísticos permitem de uma forma rápida ver a quantidade de processos ativos do Devedor em causa e os totais de capital em dívida e de capital recuperado de todos os processos ativos e terminados do mesmo Devedor.

⁷<https://www.codecademy.com/article/what-is-crud> - Consultado a 25-07-2022



(a) Listagem dos Advogados criados pelo Credor



(b) Detalhes de um Advogado

Figura 4.7: Fluxio - Listagem e detalhes da entidade Advogado

UNREGISTERED

Carregar / procurar

Painel de Controlo

- Processos
- Devedores
- Lotes de Injunções
- Advogados**

Update Advogado: Lisandro Matheus Gaspar

Nome *	Lisandro Matheus Gaspar
E-mail *	sandro68@jesus.net
NIF	177951758
Phone	141546305
Fax	351193813
Cédula Profissional	581674
Morada *	Rua de Lourenço, n.º 246, Queluz, ENTRONCAMENTO, SANTAREM

Cancelar Atualizar e Continuar a Edição Update Advogado

Powered by **Laravel Nova** - v4.9.2 (Silver Surfer)
© 2022 Laravel LLC - by Taylor Otwell and David Hemphill.

Figura 4.8: Fluxio - Formulário de atualização dos dados de um Advogado existente

UNREGISTERED

Carregar / procurar

Painel de Controlo

- Processos
- Devedores**
- Lotes de Injunções
- Advogados

Create Devedor

Nome *	Nome
NIF *	NIF
Domicílio convencionado	☐
Taxa de juro (%) *	Taxa de juro (%)
Morada *	—

Cancelar Criar & Adicionar Outro Create Devedor

Powered by **Laravel Nova** - v4.9.2 (Silver Surfer)
© 2022 Laravel LLC - by Taylor Otwell and David Hemphill.

Figura 4.9: Fluxio - Registo de um novo Devedor

UNREGISTERED

Carregar / procurar

Credor ▾

Painel de Controlo

Processos

Devedores

Lotes de Injunções

Advogados

Devedores

Pesquisa

Create Devedor

#	NOME	NIF	DOMICÍLIO CONVENCIIONADO	TAXA DE JURO (%)	QUANTIDADE DE PROCESSOS	
☐	20 Santiago Leonardo Simões de Lopes	338415781		3.20	1	⋮
☐	19 Santiago Jaime de Leal	709288312		1.90	1	⋮
☐	18 Cristiana Magalhães	798772435		2.70	1	⋮
☐	17 Sandro Mota	400293290		1.10	1	⋮
☐	16 Cristiana Miriam Esteves Neto Lopes	495016268		3.90	1	⋮
☐	15 Salomé Eva Vicente de Carvalho	240633318		1.70	1	⋮
☐	14 Catarina Sá	778960136		3.30	1	⋮
☐	13 Clara Anna de Freitas	788789872		4.10	1	⋮
☐	12 Pedro Márcio de Rodrigues	253921717		2.50	1	⋮
☐	11 Rúben Tomé Pinheiro Soares	884698137		3.20	1	⋮
☐	10 Alice Letícia de Macedo	461961651		3.00	1	⋮
☐	9 Mara Domingues de Lima	394036898		1.90	1	⋮
☐	8 Luís Francisco Leite Jesus Sá	416753390		4.60	1	⋮
☐	7 Gabriel Vaz de Faria	833647155		2.60	1	⋮
☐	6 Lorena Branco de Gomes	567885817		3.40	1	⋮
☐	5 Nelson Samuel Sá Garcia Amaral	907546744		1.20	1	⋮
☐	4 Isabela Miranda Pires	873599097		2.80	1	⋮
☐	3 Alice Ângela Mendes Brito	778869938		4.10	1	⋮
☐	2 João Cardoso de Magalhães	853475904		3.70	1	⋮
☐	1 Leandro Rafael Rocha Ferreira	818504135		4.70	1	⋮

Anterior 1-20 de 20 Próximo

Powered by **Laravel Nova** -v4.9.2 (Silver Surfer)
© 2022 Laravel LLC - by Taylor Otwell and David Hemphill.

(a) Listagem dos Devedores previamente registados

UNREGISTERED

Carregar / procurar

Credor ▾

Painel de Controlo

Processos

Devedores

Lotes de Injunções

Advogados

Quantidade de processos ativos

0

Sem Dados

Total de capital em dívida

€ 837,78

Total de capital recuperado

€ 0

Sem Dados Actuais

Devedor Details: Santiago Leonardo Simões de Lopes

⋮

#	20
Nome	Santiago Leonardo Simões de Lopes
NIF	338415781
Domicílio convenconado	
Taxa de juro (%)	3.20
Morada	Rua St. Victória Teixeira, Vila Franca de Xira, LAGOA (ALGARVE), FARO

Processos ▾

Pesquisa

Create Processo

ID	ESTADO	CAPITAL EM DÍVIDA	OUTROS CUSTOS	JUROS DE MORA	INÍCIO DO CONTRATO	FIM DO CONTRATO	TERMINOS CONTRATUAIS GENÉRICOS	ENVIAR À DISTRIBUIÇÃO
☐	20	Processo aberto	€ 837,78	€ 222,72	€ 57,01	12/06/2019	12/04/2021	

Anterior 1-1 de 1 Próximo

Powered by **Laravel Nova** -v4.9.2 (Silver Surfer)
© 2022 Laravel LLC - by Taylor Otwell and David Hemphill.

(b) Detalhes de um Devedor

Figura 4.10: Fluxio - Páginas de listagem e detalhes da entidade Devedor

UNREGISTERED

Carregar / procurar

Credor

Painel de Controlo

Processos

Devedores

Lotes de Injunções

Advogados

Update Devedor: Santiago Leonardo Simões de Lopes

Nome * Santiago Leonardo Simões de Lopes

NIF * 338415781

Domicilio convenionado ☒

Taxa de juro (%) * 3.20

Morada * Rua St. Victória Teixeira, Vila Franca de Xira, LAGOA (ALGARVE), FARO

Cancelar Atualizar e Continuar a Edição Update Devedor

Powered by **Laravel Nova** - v4.9.2 (Silver Surfer)
© 2022 Laravel LLC - by Taylor Otwell and David Hemphill.

Figura 4.11: Fluxio - Formulário de atualização dos dados de um Devedor existente

A obtenção e apresentação destes dados estatísticos faz uso das *Metrics*⁸ do Laravel Nova, e para tal basta registar uma *Class* que estenda a *Class Laravel/Nova/Metrics/Value* e definir o método *calculate*, como se pode ver no excerto de Código 4.2.

```

1 class ActiveProcessCount extends Value
2 {
3     public $name = 'Quantidade de processos ativos';
4
5     /**
6      * Calculate the value of the metric.
7      *
8      * @param \Laravel\Nova\Http\Requests\NovaRequest $request
9      * @return mixed
10     */
11     public function calculate(NovaRequest $request): mixed
12     {
13         if($request->resource === 'debtors'){
14             return $this->count($request, Process::fromDebtor($request->resourceId)
15                               ->withOpenStatus());
16         }
17         return $this->count($request, Process::fromAuthCreditor()->withOpenStatus())
18         ;
19     }
20 }

```

⁸<https://nova.laravel.com/docs/4.0/metrics/defining-metrics.html#defining-metrics> - Consultado a 17/02/2022

UNREGISTERED

Carregar / procurar

Painel de Controlo

Processos

Devedores

Lotes de Injunções

Advogados

Create Processo

Devedor *

Início do contrato *

Fim do contrato

Argumentação *

Termos contratuais genéricos

Enviar à distribuição

Causa a Pedir

Decreto-lei da transação comercial

Advogado *

Nº processo Citius

Referência injunção Citius

Referência lote Citius

Cancelar

Criar & Adicionar Outro

Create Processo

Powered by **Laravel Nova** - v4.9.2 (Silver Surfer)

© 2022 Laravel LLC - by Taylor Otwell and David Hemphill

Figura 4.12: Fluxio - Registo de um novo Processo

19 }

Código 4.2: Implementação da estatística “Processos ativos”

4.3.3.3 Criação e gestão de processos de cobrança de dívida

Processos de cobrança de dívida são a entidade central da aplicação e, por esse motivo, o CRUD desta entidade é mais complexo uma vez que tem mais entidades relacionadas. No entanto, possui todas as funcionalidades dos CRUD anteriores, ou seja, registar (Figura 4.12), listar e visualizar detalhes (Figuras 4.13 e 4.14), atualizar (Figura 4.15) e apagar processos.

A definição dos atributos e das relações da entidade Processo (e de qualquer outra entidade no projeto) é feita no método *fields* da classe que define o recurso para o Laravel Nova, como se pode ver no excerto de Código 4.3. Quando se define uma relação entre duas entidades, define-se o tipo

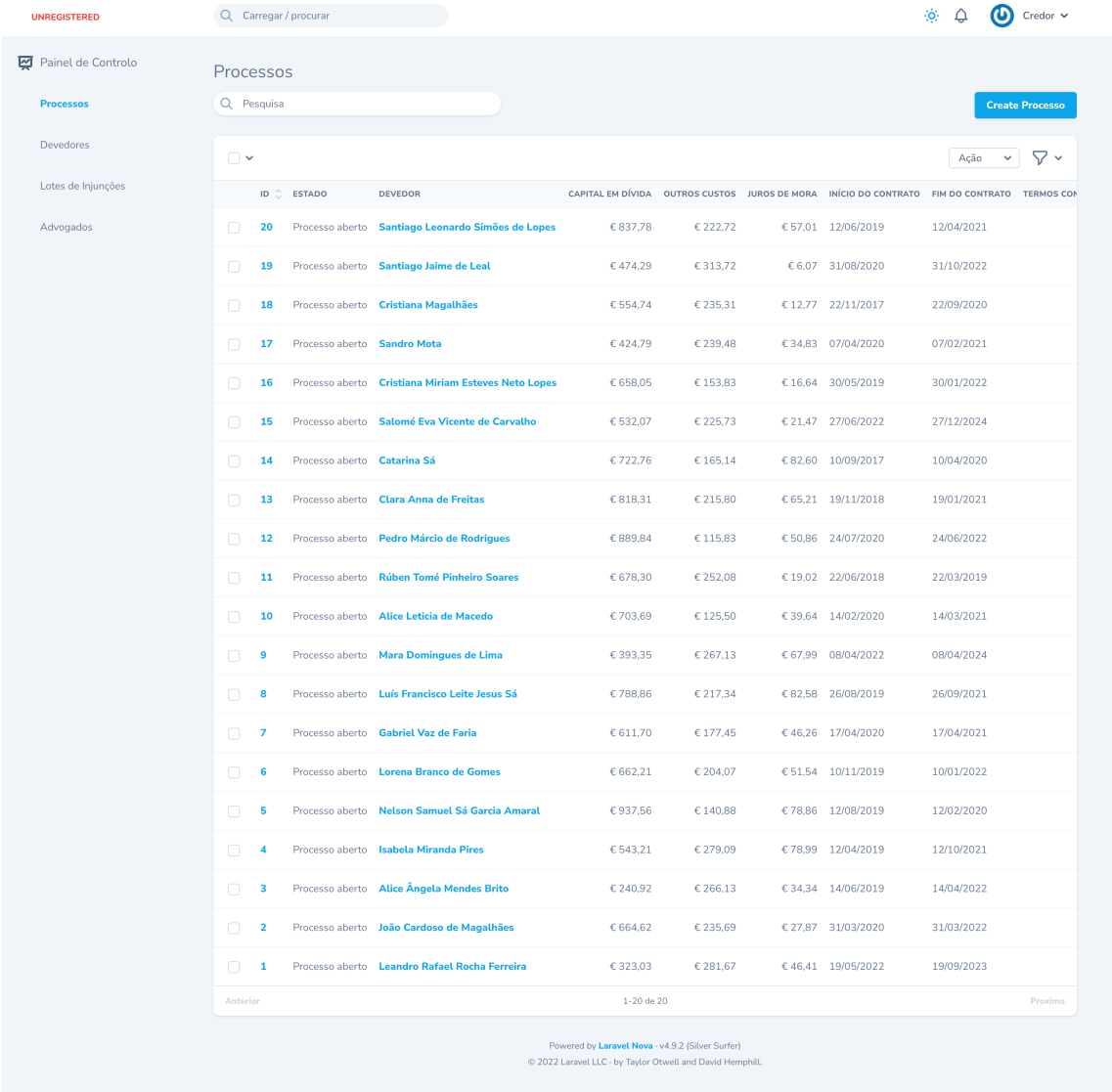


Figura 4.13: Fluxio - Listagem dos Processos previamente registados

UNREGISTERED

Carregar / procurar

 Credor

 Painel de Controlo

Processos

Devedores

Lotes de Injunções

Advogados

Processo Details: 20

ID

20

Estado

Processo aberto

Devedor

Santiago Leonardo Simões de Lopes

Capital em dívida

€ 837,78

Outros custos

€ 222,72

Juros de mora

€ 57,01

Início do contrato

12/06/2019

Fim do contrato

12/04/2021

Argumentação

Mostrar O Conteúdo

Termos contratuais genéricos



Enviar à distribuição



Causa a Pedir

Locação Financeira

Decreto-lei da transação comercial

DL n.º 62/2013, de 10 de maio

Advogado

Lisandro Matheus Gaspar

N.º processo Citius

—

Referência injunção Citius

—

Referência lote Citius

—

Faturas

Pesquisa

Create Fatura

ID	DESCRITIVO	CAPITAL	DATA DE EMISSÃO	N.º DIAS PARA PAGAMENTO	DATA INÍCIO CONTAGEM DE JUROS	JUROS DE MORA	
40	Quia quia sit est ratione veniam sit.	€ 479,94	13/09/2021	88	10/12/2021	€ 13,36	   
39	Suscipit sequi aut minima.	€ 357,84	02/08/2019	1	03/08/2019	€ 43,65	   

Anterior1-2 de 2Proximo

Outros custos

Pesquisa

Create Outro custo

ID	DESCRITIVO	CAPITAL	
80	Ut omnis blanditiis in.	€ 94,89	   
79	Est iure fugit aut dolorum consequatur.	€ 91,83	   
78	Doloremque ex qui tenetur.	€ 27,49	   
77	Quam est est ut quasi.	€ 8,51	   

Anterior1-4 de 4Proximo

Powered by **Laravel Nova** - v4.9.2 (Silver Surfer)

© 2022 Laravel LLC - by Taylor Otwell and David Hemphill.

Figura 4.14: Fluxio - Detalhes de um Processo

UNREGISTERED

Carregar / procurar

   Credor ▾

 Painel de Controlo

Processos

Devedores

Lotes de Injunções

Advogados

Update Processo: 20

Estado *

Processo aberto ▾

Devedor *

Santiago Leonardo Simões de Lopes ▾

Início do contrato *

12/06/2019  Europe/Lisbon

Fim do contrato

12/04/2021  Europe/Lisbon

Argumentação *

Nostrum voluptatem praesentium aperiam itaque. Voluptatum aut quidem nobis beatae magnam autem illo. Et nihil voluptatibus nulla. Odit aut sint non. Velit harum porro ipsa. Et consectetur dolorem vel voluptate assumenda aperiam qui. Sed dicta rerum voluptatem. Qui voluptatibus accusantium excepturi qui fuga.

Termos contratuais genéricos

☐

Enviar à distribuição

☒

Causa a Pedir

Locação Financeira ▾

Decreto-lei da transação comercial

DL n.º 62/2013, de 10 de maio ▾

Advogado *

Lisandro Matheus Gaspar ▾

Nº processo Citius

Nº processo Citius

Referência injunção Citius

Referência injunção Citius

Referência lote Citius

Referência lote Citius

Cancelar

Atualizar e Continuar a Edição

Update Processo

Powered by [Laravel Nova](#) - v4.9.2 (Silver Surfer)

© 2022 Laravel LLC - by Taylor Otwell and David Hemphill.

Figura 4.15: Fluxio - Formulário de atualização dos dados de um Processo existente

de relação em causa, por exemplo, um para muitos ou muitos para muitos (que irá coincidir com o modelo de domínio definido na Secção 3.3) e define-se a classe da entidade relacionada.

```

1 public function fields(Request $request): array
2 {
3     return [
4         ID::make(__('ID'), 'id')
5         ->sortable(),
6         BelongsTo::make(__('Estado'), 'processStatus', ProcessStatus::class)
7         ->hideWhenCreating()
8         ->viewable(false)
9         ->filterable(),
10        BelongsTo::make(__('Devedor'), 'debtor', Debtor::class)
11        ->rules('required'),
12        Currency::make(__('Capital em dívida'), 'debt')
13        ->hideWhenCreating()
14        ->hideWhenUpdating(),
15        Currency::make(__('Outros custos'), 'otherCostsTotal')
16        ->hideWhenCreating()
17        ->hideWhenUpdating(),
18        Currency::make(__('Juros de mora'), 'late_payment_interest')
19        ->hideWhenCreating()
20        ->hideWhenUpdating(),
21        Date::make(__('Início do contrato'), 'contract_date')
22        ->rules('required')
23        ->filterable(),
24        Date::make(__('Fim do contrato'), 'contract_end_date')
25        ->rules('after:contract_date')
26        ->filterable(),
27        Textarea::make(__('Argumentação'), 'reasoning')
28        ->hideFromIndex()
29        ->rules('required'),
30        Boolean::make(__('Termos contratuais genéricos'), '
            has_generic_contractual_terms'),
31        Boolean::make(__('Enviar à distribuição'), 'is_sent_at_distribution'),
32        Select::make(__('Causa a Pedir'), 'cause_of_action')
33        ->options(CauseOfActionEnum::map())
34        ->displayUsingLabels()
35        ->hideFromIndex(),
36        Select::make(__('Decreto-lei da transação comercial'), '
            commercial_transaction_decree_law')
37        ->options(CommercialTransactionDecreeLawEnum::map())
38        ->displayUsingLabels()
39        ->hideFromIndex(),
40        BelongsTo::make(__('Advogado'), 'lawyer', Lawyer::class)
41        ->hideFromIndex(),
42        Text::make(__('Nº processo Citius'), 'citius_process_id')
43        ->rules(['unique', 'nullable']),

```

Figura 4.16: Fluxio - Registo de uma nova Fatura

```

44     Text::make(__('Referência injunção Citius'), 'citius_injunction_reference')
45         ->rules([
46             'unique',
47             'nullable'
48         ])
49         ->hideFromIndex(),
50     Text::make(__('Referência lote Citius'), 'citius_batch_reference')
51         ->hideFromIndex(),
52     HasMany::make(__('Faturas'), 'invoices', ProcessInvoice::class)
53         ->hideFromIndex(),
54     HasMany::make(__('Outros custos'), 'otherProcessCosts', OtherProcessCost::
55         class)
56         ->hideFromIndex(),
57 ];

```

Código 4.3: Registo dos campos da entidade Processo

Faz também parte da gestão de processos de cobrança de dívida o controlo das várias faturas associadas a cada processo. Como se pode ver pela Figura 4.13, na página de detalhes de um Processo é apresentada uma tabela com todas as faturas e ao adicionar uma nova fatura o utilizador tem de adicionar alguns dados (Figura 4.16) fulcrais para o cálculo do capital em dívida e para o cálculo de juros, nomeadamente o valor da fatura, data de emissão e número de dias para pagamento.

Esta abordagem faz com que o valor dos juros seja calculado sempre que os dados de uma fatura sejam acedidos, garantindo que será sempre tido em conta o dia correto (dia de acesso aos

dados da fatura) para o cálculo dos juros.

```

1 public function getLatePaymentInterestAttribute(): float
2 {
3     $latePaymentInterest = 0;
4     $stopCountingDaysAt = now();
5
6     if($this->interests_start_counting_at->greaterThan($stopCountingDaysAt)) {
7         return $latePaymentInterest;
8     }
9
10    foreach (LatePaymentInterestRate::allFromCache() as $interestRate) {
11
12        if ($interestRate['year'] >= $this->interests_start_counting_at->year &&
13            $interestRate['year'] <= $stopCountingDaysAt->year) {
14
15            $previousDate = $interestRate['year'] === $this->
16                interests_start_counting_at->year ? $this->
17                interests_start_counting_at : Carbon::createFromDate($interestRate['
18                year'], 1, 1);
19
20            $lastDayToCount = Carbon::createFromDate($interestRate['year'], 12, 31);
21
22            if ($lastDayToCount->greaterThan($stopCountingDaysAt)) {
23                $lastDayToCount = $stopCountingDaysAt;
24            }
25
26            $delayDays = $lastDayToCount->diffInDays($previousDate) + 1;
27
28            $latePaymentInterest += $this->value * ($interestRate['rate'] / 100 /
29                $lastDayToCount->daysInYear) * $delayDays;
30        }
31    };
32
33    return round($latePaymentInterest, 2);
34 }

```

Código 4.4: *Accessor* da entidade Fatura responsável for calcular o valor dos juros

Para além do *Accessor* na entidade Fatura, foi definido também um outro *Accessor* na entidade Processo (excerto de Código 4.5), que basicamente soma o valor retornado pelo *Accessor* de cada uma das faturas associadas.

```

1 public function getLatePaymentInterestAttribute(): float
2 {
3     return round($this->invoices->sum('late_payment_interest'), 2);
4 }

```

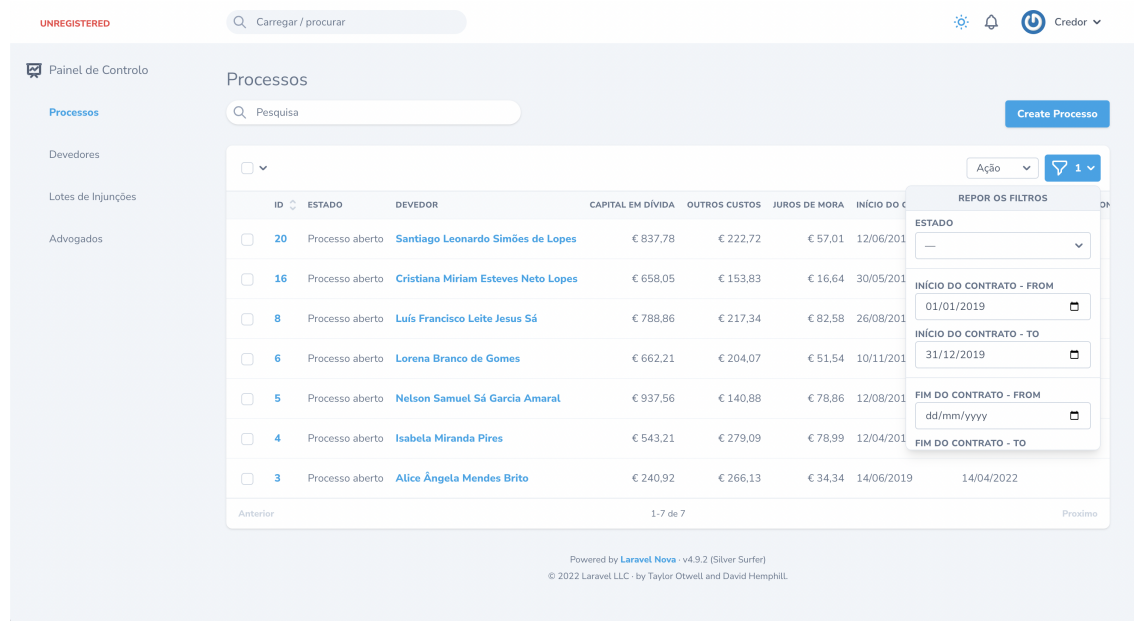


Figura 4.17: Fluxio - Filtros para filtragem dos processos mostrados

Código 4.5: *Accessor* da entidade Processo responsável por somar o valor dos juros de todas as faturas associadas

Nesta entidade foi também implementado um conjunto de filtros, nomeadamente estado, data de início e data de fim de contrato para facilitar a utilização e pesquisa de um processo, como se pode ver na Figura 4.17.

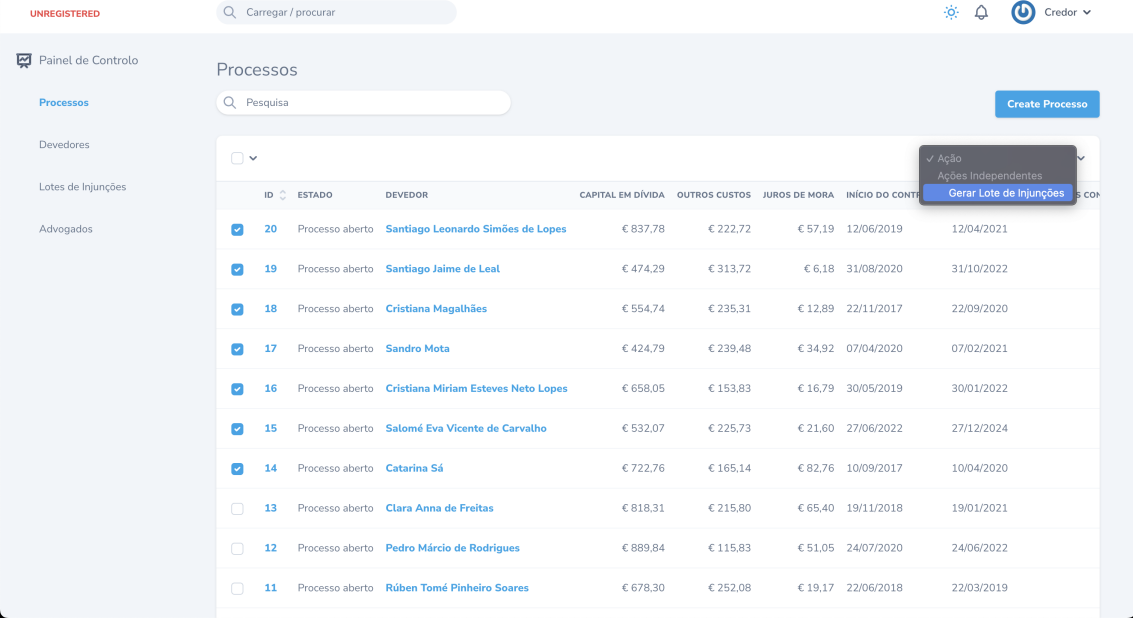
4.3.3.4 Criação de lotes de injunções

O último cenário a ser descrito é o mais importante do MVP uma vez que é a funcionalidade que visa colmatar a lacuna existente no mercado na geração de lotes de injunções para submissão no Citius como foi evidenciado na Secção 2.4.3.

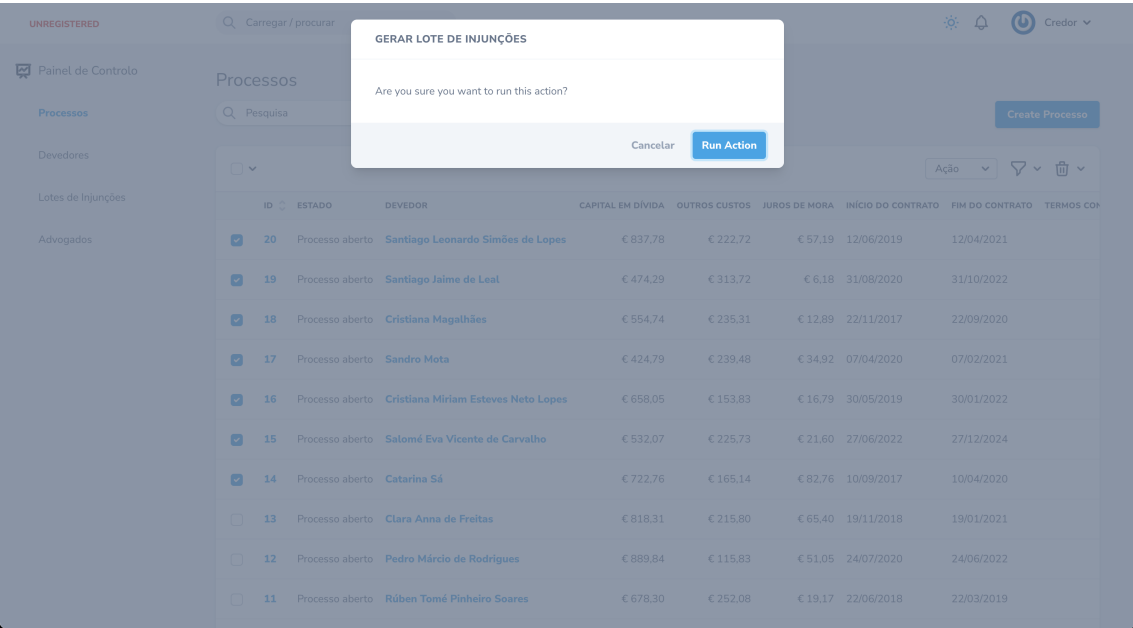
Esta funcionalidade permite que o utilizador selecione quantos processos abertos desejar, independentemente dos devedores associados e, em dois cliques, gerar um ficheiro XML com todos esses processos (Figura 4.18).

Dado que o utilizador pode seleccionar um número ilimitado de processos para adicionar ao lote (podem inclusive ser todos os processos abertos), a preparação dos dados, a geração e validação do XML acontece em plano de fundo utilizando o sistema de *queues*⁹ de Laravel que por sua vez usa um tipo de persistência tipicamente mais rápido que, no caso do *Fluxio*, é o Redis. Esta abordagem permite que o utilizador não tenha que permanecer na página onde está (pode continuar a navegar na plataforma livremente) e, acima de tudo, não enfrente erros como sessão expirada, *timeout*

⁹<https://laravel.com/docs/9.x/queues#main-content> - Consultado a 22/06/2022



(a) Seleção dos processos a fazer parte do lote de injunções



(b) Ecrã de confirmação antes de proceder à geração do ficheiro XML

Figura 4.18: Fluxio - Criação do ficheiro XML a partir dos processos seleccionados

por parte do PHP ou por parte do Nginx, uma vez que a geração destes ficheiros pode demorar milissegundos ou horas, dependendo do volume de dados em causa. Face a isto, o utilizador precisa de ser notificado quando o ficheiro estiver pronto para ser transferido e para isso utilizou-se o sistema de notificações¹⁰ de Laravel. Assim que o ficheiro estiver pronto, o utilizador receberá uma notificação através de duas vias (excerto de Código 4.6): uma notificação dentro da própria aplicação (Figura 4.19) e uma mensagem de correio eletrónico para o endereço de email que o utilizador usa para se autenticar na aplicação (Figura 4.20).

```
1 class CitiusInjunctionsBatchGenerated extends Notification implements ShouldQueue
2 {
3     use Queueable;
4
5     private CitiusInjunctionBatch $citiusInjunctionBatch;
6
7     /**
8      * Create a new notification instance.
9      *
10     * @param array $file
11     */
12     public function __construct(CitiusInjunctionBatch $citiusInjunctionBatch)
13     {
14         $this->citiusInjunctionBatch = $citiusInjunctionBatch;
15     }
16
17     /**
18      * Get the notification's delivery channels.
19      *
20      * @param mixed $notifiable
21      * @return array
22      */
23     public function via($notifiable)
24     {
25         return ['mail', NovaChannel::class];
26     }
27
28     /**
29      * Get the mail representation of the notification.
30      *
31      * @param mixed $notifiable
32      * @return \Illuminate\Notifications\Messages\MailMessage
33      */
34     public function toMail($notifiable)
35     {
36         return (new MailMessage)
37             ->line('Lote de injunções '. $this->citiusInjunctionBatch->
38                 filename .' disponível para transferir.')
```

¹⁰<https://laravel.com/docs/9.x/notifications#main-content> - Consultado a 23/06/2022

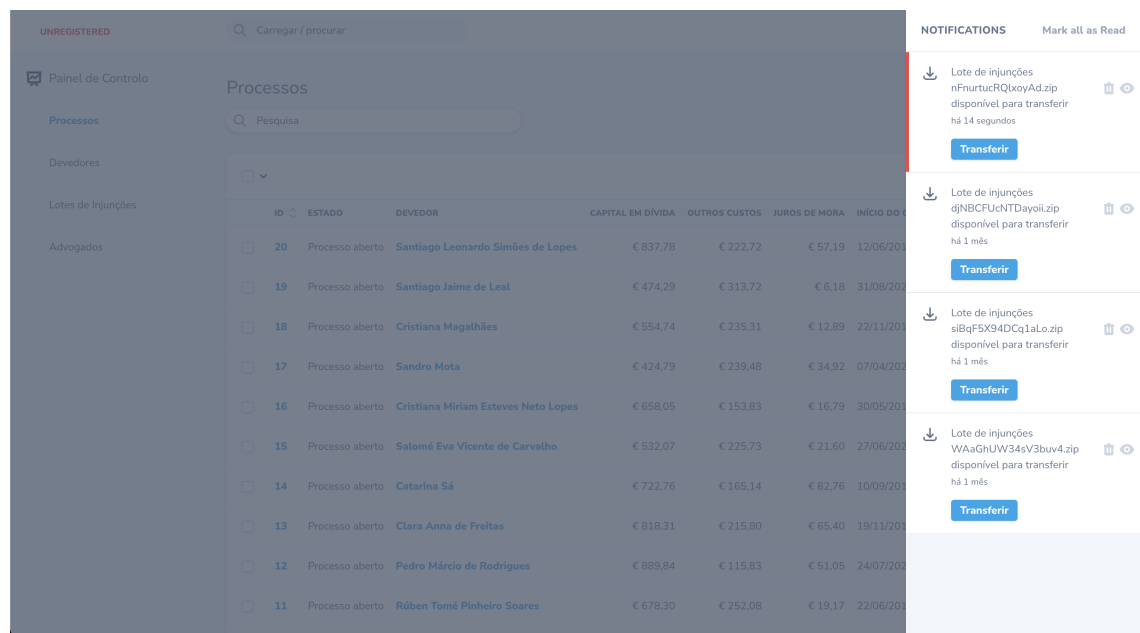


Figura 4.19: Fluxio - Notificação de ficheiro disponível para transferir dentro da aplicação

```

38         ->action('Transferir', $this->citiusInjunctionBatch->downloadUrl
39             ());
40     }
41     /**
42      * Get the array representation of the notification.
43      *
44      * @param mixed $notifiable
45      * @return NovaNotification
46      */
47     public function toNova($notifiable)
48     {
49         return (new NovaNotification())
50             ->message('Lote de injunções '. $this->citiusInjunctionBatch->filename .
51                 ' disponível para transferir')
52             ->action('Transferir', URL::remote($this->citiusInjunctionBatch->
53                 downloadUrl()))
54             ->icon('download')
55             ->type('info');
56     }
57 }

```

Código 4.6: Class responsável pelo despacho das notificações de ficheiro XML disponível para transferir

Assim que o utilizador receber estas notificações, pode então proceder à transferência (exemplo de um ficheiro gerado pela aplicação no Anexo B) do ficheiro com formato ZIP.

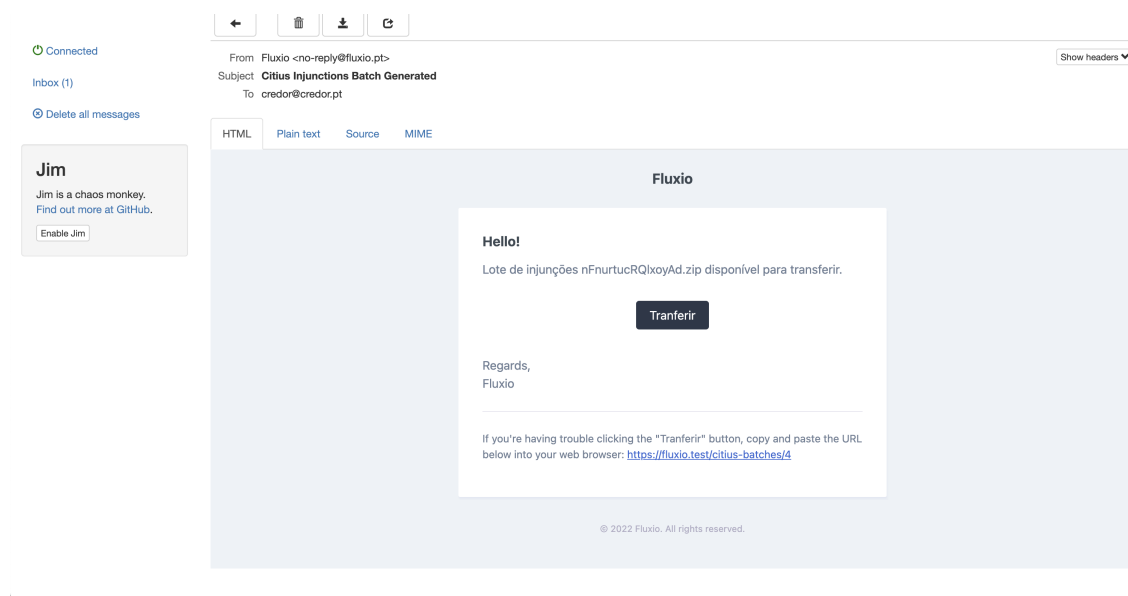


Figura 4.20: Fluxio - Notificação de ficheiro disponível para transferir através de e-mail

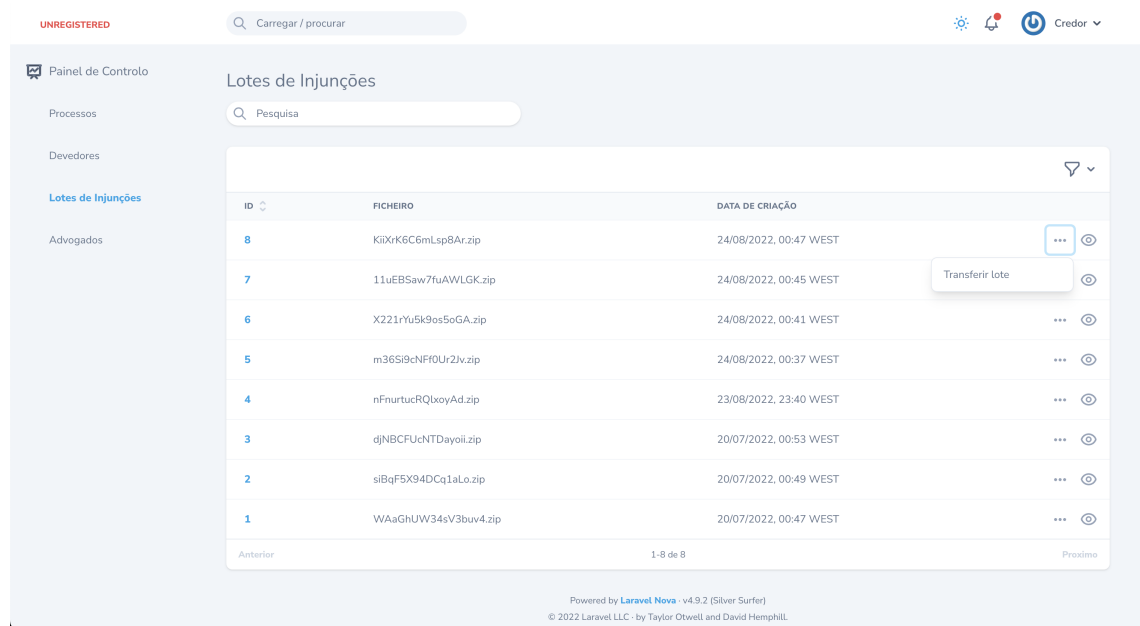
A qualquer momento o utilizador pode aceder a todos os lotes gerados e descarregá-los novamente (Figura 4.21), bastando para isso aceder ao menu “**Lotes de Injunções**” que pode ser encontrado no menu de navegação principal da aplicação.

Para além da geração do lote de injunções ser feita numa *queue*, foram tidas em consideração outras otimizações principalmente no que respeita ao acesso da base de dados relacional (MariaDB), nomeadamente o uso de *Eager Loading*¹¹ providenciado pelo *Eloquent* (o *Object-relational Mapper* usado pelo Laravel). Ao garantir que todas as entidades relacionadas com a entidade Processo (que serão necessárias para gerar o ficheiro XML) são obtidas numa só *query* à base de dados e guardadas em memória, reduz-se ao mínimo o número de *queries* totais e o acesso à memória será sempre mais rápido que à persistência. Para além disto, em termos de infraestrutura de servidor, aumentar os recursos em termos de memória torna-se financeiramente mais viável do que aumentar a capacidade de um servidor/serviço de base de dados.

4.4 Sumário

O MVP Fluxio realizado como parte desta dissertação foi desenvolvido usando a *framework* Laravel. Para além disso, utilizou-se uma ferramenta de geração de painéis de administração, Laravel Nova, que foi desenvolvida e é mantida pela empresa Laravel. A utilização desta ferramenta respeita a arquitetura do sistema pretendida, isto é, uma SPA no *frontend* que comunica com a camada lógica através de uma REST API, também providenciada pela ferramenta.

¹¹<https://laravel.com/docs/9.x/eloquent-relationships#eager-loading> - Consultado a 03/08/2022



UNREGISTERED

Carregar / procurar

Credor

Painel de Controlo

Processos

Devedores

Lotes de Injunções

Advogados

Lotes de Injunções

Pesquisa

ID	FICHEIRO	DATA DE CRIAÇÃO
8	KiXrK6C6mLsp8Ar.zip	24/08/2022, 00:47 WEST
7	11uEBSaw7fuAWLGK.zip	24/08/2022, 00:45 WEST
6	X221rYu5k9os5oGA.zip	24/08/2022, 00:41 WEST
5	m36Si9cNF0Ur2Jv.zip	24/08/2022, 00:37 WEST
4	nFnurtucRQxoyAd.zip	23/08/2022, 23:40 WEST
3	djNBCFUcNTDayoil.zip	20/07/2022, 00:53 WEST
2	siBqF5X94DCq1aLo.zip	20/07/2022, 00:49 WEST
1	WAaGhUW34sV3buv4.zip	20/07/2022, 00:47 WEST

Anterior 1-8 de 8 Próximo

Transferir lote

Powered by **Laravel Nova** - v4.9.2 (Silver Surfer)
© 2022 Laravel LLC - by Taylor Otwell and David Hemphill.

Figura 4.21: Fluxio - Listagem de todos os ficheiros XML gerados pelo sistema

Em termos de persistência dos dados foram utilizados dois tipos: uma base de dados relacional (MariaDB) para persistência de todos os dados e uma base de dados em memória, *Redis*, essencialmente usada como *cache*.

No que respeita à implementação, foram abordadas as tarefas com prioridade alta e média definidas no momento de levantamento de requisitos e definição das *User Stories* apresentadas na Secção 3.2.2. Tendo isto em conta, implementaram-se cenários de autenticação e autorização, criação e gestão de utilizadores no domínio do Administrador e, por fim, no domínio do Credor desenvolveram-se as funcionalidades criação e gestão de advogados, criação e gestão de devedores, criação e gestão de processos de cobrança de dívida e criação de lotes de injunções.

Durante o desenvolvimento, principalmente da funcionalidade de criação de lotes de injunções, foram tidas em conta questões de desempenho e escalabilidade, uma vez que o Credor pode gerar um Lote de Injunções com um número ilimitado de processos.

Capítulo 5

Validação

5.1 Introdução

De forma sucinta os objetivos para o MVP proposto neste documento foram concretizados com sucesso, no entanto a sua validação apresentou-se como um grande desafio, principalmente, por ser algo novo no contexto de processos de cobrança de dívida no panorama da justiça portuguesa. Exemplos concretos são o facto de não existir nenhum tipo de solução semelhante que possa servir como termo de comparação e o próprio Citius não fornecer uma metodologia de integração mais típica no panorama atual de desenvolvimento de software. Citando Scott Rosenberg, *“Software é fácil de fazer, exceto quando se pretende que faça algo novo. Claro que, depois, existe um corolário: O único software que vale a pena fazer é o que faz algo novo.”* [7].

5.2 Validação dos requisitos

Em termos de validação dos requisitos do ponto de vista das regras de negócio, a mesma foi realizada essencialmente em sessões de demonstração com o Dr. Tiago Lopes de Azevedo, co-orientador deste trabalho, uma vez que foi quem teve a ideia em primeiro lugar e tem conhecimento do funcionamento dos processos de injunção, para além de conhecimento acerca do funcionamento e limitações do Citius. Para além disso, como advogado, tem acesso ao Citius, o que permitiu verificar que o XML gerado pelo Fluxio é de facto suportado pela mesma, como se pode ver na Figura 5.1. Claro está que os dados utilizados pelo Fluxio são dados ficcionados e por isso na Figura 5.1 vê-se que os dados do mandatário (agente de justiça) não coincidem com os dados guardados no Citius, o que foi propositado, uma vez que assegurou que os processos de injunção não foram efetivamente criados na plataforma, até porque para tal acontecer é necessário proceder ao pagamento da taxa de justiça. Isto apenas realça as várias limitações existentes no panorama de cobrança de dívidas a nível nacional, em especial na integração com plataformas do

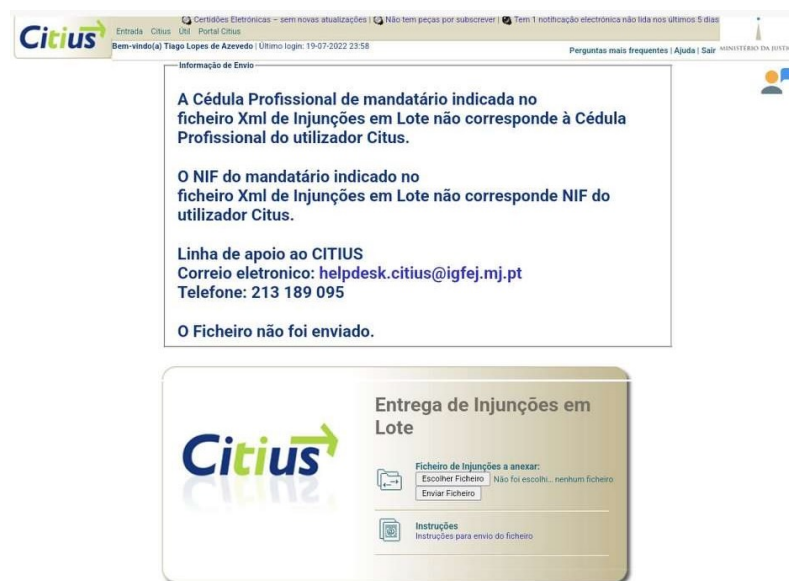


Figura 5.1: Submissão do ficheiro XML gerado pelo Fluxio no Citius com sucesso

Ministério da Justiça, que para além de ainda não possuírem API de integração, também não têm ambientes de teste para desenvolvedores.

No entanto, mesmo com todas as limitações durante o desenvolvimento e validação, todas *User Stories* com prioridades alta e média foram concretizadas com sucesso à exceção das *User Stories*: US211, US214 e US219, principalmente porque durante as sessões de demonstração as prioridades foram re-avaliadas em detrimento de outros requisitos.

No que diz respeito à validação da aplicação, do ponto de vista técnico, a utilização do Laravel Nova apresenta-se como uma mais valia porque é uma ferramenta já testada e validada pela empresa Laravel e pela comunidade de programadores Laravel. Em termos práticos, tal significa que se a ferramenta for utilizada seguindo a documentação e se não se estender ou reescrever partes de funcionalidades, não há necessidade em implementar qualquer tipo de testes automáticos, uma vez que seriam redundantes. Contudo, qualquer tipo de funcionalidade que não seja nativa ao Laravel Nova, deve ser validada.

5.3 Validação do lote de injunções

Com o princípio de que funcionalidades extra ao *Laravel Nova* devem ser devidamente validadas, a única funcionalidade presente neste MVP que requer a implementação de algum tipo de validação automática é a geração do lote de injunções no formato de um ficheiro XML.

A solução encontrada para validar este ficheiro XML foi a implementação de um validador em PHP (Laravel) que usa o XSD providenciado pelo Citius (Anexo A) de forma a validar a estrutura do XML gerado. É importante realçar que esta mesma verificação é feita em todos os XML gerados e não apenas como parte de um *pipeline* de integração contínua. Ao fazer esta validação,

é aumentado o tempo que um ficheiro demora a estar disponível para o utilizador final, no entanto, permite mitigar falhas na geração do XML, principalmente relacionadas com dados corrompidos ou formatos de dados inválidos.

No excerto de Código 5.1 é possível ver que a validação é executada imediatamente a seguir ao ficheiro XML ser gerado e antes de ser comprimido. O validador implementado pode ser encontrado no Anexo C.

```
1 public function generate(): CitiusInjunctionBatch
2 {
3     if ($this->injunctions->isEmpty()) {
4         throw new EmptyInjunctionsCollectionException();
5     }
6
7     return $this->setXmlHeader()
8         ->setCreditorNode()
9         ->setInjunctionsNodes()
10        ->setLatePaymentInterestsSummaryNode()
11        ->setXmlFooter()
12        ->saveXmlToFile()
13        ->zip();
14 }
15
16 private function saveXmlToFile(): CitiusInjunctionBatchGenerator
17 {
18     $this->xmlFilename = 'injuncoes_lote_' . now()->toDateTimeString() . '.xml';
19     $this->fullXmlPath = $this->filesPath . $this->xmlFilename;
20     Storage::disk('clients_tmp')->put($this->fullXmlPath, $this->xml);
21     $this->validateXml();
22
23     return $this;
24 }
25
26 private function validateXml(): void
27 {
28     $validator = new CitiusInjunctionBatchXmlValidator;
29
30     if (!$validator->validate(Storage::disk('clients_tmp')->path($this->fullXmlPath))) {
31         Storage::disk('clients_tmp')->delete($this->fullXmlPath);
32         throw new Exception(implode(PHP_EOL, $validator->displayErrors()));
33     }
34 }
```

Código 5.1: Criação e validação do ficheiro XML antes do mesmo ser comprimido para ser disponibilizado ao utilizador.



Figura 5.2: Quantidade de *queries* efetuadas e tempo total do pedido quando é utilizado *Eager Loading*

5.4 Testes de desempenho

Para efeitos de teste de desempenho da funcionalidade principal da aplicação, US116, utilizou-se uma ferramenta de *debugging* e *profiling*¹ para medir a diferença entre gerar um lote com e sem o uso de *Eager Loading*. Para este teste, a base de dados continha um total de 100 injunções cada uma com 50 faturas associadas e gerou-se um lote de injunções com apenas cinco dessas injunções.

Como se pode ver na Figura 5.2, ao fazer-se *Eager Loading* de todas as relações necessárias para gerar o XML (ver excerto de Código 5.2), o pedido demorou 1,8 segundos a terminar e foram feitas um total de 637 *queries*. Já no segundo caso, fazendo *Lazy Loading* (obter as relações apenas quando são precisas), o pedido demorou um total de 4,76 segundos e realizaram-se 1429 *queries* à base de dados, como se pode confirmar na Figura 5.3. A partir destes valores conclui-se que existe uma melhoria de 37,82 % no tempo do pedido e 44,58 % de melhoria na quantidade de *queries* à base de dados, quando se faz *Eager Loading* das relações necessárias à criação do ficheiro XML.

```

1 $this->injunctions = $injunctions->load([
2     'client.address',
3     'client.user',
4     'debtor.address',
5     'lawyer.address',
6     'invoices',
7 ]);

```

Código 5.2: *Eager Loading* de todas as relações da entidade Processo necessárias para gerar o ficheiro XML.

Neste caso concreto, o elevado número de *queries*, quando se recorre ao *Lazy Loading*, é um caso típico do anti-padrão de desempenho conhecido por N+1². Este anti-padrão ocorre em relações do tipo **um para muitos**, quando a entidade pai tem de carregar múltiplos filhos. No caso particular do cenário apresentado, existe uma *query* inicial que obtém as 5 injunções e quando as

¹<https://github.com/barryvdh/laravel-debugbar>

²<https://medium.com/geekculture/performance-matters-n-1-problem-a6a072525149> - Publicado a 18/07/2022



Figura 5.3: Quantidade de *queries* efetuadas e tempo total do pedido quando é utilizado *Lazy Loading*

faturas são necessárias, por cada injunção, é feita uma *query* por cada fatura (num total de 250). Ao fazermos *Eager Loading* da primeira *query* para obter as 5 injunções, apenas é feita mais uma *query* para obter todas faturas das 5 injunções.

É importante salientar que os testes desenvolvidos foram realizados em ambiente de desenvolvimento local, com a *framework* em modo de desenvolvimento local, onde não existe otimização ao nível de *cache* e existem processos no próprio Laravel que em modo de produção não acontecem. Isto significa que, de forma geral, a velocidade do pedido num servidor, em modo de produção, irá traduzir-se em melhorias substanciais em termos de tempo dos pedidos. No entanto, o número de *queries* e a relação entre os dois cenários manter-se-ão.

5.5 Sumário e conclusões

Dadas as escolhas feitas em termos de tecnologias para este trabalho, a validação na perspetiva técnica e automática ficou muito simplificada, uma vez que quer o Laravel quer o Laravel Nova são bem testados antes de serem disponibilizados ao público, o que significa que se apenas forem usadas funcionalidades existentes da forma como estão documentadas não há necessidade de repetir os mesmos testes. No entanto, se essas mesmas funcionalidades forem reescritas ou alteradas de qualquer forma, testes automáticos (unitários ou de integração) devem ser implementados.

Por outro lado, na perspetiva das regras de negócio, a validação tornou-se mais complexa, uma vez que não existem outros sistemas semelhantes que possam servir de comparação e o Citius não possui qualquer tipo de ambiente de teste. Por isso, a validação foi feita essencialmente em sessões de demonstração e discussão de requisitos.

Contudo, mesmo com estas dificuldades foram concluídas todas as *User Stories* de prioridade alta e média à exceção de apenas três. Foi também desenvolvido um validador dentro da aplicação para verificar o ficheiros XML que contém os lotes de injunções a ser submetidos no Citius antes de serem disponibilizados ao utilizador.

Para além do objetivo de cumprir os requisitos, foi também tido em conta a performance e escalabilidade da aplicação ao aplicar técnicas de leitura de dados que evitem problemas de desempenho do tipo N+1 verificando-se melhorias de cerca de 38 % e 45 % em termos do tempo do pedido e quantidade de *queries*.

Capítulo 6

Conclusões e trabalho futuro

6.1 Resultados

Desde o momento em que este trabalho foi proposto e que os primeiros passos de investigação foram dados, o objetivo tornou-se bastante claro: construir uma aplicação *Web* onde credores podem gerir os seus processos de dívida com os seus diferentes devedores e, mais importante ainda, onde possam criar ficheiros de lotes de injunções com o intuito de serem submetidos e aceites no Citius. Para além disto, sabendo que não existem serviços/*software* no mercado português para satisfazer esta necessidade, abordou-se o desenvolvimento desta solução com a perspectiva de uma nova sociedade comercial e não apenas do ponto de vista académico.

Face a isto, e comparando os requisitos definidos na Secção 3.2 com o que foi concretizado e validado, conclui-se que o MVP foi efetivamente conseguido, com a concretização de praticamente todas as *User stories* de prioridade alta e média como evidenciado na Secção 5.2, mas principalmente pelo facto de a aplicação ser capaz de gerar um ficheiro XML com um lote de injunções que é, sem qualquer dúvida, a funcionalidade que mais benefícios dá ao utilizador final e que terá um papel fulcral na venda da aplicação enquanto serviço. Ainda, apesar de estarmos perante um MVP onde a usabilidade da plataforma não era prioridade, chegou-se ao final com uma plataforma que se adapta a dispositivos móveis e com alguns cuidados no que diz respeito a ferramentas de acessibilidade para utilizadores com certos tipos de incapacidade.

Durante o desenvolvimento das funcionalidades mais importantes para o MVP foi tida em consideração a escalabilidade do sistema, em particular a otimização das *queries* à base de dados, aquando a criação do ficheiro XML, uma vez que é a funcionalidade principal e a que trata o maior volume de dados ao mesmo tempo, conseguindo-se otimizações na ordem dos 40 % como comprovado na Secção 5.4. Claro está, que nas restantes funcionalidades da aplicação estes cuidados também foram tidos em conta, doravante com menos ênfase.

6.2 Trabalho futuro

Como qualquer MVP, pretendeu-se no final desta dissertação ter o menor número de funcionalidades que se traduzissem no maior quantidade de benefícios para o utilizador. Além de mais, o *Fluxiopartilha* o mesmo princípio que qualquer *software* moderno: nunca está terminado, isto é, ou está abandonado ou está em constante evolução. Tendo isto em conta, o próximo passo da aplicação será o de implementar as *User stories* de prioridade baixa do Credor (Secção 3.5 e antes de passar para as *User stories* por implementar do Administrador, testar a aplicação em contexto real, o que possivelmente acontecerá através do Dr. Tiago Lopes de Azevedo, que usará a aplicação juntamente de alguns dos clientes que representa permitindo a deteção de erros que até ao momento não foram detetados, principalmente pelo facto do Citius não possuir qualquer tipo de *sandbox*.

Passando esta fase de testes em contexto real e com dados reais, irá proceder-se à correção de erros e posteriormente será dado ênfase às restantes funcionalidades que tornarão a aplicação numa plataforma funcional, à base de subscrição. Essas funcionalidades passam pela integração de métodos de pagamento e faturação, gestão de clientes do serviço, aplicação do manual de imagem da marca *Fluxio* e terminar algumas das traduções para português do conteúdo gerado automaticamente pelo Laravel Nova. Para se efetivar a comercialização da aplicação, tem que se garantir que a mesma é fiável através da elaboração de mais testes unitários e de funcionalidade e também a construção de uma infraestrutura de servidor estável e escalável na *cloud*.

Por fim, assim que a aplicação comece a apresentar sinais de sucesso e rentabilidade no mercado nacional, será sempre tido em conta uma abordagem de melhoria contínua, apostando-se em novas funcionalidades e novos benefícios para o utilizador, a título de exemplo, desdobramento do ator Credor em diferentes sub-atores (Secção 3.2.1), continuar a apostar num maior desempenho da base de dados (analisar padrões de leitura e criar mais índices) e aumentar a acessibilidade a utilizadores com algum tipo de incapacidades.

Referências

- [1] Mike Cohn. *User Stories Applied: For Agile Software Development*. Addison-Wesley Professional, 8 2004.
- [2] Salvador da Costa. *A Injunção e as Conexas Ação e Execução*. Almedina, 8 edição, 2020.
- [3] Pop Dragos-Paul e Altar Adam. Designing an MVC Model for Rapid Web Application Development. Em *Designing an MVC Model for Rapid Web Application Development*, Bucharest, 2013. Procedia Engineering.
- [4] David Graeber. *Debt - The first 5000 years*. Melville House, 2011.
- [5] Hermes Euler. Debt Collection. 2 2018.
- [6] Mohamad Kassab. The Changing Landscape of Requirements Engineering Practices Over The Past Decade. Em *The Changing Landscape of Requirements Engineering Practices Over The Past Decade*, 8 2015.
- [7] Scott Rosenberg. *Dreaming in Code: Two Dozen Programmers, Three Years, 4,732 Bugs, and One ...*. Three Rivers Press, 2008.
- [8] Daniel Lord Smail. Legal plunder : households and debt collection in late Medieval Europe. *Harvard University Press*, páginas 235–244, 2016.
- [9] Xinyu Wang, L Zhao, Ye Wang, e Jie Sun. The Role of Requirements Engineering Practices in Agile Development: An Empirical Study. Em *Communications in Computer and Information Science*, volume 432, páginas 195–209. Springer, Berlin, Heidelberg, 8 2014.

Anexo A

XSD de validação do ficheiro XML das injunções em lote (V1.5)

```
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <!-- 02-12-2008 - Morada min 2 - validação de datas min e max e Versao de validação
   1.5 com 12.000 caracteres -->
3 <xsd:schema xmlns:inj="http://www.tribunaisnet.mj.pt/xml/schemas/dgaj/injun/1.5/
   Injun.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:dgaj="http://www.
   tribunaisnet.mj.pt/xml/schemas/dgaj/general/enums/tribunais/1.1/tribunais.xsd"
   targetNamespace="http://www.tribunaisnet.mj.pt/xml/schemas/dgaj/injun/1.5/Injun
   .xsd" elementFormDefault="qualified" attributeFormDefault="qualified" version="
   1.5">
4 <xsd:import namespace="http://www.tribunaisnet.mj.pt/xml/schemas/dgaj/general/
   enums/tribunais/1.1/tribunais.xsd" schemaLocation="Tribunais.xsd"/>
5 <xsd:element name="loteInjuncoes">
6   <xsd:annotation>
7     <xsd:documentation>Ficheiro de Injunções em Lote</xsd:documentation>
8   </xsd:annotation>
9   <xsd:complexType>
10    <xsd:sequence>
11      <xsd:element name="requerente" maxOccurs="unbounded">
12        <xsd:complexType>
13          <xsd:complexContent>
14            <xsd:extension base="inj:entidade">
15              <xsd:sequence minOccurs="0">
16                <xsd:element name="mandatario">
17                  <xsd:complexType>
18                    <xsd:complexContent>
19                      <xsd:extension base="inj:entidade">
20                        <xsd:attribute name="cedula" use="required">
21                          <xsd:simpleType>
22                            <xsd:restriction base="xsd:string">
23                              <xsd:maxLength value="10"/>
24                            </xsd:restriction>

```



```

69         <xsd:extension base="inj:requerido">
70             <xsd:sequence>
71                 <xsd:choice>
72                     <xsd:element name="notifTribunal">
73                         <xsd:annotation>
74                             <xsd:documentation>A existência deste
                                elemento corresponde à ausência de
                                manifestação do disposto no artº 10º nº2
                                al.) 1 pelo que a notificação será
                                efectuada pelo Tribunal</
                                xsd:documentation>
75                         </xsd:annotation>
76                     </xsd:element>
77                     <xsd:element name="notifMandatario">
78                         <xsd:annotation>
79                             <xsd:documentation>A existência deste
                                elemento corresponde à manifestação do
                                disposto no artº 10º nº2 al.) 1
                                relativamente à notificação por Mandatá
                                rio do Requerente</xsd:documentation>
80                         </xsd:annotation>
81                     </xsd:element>
82                     <xsd:element name="notifSolicitador">
83                         <xsd:annotation>
84                             <xsd:documentation>A existência deste
                                elemento corresponde à manifestação do
                                disposto no artº 10º nº2 al.) 1
                                relativamente à notificação por
                                Solicitador de Execução</
                                xsd:documentation>
85                             </xsd:annotation>
86                         <xsd:complexType>
87                             <xsd:complexContent>
88                                 <xsd:extension base="inj:entidade">
89                                     <xsd:attribute name="cedula" use="
                                                required">
90                                         <xsd:simpleType>
91                                             <xsd:restriction base="xsd:integer">
92                                                 <xsd:maxInclusive value="99999"/>
93                                             </xsd:restriction>
94                                         </xsd:simpleType>
95                                     </xsd:attribute>
96                                 </xsd:extension>
97                             </xsd:complexContent>
98                         </xsd:complexType>
99                     </xsd:element>
100                 </xsd:choice>
101             </xsd:sequence>

```

```

102         <xsd:attribute name="domicilioConvencionado" type="
103             xsd:boolean" use="required"/>
104     </xsd:extension>
105 </xsd:complexContent>
106 </xsd:complexType>
107 </xsd:element>
108 </xsd:sequence>
109 <xsd:attribute name="numSeq" use="required">
110     <xsd:annotation>
111         <xsd:documentation>Nº sequencial da Injunção no ficheiro
112             actual</xsd:documentation>
113     </xsd:annotation>
114     <xsd:simpleType>
115         <xsd:restriction base="xsd:positiveInteger">
116             <xsd:minInclusive value="0001"/>
117             <xsd:maxInclusive value="9998"/>
118         </xsd:restriction>
119     </xsd:simpleType>
120 </xsd:attribute>
121 <xsd:attribute name="nReferencia">
122     <xsd:annotation>
123         <xsd:documentation>referencia opcional indicada pelo
124             utilizador</xsd:documentation>
125     </xsd:annotation>
126     <xsd:simpleType>
127         <xsd:restriction base="xsd:string">
128             <xsd:maxLength value="100"/>
129             <xsd:whiteSpace value="collapse"/>
130         </xsd:restriction>
131     </xsd:simpleType>
132 </xsd:attribute>
133 <xsd:attribute name="capital" type="inj:valorType" use="
134     required"/>
135 <xsd:attribute name="juros" type="inj:valorType"/>
136 <xsd:attribute name="jurosTaxa" type="inj:valorType"/>
137 <xsd:attribute name="jurosData">
138     <xsd:annotation>
139         <xsd:documentation>Juros contados desde...</
140             xsd:documentation>
141     </xsd:annotation>
142     <xsd:simpleType>
143         <xsd:restriction base="xsd:date">
144             <xsd:minExclusive value="1950-12-31"/>
145             <xsd:maxInclusive value="2079-06-06"/>
146         </xsd:restriction>
147     </xsd:simpleType>
148 </xsd:attribute>
149 <xsd:attribute name="outroValor" type="inj:valorType"/>

```

```

145         <xsd:attribute name="taxaJustica" type="inj:valorType" use="
146             required"/>
147         <xsd:attribute name="causaPedir" type="inj:enumCausaPedir" use=
148             "required"/>
149         <xsd:attribute name="contratoData">
150             <xsd:simpleType>
151                 <xsd:restriction base="xsd:date">
152                     <xsd:minExclusive value="1910-12-31"/>
153                     <xsd:maxInclusive value="2079-06-06"/>
154                 </xsd:restriction>
155             </xsd:simpleType>
156         </xsd:attribute>
157         <xsd:attribute name="contratoDataInicio">
158             <xsd:simpleType>
159                 <xsd:restriction base="xsd:date">
160                     <xsd:minExclusive value="1910-12-31"/>
161                     <xsd:maxInclusive value="2079-06-06"/>
162                 </xsd:restriction>
163             </xsd:simpleType>
164         </xsd:attribute>
165         <xsd:attribute name="contratoDataFim">
166             <xsd:simpleType>
167                 <xsd:restriction base="xsd:date">
168                     <xsd:minExclusive value="1950-12-31"/>
169                     <xsd:maxInclusive value="2079-06-06"/>
170                 </xsd:restriction>
171             </xsd:simpleType>
172         </xsd:attribute>
173         <xsd:attribute name="notifEmail" type="xsd:boolean" use="
174             required"/>
175         <xsd:attribute name="tipoTransComercial" type="
176             inj:enumTransComercial" use="required"/>
177         <xsd:attribute name="distribuicao" type="xsd:boolean" use="
178             required">
179             <xsd:annotation>
180                 <xsd:documentation>O conteúdo True corresponde à declaração
181                     do artigo 10º nº 2 alínea j) do regime anexo ao D.L
182                     .269/98 de 1 de Setembro</xsd:documentation>
183             </xsd:annotation>
184         </xsd:attribute>
185         <xsd:attribute name="distribCodTrib" type="dgaj:enumTribunais"
186             use="required">
187             <xsd:annotation>
188                 <xsd:documentation>A existência deste atributo é obrigató
189                     ria e o seu conteúdo deve corresponder a um dos
190                     tribunais enumerados em Tribunais.xsd</
191                     xsd:documentation>
192             </xsd:annotation>
193         </xsd:attribute>

```

```

183         </xsd:complexType>
184     </xsd:element>
185 </xsd:sequence>
186 </xsd:complexType>
187 <xsd:unique name="uniqueNumSeq">
188     <xsd:selector xpath="./inj:injuncao"/>
189     <xsd:field xpath="@inj:numSeq"/>
190 </xsd:unique>
191 </xsd:element>
192 <xsd:element name="resumoTaxas">
193     <xsd:complexType>
194         <xsd:sequence maxOccurs="unbounded">
195             <xsd:element name="taxa">
196                 <xsd:complexType>
197                     <xsd:attribute name="taxaValor" use="required">
198                         <xsd:simpleType>
199                             <xsd:restriction base="inj:valorType"/>
200                         </xsd:simpleType>
201                     </xsd:attribute>
202                     <xsd:attribute name="taxaQuantidade" use="required">
203                         <xsd:simpleType>
204                             <xsd:restriction base="xsd:positiveInteger">
205                                 <xsd:minInclusive value="0001"/>
206                                 <xsd:maxInclusive value="9998"/>
207                             </xsd:restriction>
208                         </xsd:simpleType>
209                     </xsd:attribute>
210                 </xsd:complexType>
211             </xsd:element>
212         </xsd:sequence>
213     </xsd:complexType>
214     <xsd:unique name="uniqueTaxaValor">
215         <xsd:selector xpath="./inj:taxa"/>
216         <xsd:field xpath="@inj:taxaValor"/>
217     </xsd:unique>
218 </xsd:element>
219 </xsd:sequence>
220 </xsd:sequence>
221 <xsd:attribute name="versaoSchema" type="xsd:string" use="required" fixed="
222     1.5"/>
223 <xsd:attribute name="dataFicheiro" use="required">
224     <xsd:simpleType>
225         <xsd:restriction base="xsd:date">
226             <xsd:minExclusive value="1950-12-31"/>
227             <xsd:maxInclusive value="2079-06-06"/>
228         </xsd:restriction>
229     </xsd:simpleType>
230 </xsd:attribute>
231 </xsd:complexType>

```

```
231 </xsd:element>
232 <xsd:complexType name="entidade">
233   <xsd:annotation>
234     <xsd:documentation>Define uma Entidade</xsd:documentation>
235   </xsd:annotation>
236   <xsd:attribute name="nif" type="inj:NIFType" use="required"/>
237   <xsd:attribute name="bi">
238     <xsd:simpleType>
239       <xsd:restriction base="xsd:string">
240         <xsd:minLength value="5"/>
241         <xsd:maxLength value="10"/>
242       </xsd:restriction>
243     </xsd:simpleType>
244   </xsd:attribute>
245   <xsd:attribute name="nome" use="required">
246     <xsd:simpleType>
247       <xsd:restriction base="xsd:string">
248         <xsd:maxLength value="80"/>
249         <xsd:minLength value="3"/>
250       </xsd:restriction>
251     </xsd:simpleType>
252   </xsd:attribute>
253   <xsd:attribute name="morada" use="required">
254     <xsd:simpleType>
255       <xsd:restriction base="xsd:string">
256         <xsd:maxLength value="60"/>
257         <xsd:minLength value="1"/>
258       </xsd:restriction>
259     </xsd:simpleType>
260   </xsd:attribute>
261   <xsd:attribute name="moradaComplemento">
262     <xsd:simpleType>
263       <xsd:restriction base="xsd:string">
264         <xsd:maxLength value="60"/>
265         <xsd:minLength value="1"/>
266       </xsd:restriction>
267     </xsd:simpleType>
268   </xsd:attribute>
269   <xsd:attribute name="cp4" use="required">
270     <xsd:simpleType>
271       <xsd:restriction base="xsd:positiveInteger">
272         <xsd:minInclusive value="1000"/>
273         <xsd:maxInclusive value="9999"/>
274       </xsd:restriction>
275     </xsd:simpleType>
276   </xsd:attribute>
277   <xsd:attribute name="cp3" use="required">
278     <xsd:simpleType>
279       <xsd:restriction base="xsd:integer">
```

```

280         <xsd:minInclusive value="000"/>
281         <xsd:maxInclusive value="999"/>
282         <xsd:pattern value="[0-9]{3}"/>
283     </xsd:restriction>
284 </xsd:simpleType>
285 </xsd:attribute>
286 <xsd:attribute name="cpLocalidade" use="required">
287     <xsd:simpleType>
288         <xsd:restriction base="xsd:string">
289             <xsd:maxLength value="60"/>
290             <xsd:minLength value="1"/>
291         </xsd:restriction>
292     </xsd:simpleType>
293 </xsd:attribute>
294 <xsd:attribute name="telefone">
295     <xsd:simpleType>
296         <xsd:restriction base="xsd:positiveInteger">
297             <xsd:whiteSpace value="collapse"/>
298             <xsd:pattern value="[0-9]{9}"/>
299         </xsd:restriction>
300     </xsd:simpleType>
301 </xsd:attribute>
302 <xsd:attribute name="fax">
303     <xsd:simpleType>
304         <xsd:restriction base="xsd:positiveInteger">
305             <xsd:whiteSpace value="collapse"/>
306             <xsd:pattern value="[0-9]{9}"/>
307         </xsd:restriction>
308     </xsd:simpleType>
309 </xsd:attribute>
310 <xsd:attribute name="email">
311     <xsd:simpleType>
312         <xsd:restriction base="xsd:string">
313             <xsd:maxLength value="80"/>
314             <xsd:minLength value="5"/>
315             <xsd:pattern value="([a-zA-Z0-9_\-\.]+)@
                ((\[[0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. )|
                (([a-zA-Z0-9\-\ ]+\. )+)) ([a-
                zA-Z]{2,4}| [0-9]{1,3}) (\ )?"/>
316         </xsd:restriction>
317     </xsd:simpleType>
318 </xsd:attribute>
319 <xsd:attribute name="nib">
320     <xsd:simpleType>
321         <xsd:restriction base="xsd:string">
322             <xsd:maxLength value="34"/>
323             <xsd:minLength value="15"/>
324         </xsd:restriction>
325     </xsd:simpleType>
326 </xsd:attribute>

```

```
327 </xsd:complexType>
328 <xsd:complexType name="requerido">
329   <xsd:annotation>
330     <xsd:documentation>Define um Requerido</xsd:documentation>
331   </xsd:annotation>
332   <xsd:attribute name="nif" type="inj:NIFType"/>
333   <xsd:attribute name="bi">
334     <xsd:simpleType>
335       <xsd:restriction base="xsd:string">
336         <xsd:minLength value="5"/>
337         <xsd:maxLength value="10"/>
338       </xsd:restriction>
339     </xsd:simpleType>
340   </xsd:attribute>
341   <xsd:attribute name="nome" use="required">
342     <xsd:simpleType>
343       <xsd:restriction base="xsd:string">
344         <xsd:maxLength value="80"/>
345         <xsd:minLength value="3"/>
346       </xsd:restriction>
347     </xsd:simpleType>
348   </xsd:attribute>
349   <xsd:attribute name="morada" use="required">
350     <xsd:simpleType>
351       <xsd:restriction base="xsd:string">
352         <xsd:maxLength value="60"/>
353         <xsd:minLength value="2"/>
354       </xsd:restriction>
355     </xsd:simpleType>
356   </xsd:attribute>
357   <xsd:attribute name="moradaComplemento">
358     <xsd:simpleType>
359       <xsd:restriction base="xsd:string">
360         <xsd:maxLength value="60"/>
361         <xsd:minLength value="1"/>
362       </xsd:restriction>
363     </xsd:simpleType>
364   </xsd:attribute>
365   <xsd:attribute name="cp4" use="required">
366     <xsd:simpleType>
367       <xsd:restriction base="xsd:positiveInteger">
368         <xsd:minInclusive value="1000"/>
369         <xsd:maxInclusive value="9999"/>
370       </xsd:restriction>
371     </xsd:simpleType>
372   </xsd:attribute>
373   <xsd:attribute name="cp3" use="required">
374     <xsd:simpleType>
375       <xsd:restriction base="xsd:integer">
```

```

376         <xsd:minInclusive value="000"/>
377         <xsd:maxInclusive value="999"/>
378         <xsd:pattern value="[0-9]{3}"/>
379     </xsd:restriction>
380 </xsd:simpleType>
381 </xsd:attribute>
382 <xsd:attribute name="cpLocalidade" use="required">
383     <xsd:simpleType>
384         <xsd:restriction base="xsd:string">
385             <xsd:maxLength value="60"/>
386             <xsd:minLength value="1"/>
387         </xsd:restriction>
388     </xsd:simpleType>
389 </xsd:attribute>
390 <xsd:attribute name="telefone">
391     <xsd:simpleType>
392         <xsd:restriction base="xsd:positiveInteger">
393             <xsd:whiteSpace value="collapse"/>
394             <xsd:pattern value="[0-9]{9}"/>
395         </xsd:restriction>
396     </xsd:simpleType>
397 </xsd:attribute>
398 <xsd:attribute name="fax">
399     <xsd:simpleType>
400         <xsd:restriction base="xsd:positiveInteger">
401             <xsd:whiteSpace value="collapse"/>
402             <xsd:pattern value="[0-9]{9}"/>
403         </xsd:restriction>
404     </xsd:simpleType>
405 </xsd:attribute>
406 <xsd:attribute name="email">
407     <xsd:simpleType>
408         <xsd:restriction base="xsd:string">
409             <xsd:maxLength value="80"/>
410             <xsd:minLength value="5"/>
411             <xsd:pattern value="([a-zA-Z0-9_\-\.]+)@
                ((\[[0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. )|
                ((([a-zA-Z0-9\-\ ]+\. )+)) ([a-
                zA-Z]{2,4}|[0-9]{1,3}) (\.?) )"/>
412         </xsd:restriction>
413     </xsd:simpleType>
414 </xsd:attribute>
415 </xsd:complexType>
416 <xsd:simpleType name="NIFType">
417     <xsd:restriction base="xsd:string">
418         <xsd:maxLength value="25"/>
419         <xsd:whiteSpace value="collapse"/>
420         <xsd:minLength value="6"/>
421     </xsd:restriction>
422 </xsd:simpleType>

```

```
423 <xsd:simpleType name="enumCausaPedir">
424   <xsd:annotation>
425     <xsd:documentation>Enumeração das Causa de Pedir</xsd:documentation>
426   </xsd:annotation>
427   <xsd:restriction base="xsd:short">
428     <xsd:enumeration value="1">
429       <xsd:annotation>
430         <xsd:documentation source="Abertura de Crédito"/>
431       </xsd:annotation>
432     </xsd:enumeration>
433     <xsd:enumeration value="2">
434       <xsd:annotation>
435         <xsd:documentation source="Aluguer"/>
436       </xsd:annotation>
437     </xsd:enumeration>
438     <xsd:enumeration value="3">
439       <xsd:annotation>
440         <xsd:documentation source="Aluguer de Longa Duração"/>
441       </xsd:annotation>
442     </xsd:enumeration>
443     <xsd:enumeration value="4">
444       <xsd:annotation>
445         <xsd:documentation source="Arrendamento"/>
446       </xsd:annotation>
447     </xsd:enumeration>
448     <xsd:enumeration value="5">
449       <xsd:annotation>
450         <xsd:documentation source="Compra e Venda"/>
451       </xsd:annotation>
452     </xsd:enumeration>
453     <xsd:enumeration value="6">
454       <xsd:annotation>
455         <xsd:documentation source="Compra e Venda a Prestações"/>
456       </xsd:annotation>
457     </xsd:enumeration>
458     <xsd:enumeration value="7">
459       <xsd:annotation>
460         <xsd:documentation source="Empreitada"/>
461       </xsd:annotation>
462     </xsd:enumeration>
463     <xsd:enumeration value="8">
464       <xsd:annotation>
465         <xsd:documentation source="Financiamento para Aquisições a Crédito"/>
466       </xsd:annotation>
467     </xsd:enumeration>
468     <xsd:enumeration value="9">
469       <xsd:annotation>
470         <xsd:documentation source="Financiamento de Bens ou Serviços"/>
471       </xsd:annotation>
```

```
472     </xsd:enumeration>
473     <xsd:enumeration value="10">
474       <xsd:annotation>
475         <xsd:documentation source="Locação Financeira"/>
476       </xsd:annotation>
477     </xsd:enumeration>
478     <xsd:enumeration value="11">
479       <xsd:annotation>
480         <xsd:documentation source="Mútuo"/>
481       </xsd:annotation>
482     </xsd:enumeration>
483     <xsd:enumeration value="12">
484       <xsd:annotation>
485         <xsd:documentation source="Seguro"/>
486       </xsd:annotation>
487     </xsd:enumeration>
488     <xsd:enumeration value="13">
489       <xsd:annotation>
490         <xsd:documentation source="Utilização de Cartão de Crédito"/>
491       </xsd:annotation>
492     </xsd:enumeration>
493     <xsd:enumeration value="14">
494       <xsd:annotation>
495         <xsd:documentation source="Outra"/>
496       </xsd:annotation>
497     </xsd:enumeration>
498   </xsd:restriction>
499 </xsd:simpleType>
500 <xsd:simpleType name="enumTransComercial">
501   <xsd:annotation>
502     <xsd:documentation>Enumeração da Obrigação emergente de transação comercial
503       abrangida pelo:</xsd:documentation>
504   </xsd:annotation>
505   <xsd:restriction base="xsd:short">
506     <xsd:enumeration value="0">
507       <xsd:annotation>
508         <xsd:documentation source="Sem Transação comercial"/>
509       </xsd:annotation>
510     </xsd:enumeration>
511     <xsd:enumeration value="1">
512       <xsd:annotation>
513         <xsd:documentation source="DL n.º 32/2003, de 17 de fevereiro"/>
514       </xsd:annotation>
515     </xsd:enumeration>
516     <xsd:enumeration value="2">
517       <xsd:annotation>
518         <xsd:documentation source="DL n.º 62/2013, de 10 de maio"/>
519       </xsd:annotation>
520     </xsd:enumeration>
```

```
520     </xsd:restriction>
521 </xsd:simpleType>
522 <xsd:simpleType name="valorType">
523     <xsd:restriction base="xsd:decimal">
524         <xsd:fractionDigits value="2"/>
525         <xsd:maxInclusive value="999999999"/>
526         <xsd:minInclusive value="0.01"/>
527     </xsd:restriction>
528 </xsd:simpleType>
529 </xsd:schema>
```

Anexo B

Ficheiro XML gerado pelo Fluxio com um lote the injunções

```
1      <?xml version="1.0" encoding="utf-8"?>
2      <inj:loteInjuncoes inj:dataFicheiro="2020-01-28" inj:versaoSchema="1.5"
3      xmlns:inj="http://www.tribunaisnet.mj.pt/xml/schemas/dgaj/injun/1.5/Injun.xsd"
4      xmlns:dgaj="http://www.tribunaisnet.mj.pt/xml/schemas/dgaj/general/enums/
5      tribunais/1.1/tribunais.xsd"
6      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
7      xsi:schemaLocation="http://www.tribunaisnet.mj.pt/xml/schemas/dgaj/injun/1.5/
8      Injun.xsd"><inj:requerente inj:nif="532967974" inj:nib="
9      PT40961905291785785277480" inj:nome="Credor" inj:morada="Lg. St. Carlos
10     Esteves, 7" inj:email="credor@credor.pt" inj:cp4="5696" inj:cp3="251"
11     inj:cpLocalidade="Abrantes" inj:telefone="250656537"><inj:mandatario inj:nif=
12     "903407039" inj:cedula="650035" inj:nome="Verónica Henriques de Leal"
13     inj:morada="Rua Cristiano Batista, 71, 61º Esq." inj:cp4="3350" inj:cp3="982"
14     inj:cpLocalidade="Sines" inj:email="gpereira@goncalves.pt" inj:telefone="
15     266419413" inj:fax="045526854"/></inj:requerente><inj:injuncoes><inj:injuncao
16     inj:numSeq="1" inj:nReferencia="1" inj:capital="323.03" inj:juros="46.48"
17     inj:outroValor="281.67" inj:jurosTaxa="4.70" inj:jurosData="2018-05-11"
18     inj:taxaJustica="51" inj:causaPedir="14" inj:contratoData="2022-05-19"
19     inj:tipoTransComercial="1" inj:distribuicao="0" inj:distribCodTrib="2160"
20     inj:contratoDataInicio="2022-05-19" inj:contratoDataFim="2023-09-19"
21     inj:notifEmail="1"><inj:complementoCausaPedir>Quia maxime quis expedita nihil
22     sed. Nulla omnis inventore perspiciatis dolore qui magnam quis. Neque minus
23     omnis eos quam sit ab. Atque voluptatem beatae sit.</
24     inj:complementoCausaPedir><inj:contratoCelebradoComConsumidor
25     inj:ComportaClausulasGerais="1"/><inj:requerido inj:nome="Leandro Rafael
26     Rocha Ferreira" inj:morada="Av. Alexandre Gaspar, 946, 44º Eq." inj:cp4="4314
27     " inj:cp3="582" inj:cpLocalidade="Loures" inj:domicilioConvencionado="0"><
28     inj:notifTribunal/></inj:requerido></inj:injuncao><inj:injuncao inj:numSeq="2
29     " inj:nReferencia="2" inj:capital="664.62" inj:juros="28.01" inj:outroValor="
30     235.69" inj:jurosTaxa="3.70" inj:jurosData="2022-06-18" inj:taxaJustica="51"
31     inj:causaPedir="1" inj:contratoData="2020-03-31" inj:tipoTransComercial="0"
```

```
inj:distribuicao="1" inj:distribCodTrib="2186" inj:contratoDataInicio="
2020-03-31" inj:contratoDataFim="2022-03-31" inj:notifEmail="1"><
inj:complementoCausaPedir>A dolor quisquam facilis. Blanditiis eligendi
soluta est libero dolorum quidem magni. Sit ex quia consequuntur pariat
esse dicta. Mollitia voluptate ut accusamus sed minima magni et.</
inj:complementoCausaPedir><inj:contratoNaoCelebradoComConsumidor/><
inj:requerido inj:nome="João Cardoso de Magalhães" inj:morada="Rua Leite, nº
954" inj:cp4="3995" inj:cp3="880" inj:cpLocalidade="Lagos"
inj:domicilioConvencionado="0"><inj:notifTribunal/></inj:requerido></
inj:injuncao><inj:injuncao inj:numSeq="3" inj:nReferencia="3" inj:capital="
240.92" inj:juros="34.4" inj:outroValor="266.13" inj:jurosTaxa="4.10"
inj:jurosData="2019-02-05" inj:taxaJustica="51" inj:causaPedir="1"
inj:contratoData="2019-06-14" inj:tipoTransComercial="0" inj:distribuicao="1"
inj:distribCodTrib="2057" inj:contratoDataInicio="2019-06-14"
inj:contratoDataFim="2022-04-14" inj:notifEmail="1"><
inj:complementoCausaPedir>Esse amet numquam qui eos. Voluptas eveniet sint
cumque consequatur. Consectetur consectetur et eos animi optio. Iure eos fuga
et non voluptas placeat totam. Porro nihil alias ut.</
inj:complementoCausaPedir><inj:contratoCelebradoComConsumidor
inj:ComportaClausulasGerais="1"/><inj:requerido inj:nome="Alice Ângela Mendes
Brito" inj:morada="Avenida Pinho, 12" inj:cp4="5725" inj:cp3="787"
inj:cpLocalidade="Castelo Branco" inj:domicilioConvencionado="0"><
inj:notifTribunal/></inj:requerido></inj:injuncao><inj:injuncao inj:numSeq="4
" inj:nReferencia="4" inj:capital="543.21" inj:juros="79.11" inj:outroValor="
279.09" inj:jurosTaxa="2.80" inj:jurosData="2018-06-06" inj:taxaJustica="51"
inj:causaPedir="12" inj:contratoData="2019-04-12" inj:tipoTransComercial="0"
inj:distribuicao="1" inj:distribCodTrib="2112" inj:contratoDataInicio="
2019-04-12" inj:contratoDataFim="2021-10-12" inj:notifEmail="1"><
inj:complementoCausaPedir>Sit culpa ullam quisquam aut non id. Alias adipisci
enim natus et aliquid maxime deleniti consectetur. Cumque et dolore
consequatur autem id. Ut vitae at ut consequatur in iusto. Vel libero in
dolores voluptate eligendi. Consequatur tempora reiciendis quia unde
doloremque.</inj:complementoCausaPedir><inj:contratoCelebradoComConsumidor
inj:ComportaClausulasGerais="1"/><inj:requerido inj:nome="Isabela Miranda
Pires" inj:morada="Tv. Neves" inj:cp4="8592" inj:cp3="622" inj:cpLocalidade="
Penafiel" inj:domicilioConvencionado="0"><inj:notifTribunal/></inj:requerido>
</inj:injuncao><inj:injuncao inj:numSeq="5" inj:nReferencia="5" inj:capital="
937.56" inj:juros="79.07" inj:outroValor="140.88" inj:jurosTaxa="1.20"
inj:jurosData="2020-12-26" inj:taxaJustica="51" inj:causaPedir="3"
inj:contratoData="2019-08-12" inj:tipoTransComercial="2" inj:distribuicao="1"
inj:distribCodTrib="2097" inj:contratoDataInicio="2019-08-12"
inj:contratoDataFim="2020-02-12" inj:notifEmail="1"><
inj:complementoCausaPedir>Quam aut facere sit natus eum atque quia. Suscipit
debitis cupiditate enim rerum ipsam. Qui ea sit iusto dolorum earum et optio
eligendi. Sit fuga consectetur et voluptatem.</inj:complementoCausaPedir><
inj:contratoCelebradoComConsumidor inj:ComportaClausulasGerais="1"/><
inj:requerido inj:nome="Nelson Samuel Sá Garcia Amaral" inj:morada="Tv. de Sá
, nº 745, 9º Dr." inj:cp4="8417" inj:cp3="714" inj:cpLocalidade="Vila do
Conde" inj:domicilioConvencionado="0"><inj:notifTribunal/></inj:requerido></
```

```
inj:injuncao><inj:injuncao inj:numSeq="6" inj:nReferencia="6" inj:capital="
662.21" inj:juros="51.68" inj:outroValor="204.07" inj:jurosTaxa="3.40"
inj:jurosData="2020-09-16" inj:taxaJustica="51" inj:causaPedir="14"
inj:contratoData="2019-11-10" inj:tipoTransComercial="0" inj:distribuicao="1"
inj:distribCodTrib="2107" inj:contratoDataInicio="2019-11-10"
inj:contratoDataFim="2022-01-10" inj:notifEmail="1"><
inj:complementoCausaPedir>Aut voluptatibus illum neque qui eum. Illo sint
quia sed. Nihil est quia aut doloremque unde laudantium tenetur. Tempore
dolores officia ipsa rerum rerum officia. In fugit sit necessitatibus iure
maiores error.</inj:complementoCausaPedir><
inj:contratoNaoCelebradoComConsumidor/><inj:requerido inj:nome="Lorena Branco
de Gomes" inj:morada="Av. São. Sérgio" inj:cp4="2464" inj:cp3="077"
inj:cpLocalidade="Vila Real de Santo António" inj:domicilioConvencionado="1">
<inj:notifTribunal/></inj:requerido></inj:injuncao><inj:injuncao inj:numSeq="
7" inj:nReferencia="7" inj:capital="611.70" inj:juros="46.39" inj:outroValor=
"177.45" inj:jurosTaxa="2.60" inj:jurosData="2019-08-03" inj:taxaJustica="51"
inj:causaPedir="7" inj:contratoData="2020-04-17" inj:tipoTransComercial="2"
inj:distribuicao="1" inj:distribCodTrib="2189" inj:contratoDataInicio="
2020-04-17" inj:contratoDataFim="2021-04-17" inj:notifEmail="1"><
inj:complementoCausaPedir>Dolor minima quis et id commodi. Ipsum dolorem
repudiandae quas. Voluptatem eveniet sed rerum ex fuga suscipit vero dolor.
Cumque id aut voluptas consequatur libero voluptates modi. Tempore odit natus
distinctio.</inj:complementoCausaPedir><inj:contratoCelebradoComConsumidor
inj:ComportaClausulasGerais="1"/><inj:requerido inj:nome="Gabriel Vaz de
Faria" inj:morada="Travessa St. Cristiano Garcia" inj:cp4="2518" inj:cp3="659
" inj:cpLocalidade="Cartaxo" inj:domicilioConvencionado="1"><
inj:notifTribunal/></inj:requerido></inj:injuncao><inj:injuncao inj:numSeq="8
" inj:nReferencia="8" inj:capital="788.86" inj:juros="82.75" inj:outroValor="
217.34" inj:jurosTaxa="4.60" inj:jurosData="2020-10-06" inj:taxaJustica="51"
inj:causaPedir="9" inj:contratoData="2019-08-26" inj:tipoTransComercial="1"
inj:distribuicao="0" inj:distribCodTrib="2057" inj:contratoDataInicio="
2019-08-26" inj:contratoDataFim="2021-09-26" inj:notifEmail="1"><
inj:complementoCausaPedir>Autem eveniet aliquam alias soluta est est. Hic vel
pariatur veritatis magni. Et sunt voluptas tenetur id aut illum pariatur
esse. Inventore id labore repellat dicta voluptas. Non omnis debitis ducimus
enim cum repudiandae quaerat.</inj:complementoCausaPedir><
inj:contratoNaoCelebradoComConsumidor/><inj:requerido inj:nome="Luís
Francisco Leite Jesus Sá" inj:morada="R. São. Luciana, nº 7, 68º Dir."
inj:cp4="9592" inj:cp3="598" inj:cpLocalidade="Tondela"
inj:domicilioConvencionado="1"><inj:notifTribunal/></inj:requerido></
inj:injuncao><inj:injuncao inj:numSeq="9" inj:nReferencia="9" inj:capital="
393.35" inj:juros="68.08" inj:outroValor="267.13" inj:jurosTaxa="1.90"
inj:jurosData="2017-07-30" inj:taxaJustica="51" inj:causaPedir="6"
inj:contratoData="2022-04-08" inj:tipoTransComercial="0" inj:distribuicao="0"
inj:distribCodTrib="2131" inj:contratoDataInicio="2022-04-08"
inj:contratoDataFim="2024-04-08" inj:notifEmail="1"><
inj:complementoCausaPedir>Nihil consectetur qui ad rerum alias soluta
cupiditate. Vel ut consequatur assumenda omnis quas culpa. Amet sed molestiae
ducimus rem exercitationem animi quibusdam. Excepturi aliquam in sapiente
```

```

tempora.</inj:complementoCausaPedir><inj:contratoCelebradoComConsumidor
inj:ComportaClausulasGerais="1"/><inj:requerido inj:nome="Mara Domingues de
Lima" inj:morada="Tv. St. Joana Silva, nº 6" inj:cp4="3890" inj:cp3="050"
inj:cpLocalidade="Odivelas" inj:domicilioConvencionado="0"><inj:notifTribunal
/></inj:requerido></inj:injuncao><inj:injuncao inj:numSeq="10"
inj:nReferencia="10" inj:capital="703.69" inj:juros="39.7" inj:outroValor="
125.50" inj:jurosTaxa="3.00" inj:jurosData="2022-10-01" inj:taxaJustica="51"
inj:causaPedir="10" inj:contratoData="2020-02-14" inj:tipoTransComercial="2"
inj:distribuicao="0" inj:distribCodTrib="2157" inj:contratoDataInicio="
2020-02-14" inj:contratoDataFim="2021-03-14" inj:notifEmail="1"><
inj:complementoCausaPedir>Esse itaque eaque cum molestias nemo quia. Quo aut
ipsam rerum qui aut. Consequatur nihil temporibus numquam quia possimus sit
pariatur velit. Voluptas quia labore eaque eum. Atque magni et adipisci
voluptatem.</inj:complementoCausaPedir><inj:contratoNaoCelebradoComConsumidor
/><inj:requerido inj:nome="Alice Leticia de Macedo" inj:morada="Lg. São.
Manuel, 6" inj:cp4="3605" inj:cp3="525" inj:cpLocalidade="Vila Franca de Xira
" inj:domicilioConvencionado="1"><inj:notifTribunal/></inj:requerido></
inj:injuncao><inj:injuncao inj:numSeq="11" inj:nReferencia="11" inj:capital="
678.30" inj:juros="19.17" inj:outroValor="252.08" inj:jurosTaxa="3.20"
inj:jurosData="2022-06-11" inj:taxaJustica="51" inj:causaPedir="1"
inj:contratoData="2018-06-22" inj:tipoTransComercial="0" inj:distribuicao="1"
inj:distribCodTrib="2087" inj:contratoDataInicio="2018-06-22"
inj:contratoDataFim="2019-03-22" inj:notifEmail="1"><
inj:complementoCausaPedir>Voluptatem culpa sunt optio soluta quia soluta
reprehenderit. Labore velit voluptatem aperiam qui ipsa ea doloremque. Nemo
nemo impedit aut sed odit nostrum sit. Laboriosam facere harum accusamus
officia quisquam. Sed non quia molestiae cumque qui dignissimos. Sed
consectetur exercitationem ullam veniam earum illum. Et animi aspernatur
voluptatem saepe nostrum vitae consectetur.</inj:complementoCausaPedir><
inj:contratoNaoCelebradoComConsumidor/><inj:requerido inj:nome="Rúben Tomé
Pinheiro Soares" inj:morada="Travessa St. Rodrigo Teixeira, 610" inj:cp4="
8955" inj:cp3="718" inj:cpLocalidade="Espinho" inj:domicilioConvencionado="0"
><inj:notifTribunal/></inj:requerido></inj:injuncao><inj:injuncao inj:numSeq=
"12" inj:nReferencia="12" inj:capital="889.84" inj:juros="51.05"
inj:outroValor="115.83" inj:jurosTaxa="2.50" inj:jurosData="2022-05-05"
inj:taxaJustica="51" inj:causaPedir="4" inj:contratoData="2020-07-24"
inj:tipoTransComercial="1" inj:distribuicao="0" inj:distribCodTrib="2087"
inj:contratoDataInicio="2020-07-24" inj:contratoDataFim="2022-06-24"
inj:notifEmail="1"><inj:complementoCausaPedir>Tempora eum qui repellat
laborum deleniti. Dolor est recusandae ipsa cupiditate doloribus. Molestias
id labore omnis voluptate perspiciatis dicta. Alias quis non tempore
distinctio rem et. Nihil reiciendis placeat sed nesciunt omnis vel.</
inj:complementoCausaPedir><inj:contratoCelebradoComConsumidor
inj:ComportaClausulasGerais="1"/><inj:requerido inj:nome="Pedro Márcio de
Rodrigues" inj:morada="R. St. Bernardo Campos, 427, 90º Dr." inj:cp4="8402"
inj:cp3="302" inj:cpLocalidade="Pinhel" inj:domicilioConvencionado="0"><
inj:notifTribunal/></inj:requerido></inj:injuncao><inj:injuncao inj:numSeq="
13" inj:nReferencia="13" inj:capital="818.31" inj:juros="65.4" inj:outroValor
="215.80" inj:jurosTaxa="4.10" inj:jurosData="2021-03-30" inj:taxaJustica="51

```

```

" inj:causaPedir="7" inj:contratoData="2018-11-19" inj:tipoTransComercial="0"
  inj:distribuicao="0" inj:distribCodTrib="2160" inj:contratoDataInicio="
2018-11-19" inj:contratoDataFim="2021-01-19" inj:notifEmail="1"><
inj:complementoCausaPedir>Dolores nulla qui a eligendi possimus consequuntur.
  Et tempore explicabo ratione consequatur asperiores. Facere tempore
consectetur atque occaecati est. Non sint commodi maxime temporibus aut.
Consectetur aliquid aut voluptatem iste.</inj:complementoCausaPedir><
inj:contratoCelebradoComConsumidor inj:ComportaClausulasGerais="1"/><
inj:requerido inj:nome="Clara Anna de Freitas" inj:morada="Lg. Esteves"
inj:cp4="1026" inj:cp3="955" inj:cpLocalidade="Alcobaça"
inj:domicilioConvencionado="1"><inj:notifTribunal/></inj:requerido></
inj:injuncao><inj:injuncao inj:numSeq="14" inj:nReferencia="14" inj:capital="
722.76" inj:juros="82.76" inj:outroValor="165.14" inj:jurosTaxa="3.30"
inj:jurosData="2018-04-07" inj:taxaJustica="51" inj:causaPedir="6"
inj:contratoData="2017-09-10" inj:tipoTransComercial="2" inj:distribuicao="0"
  inj:distribCodTrib="2107" inj:contratoDataInicio="2017-09-10"
inj:contratoDataFim="2020-04-10" inj:notifEmail="1"><
inj:complementoCausaPedir>Officiis quisquam aut et ipsum molestias sunt optio
. Repellendus est adipisci inventore laboriosam culpa non. Magnam adipisci
molestiae illo sint sit. Itaque dignissimos et aut voluptates nobis voluptas
culpa. Vel aut sit et. Et occaecati voluptas qui rerum molestias dolor
aperiam. Quis magnam vero pariatum tenetur officia ipsa cum.</
inj:complementoCausaPedir><inj:contratoCelebradoComConsumidor
inj:ComportaClausulasGerais="1"/><inj:requerido inj:nome="Catarina Sá"
inj:morada="Rua Lisandro Lima, nº 4" inj:cp4="7607" inj:cp3="349"
inj:cpLocalidade="Aqualva-Cacém" inj:domicilioConvencionado="0"><
inj:notifTribunal/></inj:requerido></inj:injuncao><inj:injuncao inj:numSeq="
15" inj:nReferencia="15" inj:capital="532.07" inj:juros="21.6" inj:outroValor
="225.73" inj:jurosTaxa="1.70" inj:jurosData="2021-01-19" inj:taxaJustica="51
" inj:causaPedir="10" inj:contratoData="2022-06-27" inj:tipoTransComercial="1
" inj:distribuicao="1" inj:distribCodTrib="2127" inj:contratoDataInicio="
2022-06-27" inj:contratoDataFim="2024-12-27" inj:notifEmail="1"><
inj:complementoCausaPedir>Et facere totam numquam sed modi ullam. Quibusdam
dolore possimus illum. Omnis veritatis sint deleniti repellendus. Cupiditate
molestias esse et et sed. Deserunt ipsum officii eos porro iure inventore
totam. Ut voluptates officii ratione ratione voluptatem.</
inj:complementoCausaPedir><inj:contratoCelebradoComConsumidor
inj:ComportaClausulasGerais="1"/><inj:requerido inj:nome="Salomé Eva Vicente
de Carvalho" inj:morada="Rua Rodrigues, 21" inj:cp4="3325" inj:cp3="205"
inj:cpLocalidade="Torres Novas" inj:domicilioConvencionado="1"><
inj:notifTribunal/></inj:requerido></inj:injuncao><inj:injuncao inj:numSeq="
16" inj:nReferencia="16" inj:capital="658.05" inj:juros="16.79"
inj:outroValor="153.83" inj:jurosTaxa="3.90" inj:jurosData="2021-11-28"
inj:taxaJustica="51" inj:causaPedir="12" inj:contratoData="2019-05-30"
inj:tipoTransComercial="2" inj:distribuicao="0" inj:distribCodTrib="2097"
inj:contratoDataInicio="2019-05-30" inj:contratoDataFim="2022-01-30"
inj:notifEmail="1"><inj:complementoCausaPedir>Est similique rerum ut
consequuntur rerum earum voluptates. Vel repudiandae qui corrupti veritatis
alias enim. Provident excepturi numquam minus est. Cum expedita ex culpa amet

```

atque voluptatem. Qui labore ducimus officia velit. Est doloribus quae aut temporibus rerum aspernatur non. Pariatur placeat voluptatem architecto aut. Voluptas voluptas modi nihil pariatur fugiat dolore quae nesciunt.</inj:complementoCausaPedir><inj:contratoNaoCelebradoComConsumidor/><inj:requerido inj:nome="Cristiana Miriam Esteves Neto Lopes" inj:morada="Tv. St. Bárbara Costa, nº 6, 27º Dir." inj:cp4="3661" inj:cp3="285" inj:cpLocalidade="Guarda" inj:domicilioConvencionado="1"><inj:notifTribunal/></inj:requerido></inj:injuncao><inj:injuncao inj:numSeq="17" inj:nReferencia="17" inj:capital="424.79" inj:juros="34.92" inj:outroValor="239.48" inj:jurosTaxa="1.10" inj:jurosData="2019-04-25" inj:taxaJustica="51" inj:causaPedir="10" inj:contratoData="2020-04-07" inj:tipoTransComercial="1" inj:distribuicao="0" inj:distribCodTrib="2160" inj:contratoDataInicio="2020-04-07" inj:contratoDataFim="2021-02-07" inj:notifEmail="1"><inj:complementoCausaPedir>Dignissimos a nesciunt cupiditate quae aut. Voluptate rerum delectus harum voluptatem magnam voluptate odit fugiat. Error quae quasi culpa praesentium tempora dolor perferendis. Commodi minima aut autem harum sapiente vero. Dolorem nobis commodi eum eaque quo. Illo consequatur praesentium adipisci assumenda suscipit. Et in tempore magnam animi.</inj:complementoCausaPedir><inj:contratoNaoCelebradoComConsumidor/><inj:requerido inj:nome="Sandro Mota" inj:morada="R. St. Francisco Campos" inj:cp4="2330" inj:cp3="953" inj:cpLocalidade="Alcácer do Sal" inj:domicilioConvencionado="1"><inj:notifTribunal/></inj:requerido></inj:injuncao><inj:injuncao inj:numSeq="18" inj:nReferencia="18" inj:capital="554.74" inj:juros="12.89" inj:outroValor="235.31" inj:jurosTaxa="2.70" inj:jurosData="2020-01-01" inj:taxaJustica="51" inj:causaPedir="8" inj:contratoData="2017-11-22" inj:tipoTransComercial="0" inj:distribuicao="1" inj:distribCodTrib="2186" inj:contratoDataInicio="2017-11-22" inj:contratoDataFim="2020-09-22" inj:notifEmail="1"><inj:complementoCausaPedir>Nemo sed omnis ducimus modi enim nostrum. Voluptate odit praesentium magnam ut nesciunt neque fugit. Mollitia ad fugit aut omnis unde quaerat. Est qui provident repellat eum et. Eos nam ea laboriosam ab iure porro ipsa. Fugit dolores facere asperiores.</inj:complementoCausaPedir><inj:contratoCelebradoComConsumidor inj:ComportaClausulasGerais="1"/><inj:requerido inj:nome="Cristiana Magalhães" inj:morada="Avenida St. Telmo Sá , nº 51, 1º Esq." inj:cp4="7681" inj:cp3="624" inj:cpLocalidade="Mirandela" inj:domicilioConvencionado="1"><inj:notifTribunal/></inj:requerido></inj:injuncao><inj:injuncao inj:numSeq="19" inj:nReferencia="19" inj:capital="474.29" inj:juros="6.18" inj:outroValor="313.72" inj:jurosTaxa="1.90" inj:jurosData="2022-01-27" inj:taxaJustica="51" inj:causaPedir="4" inj:contratoData="2020-08-31" inj:tipoTransComercial="0" inj:distribuicao="1" inj:distribCodTrib="2013" inj:contratoDataInicio="2020-08-31" inj:contratoDataFim="2022-10-31" inj:notifEmail="1"><inj:complementoCausaPedir>Amet error quia omnis dicta. Velit qui consecetur veritatis qui aliquam. Dolores et consequuntur totam iure voluptatem ipsum est. Explicabo similique quo illo minus explicabo accusamus aut. Autem incidunt sit quod quisquam quia corporis eveniet. Nihil ut aut rerum beatae eos rerum quam.</inj:complementoCausaPedir><inj:contratoNaoCelebradoComConsumidor/><inj:requerido inj:nome="Santiago Jaime de Leal" inj:morada="Travessa Pinho, nº 83" inj:cp4="6012" inj:cp3="034

```
" inj:cpLocalidade="Marinha Grande" inj:domicilioConvencionado="0"><
inj:notifTribunal/></inj:requerido></inj:injuncao><inj:injuncao inj:numSeq="
20" inj:nReferencia="20" inj:capital="837.78" inj:juros="57.19"
inj:outroValor="222.72" inj:jurosTaxa="3.20" inj:jurosData="2019-08-03"
inj:taxaJustica="51" inj:causaPedir="10" inj:contratoData="2019-06-12"
inj:tipoTransComercial="2" inj:distribuicao="1" inj:distribCodTrib="2124"
inj:contratoDataInicio="2019-06-12" inj:contratoDataFim="2021-04-12"
inj:notifEmail="1"><inj:complementoCausaPedir>Nostrum voluptatem praesentium
aperiam itaque. Voluptatum aut quidem nobis beatae magnam autem illo. Et
nihil voluptatibus nulla. Odit aut sint non. Velit harum porro ipsa. Et
consectetur dolorem vel voluptate assumenda aperiam qui. Sed dicta rerum
voluptatem. Qui voluptatibus accusantium excepturi qui fuga.</
inj:complementoCausaPedir><inj:contratoNaoCelebradoComConsumidor/><
inj:requerido inj:nome="Santiago Leonardo Simões de Lopes" inj:morada="Rua St
. Victória Teixeira" inj:cp4="3393" inj:cp3="893" inj:cpLocalidade="Vila
Franca de Xira" inj:domicilioConvencionado="1"><inj:notifTribunal/></
inj:requerido></inj:injuncao></inj:injuncoes><inj:resumoTaxas><inj:taxa
inj:taxaValor="51" inj:taxaQuantidade="20"/></inj:resumoTaxas></
inj:loteInjuncoes>
```

Anexo C

Validador do ficheiro XML que contém o lote de injunções a ser submetido no Citius

```
1 class CitiusInjunctionBatchXmlValidator
2 {
3     protected string $xsdSchema;
4
5     protected XMLReader $handler;
6
7     public int $feedErrors = 0;
8
9     public array $errorDetails;
10
11     public function __construct()
12     {
13         $this->handler = new XMLReader();
14         $this->xsdSchema = config('citius.xsd');
15         $this->errorDetails = [];
16     }
17
18     /** @throws Exception */
19     public function validate($xml): bool
20     {
21         if (!class_exists('XMLReader')) {
22             throw new \DOMException("'XMLReader' class not found!");
23         }
24
25         if (!file_exists($this->xsdSchema)) {
26             throw new Exception('Schema is Missing, Please add schema to feedSchema property');
27         }
28     }
29 }
```

```
28
29     $this->handler->open($xml);
30     $this->handler->setSchema($this->xsdSchema);
31     libxml_use_internal_errors(true);
32
33     while($this->handler->read()) {
34         if (!$this->handler->isValid()) {
35             $this->errorDetails = array_merge($this->errorDetails, $this->
                libxmlDisplayErrors());
36             $this->feedErrors = 1;
37         }
38     }
39
40     return $this->feedErrors === 0;
41 }
42
43 public function displayErrors(): array
44 {
45     return $this->errorDetails;
46 }
47
48 private function libxmlDisplayError($error): string
49 {
50     return Str::remove('http://www.tribunaisnet.mj.pt/xml/schemas/dgaj/injun
        /1.5/Injun.xsd', $error->message);
51 }
52
53 private function libxmlDisplayErrors(): array
54 {
55     $errors = libxml_get_errors();
56     $result = [];
57
58     foreach ($errors as $error) {
59         $result[] = $this->libxmlDisplayError($error);
60     }
61
62     libxml_clear_errors();
63
64     return $result;
65 }
66 }
```