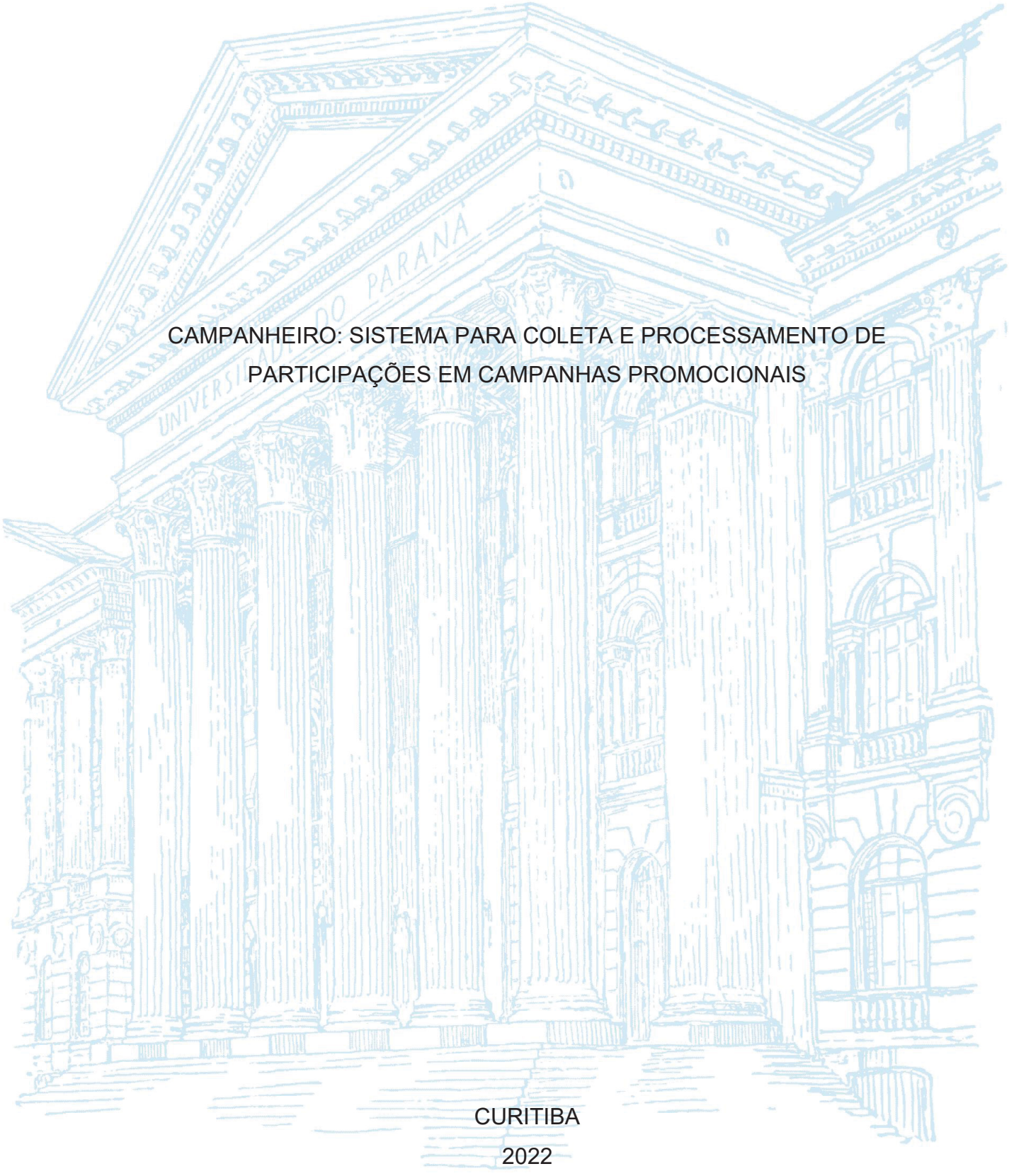
UNIVERSIDADE FEDERAL DO PARANÁ

REGIS ANDRE GABOARDI



CAMPANHEIRO: SISTEMA PARA COLETA E PROCESSAMENTO DE PARTICIPAÇÕES EM CAMPANHAS PROMOCIONAIS

CURITIBA

2022

REGIS ANDRE GABOARDI

CAMPANHEIRO: SISTEMA PARA COLETA E PROCESSAMENTO DE PARTICIPAÇÕES EM CAMPANHAS PROMOCIONAIS

Monografia apresentada ao curso de Pós-Graduação em Engenharia de Software, Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná, como requisito parcial à obtenção do título de Especialista em Engenharia de Software.

Orientadora: Profa. Dra. Rafaela Mantovani Fontana.

CURITIBA

2022

MINISTÉRIO DA EDUCAÇÃO
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
CURSO DE PÓS-GRADUAÇÃO ENGENHARIA DE SOFTWARE - 40001016231E1

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação ENGENHARIA DE SOFTWARE da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **REGIS ANDRE GABOARDI** intitulada: **Campanheiro: Sistema para coleta e processamento de participações em campanhas promocionais**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.
A outorga do título de especialista está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 05 de Abril de 2022.

RAFAELA MANTOVANI FONTANA
Presidente da Banca Examinadora

RAZER ANTHOM NIZER ROJAS MONTAÑO
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

AGRADECIMENTOS

Em primeiro lugar agradeço à Luiza, que plantou, nutriu e cuidou dessa ideia como se fosse dela própria. Também à Lana e Cacau, que afastaram a solidão e nos ajudaram a manter a sanidade em dias de distanciamento.

Agradeço, também, aos professores que compõem o corpo docente da Especialização em Engenharia de Software da Universidade Federal do Paraná, por terem nos recebido em seus lares com a mesma atenção e dedicação, mesmo em tempos desafiadores.

Gostaria de agradecer especialmente à orientadora: Professora Dra. Rafaela Mantovani Fontana, pela cordialidade, ajuda, incentivo e por fazer uma diferença tão positiva no desenvolvimento acadêmico de seus alunos.

It's all extensions of what's coming out of our heads.
You've got to remember that you've got to have it inside your head to be able to get it out at all anyway.
The equipment isn't actually thinking of what to do any of the time; it couldn't control itself.
(David Gilmour, 1972)

RESUMO

Promoções existem em todas as cores, tamanhos e formatos para atrair as intenções de compra de tipos diferentes de consumidores. Elas dão competitividade ao mercado e possibilitam que ele esteja sempre fluindo e evoluindo. Como resultado de suas evoluções, o mercado encontrou novo abrigo em meios virtuais – com o avanço do comércio eletrônico – onde bens de consumo são negociados sem que o consumidor tenha contato direto com o vendedor, abrindo espaço para que novos tipos de promoção sejam explorados, uma vez que o consumidor não pode mais pechinchar por uma melhor condição de negócio. Alguns tipos de promoção são mais ambiciosas e oferecem benefícios que serão concedidos depois da aquisição, pois estão atreladas a uma participação e ao processamento dessa participação para que o prêmio seja garantido ao consumidor. Esse tipo de promoção – sendo chamada de “pós-compra” – está em foco neste trabalho, que propõe um sistema que agiliza tanto o processo de registro e participação, quanto o processamento dos dados dessa participação, para que o consumidor final perceba uma melhor experiência e maior agilidade na avaliação e recebimento do prêmio de sua participação. Neste projeto, está documentado o processo de concepção e desenvolvimento de um sistema que recebe os dados de participação do consumidor e os processa de acordo com as regras definidas pelas campanhas existentes. As campanhas são configuradas no próprio sistema desenvolvido, resultando em uma avaliação automática da participação com a possibilidade de correção de dados caso violações ou inconformidades sejam detectadas. O sistema desenvolvido é um motor de processamento feito em Java e não dispõe de uma interface para o cliente – visto que cada entidade que promove uma campanha pode utilizar seu próprio estilo visual – e usa banco de dados relacional para persistência das informações relativas às campanhas e às participações.

Palavras-chave: Campanha Promocional. Processamento de Participações. Pós-compra. Motor de Processamento. Comércio Eletrônico.

ABSTRACT

Promotions comes in all shapes, colors and sizes to get the buying intentions from different types of consumers, they make the market competitive and allow it to flow and evolve. As a result of the market's evolution, it found new home in virtual areas – as the e-commerce advances – where trade goods are negotiated without the direct contact between consumer and seller, creating a space so that new kinds of promotions can be explored once the consumer can't bargain for a better deal. Some types of promotion are so ambitious that they offer a bonus after the purchase, as they are bound to a participation and the processing of this participation so that the bonus can be awarded to the consumer. This type of promotion is in focus throughout this project – being referred as "pos-purchase" – that propose a system to make both the processes of registration and participation more agile. This project documents the conception and development process of a system that receives data from consumer participation and process these data according to rules defined by existing campaigns that are configured in the very same system. The system itself is a processing engine developed in Java that doesn't offer a client interface – given that each entity that should promote a campaign would use their own interface and visual style – and employ a relational database to persist both campaign and participation information.

Keywords: Promotional Campaign. Participation Processing. Rebate. Pos-Purchase. E-commerce.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE ABREVIATURAS OU SIGLAS

API	- Application Programming Interface
CRUD	- Create, Read, Update, Delete
HTML	- Hyper Text Markup Language
IDE	- Integrated Development Environment
POC	- Prova de Conceitos
PU	- Processo Unificado
SEPT	- Setor de Educação Profissional e Tecnológica
SQL	- Structured Query Language
UFPR	- Universidade Federal do Paraná
UML	- Unified Modeling Language

LISTA DE SÍMBOLOS

© - copyright

@ - arroba

® - marca registrada

Σ - somatório de números

Π - produtório de números

SUMÁRIO

1 INTRODUÇÃO

Muitos tipos de promoção circulam pelo mercado, visando, dentre outros objetivos, o aumento do fluxo de vendas e a fidelização do consumidor. Há uma categoria de promoção que estará em foco neste trabalho: as promoções que têm seu efeito no período pós-compra. O consumidor tende a escolher promoções onde a economia dá-se mais rápido, e se torna mais disposto a participar de promoções pós-compra conforme seu valor (economia) melhora em comparação com a promoção feita em loja (CURRIE, 2014).

Para balancear o valor econômico de uma promoção, de modo que seja atrativa para o consumidor, mas continue sendo rentável como negócio, lojistas e fabricantes precisam fazer um investimento maior nesta modalidade do que fariam no caso de uma promoção de desconto convencional. Segundo Bulkeley (1998), aos olhos de lojistas e fabricantes, as promoções de pós-compra tornam-se rentáveis, pois, apesar de oferecerem um desconto maior do que as promoções convencionais, muitos dos consumidores acabam não requerendo o prêmio oferecido pela promoção; essa ausência de reclamação de prêmios viabiliza promoções desse tipo e possibilita que sejam mais ambiciosas. Outras estratégias que cercam promoções pós-compra incluem a oferta de aparelhos que compõe um combo junto ao produto principal, como *Smartphone* e fones de ouvido, por exemplo.

Os benefícios das promoções pós-compra para os consumidores são a redução do preço para todos os produtos adquiridos. Essas promoções também influenciam os consumidores a comprar mais cedo que o esperado (21%), comprar produtos mais caros do que comprariam sem a promoção (17%) e de marcas que eles não comprariam sem a promoção (9%) (CURRIE, 2014).

Os consumidores, por sua vez, demonstram uma linha de aprendizado sobre promoções pós-compra, fazendo com que fabricantes e lojistas fiquem limitados a oferecer benefícios em longo prazo, para que o consumidor fique preso em uma rede de promoções deste tipo (CHOI, KISSAN, LIMIEUX, 2010). É o caso de promoções do tipo *cashback,* que consistem em oferecer parte do valor pago pelo produto de volta ao consumidor, seja em forma de moeda ou como saldo para compras futuras na mesma rede de lojas ou marca.

De acordo com Kabango e Asa (2015) o comercio eletrônico (*e-commerce*) é o verdadeiro motor da nova economia por ser uma fonte de competitividade

econômica notável para os comerciantes e um novo espaço para consumidores. Este novo espaço refere-se à facilidade de encontrar o mesmo produto com diferentes preços e conjuntos de informações e detalhes em diferentes lojas virtuais. Desta forma, o consumidor pode fazer comparações de detalhes e até mesmo estoque de produtos antes de decidir de qual loja fará sua compra.

Passar mais tempo comparando produtos e buscando promoções alternativas na Internet fortalece o próprio *e-commerce*, uma vez que a relação entre tempo gasto na internet e probabilidade de consumo é positiva (MENDES, 2013). Apesar de ser uma força que influencia o mercado, o *e-commerce* representa uma fatia pequena na participação total do varejo no Brasil, mas tem grande potencial de crescimento.

Há, então, uma tendência natural para que o *e-commerce* passe a governar as regras e movimentos dos mercados tanto físicos, quanto eletrônicos. Para Mendes (2013, p. 44) “é inegável que a internet, e o mercado criado por ela, influenciaram e ainda influenciam todos os mercados e empresas que buscam se tornar ou permanecer competitivas.” Assim, essa influência é benéfica ao consumidor, pois este dispõe de informações e maior poder de pesquisa pela internet do que por meios convencionais.

1.1 PROBLEMA

Mesmo com as vantagens econômicas oferecidas pelas promoções pós-compra, essa modalidade de promoção também envolve outro tipo de motivação para obter resultados. De acordo com Tat e Lee (1993), essas promoções utilizam três pilares motivacionais para adesão por parte do consumidor: (1) consciência de preço, (2) percepção do tempo e esforço associados à participação, e (3) a percepção da economia e satisfação resultante da participação.

Tanto os consumidores que optam por aderir, quanto os que evadem, concordam que há muito tempo e esforço relacionado às participações em campanhas dessa modalidade (TAT; LEE, 1993).

> O processo da redenção de prêmios em promoções pós-compra é complicado e demorado. O participante precisa coletar certificados, preencher formulários, enviar os dados, provas da compra e recibos ao centro de processamento antes do encerramento da promoção. Tendo completado todas essas tarefas, o participante deve esperar de quatro a seis semanas para receber o prêmio oferecido pela promoção. Em alguns casos, esse processo é tão difícil e inconveniente que se torna frustrante para o participante. (TAT; LEE, 1993, p. 54-55).

O consumidor usa um mecanismo de ajuste e ancoragem para definir sua participação ou evasão da promoção pós-compra, atrelado à sua motivação inicial para a aquisição do produto promocional. Caso o consumidor esteja inclinado a obter o produto, ancorará sua motivação no cenário de sucesso na participação. Caso contrário, a motivação estará ancorada na má experiência nos processos da participação (SOMAN; GOURVILLE, 2005).

O Campanheiro é um sistema que tem a proposta de centralizar o recebimento e processamento dos dados de participações em campanhas pós-compra *online*, classificadas em variadas modalidades (*cashback*, garantia, presente, etc), com a finalidade de tornar o processo de participação mais simples para o participante e garantir benefícios no formato de prêmios oferecidos pelo promotor da campanha.

1.2 OBJETIVO GERAL

O objetivo do sistema proposto, denominado “Campanheiro”, é oferecer um sistema *back end* que centraliza o gerenciamento, coleta e processamento de dados de promoções pós-compra, para que, com base nas regras e termos de participação de cada campanha, o benefício seja concedido ou negado ao consumidor final.

1.3 OBJETIVOS ESPECÍFICOS

Especificamente, o Campanheiro oferece um conjunto de APIs que:

1. Permitem a criação e manutenção de Marcas, Produtos e Campanhas, que são as entidades principais do sistema;
2. Coletam as informações de Participações associando-as às Campanhas adequadas;

3. Processam as Participações recebidas, as classificando dentre um conjunto de *estados*;
4. Oferecem um canal para que a Participação seja corrigida, de acordo com os erros identificados durante o processamento;
5. Oferecem relatórios para os números das Participações de cada Campanha.

1.4 JUSTIFICATIVA

Em seu estudo, Kabango e Asa (2015) avaliam que o estado do *e-commerce* em países em desenvolvimento revela oportunidades que devem ser aproveitadas pelas organizações, para que estas sobrevivam as consequências da globalização e dos mercados livres.

A redução de custos (para fabricantes e lojistas) é obtida por meio de novos métodos logísticos e com o planejamento detalhado do ciclo de vida dos produtos (MENDES, 2013). É com esta redução de custos em mente que se dá a concepção de um sistema que possa gerenciar promoções – tanto para comerciantes quando para fabricantes – de forma que a disponibilização da campanha promocional para o mercado e o processamento de seus dados e de seus clientes seja viabilizado, oferecendo um produto atrelado a um benefício que venha a agregar valor para o consumidor.

O Campanheiro, visa ser um sistema que viabiliza o oferecimento de campanhas promocionais pós-compra e acelera o processamento de participações para conceder a aprovação ou reprovação destas participações.

Os contratantes podem elaborar campanhas promocionais e oferecer brindes, cupons de desconto, *cashback*, garantia estendida, ou qualquer outro tipo de prêmio que possa ser usado como forma de barganha com o público-alvo de cada produto.

O Campanheiro objetiva sempre ser um facilitador para que a entidade ofertante tenha mais agilidade no processo de coleta e processamento de dados, sem que haja necessidade do desenvolvimento de um novo sistema para cada nova campanha.

No próximo capítulo encontra-se a contextualização sobre campanhas promocionais e sua relação com o consumidor, além da engenharia de software empregada para o desenvolvimento do sistema proposto.

O terceiro capítulo contém detalhes sobre os materiais e métodos utilizados, desde a concepção até o desenvolvimento do sistema, que é apresentado no quarto capítulo com ilustrações das requisições feitas ao servidor do sistema, representando a criação da estrutura da campanha e da criação e processamento de uma participação.

O quinto capítulo explora considerações e conclusões finais, bem como recomendações para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo estão presentes os conceitos que ancoram a teoria por trás do desenvolvimento do sistema proposto. A seguir, estão detalhados estudos sobre o consumidor e seu comportamento, sobre engenharia de software e o processo unificado, bem como uma breve comparação entre o sistema proposto e outros softwares presentes no mercado.

2.1 BASE DE CONHECIMENTO SOBRE O CONSUMIDOR

Promoções e propaganda são onipresentes, estão na televisão, shopping centers, em estradas saturadas por *outdoors* ou pontos de ônibus que exibem promoções, até mesmo latas de lixo trazem mensagens. Definir uma promoção, ou estilo, por sua vez, não é tarefa simples, por não haver uma fórmula ou receita que garanta sucesso. Muitas promoções são mais voltadas ao produto, outras ao marketing, algumas são oferecidas pelo lojista e outras pelo fabricante, mas o que todas têm em comum é que são focadas no usuário final: o consumidor (CURRIE, 2014).

Segundo Geraldo e Mainardes (2017), os maiores objetivos para empresas que buscam obter vantagem competitiva em mercados online são entender e identificar os fatores que afetam a decisão de compra por parte do usuário final. Estudos feitos para identificar esses comportamentos em países emergentes mapeiam características do perfil dos compradores e sua atitude em relação às compras online.

Pilík (2013) considera a segurança, privacidade e desconfiança que precede a adoção de um novo modelo de compras online, e há também a facilidade de uso do varejo online, utilidade, preço e até mesmo a ansiedade como fator determinante (PANDA; SWAR, 2013). Em um estudo que considera dados levantados em pesquisas anteriores, Geraldo e Mainardes (2017) definem quatro fatores que influenciam a intenção de compra por parte do consumidor: loja virtual, conveniência, confiança e promoção.

Por este trabalho sugerir um sistema que oferece o acesso, participação e o cumprimento da promoção, sem contemplar a venda, informações intrínsecas ao produto promocional ou ter relação com o nível de confiança da loja pela qual se dá

a venda (seja virtual ou física), utiliza-se então a quarta hipótese de Geraldo e Mainardes (2017) para verificar como as promoções podem melhorar os resultados para lojistas, fabricantes e para os próprios consumidores. Esta análise propõe que as promoções estão associadas de forma positiva à decisão de compras dos consumidores e os fatores que contribuem para esta conclusão são quatro:

> Os autores sugerem que (1) a adoção de procedimentos de aprimoramento leva a um aumento considerável, superior ao das promoções personalizadas e indiferenciadas; (2) promoções de fidelidade oferecidas aos consumidores são mais rentáveis em lojas online do que nas tradicionais; (3) promoções online personalizadas são mais rentáveis e; (4) para consumidores sensíveis às promoções online, as promoções personalizadas por segmento levam a um aumento considerável do lucro. (GERALDO; MAINARDES, 2017, página 186).

O termo “rebate” é utilizado por Currie (2014) para categorizar as promoções pós-compra. Essas promoções ficam atrás somente dos cupons de desconto entre os consumidores de bens materiais, e para os fabricantes, sua popularidade é atribuída ao fato de permitir que preços sejam alcançados de modo que engajem o mercado (CURRIE, 2014). Essas promoções devem ser registradas e seguir regras bem definidas pelo fabricante, o que faz com que o consumidor seja influenciado na hora da compra, mas caso não siga as etapas para o resgate, gera a margem de lucro que viabiliza a promoção por parte do fabricante. Entre outras vantagens do “rebate” – em comparação com cupons – são que o rebate tem risco mínimo para o consumidor, é fácil de se programar enquanto negócio, exige processamento e burocracia mínimos e estabelece uma relação entre o consumidor e o fabricante (CURRIE, 2014).

2.2 WEB SERVICES

A internet é composta, basicamente, pelo conjunto de mensagens trocadas entre um cliente e um servidor para atender um proposito definido. O REST é um modelo arquitetônico que define condições a serem observadas para se desenhar e construir aplicações distribuídas na internet.

REST é a sigla para o inglês *REpresentational State Transfer*, traduzido para “Transferência do Estado Representacional”, ou, literalmente, “repouso”, pois trata de uma conversação que representa o estado dos recursos transferidos, e não os

recursos em si. O REST tem base em cinco condições, inicialmente descritas por Fielding (2000), a seguir:

Primeiramente, a comunicação dá-se entre um cliente e um servidor e tem a separação de interesses como um interesse básico. Com essa separação, os componentes podem evoluir independentes entre si.

Em segundo lugar, a comunicação não deve apresentar estados, de tal modo que toda requisição por parte do cliente deve conter todos os dados necessários para o seu processamento por parte do servidor. Isso implica na melhoria da visibilidade, confiabilidade e escalabilidade de um sistema, pois os dados de cada requisição são facilmente visualizados, sem que haja a necessidade de rastrear múltiplas requisições para determinar a natureza de uma transação entre cliente e servidor.

A terceira condição prescreve que requisições devem ser rotuladas e retidas quando possível, eliminando futuras requisições recorrentes, uma vez que o cliente é autorizado a lembrar-se de uma resposta emitida pelo servidor anteriormente.

A quarta condição diz respeito à uniformidade entre diferentes componentes do sistema. A implementação é desacoplada do serviço que a provê, encorajando a evolução independente ao passo que reduz a eficiência uma vez que a informação é transmitida em formato uniforme, em vez de ser especializada para cada necessidade.

A última condição descrita por Fielding (2000) define que o sistema deve ter diferentes níveis de hierarquia e que cada nível não tenha visibilidade ao próximo nível com quem interage.

O REST trata da comunicação através de imagens do recurso e não do recurso em si. Alguns recursos podem ser vazios, caracterizando a ausência da entidade, ou a entidade “em branco”. Outros recursos podem ser representados estaticamente, significando que sua representação será sempre a mesma em qualquer momento da execução. A única parte necessariamente estática na representação de um recurso é a semântica do mapeamento, uma vez que é essa semântica que distingue os recursos trafegados entre si (FIELDING, 2000).

Observando as condições descritas por Fielding se constrói uma aplicação amigável à web, como foi posteriormente proposto por Leonard Richardson em seu modelo para níveis de maturidade dos web services.

Richardson sugeriu, em 2008, três níveis para a maturidade da arquitetura aplicada no desenvolvimento de um sistema web adequado ao REST. O primeiro nível de maturidade caracteriza-se pela existência de diferentes recursos para o recebimento de consultas por parte do servidor. O segundo nível considera o primeiro e a utilização dos verbos HTTP adequados para cada consulta (POST, GET, PUT, etc) e deve disponibilizar as operações de criação, leitura, atualização e deleção dos recursos que representam as entidades do negócio (WEBBER; PARASTATIDIS; ROBINSON, 2010).

O terceiro e mais alto nível de maturidade considera o que é proposto no segundo, acompanhado de uma camada de descoberta da própria aplicação por meio do uso do *Hypermedia*, assim o usuário pode conhecer e utilizar todos os recursos disponíveis no servidor, sem ter o conhecimento prévio sobre eles (WEBBER; PARASTATIDIS; ROBINSON, 2010).

Há, ainda, um “nível zero” de maturidade, para aplicações que não se adequam ao primeiro nível sugerido por Richardson.

O sistema proposto e desenvolvido neste projeto está adequado ao segundo nível de maturidade sugerido por Richardson (2008).

2.3 SOFTWARES SEMELHANTES

Entre os sistemas semelhantes ao proposto neste trabalho, há o sistema da alemã Marken Mehrwert[1], que serviu como inspiração para o modelo de processamento desenvolvido neste projeto, e duas promoções promovidas por redes de supermercados paranaenses: A promoção “Selinhos de Desconto”[2] promovida pela rede Muffato, e a “Promoção Super Marcas”[3] promovida pela rede Condor, no Paraná e em algumas cidades de Santa Catarina.

O sistema desenvolvido pela rede de supermercados Condor pode ser considerado uma virtualização de um sistema promocional tradicional, como é praticado pelo Muffato.

Na promoção tradicional, promovida pelo Muffato, o cliente deve colecionar e preencher uma cartela de adesivos e, assim que a cartela estiver completa, o cliente

---

[1] https://www.markenmehrwert.com/start/konsumenten/?language=en
[2] https://selinhosdedescontosincriveis.supermuffato.com.br/
[3] https://www.promocaosupermarcas.com.br/

pode fazer a troca pelo prêmio escolhido. O cliente deve estar cadastrado no sistema da rede de mercados e receberá selinhos adesivos de acordo com o valor de sua compra. Esta promoção visa a fidelização do cliente por meio da oferta de um brinde que poderá ser obtido ao longo do tempo que o cliente faz compras na mesma rede de mercados.

Para participar da promoção dos supermercados da rede Condor, o cliente deve estar cadastrado no sistema da rede e, dentre os produtos adquiridos, deve haver produtos das marcas patrocinadoras da promoção. Os códigos do sorteio são gerados na hora da compra, considerando o valor mínimo e a presença de produtos das marcas apoiadoras, e devem ser cadastrados no site da promoção. Uma vez informados ao sistema, o participante pode, via sorteio, receber um dos prêmios oferecidos. O site da promoção oferece, ainda, códigos de sorteio adicionais, caso o participante informe quais marcas foram adquiridas em cada compra.

Figura 1 - Imagem descritiva da Campanha promocional

Fonte: https://www.promocaosupermarcas.com.br/

A alemã Marken Mehrwert, doravante MMW, possui um sistema para processamento de participações em campanhas promocionais completo. Conta com interface para o usuário, busca de campanhas por marcas e até mesmo a parte logística para o pagamento de prêmios em qualquer tipo de promoção. Por possuir um sistema que abrange todas as etapas promocionais, o usuário pode utilizar das interfaces de cada promoção para fazer seu registro buscando um prêmio específico, e as páginas de cada promoção promovem o "*look and feel*" das marcas

promotoras da campanha, ou seja, buscam imitar o ambiente da marca para que o usuário sinta-se interagindo com a própria marca, e não com a MMW.

A API da MMW, no entanto, é personalizada para cada marca que a contrata. Diferentes marcas optam por enviar os dados de suas participações por meios distintos, dentre eles pode-se citar os seguintes:

1. O envio em lotes pré-processados: quando a marca possui a própria interface para a coleta de dados, mas usa o motor da MMW para fazer o processamento.
2. O envio dos dados diretamente: quando a marca possui a própria interface para a coleta de dados, e essa interface comunica-se diretamente com a API da MMW para a transferência da participação.
3. A coleta feita pela MMW: quando a MMW coleta e processa os dados das participações, produzindo uma nova interface personalizada para a campanha promocional daquele contratante.

Na Tabela 1 é feita uma comparação entre os sistemas oferecidos pela rede de mercados Condor, uma promoção personalizada pela MMW e pelo sistema de APIs proposto neste trabalho.

Quadro 1 - Comparativo de Funcionalidades

	Promoção Super Marcas	MMW	Campanheiro
Interface personalizada	Sim	Sim	Não
Código reutilizável	Sim	Sim	Sim
Tipos de campanha diferentes	Não	Sim	Sim
Requer validação manual	Não	Sim	Sim
Possui conjunto de termos e regras	Sim	Sim	Sim

Fonte: O Autor (2022)

A promoção “Selinhos de Desconto”, em um cenário ilustrativo, poderia se beneficiar ao utilizar um sistema de processamento como é o proposto neste trabalho. A rede de mercados já possui um aplicativo para dispositivos móveis, e a coleção dos selinhos poderia ser gerenciada por este app, que também exibe outros detalhes da promoção, tais como qual o preço em selinhos de cada prêmio e quais combinações de produtos geram selos adicionais quando adquiridas. O aplicativo serviria, nesse caso, como interface para o usuário, e se comunicaria com a API do Campanheiro para o registro e processamento da participação.

3 MATERIAIS E MÉTODOS

Na década de 1960 havia a percepção de que as técnicas de programação já não acompanhavam o amadurecimento de softwares, tanto em tamanho, quanto em complexidade. Profissionais não tinham treinamento formal: aprendiam com a prática e os problemas mais persistentes eram resolvidos com a adição de mais programadores à equipe. Como resultado, os sistemas eram entregues com atraso, não se comportavam como o previsto, não eram adaptáveis e muitos problemas só seriam identificados após a entrega. Com este contexto, o termo “Engenharia de Software” foi concebido, no sentido de que deveria ser possível construir sistemas como se constrói uma casa: começando com uma base teórica e utilizando-se de técnicas de construção comprovadas, como nas outras engenharias (VLIET, 2007).

Nas décadas seguintes uma variedade de técnicas, métodos e notações foram concebidos para viabilizar o desenvolvimento de sistemas de forma profissional. Segundo Sommerville (2011), a Engenharia de Software tem o objetivo de apoiar o desenvolvimento profissional de sistemas, incluindo técnicas de especificação, desenho e evolução. Nenhuma destas técnicas é normalmente considerada para o desenvolvimento amador ou pessoal de um sistema. Em acordo com Sommerville (2011), um sistema profissional é aquele que não será utilizado pelo próprio desenvolvedor, é construído com um propósito específico por um time e será manutenido depois de pronto. A Engenharia de Software é uma diferença importante entre o desenvolvimento profissional e amador de sistemas.

Mesmo sendo desenvolvido individualmente, o sistema proposto por este trabalho emprega conceitos da Engenharia de Software para torná-lo mais próximo de um sistema profissional. Diferentes tipos de sistemas requerem fundamentos diferentes da Engenharia para sua produção, mas alguns fundamentos estão presentes em qualquer tipo de sistema. Tais como o planejamento: a possibilidade de estimar as datas de início e fim de cada fase de desenvolvimento; a confiança de performance: o sistema se comporta como o esperado, é seguro e disponível para uso quando necessário; o entendimento e gerenciamento da especificação e requisitos: o conhecimento necessário para que o sistema seja entregue dentro do prazo e orçamento; e a efetividade no uso de recursos: ferramentas que já existem devem ser reutilizadas em vez de re-desenvolvidas (SOMMERVILLE, 2011).

Desde seu surgimento, a Engenharia de Software ramificou-se em várias modalidades para atender às necessidades dos diferentes tipos de sistemas, e do amadurecimento desses sistemas, que estavam sendo construídos considerando suas práticas. Essa evolução se deu, segundo Sommerville (2011), pois não há sentido em buscar um modelo universal para o desenvolvimento de sistemas, quando estes apresentam-se com diferentes portes e finalidades, e diferentes tipos de software requerem diferentes abordagens.

A primeira modalidade da Engenharia de Software trata do desenvolvimento de sistemas de forma linear, com etapas bem definidas para concepção, documentação, desenvolvimento, validação e evolução (VLIET, 2007). Por ter essa característica de seguir sempre “em frente”, ficou, ainda na década de 1970, conhecida como “cascata”. Com a evolução da base de conhecimentos sobre sistemas e da própria Engenharia, em 2001 foi publicado o Manifesto Ágil para Desenvolvimento de Sistemas; uma série de princípios que fundamentam o desenvolvimento ágil.

3.1 PROCESSO ÁGIL

Para a realização deste trabalho foi utilizado o *Scrum*, um método ágil que, segundo Schwaber e Sutherland (2020, p. 3), “é um *framework* que viabiliza a construção de produtos através de várias práticas e técnicas”. O *Scrum* é um processo de conhecimento empírico, que se utiliza de um formato iterativo e incremental para melhora a previsibilidade e facilita o controle de riscos (SCHWABER; SUTHERLAND, 2020).

Dentre as características do *Scrum*, destaca-se neste trabalho a adaptabilidade: quando o progresso do desenvolvimento é analisado em relação ao objetivo da entrega, e adaptações são feitas para reduzir o desvio do produto desenvolvido em relação ao produto esperado, otimizando assim o valor do próximo *sprint* (SCHWABER; SUTHERLAND, 2020).

3.1.1 Execução dos Sprints

Para o planejamento e execução dos sprints, foi utilizada a técnica do *Planning Poker*, que é um método que consiste em avaliar a complexidade das histórias do usuário relativamente umas com as outras (MAXIMINI, 2018).

O *Planning Poker* é uma técnica empregada em equipes ágeis para se chegar ao consenso sobre o peso ou tamanho de cada história no *sprint*. Neste projeto, foi utilizada a técnica de aproximação - normalmente utilizada em equipe - uma vez que as histórias foram comparadas pelo autor para se definir suas complexidades, relativamente. Dessa forma, as histórias foram pontuadas e agrupadas no cronograma de desenvolvimento, como é exibido no Quadro 1.

Outra métrica importante é a velocidade de desenvolvimento, que segundo Maximini (2018), “representa a agilidade de um time. Ela [a velocidade] deriva da soma das estimativas do *Sprint* e é avaliada em vários *Sprints*.” (2018, p.319). A velocidade é usada para construir planos de entrega e de iteração e possibilitar uma previsão de quais ferramentas serão entregues em quais datas.

As histórias do usuário são baseadas nos casos de uso definidos para o projeto, desta forma, foi possível avaliar a complexidade e quantos *story points* cada história vale para que fosse feita a relação de cada *story point* em horas de trabalho.

Os *sprints* foram divididos conforme a Tabela 2 - Divisão de Sprints para o desenvolvimento.

Quadro 2 - Divisão de Sprints para o desenvolvimento

Título do *sprint*	Conteúdo do *sprint*
1, Prova de Conceito	Criação das estruturas, APIs e suas dependências para o CRUD de Endereços, bem como conexões com *Hibernate* e testes com *JUnit*.
2, Base da Campanha	Criação das estruturas, APIs e suas dependências para o CRUD das entidades relacionadas à Campanha (Marca, Produto e Campanha).
3, Busca e Registro	Criação das estruturas, APIs e suas dependências para a busca de Campanhas e criação de uma Participação em uma Campanha.
4, Dados da Compra	Criação das APIs para a criação e atualização dos dados da Participação antes do primeiro

	processamento.
5, Processamento	Criação das APIs e suas dependências para o usuário possa enviar a imagem da nota fiscal, para que seja feito o processamento de uma Participação.
6, Notificação e Correção	Criação de APIs e dependências para que e-mails sejam enviados ao usuário, seja para notificação do progresso do processamento ou para a solicitação de correção de erros encontrados durante as validações.

Fonte: O Autor (2022)

Os *Sprints* foram planejados para que ocupassem um espaço de tempo de quatro semanas, estimando-se um esforço de duas horas por dia (14 horas por semana). Um *story point* foi avaliado como oito horas de trabalho, em um *Sprint* de cinquenta e quatro horas cabem 7 pontos.

Os *Sprints* foram alocados de tal forma que o próximo *sprint* sempre adiciona funções ao sistema que são dependentes das funções desenvolvidas no *sprint* atual. Ajustes podem ser feitos ao planejamento do projeto em inúmeros momentos e por várias razões. Este projeto teve seu escopo ajustado quando uma ferramenta de login de usuário, inicialmente planejada, foi substituída por outra ferramenta que permite a atualização de uma Participação sem que haja necessidade de se criar uma conta (nos *sprints* 5 e 6).

O cronograma de desenvolvimento segue o que é demonstrado no Quadro 2. O projeto foi criado e desenvolvido ao longo do ano de 2021, finalizado e apresentado no início de 2022.

Quadro 3 - Cronograma de desenvolvimento

Data Início	Funcionalidades
01/02/2021	Configuração Hibernate 5.4.27, Manter País, Manter Estado, Manter Cidade e Manter Endereço.
03/05/2021	Manter Marca, Manter Produto e Manter Campanha, Associar Produtos à Campanha.
31/05/2021	Criar Participação (dados pessoais), Buscar Campanha
05/07/2021	Adicionar Produto à Participação, Adicionar Comprovantes à

	Participação
20/08/2021	Motor de processamento
27/09/2021	Correção de dados da Participação, Comunicação por e-mail
01/11/2021	Encerrar Participação, Extração de relatórios por Campanha

Fonte: O Autor (2022)

O Sprint 1 é uma prova de conceitos para testar as tecnologias escolhidas para a criação da estrutura das entidades relativas ao endereço: País, Estado, Cidade e Endereço.

O Sprint 2 é dedicado à criação da base de uma Campanha (Promoção); adicionando funções para criação da Marca que oferece, e dos Produtos que participam desta Campanha. Também foi adicionada a função de atrelar os Produtos à Campanha.

O Sprint 3 é dedicado à criação de mecanismos de Busca e do Registro (Participação) em uma Campanha.

O Sprint 4 é dedicado à apresentação dos dados da compra, onde o Cliente (Participante) deve informar dados do produto (IMEI, EAN, características, etc.) bem como da compra (número da nota fiscal, loja, etc.) e anexar as provas necessárias.

O Sprint 5 trata do processamento dos dados informados pelo Cliente durante o Registro.

O Sprint 6 é voltado à administração e estatística do sistema e suas Campanhas e à correção de informações por parte do Cliente, caso seu Registro seja indeferido para a Campanha desejada.

3.2 LINGUAGEM DE MODELAGEM UNIFICADA, UML

A UML nasceu a partir da unificação de vários outros métodos de modelagem para sistemas orientados a objeto que evoluíram entre as décadas de 1980 e 1990. A UML agrupou características das diferentes linguagens existentes sob um único modelo, focado em descrever e ilustrar sistemas de software, particularmente sistemas orientados a objetos. Em 1997, a UML substituiu as linguagens e métodos anteriores (FOWLER, 2004).

A UML foi projetada para ser abrangente em seu atendimento às necessidades de diferentes tipos de aplicação. O objetivo da UML é prover uma notação padrão que pode ser utilizada por todos os métodos orientados a objetos de desenvolvimento de sistemas e tem três categorias principais: Modelo Funcional, Modelo de Objetos e Modelo Dinâmico (BRUEGGE e DUTOID, 2009).

Para a concepção e análise do sistema proposto, foi utilizada a técnica do *Sketch* como forma de ilustrar sua estrutura. O *Sketch*, concebido por Martin Fowler, é utilizado para separar o sistema em partes que viabilizam seu desenvolvimento incremental. Neste projeto foram produzidos a partir do *Sketch*, Diagramas de Caso de Uso (Apêndice F), Diagramas de Classes (Apêndice H), Diagramas de Sequência (Apêndice I), que são considerados parte do modelo funcional e de objetos por Bruegge e Dutoid (2009).

Documentos adicionais produzidos durante a análise e o desenvolvimento do sistema proposto estão presentes nos demais apêndices desse documento, são eles: Documento de Visão (Apêndice A), Casos de Uso Negociais (Apêndice B), glossário (Apêndice C), Regras de Negócio (Apêndice D), Diagrama de Classe dos Objetos Relacionais (Apêndice E), Especificação dos Casos de Uso (Apêndice G), Modelo Físico de dados (Apêndice J) e os casos de teste (Apêndice H).

A vantagem de se utilizar o Sketch, ou esboço, é que esta técnica permite visualizar alguns problemas ou falhas no sistema antes que este seja codificado. O esboço permite visualizar e comunicar as ideias que formulam o sistema a ser desenvolvido e identificar alternativas para possíveis situações de risco. O esboço, por ser seletivo, adapta-se às metodologias ágeis, pois permite que a comunicação seja estabelecida sobre o objeto ou fração do sistema que será desenvolvida “imediatamente”, em vez de tratar do sistema como um todo. O esboço pode selecionar partes do sistema e ser utilizado em cerimônias que duram minutos e tratam da estratégia para as próximas horas, ou cerimônias que levam meio, ou o dia todo, e tratam da estratégia para as próximas semanas (FOWLER, 2004).

Dentre as ferramentas e diagramas fornecidos pela UML, durante a concepção do sistema proposto por este trabalho foram utilizados Diagramas de Caso de Uso. Esses diagramas propõem uma visão externa do funcionamento do sistema e descrevem individualmente as funcionalidades do sistema que geram um resultado para quem o usa (BRUEGGE e DUTOID, 2009). O Diagrama de Caso de Uso também ilustra quais ações implicam na execução de ações adicionais, e quais

atores podem acionar cada caso de uso. Esse tipo de diagrama prevê o uso de notações mais detalhadas, mas segundo Fowler (2004), esses detalhes devem ser ignorados durante a leitura do diagrama, pois ele tem a intensão principal de demonstrar como o usuário interage com o sistema de um ponto de vista externo.

Para ilustrar a estrutura do sistema como um todo, são utilizados Diagramas de Classes. Classes são abstrações que definem o comportamento e estrutura padrão de um conjunto de Objetos. Objetos são instâncias de Classes que atribuem valor aos seus atributos e recordam ligações com outros Objetos, durante a execução do sistema, esses Objetos são criados, modificados e destruídos (BRUEGGE e DUTOID, 2009).

Estão presentes também Diagramas de Interação, são diagramas que ilustram a comunicação entre Objetos e formalizam o dinamismo entre essa ligação. Este tipo de diagrama utiliza-se de linhas horizontais que demonstram qual Objeto origina um fluxo ou mensagem e para qual Objeto este fluxo destina-se, e linhas verticais que demonstram a linha do tempo e a ordem em que se dá cada comunicação entre Objetos. Um tipo de Diagrama de Interação é o Diagrama de Sequência, que representam o Ator e o estímulo que iniciou uma determinada cadeia de execução, definindo a ordem, os estímulos passados entre os Objetos envolvidos na ação desencadeada pelo Ator inicialmente, até que esta ação obtenha um resultado (BRUEGGE e DUTOID, 2009) seja internamente, ou produzindo uma resposta ao Ator.

3.3 FERRAMENTAS E TECNOLOGIAS

Para a análise, planejamento e desenvolvimento deste trabalho, foram utilizadas as ferramentas e tecnologias a seguir.

3.3.1 MySQL Workbench

O MySQL[4] é um banco de dados relacional amplamente conhecido e utilizado pelo mercado, escolhido por ser *open source* e de fácil utilização. O MySQL Workbench é uma ferramenta visual para o desenvolvimento de bancos de dados

---

[4] https://www.mysql.com/products/workbench/

que possibilita o design, criação, desenvolvimento, administração e manutenção de bancos de dados relacionais.

O MySQL Workbench é uma IDE para a Linguagem de Consulta Estruturada (SQL) utilizada pelo MySQL e oferece facilidades para gerenciar um banco de dados, disponibilizando uma interface de usuário para a criação e manutenção do esquema, tabelas, *triggers*, etc, e também oferecendo o processo de engenharia reversa, permitindo que os diagramas relativos ao banco de dados sejam gerados com base nas características do banco existente.

3.3.2 Spring Boot

O Spring[5] é um framework de desenvolvimento de software computacional em Java, e dentre suas versões - as mais utilizadas sendo o Spring Boot e o Spring MVC - o Spring Boot foi escolhido para o desenvolvimento deste trabalho.

O Spring Boot é um framework que possibilita a criação de aplicações independentes em Java de forma descomplicada, por ter necessidades mínimas de configurações, sejam relacionadas ao próprio Spring, ou outras bibliotecas.

Este projeto foi desenvolvido utilizando a versão 2.4.2 do Spring Boot.

3.3.3 Hibernate

O Hibernate[6] é uma ferramenta para o mapeamento relacional de entidades para a linguagem de programação Java, que teve sua primeira versão lançada em maio de 2001.

O Hibernate é um *framework* amplamente conhecido e utilizado, pois promove uma API para a persistência em Java, organizando a comunicação entre o sistema e o banco de dados.

Por ser uma ferramenta feita para o Java, o Hibernate reconhece as entidades e relacionamentos definidos na arquitetura do sistema, tais como herança, associação, composição, entre outros, e pode gerar e executar scripts do banco de dados durante a inicialização do sistema.

---

[5] https://spring.io/guides/gs/spring-boot/
[6] https://hibernate.org/

O Hibernate é um framework de integração com banco de dados que promove performance, escalabilidade e confiabilidade, além de ser uma ferramenta muito presente no mercado de TI como um todo. No desenvolvimento desse projeto, a versão 5.4.27.Final foi utilizada.

3.3.4 Postman

Postman[7] é um software para a integração de APIs que simplifica as etapas do ciclo desenvolvimento, compartilhamento e documentação de APIs. O Postman funciona com um Cliente de API, essa é sua característica fundamental, possibilitando a definições de chamadas complexas à uma API via HTTP, REST, SOAP, etc.

Dado que o sistema proposto neste projeto não oferece uma interface ao usuário que deseja manipulá-lo, o Postman torna-se útil, pois é por meio deste software que pode-se ilustrar a interação entre o usuário e o sistema. No Postman, são configuradas as chamadas que o usuário faria caso houvesse uma interface de uso, e todas as informações presentes nesta possível interface são configuradas no Cliente de API do Postman, que, por sua vez, encaminha uma chamada para o sistema, possibilitando assim a visualização do funcionamento do sistema, mesmo que não haja uma interface do usuário.

Além de ser utilizado como Cliente de API, o Postman é utilizado neste projeto como uma ferramenta de testes, pois pode detectar a resposta emitida pela API requisitada e configurar variáveis e avaliações baseadas nesta resposta. Utilizando-se desta possibilidade, alguns testes e validações foram configurados diretamente no Postman, para garantir que as chamadas efetuadas por meio do Cliente de API fossem processadas corretamente pelo sistema.

3.3.5 Astah Community

Para a produção dos diagramas da UML e documentação do sistema nesta área, optou-se pelo uso do software Astah[8] por ser uma ferramenta completa, de

---

[7] https://www.postman.com/
[8] Disponível em: https://astah.net/products/astah-community/

simples utilização e que dispõe de uma versão gratuita para estudantes, denominada Astah Community.

O Astah é voltado para o desenho de diagramas e documentação para sistemas desenvolvidos em Java, dispondo de identificação de Classes e Objetos que são nativos da linguagem. A grande maioria dos Diagramas de Classe, Diagramas de Caso de Uso e Diagramas de Sequência foram desenvolvidos utilizando deste software. Dentre as exceções, está o Diagrama de Entidade e Relacionamento que foi gerado utilizando a ferramenta de Engenharia Reversa, disponibilizada pelo MySQL Workbench.

3.3.6 IntelliJ IDEA

O IntelliJ IDEA[9] é uma IDE desenvolvida em Java e voltada para o desenvolvimento de softwares computacionais. Foi escolhida para este trabalho por ser uma ferramenta poderosa, que dispõe de integração com diferentes servidores de aplicação, bancos de dados e ferramentas de versionamento.

Além de possuir integrações e ser otimizado para o desenvolvimento, o IntelliJ oferece licença para usuários estudantes, que pode ser obtida com um e-mail atrelado à instituição de ensino reconhecida, caso da UFPR. A licença é válida por um ano e o software recebe atualizações constantes, sendo versionado semestralmente.

3.3.7 GitHub

Git é um sistema de controle de distribuição e versionamento gratuito e *open source,* designado ao gerenciamento de arquivos de projetos de qualquer tamanho com agilidade e eficiência.

GitHub[10] é um provedor de hospedagem de repositório para desenvolvimento e controle de versão de software, que utiliza o Git para verificar a integridade dos arquivos alterados e efetuar a combinação de versões diferentes, enquanto também oferece também suas próprias ferramentas. Por ser um provedor

---

[9] https://www.jetbrains.com/idea/
[10] https://github.com/

de hospedagem, o GitHub funciona como uma ferramenta de prevenção de riscos pois, uma vez que o código é versionado e enviado ao repositório online. Isso significa que a versão mais recente, ou ramificação atual, está hospedada no GitHub, e caso haja alguma falha local no ambiente de cada desenvolvedor, as versões mais recentes não são afetadas pois estão hospedadas em um provedor externo a este ambiente.

O Git é amplamente utilizado para facilitar o compartilhamento e controle de versões de sistemas em desenvolvimento. O GitHub foi o provedor escolhido pois já era familiar ao desenvolvedor do projeto.

No próximo capítulo, o sistema construído será apresentado, ilustrando como funcionam as requisições para, e as respostas geradas pelo sistema.

3.3.8 Hardware

O desenvolvimento do sistema e a execução dos testes foram efetuados em uma máquina com as seguintes características de hardware:

- Memoria: 16Gb
- Processador: Intel(R) Core(TM) i5-8400 CPU @ 2.80GHz 2.81 GHz
- Armazenamento: 1Tb
- Vídeo: Radeon RX 570S, 4Gb

4 APRESENTAÇÃO DO SISTEMA

O Campanheiro tem o objetivo de prover a estrutura para o recebimento e processamento de participações em campanhas promocionais de acordo com suas condições e termos. Para isso, oferece o conjunto de APIs para a criação e configuração dos ecossistemas de cada campanha, e APIs para a coleta, processamento e comunicação com os usuários participantes.

O sistema proposto não contempla as interfaces com o usuário (páginas web ou mobile) pois é direcionado à concepção e desenvolvimento de um sistema que possibilita a coleta e processamento automático de dados para diferentes tipos de campanhas promocionais, observando boas práticas e frameworks de back-end atuais. As interfaces com o usuário, por sua vez, são muito mais relativas ao tipo de campanha, ao conjunto de produtos oferecidos e ao seu público-alvo, por isso adicionam menos valor ao projeto desenvolvido em relação à mecânica do processamento. As interações e chamadas às APIs existentes são ilustradas por meio do Postman. Essas chamadas são as mesmas que seriam efetuadas por diferentes tipos de interfaces que podem ser construídas futuramente.

4.1 PROCESSAMENTO

O processamento de uma Participação, a partir de seu recebimento, pode ocorrer inúmeras vezes até que um estado final seja atribuído.

O primeiro estado de uma Participação será obrigatoriamente "*NOT_PROCESSED*". Esse estado indica que uma Participação foi recebida, mas que nenhum processamento foi aplicado aos seus dados.

Após a criação da Participação no banco de dados, o sistema fará o processamento dos dados dessa Participação, o que pode resultar em um dos quatro possíveis estados, que são:

1. "*NO_CAMPAIGN"*, essa Participação não se qualifica para nenhuma das Campanhas válidas e presentes no banco de dados;
2. "*DUPLICATED*", o sistema identificou outra Participação idêntica a essa no banco de dados, sem considerar o estado atual da Participação anterior.

3. “*INVALID*”, essa Participação contém uma ou mais violações identificadas. Quando o sistema identificar apenas uma violação, o estado atribuído será identificador dessa violação: “*BAD_INVOICE*”, por exemplo. Quando mais de uma violação for identificada, a Participação receberá o estado de “*INVALID*” e o detalhamento das violações identificadas será enviado ao participante, por e-mail.
4. “*VALIDATION_QUEUE*”, nenhuma violação foi identificada automaticamente pelo sistema e essa participação pode ser validada manualmente por um operador do sistema.

Uma vez que a Participação é validada pelo operador do sistema, essa Participação pode ser deferida ou indeferida. No caso de indeferimento, seguirá para o estado de “*CORRECTION_QUEUE”*, e o participante será informado das violações identificadas e receberá um código para efetuar as correções solicitadas, para que um novo processamento de sua Participação seja efetuado.

Quando o participante enviar a correção dos dados de sua Participação, o código de correção gerado pelo sistema deixará de ser válido, e a Participação voltará ao estado de “*VALIDATION_QUEUE*”, para nova validação manual. Esse processo pode repetir-se de acordo com as regras definidas pelo Promotor de cada Campanha. Se o Promotor definir um limite de tentativas de correção, a Participação seguirá para o estado “*FINAL_INVALID*”, caso este limite seja alcançado.

Se o operador do sistema não encontrar nenhuma violação durante a sua validação manual da Participação, o próximo estado será “*VALID*”, significando que aquela Participação está deferida para o recebimento do prêmio oferecido pela Campanha.

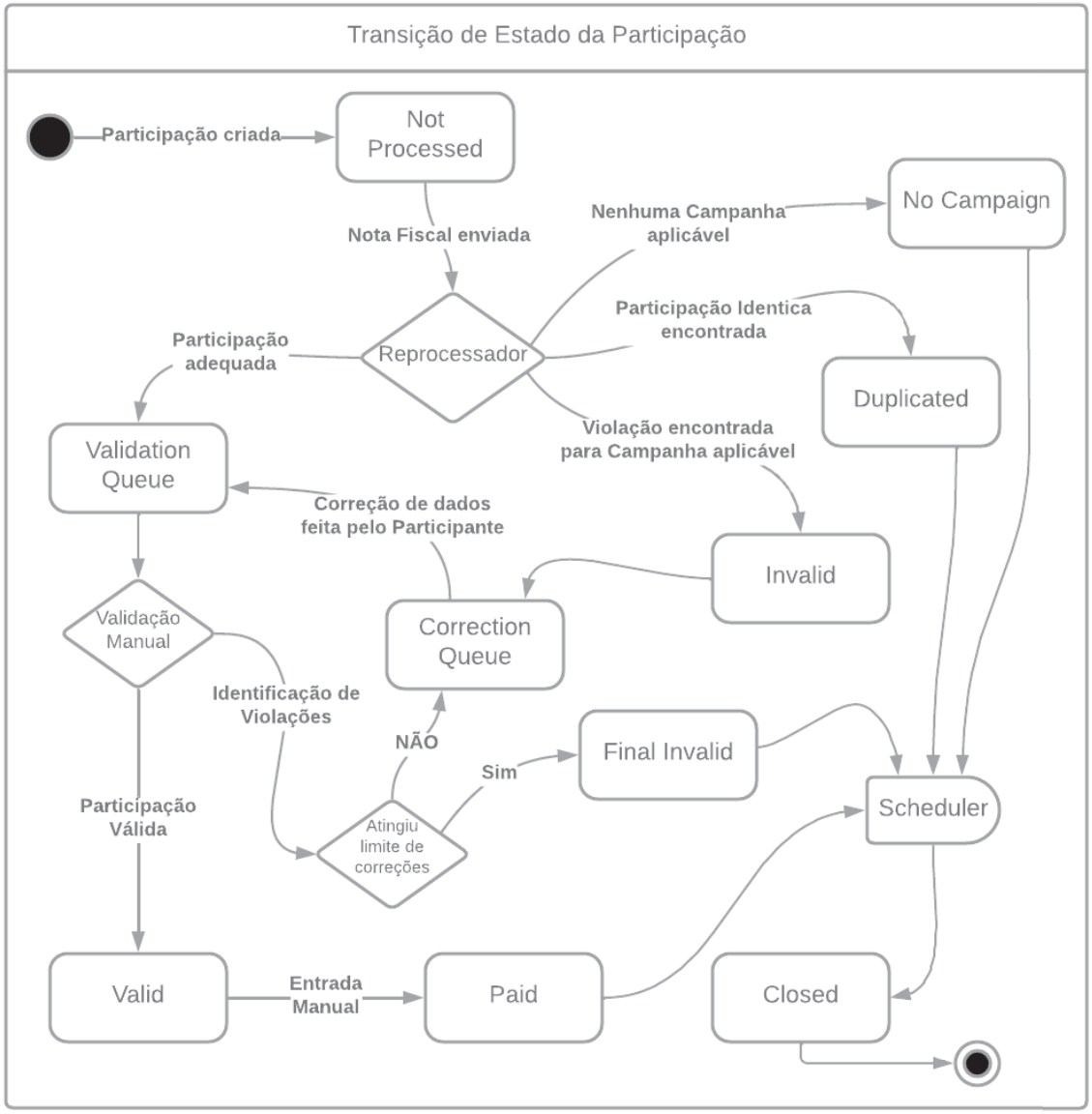
As Participações com estado “*VALID*” seguem para o estado “*PAID”*, quando seu benefício é pago. Esse estado é atualizado manualmente, ou em lotes, de acordo com a logística de pagamento de cada Campanha. Essa logística é externa ao sistema; o pagamento dos prêmios pode ser feito após o encerramento do período de validade da Campanha e do processamento de todas as Participações que se qualificam.

O sistema verificará o banco de dados automaticamente por Participações com o estado “*PAID”* ou “*FINAL_INVALID”* e fará a anonimização dos dados, de modo que seja possível recuperar as estatísticas das Campanhas, mas não seja

possível identificar quem é o participante. Depois dessa atualização, as Participações receberão o estado de "*CLOSED*" e seu ciclo de vida é encerrado.

A Figura 2 - Fluxo de Estado da Participação, exemplifica, resumidamente, os estados que uma Participação pode receber, bem como seu fluxo e sequência.

Figura 2 - Fluxo de Status da Participação



Fonte: O Autor (2022)

As transições de um estado "*PAID*" ou inválido, ou seja, "*NO_CAMPAIGN*", "*DUPLICATED*" e "*FINAL_INVALID",* passam por um processo de agendamento antes que recebam o estado "*CLOSED*", para evitar que a anonimização de dados

seja imediata, assim possibilitando a recuperação e correção manual de dados em caso de falha do sistema, independente da causa.

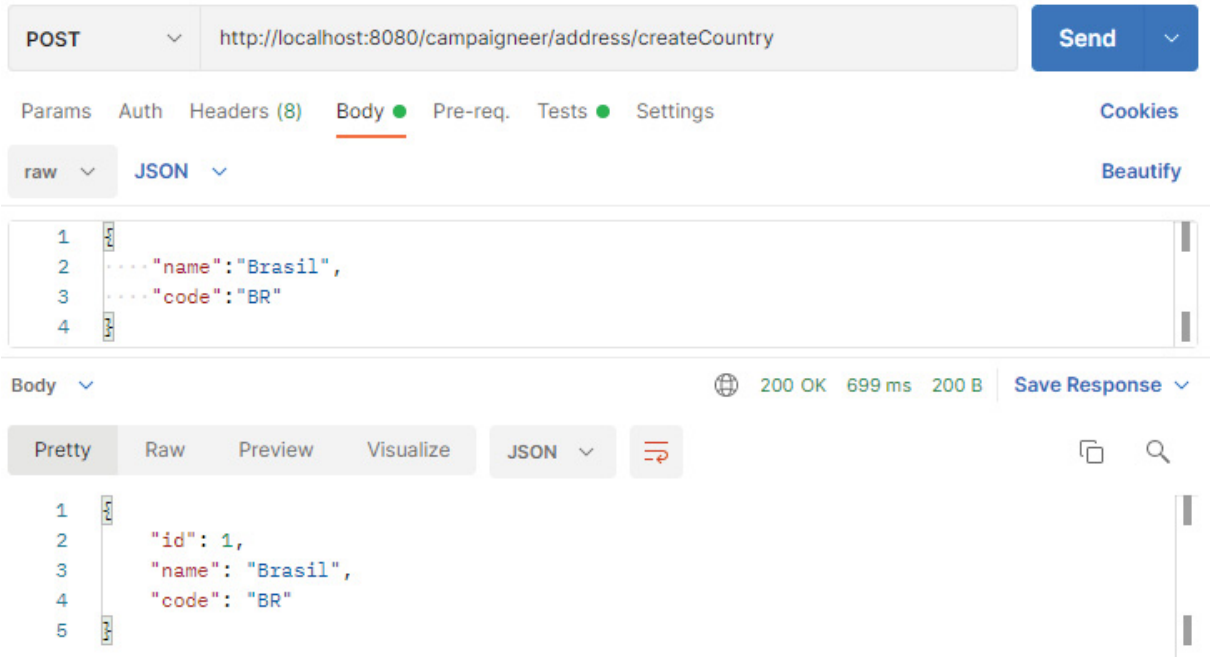
4.2 CRIAÇÃO DA ESTRUTURA

Para que Participações possam ser processadas, primeiro a estrutura de uma Campanha precisa ser criada e configurada no sistema. Essa estrutura consiste em uma Marca promotora, termos e condições.

A Marca promotora deve possuir ao menos o endereço de sua matriz. Os endereços possuem estruturas separadas para País, Estado, Cidade e o Endereço em si, contendo informações como CEP e complemento, por exemplo.

Para a criação das estruturas do Endereço, são utilizadas as APIs ilustradas nas imagens a seguir: Figura 3 - Chamada para API de criação do País; Figura 4 - Chamada para API de criação do Estado; Figura 5 - Chamada para API de criação da Cidade e Figura 6 - Chamada para API de criação do Endereço:

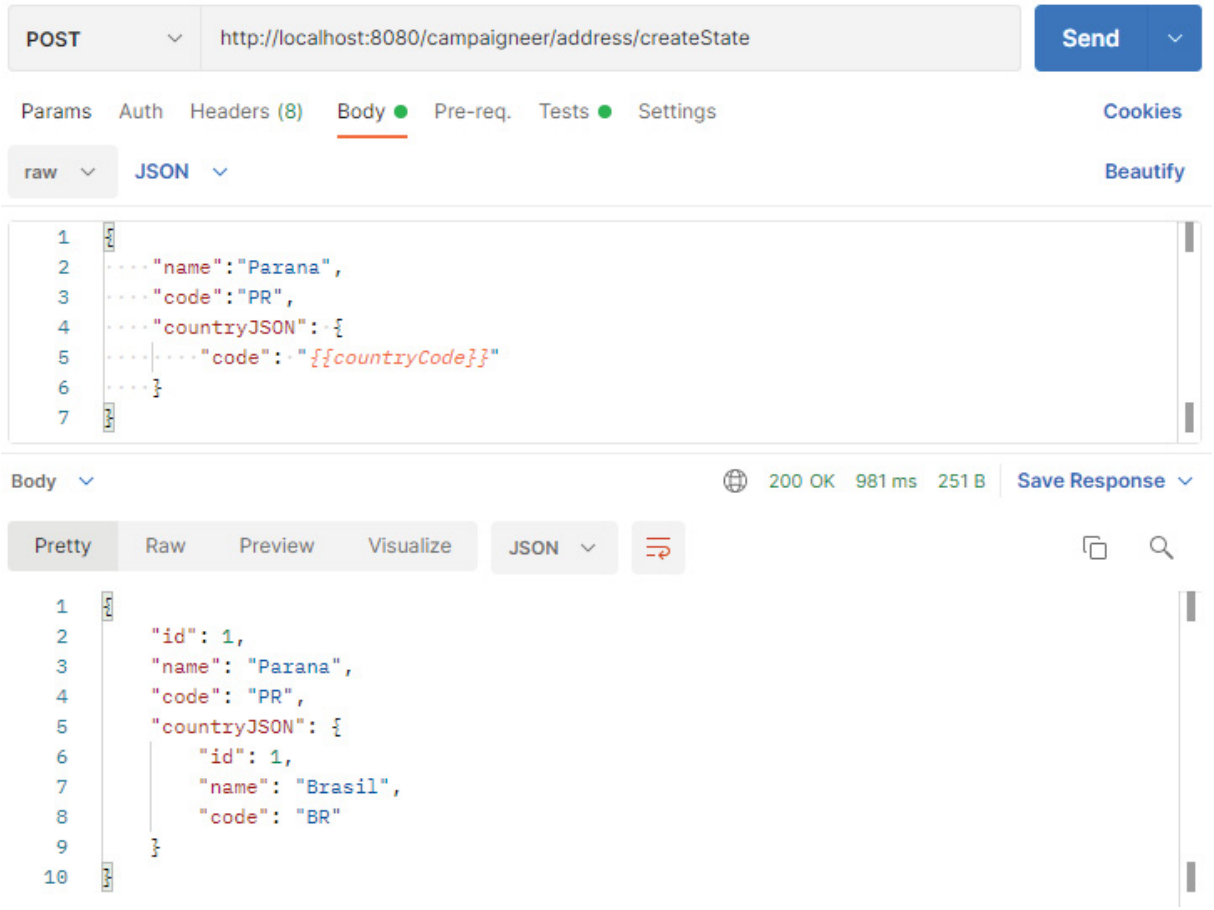
Figura 3 - Chamada para API de criação do País (UC01)



Fonte: O Autor (2022)

As APIs para a criação da estrutura de Endereço estão localizadas em /campaigneer/address/.

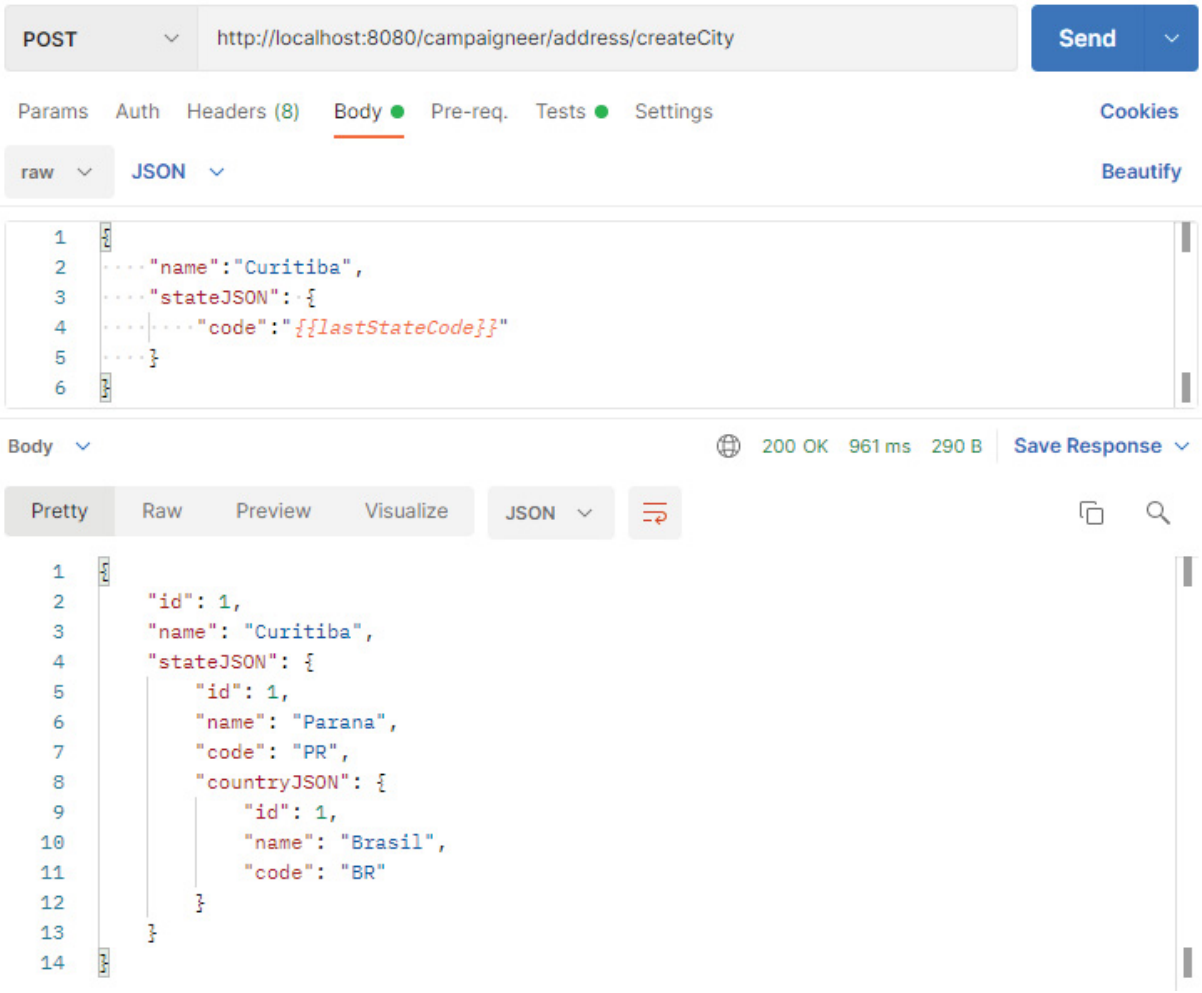
Figura 4 - Chamada para API de criação do Estado (UC05)



Fonte: O Autor (2022)

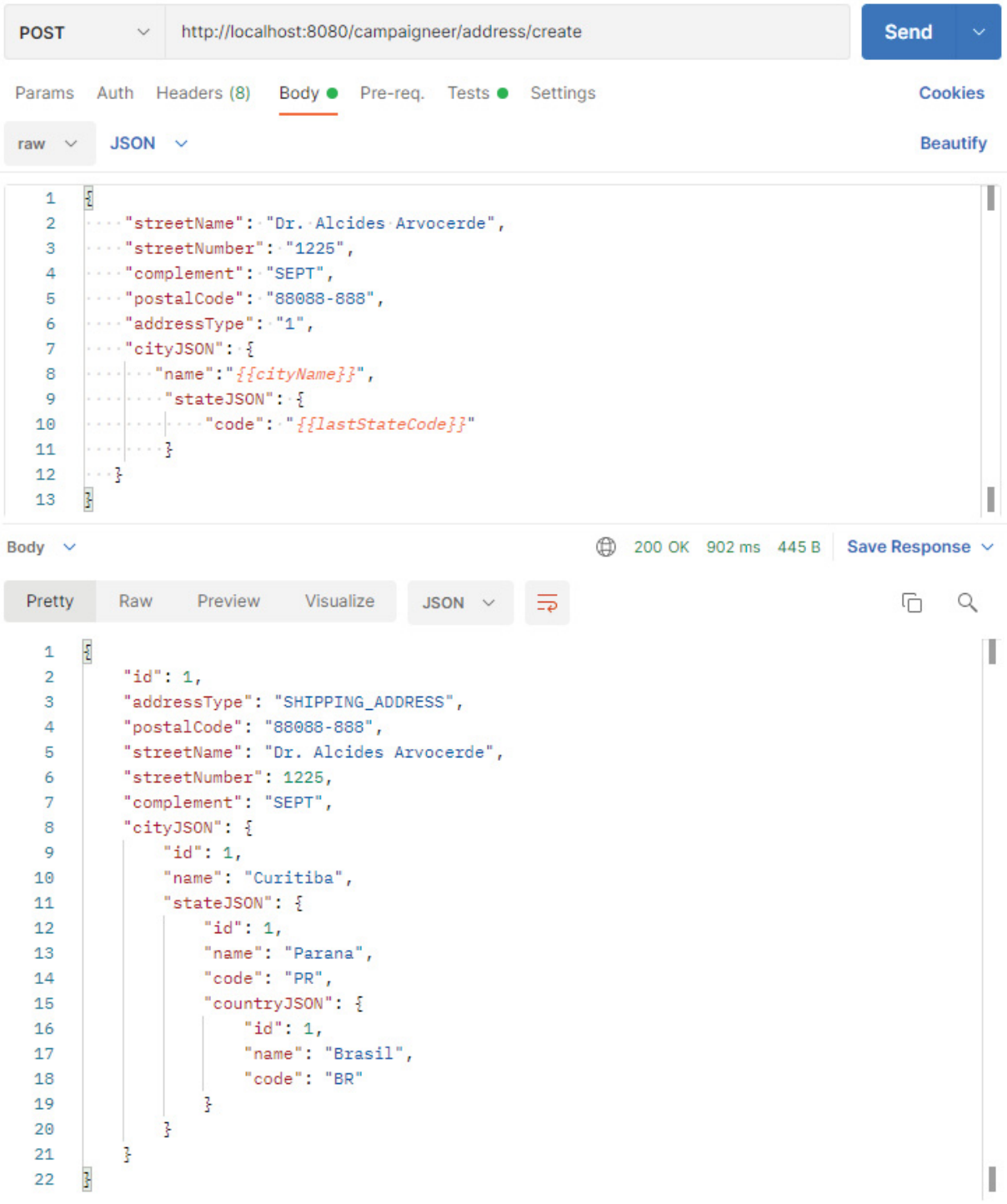
Para a ilustração das chamadas, o recurso de variáveis do Postman é utilizado, o que possibilita que partes da resposta sejam armazenadas em variáveis, para que sejam utilizados em requisições posteriores, como é o caso do *countryCode* na Figura 4.

Figura 5 - Chamada para API de criação da Cidade (UC09)



Fonte: O Autor (2022)

Figura 6 - Chamada para API de criação do Endereço (UC13)

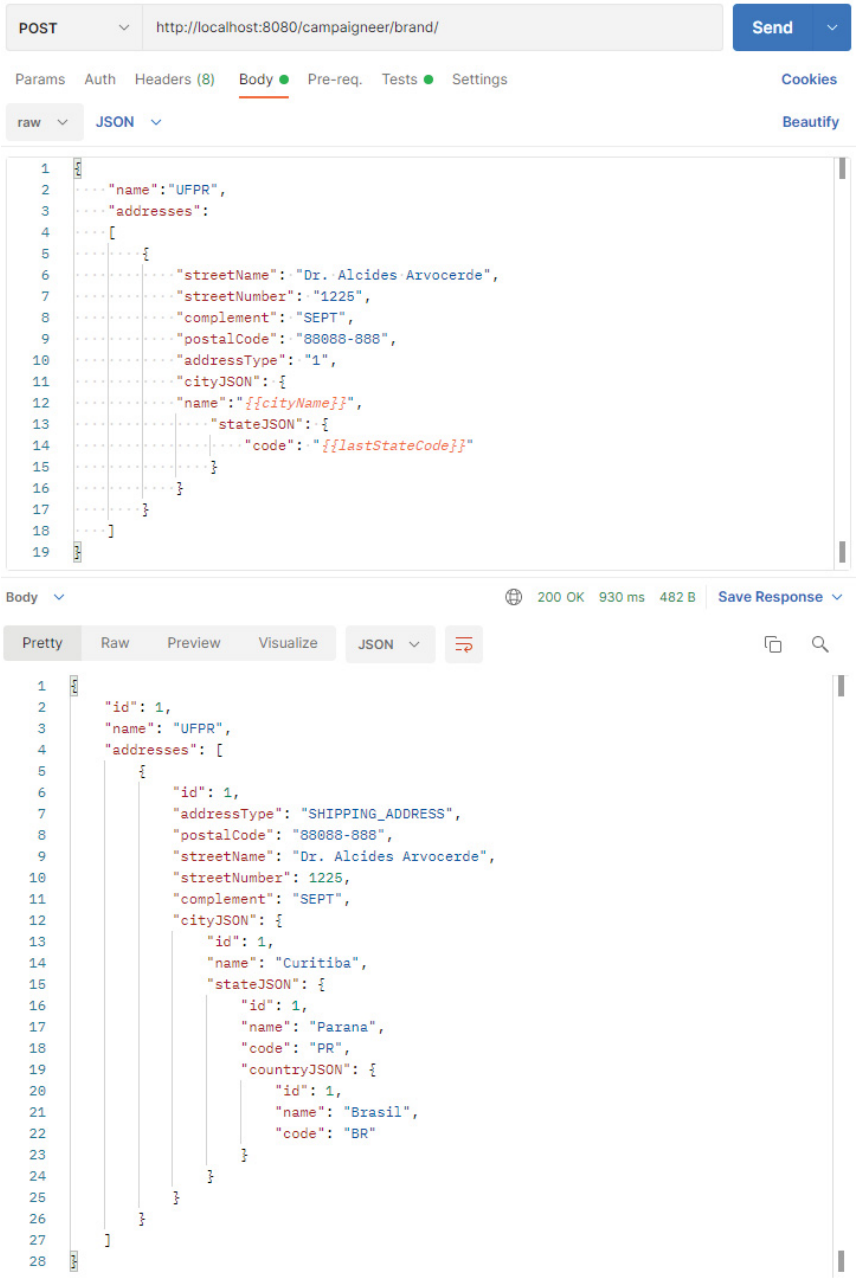


Fonte: O Autor (2022)

Para fins de exemplo, na demonstração das chamadas para o sistema, será criado como empresa a Universidade Federal do Paraná, e, como sua Campanha, o Curso de pós-graduação em Engenharia de Software.

Na Figura 7 - Chamada para API de criação da Marca, a Marca (UFPR) será criada e utilizará como endereço de sua matriz, o endereço criado anteriormente.

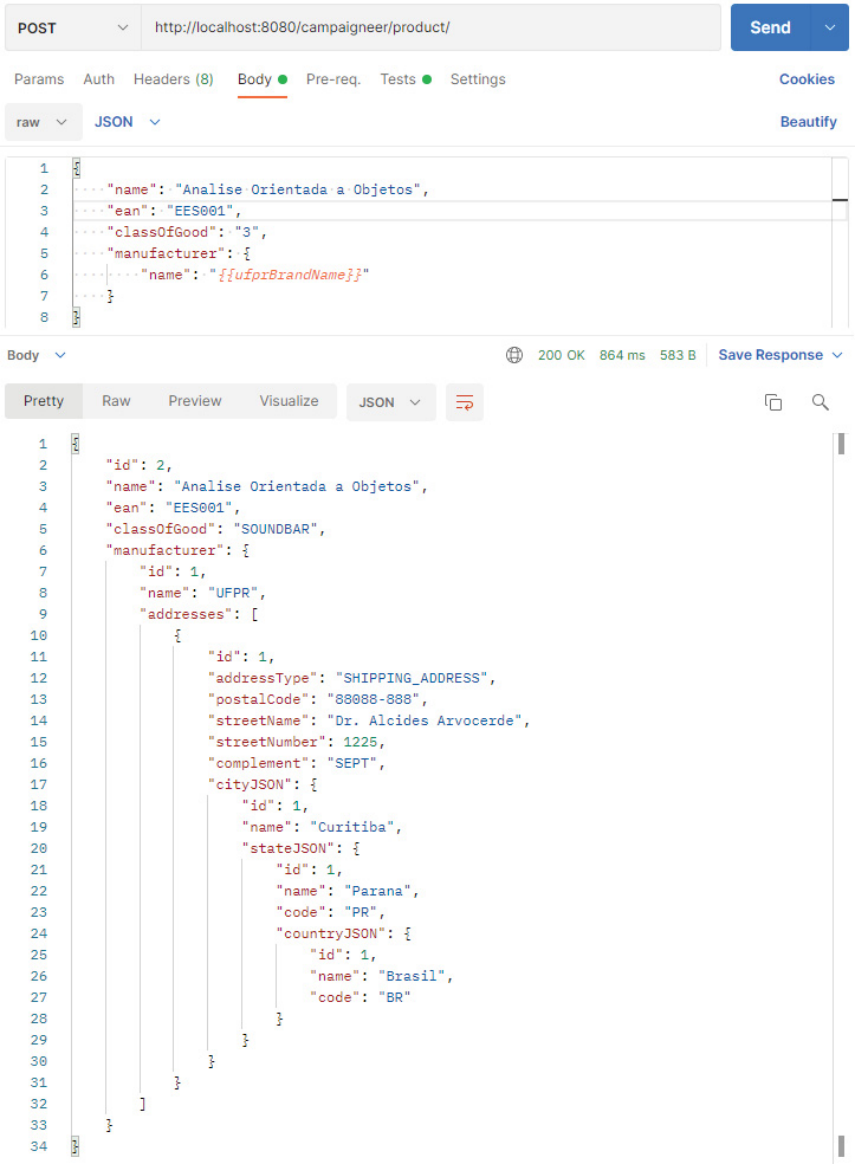
Figura 7 - Chamada para API de criação da Marca (UC17)



Fonte: O Autor (2022)

A Figura 8 - Chamada para API de criação do Produto ilustra a requisição para a criação de um Produto, que para este exemplo, será a turma de 2020 da pós-graduação em Engenharia de Software.

Figura 8 - Chamada para API de criação do Produto (UC21)



Fonte: O Autor (2022)

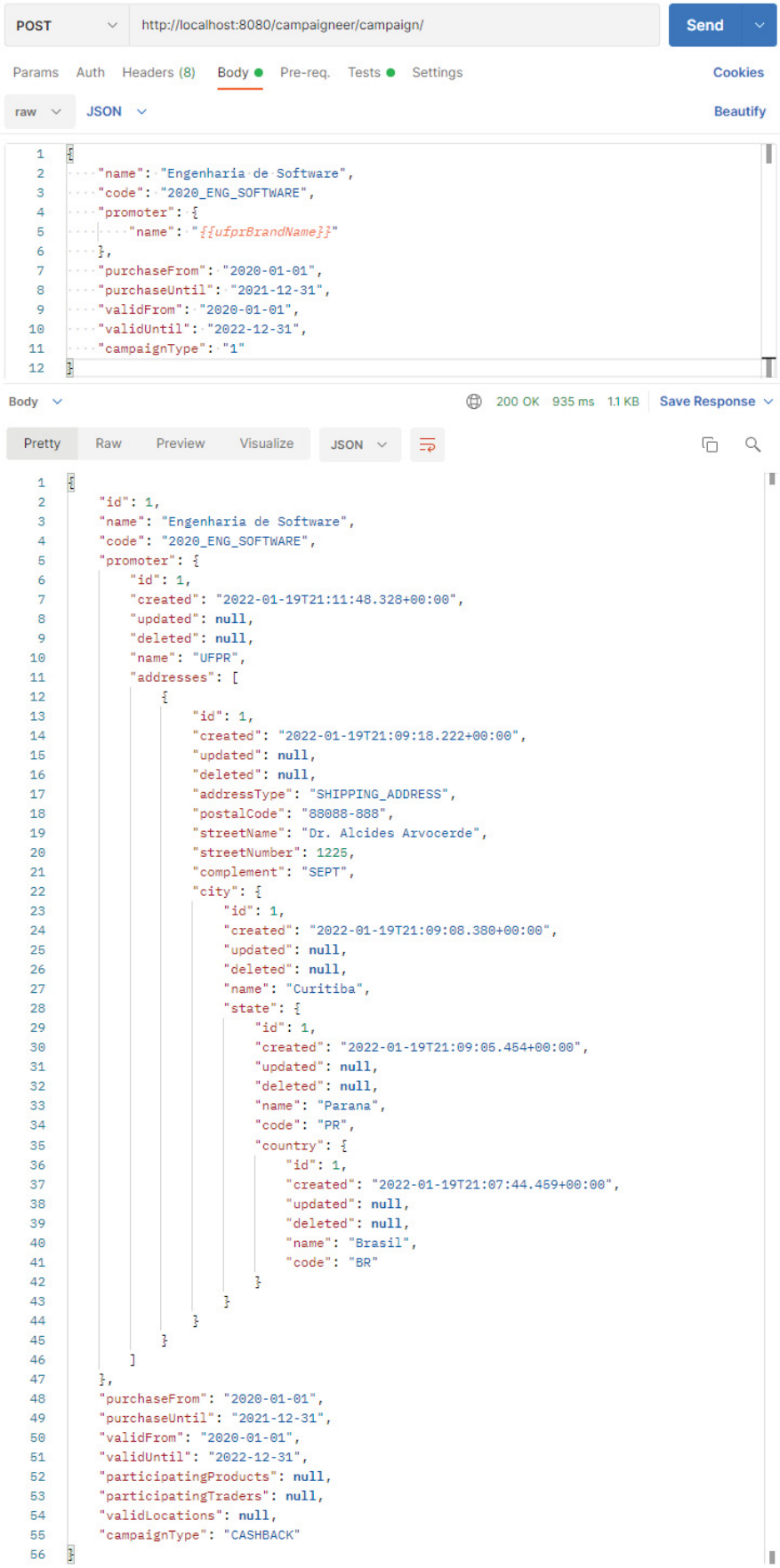
Todas as Campanhas que serão criadas no sistema precisam, no mínimo, da estrutura apresentada até agora. A campanha precisa de uma Marca promotora e de Produtos dessa marca que viabilizam o processamento da participação. A Marca promotora deve possuir um endereço matriz, que é onde a Campanha será ofertada. Caso a Campanha seja promovida em outro território, a Marca deverá ter uma filial presente neste território.

4.3 CRIAÇÃO DA CAMPANHA

Na Figura 9 - Chamada para API de criação da Campanha, é ilustrada a requisição para a criação de uma Campanha. As Campanhas podem ser criadas parcialmente: primeiro são informados as datas e o tipo da campanha, posteriormente os Produtos, termos e condições.

Nesta requisição, são enviadas ao servidor apenas o nome da Campanha, qual é a marca promotora e as datas para participação.

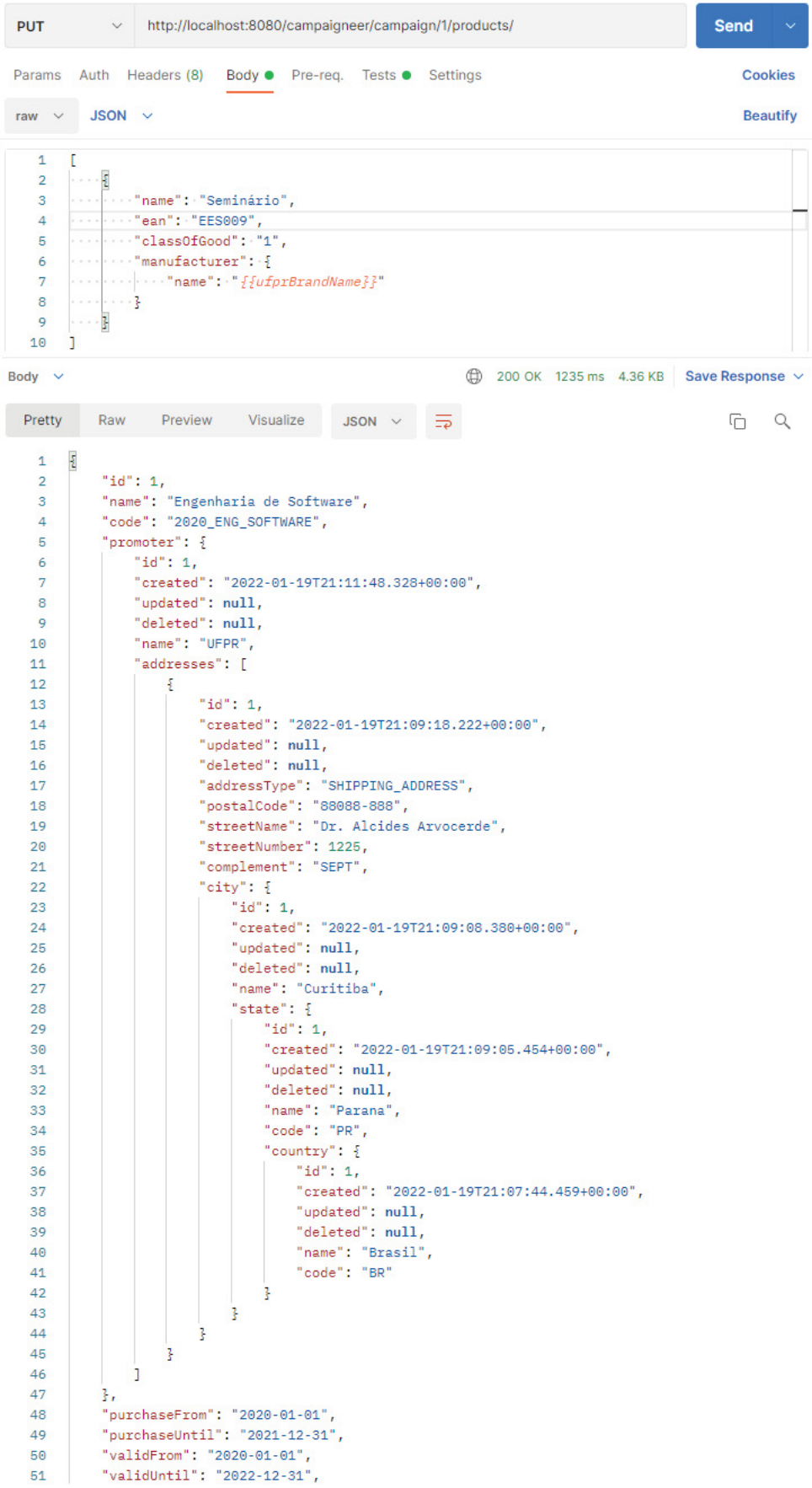
Figura 9 - Chamada para API de criação da Campanha (UC25)



Fonte: O Autor (2022)

Posteriormente pode ser feita a atualização da campanha, adicionando Produtos, termos e condições, documentos, acordos, e outras particularidades. A Figura 10 - Chamada para API que adiciona Produtos à Campanha, ilustra a adição de Produtos à uma Campanha, a API recebe lista de Produtos e os atrela à Campanha selecionada.

Figura 10 - Chamada para API que adiciona Produtos à Campanha parte 1 (UC29)



Fonte: O Autor (2022)

Figura 11 - Chamada para API que adiciona Produtos à Campanha parte 2 (UC29)

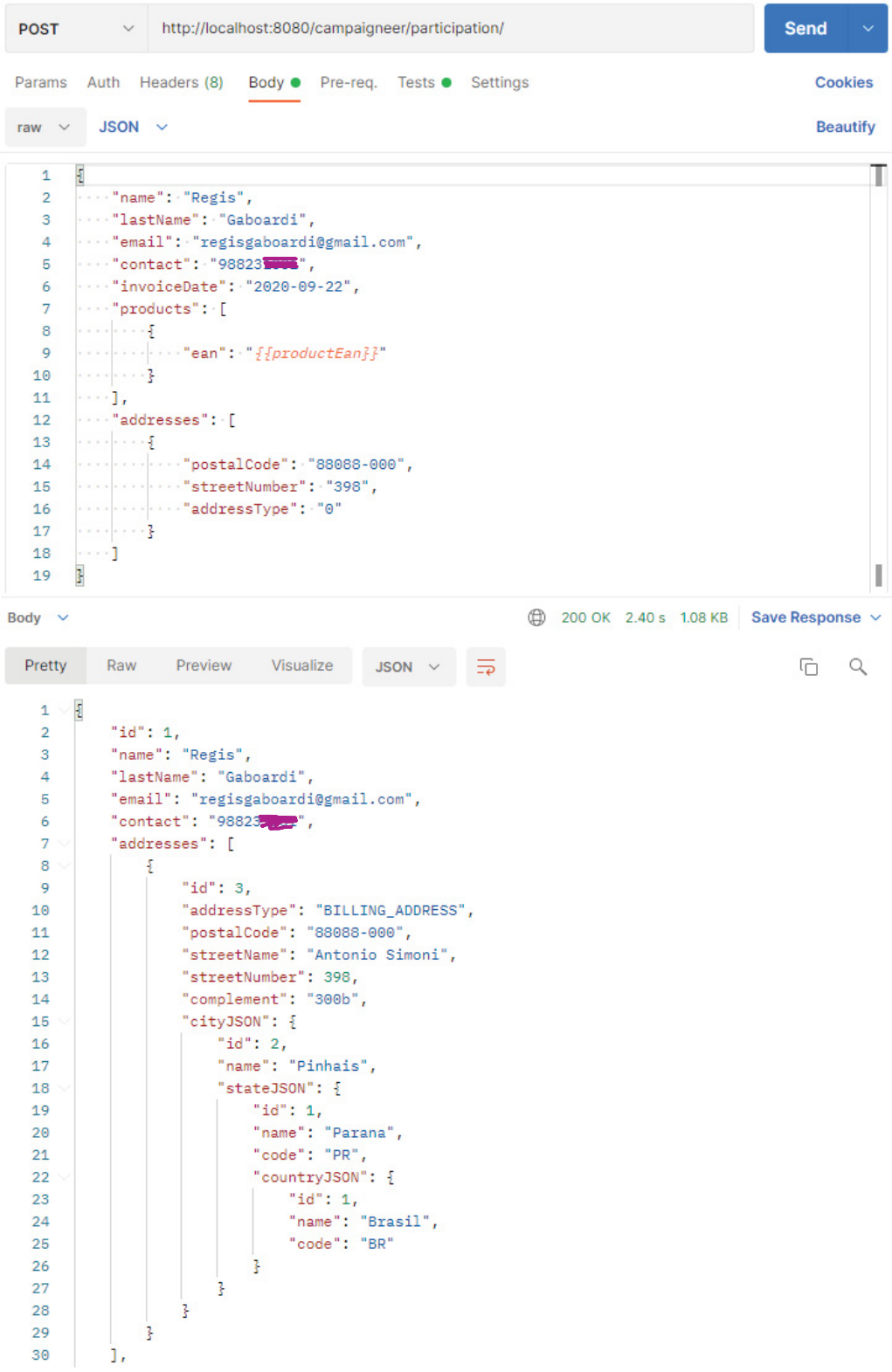

```
    "participatingProducts": [
        {
            "id": 3,
            "created": "2022-01-19T21:18:23.537+00:00",
            "updated": null,
            "deleted": null,
            "name": "Engenharia de Software",
            "ean": "EES002",
            "classOfGood": "SMARTPHONE",
            "manufacturer": {
                "id": 1,
                "created": "2022-01-19T21:11:48.328+00:00",
                "updated": null,
                "deleted": null,
                "name": "UFPR",
                "addresses": [
                    {
                        "id": 1,
                        "created": "2022-01-19T21:09:18.222+00:00",
                        "updated": null,
                        "deleted": null,
                        "addressType": "SHIPPING_ADDRESS",
                        "postalCode": "88088-888",
                        "streetName": "Dr. Alcides Arvocerde",
                        "streetNumber": 1225,
                        "complement": "SEPT",
                        "city": {
                            "id": 1,
                            "created": "2022-01-19T21:09:08.380+00:00",
                            "updated": null,
                            "deleted": null,
                            "name": "Curitiba",
                            "state": {
                                "id": 1,
                                "created": "2022-01-19T21:09:05.454+00:00",
                                "updated": null,
                                "deleted": null,
                                "name": "Parana",
                                "code": "PR",
                                "country": {
                                    "id": 1,
                                    "created": "2022-01-19T21:07:44.459+00:00",
                                    "updated": null,
                                    "deleted": null,
                                    "name": "Brasil",
                                    "code": "BR"
                                }
                            }
                        }
                    }
                ]
            }
        }
    ],
    "participatingTraders": null,
    "validLocations": [],
    "campaignType": "CASHBACK"
}
```


Fonte: O Autor (2022)

4.4 CRIAÇÃO DA PARTICIPAÇÃO

A Participação pode ser criada quando a Campanha “vai ao ar”, ou seja, após a data “validFrom”. Antes dessa data, todos os registros que eventualmente venham a ser qualificados para a campanha serão invalidados no primeiro processamento. A Figura 12 - Chamada para API que cria uma Participação, representa a criação de uma Participação.

Figura 12 - Chamada para API que cria uma Participação parte 1 (UC34)



Fonte: O Autor (2022)

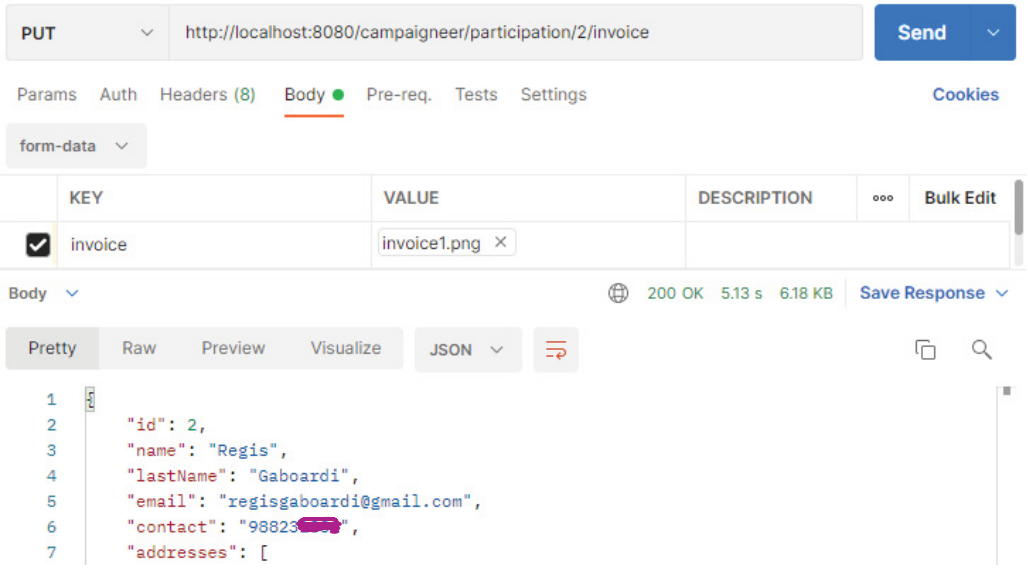
Figura 13 - Chamada para API que cria uma Participação parte 2 (UC34)


```
"products": [
    {
        "id": 9,
        "name": "Métodos Ágeis de Desenvolvimento de Software",
        "ean": "EES013",
        "classOfGood": "SOUNDBAR",
        "manufacturer": {
            "id": 1,
            "name": "UFPR",
            "addresses": [
                {
                    "id": 1,
                    "addressType": "SHIPPING_ADDRESS",
                    "postalCode": "88088-888",
                    "streetName": "Dr. Alcides Arvocerde",
                    "streetNumber": 1225,
                    "complement": "SEPT",
                    "cityJSON": {
                        "id": 1,
                        "name": "Curitiba",
                        "stateJSON": {
                            "id": 1,
                            "name": "Parana",
                            "code": "PR",
                            "countryJSON": {
                                "id": 1,
                                "name": "Brasil",
                                "code": "BR"
                            }
                        }
                    }
                }
            ]
        }
    }
],
"triggeredCampaign": null,
"campaignStatus": "NOT_PROCESSED",
"invoiceDate": "2020-09-22",
"invoice": null
}
```


Fonte: O Autor (2022)

A Participação, assim como a Campanha, pode ser criada parcialmente, mas algumas informações são mínimas. No exemplo acima, a Participação é criada sem o envio do arquivo da nota fiscal. Por padrão, as Participações são processadas pela primeira vez apenas após o envio deste arquivo, que está ilustrado na Figura 14 - Chamada para API de submissão do arquivo da nota fiscal.

Figura 14 - Chamada para API de submissão do arquivo da nota fiscal parte 1 (UC35)



Fonte: O Autor (2022)

Figura 15 - Chamada para API de submissão do arquivo da nota fiscal parte 2 (UC35)

```
        {
            "id": 4,
            "addressType": "BILLING_ADDRESS",
            "postalCode": "88088-000",
            "streetName": "Antonio Simoni",
            "streetNumber": 398,
            "complement": "300b",
            "cityJSON": {
                "id": 2,
                "name": "Pinhais",
                "stateJSON": {
                    "id": 1,
                    "name": "Parana",
                    "code": "PR",
                    "countryJSON": {
                        "id": 1,
                        "name": "Brasil",
                        "code": "BR"
                    }
                }
            }
        }
    ],
    "campaignStatus": "VALIDATION_QUEUE",
    "invoiceDate": "2020-09-22",
    "invoice": "src\\main\\resources\\invoices\\2020_ENG_SOFTWARE_2.jpg"
}
```

Fonte: O Autor (2022)

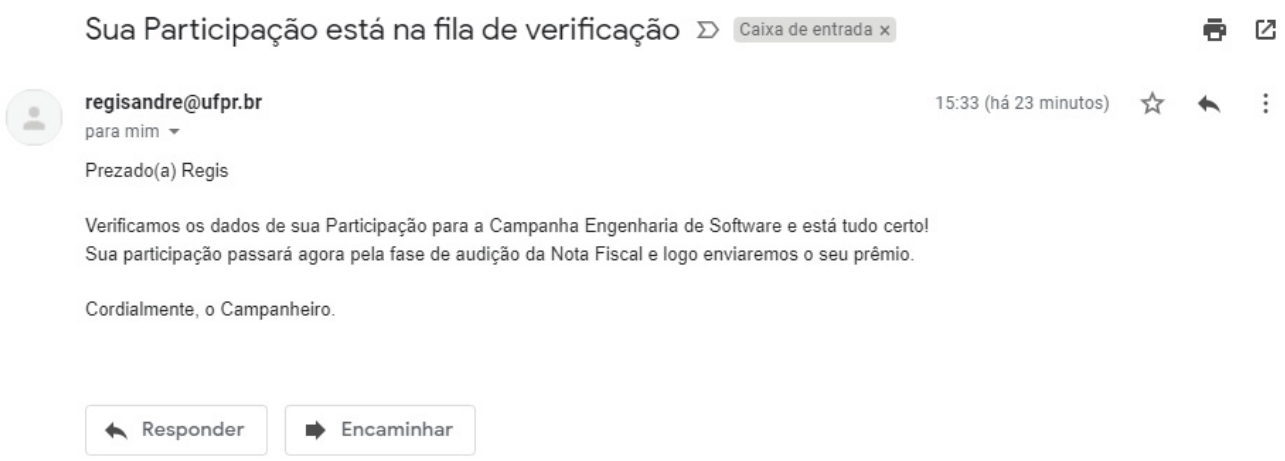
Uma vez que o arquivo da nota fiscal é enviado, a Participação será processada pela primeira vez e um e-mail será enviado para o participante informando-o do estado de seu pedido.

O arquivo será salvo no sistema de arquivos do servidor e o caminho para o arquivo é persistido no banco de dados para a Participação de qual o arquivo pertence. A resposta contém o caminho do arquivo para sinalizar que há um arquivo salvo para esta Participação.

Caso o processamento inicial não encontre nenhuma violação nas informações fornecidas, a Participação receberá o estado de "*VALIDATION_QUEUE*", como é exibido na Figura 15 - Chamada para API de submissão do arquivo da nota fiscal parte 2.

O e-mail que informa o participante que sua Participação foi recebida é representado pela Figura 16 - E-mail enviado ao Participante após o primeiro processamento da Participação.

Figura 16 - E-mail enviado ao Participante após o primeiro processamento da Participação (UC40)



Fonte: O Autor (2022)

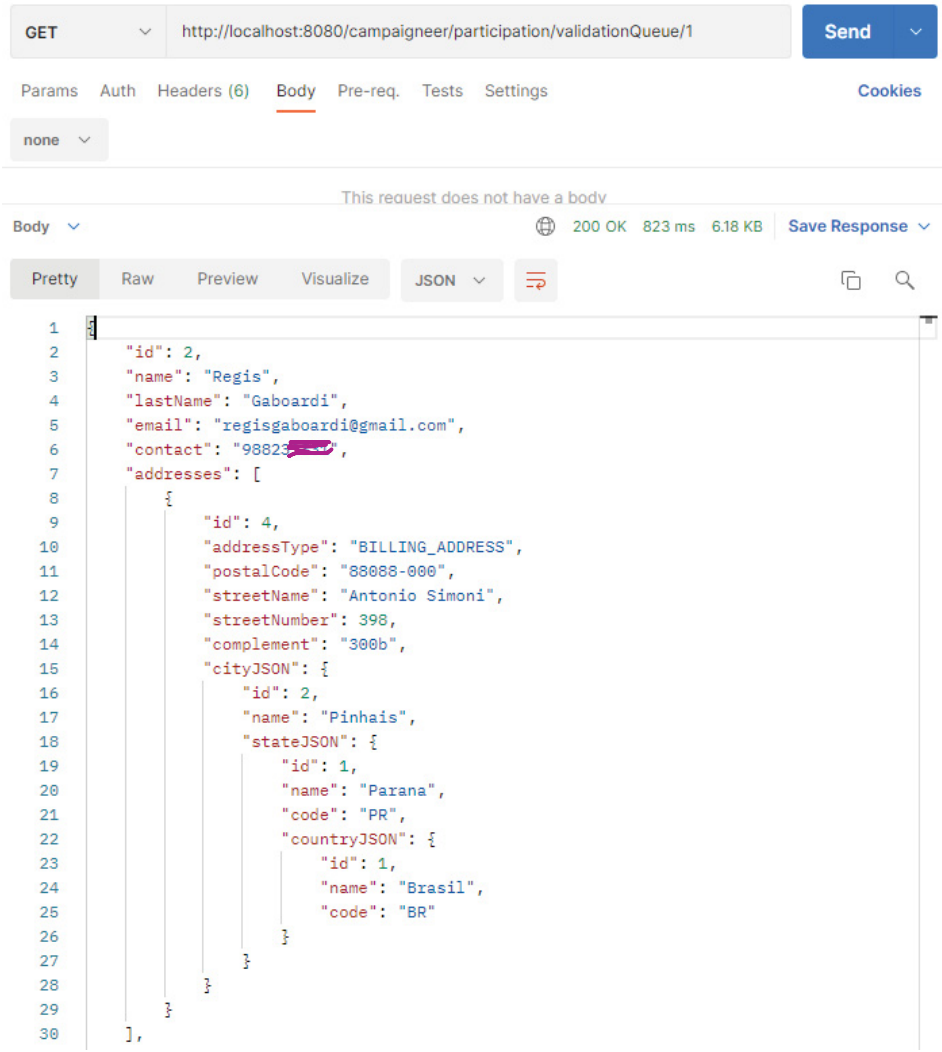
4.5 VALIDAÇÃO E CORREÇÃO

A partir do momento que a Participação é recebida, toda a comunicação com o participante será feita via e-mail. O componente de e-mails compõe a mensagem enviada identificando os dados do participante, como nome do participante, nome da campanha e quais violações foram eventualmente identificadas na Participação. Depois de recebida e processada, a Participação deve aguardar pela etapa de validação manual, ou seja, é necessário que um usuário administrador (ou operador) do sistema faça a verificação comparativa entre os dados fornecidos pelo participante em formulário, e os dados presentes no arquivo da nota fiscal submetido.

Para que seja feita esta conferência, é disponibilizada a API que recupera por Campanha, a próxima participação com estado "*VALIDATION_QUEUE*".

Esta chamada é representada pela Figura 17 - Chamada para API que recupera os dados de uma Participação para validação.

Figura 17 - Chamada para API que recupera os dados de uma Participação para validação parte 1 (UC30)



Fonte: O Autor (2022)

Figura 18 - Chamada para API que recupera os dados de uma Participação para validação parte 2 (UC30)


```
"products": [
    {
        "id": 3,
        "name": "Engenharia de Software",
        "ean": "EES002",
        "classOfGood": "SMARTPHONE",
        "manufacturer": {
            "id": 1,
            "name": "UFPR",
            "addresses": [
                {
                    "id": 1,
                    "addressType": "SHIPPING_ADDRESS",
                    "postalCode": "88088-888",
                    "streetName": "Dr. Alcides Arvocerde",
                    "streetNumber": 1225,
                    "complement": "SEPT",
                    "cityJSON": {
                        "id": 1,
                        "name": "Curitiba",
                        "stateJSON": {
                            "id": 1,
                            "name": "Parana",
                            "code": "PR",
                            "countryJSON": {
                                "id": 1,
                                "name": "Brasil",
                                "code": "BR"
                            }
                        }
                    }
                }
            ]
        }
    }
],
```


Fonte: O Autor (2022)

Com estes dados, o operador consegue verificar se há fraude nas informações providas pelo Participante. Após a verificação, o operador deve fazer uma requisição adicional informando quais violações foram identificadas. As violações são obrigatoriamente enviadas no formato de *flags* e o motor do sistema identificará, baseado neste conjunto, quais correções serão posteriormente solicitadas ao participante.

O restante da resposta é relativo à Campanha identificada e é parcialmente omitido na Figura 19 - Chamada para API que recupera os dados de uma Participação para validação parte.

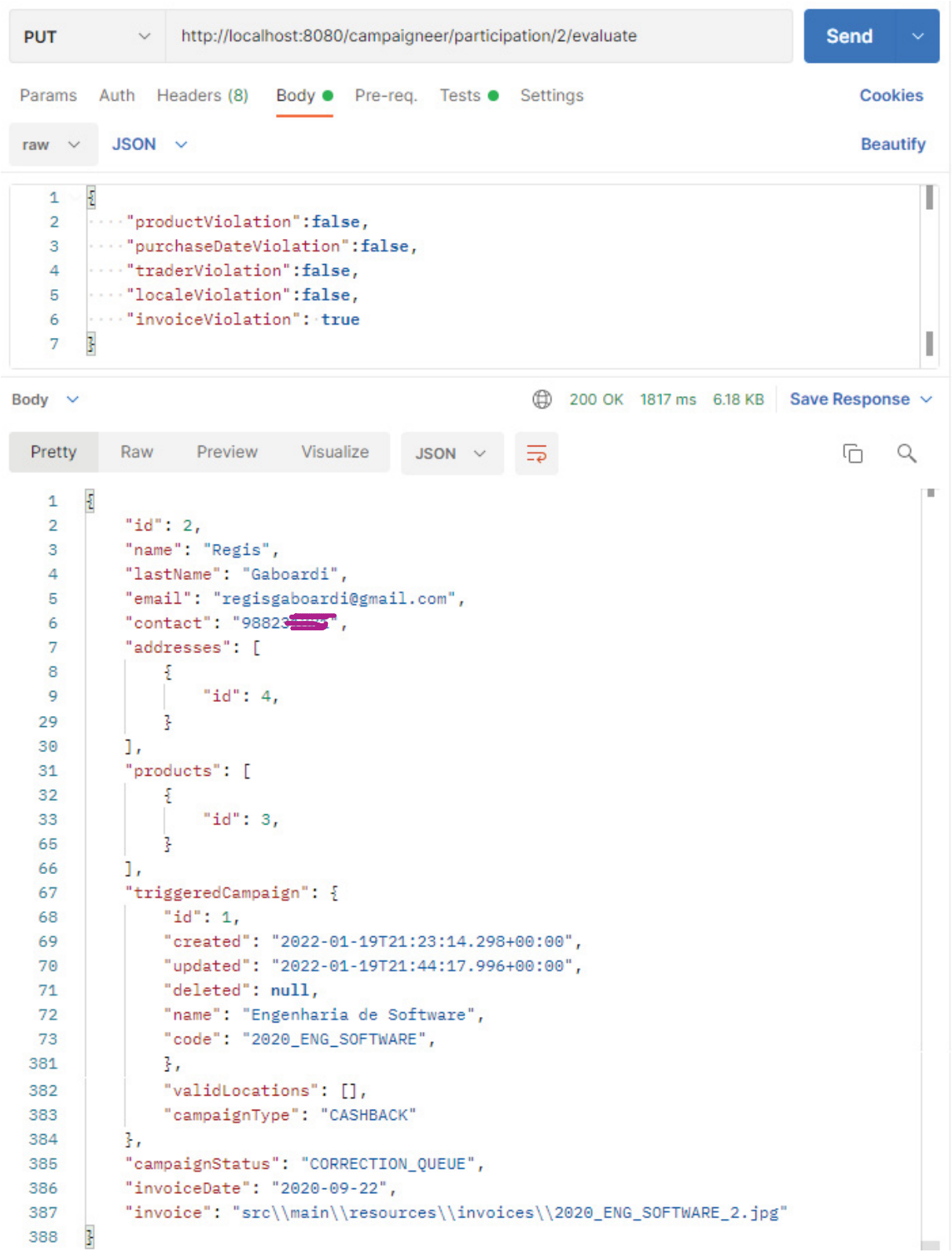
Figura 19 - Chamada para API que recupera os dados de uma Participação para validação parte 3 (UC30)


```
    "triggeredCampaign": {
        "id": 1,
        "created": "2022-01-19T21:23:14.298+00:00",
        "updated": "2022-01-19T21:44:17.996+00:00",
        "deleted": null,
        "name": "Engenharia de Software",
        "code": "2020_ENG_SOFTWARE",
        "promoter": {
            "id": 1,
            "created": "2022-01-19T21:11:48.328+00:00",
            "updated": null,
            "deleted": null,
            "name": "UFPR",
            "addresses": [
            ]
        },
        "purchaseFrom": "2020-01-01",
        "purchaseUntil": "2021-12-31",
        "validFrom": "2020-01-01",
        "validUntil": "2022-12-31",
        "participatingProducts": [
            {
                "id": 2,
            }
        ],
        "validLocations": [],
        "campaignType": "CASHBACK"
    },
    "campaignStatus": "VALIDATION_QUEUE",
    "invoiceDate": "2020-09-22",
    "invoice": "src\\main\\resources\\invoices\\2020_ENG_SOFTWARE_2.jpg"
}
```


Fonte: O Autor (2022)

A requisição para avaliação da participação envia ao servidor as *flags* de violação, conforme a Figura 20 - Chamada para API que envia o resultado da validação manual ao servidor com violações. Algumas partes do objeto retornado são relativos à Participação e à Campanha e não diferem das informações apresentadas em imagens anteriores, por isso, estão parcialmente omitidas.

Figura 20 - Chamada para API que envia o resultado da validação manual ao servidor com violações (UC31)

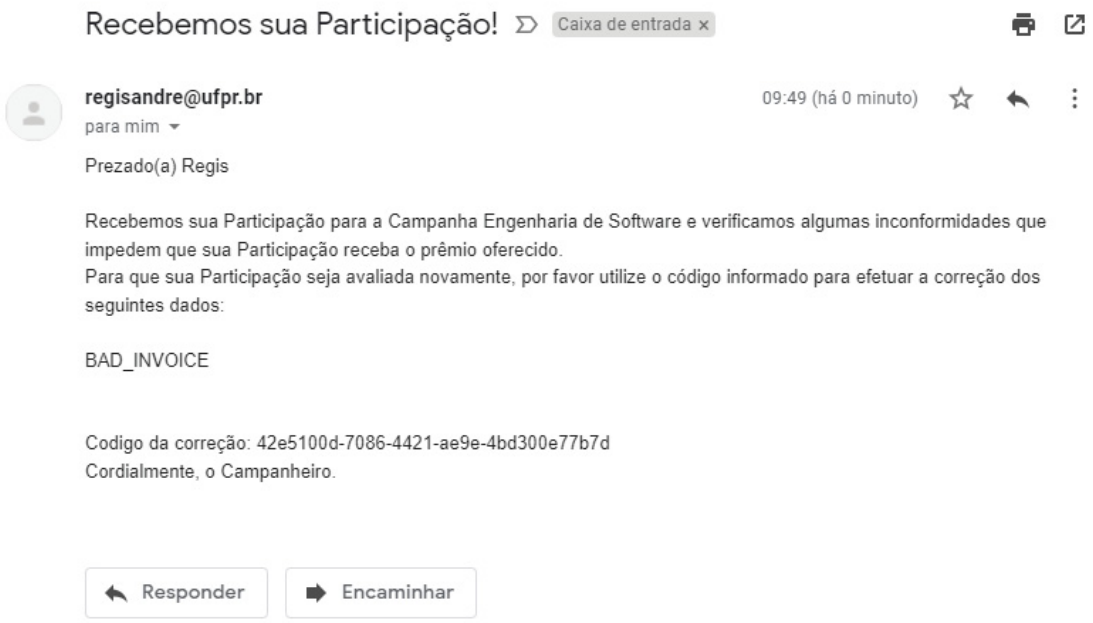


Fonte: O Autor (2022)

No exemplo acima, é enviado ao servidor a identificação de um problema com o arquivo da nota fiscal. Esse envio causa um reprocessamento da Participação avaliada e, por consequência, um e-mail será enviado ao participante para notificá-lo do estado da Participação.

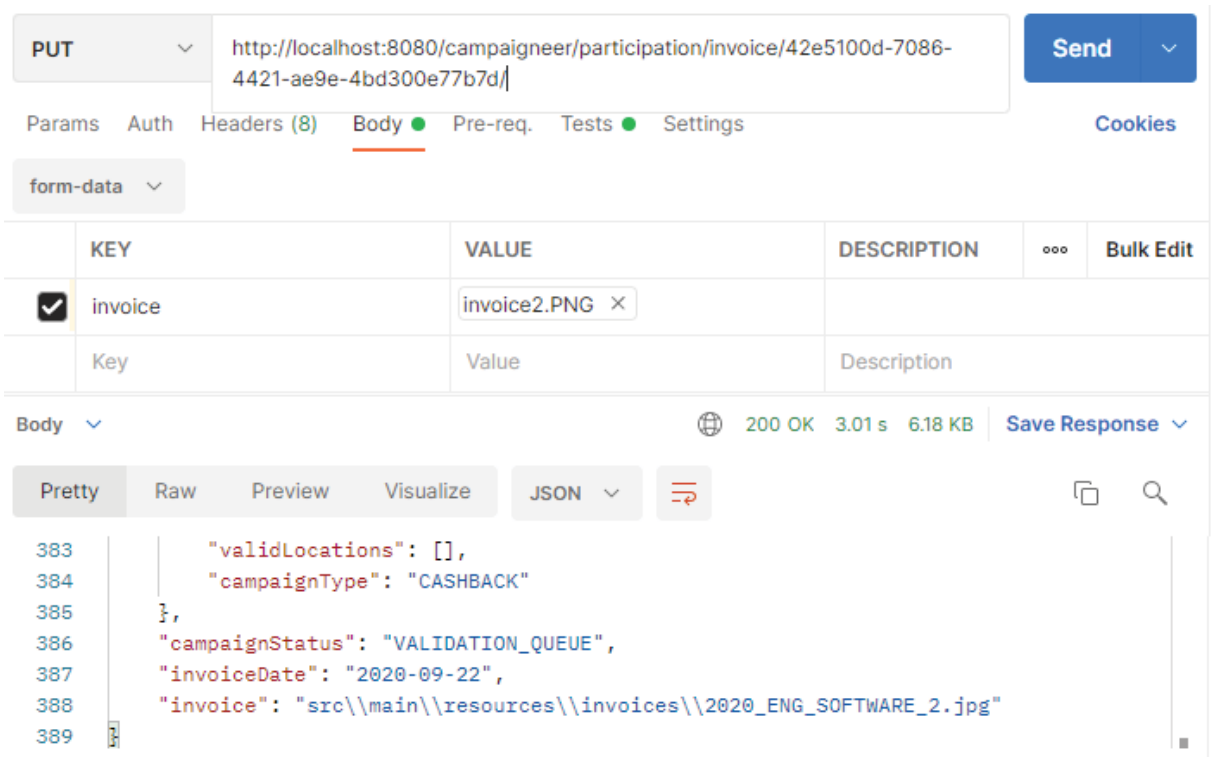
Quando há identificação de problemas na Participação, o sistema gera um código para a correção dos dados, que é enviado ao participante por e-mail. Este código deve ser utilizado na URL da requisição feita ao servidor, por qualquer interface. Para ilustração, o código gerado é enviado ao participante, e é utilizado manualmente via Postman, como mostram as figuras 21 - E-mail enviado ao participante notificando-o que há erros em sua Participação e 22 - Chamada para API de correção de dados/reenvio da nota fiscal.

Figura 21 - E-mail enviado ao participante notificando-o que há erros em sua Participação (UC40)



Fonte: O Autor (2022)

Figura 22 - Chamada para API de correção de dados/reenvio da nota fiscal (UC36)

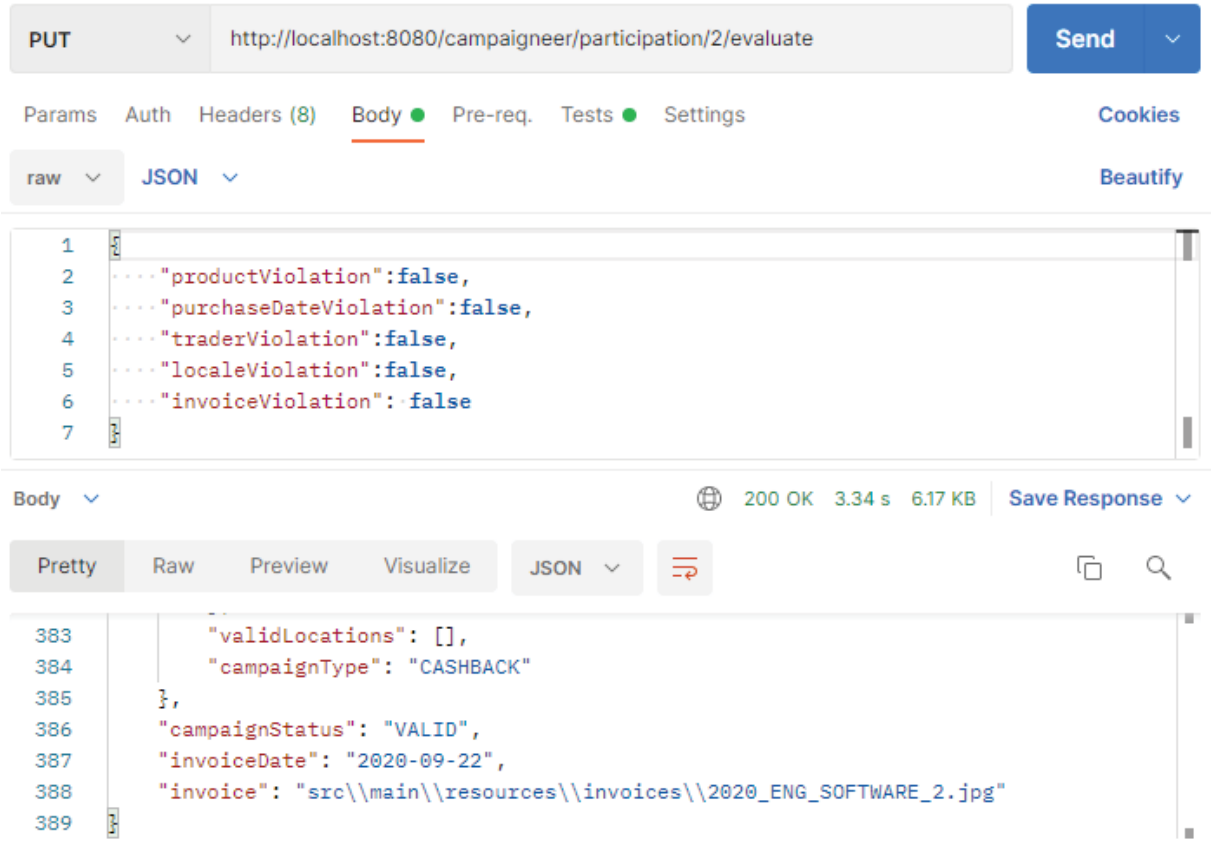


Fonte: O Autor (2022)

Após receber a correção dos dados identificados, o sistema recoloca a Participação automaticamente na fila para validação novamente, representada pelo estado "*VALIDATION_QUEUE*", exibido na imagem acima.

Quando nova validação for feita para esta Participação, mas nenhuma violação for encontrada, o reprocessamento que ocorre automaticamente identificará esta Participação com o estado "*VALID*", exibido na Figura 23 - Chamada para API que envia o resultado da validação manual ao servidor sem violações.

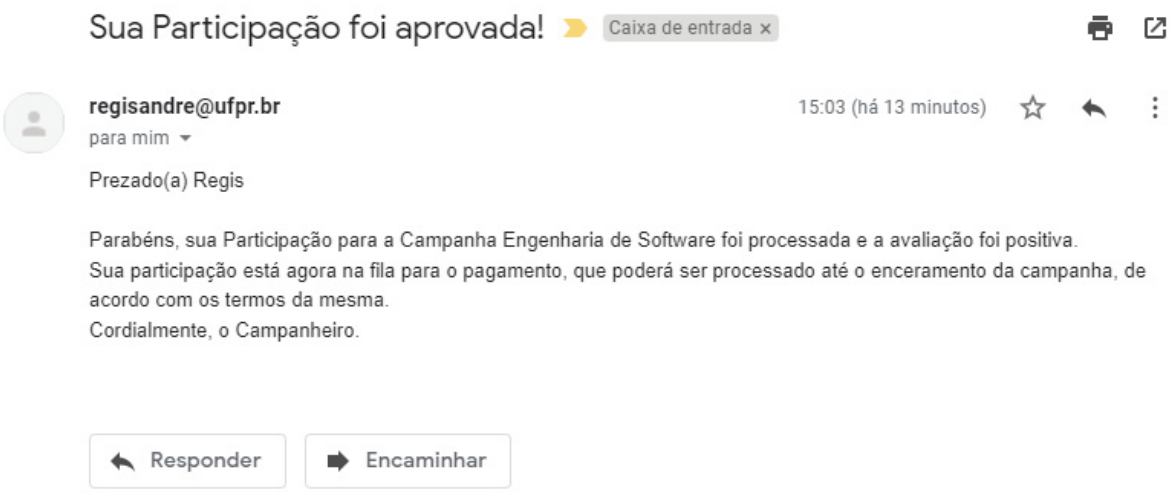
Figura 23 - Chamada para API que envia o resultado da validação manual ao servidor sem violações (UC31)



Fonte: O Autor (2022)

Uma vez que a Participação recebe o estado de “válido”, significa que os dados informados no momento em que a Participação foi criada combinam com os dados presentes na nota fiscal e nenhuma tentativa de fraude foi identificada. Com este estado, um novo e-mail é enviado ao participante com o estado atualizado (Figura 24).

Figura 24 - E-mail enviado ao participante notificando-o do processamento de sua Participação (UC40)

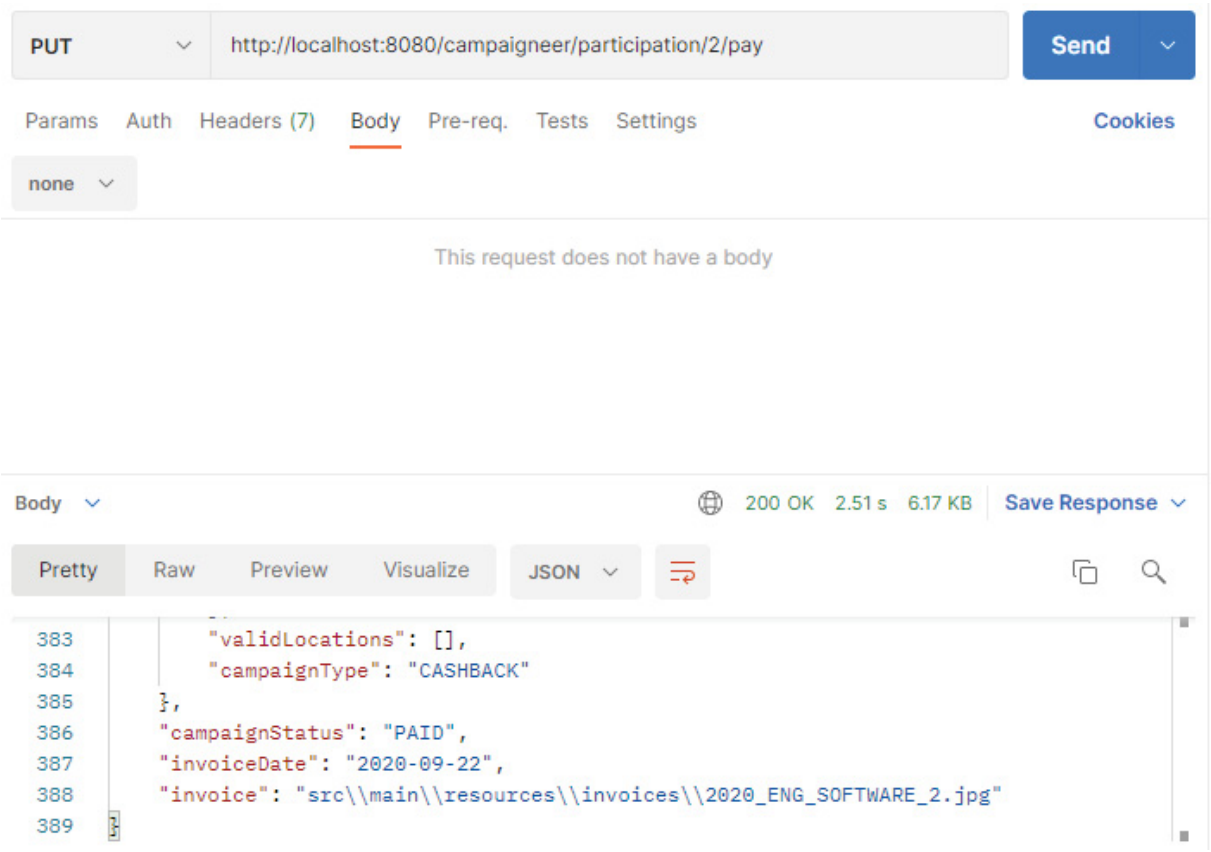


Fonte: O Autor (2022)

Uma vez aprovado, a única etapa manual restante é o pagamento, que pode ser processado individualmente ou para todas as Participações válidas da Campanha.

De acordo com os termos e condições de cada Campanha, o pagamento pode ser instantâneo ou levar algum tempo. Por exemplo, pagamentos de *cashback* ou liberação de certificado de garantia podem ser recebidos mais rapidamente do que o envio de um terceiro produto pelos Correios.

Figura 25 - Chamada para a API de pagamento de uma Participação (UD32)

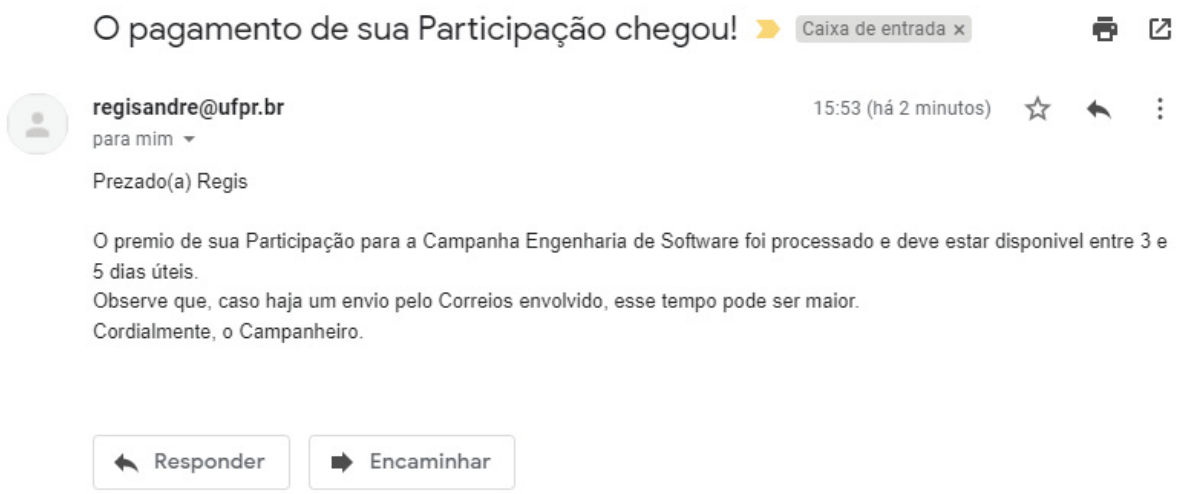


Fonte: O Autor (2022)

A Figura 25 - Chamada para a API de pagamento de uma Participação demonstra a requisição para o pagamento do prêmio de uma Participação.

Após o pagamento, um novo e-mail é enviado ao participante com informações do processamento de sua participação, como exibido na Figura 26 - E-mail enviado ao participante notificando-o do pagamento de sua Participação.

Figura 26 - E-mail enviado ao participante notificando-o do pagamento de sua Participação (UC40)



Fonte: O Autor (2022)

Após o pagamento, a Participação será encerrada e não sofrerá atualizações futuras.

O sistema tem um agente que procura por Participações com o estado "*PAID*" regularmente a cada cinco minutos. As Participações encontradas com esse estado são encerradas automaticamente, sem a necessidade da intervenção de um operador. O agente periódico é ilustrado pela Figura 27 - Imagem do agente automático para encerramento de Participações pagas.

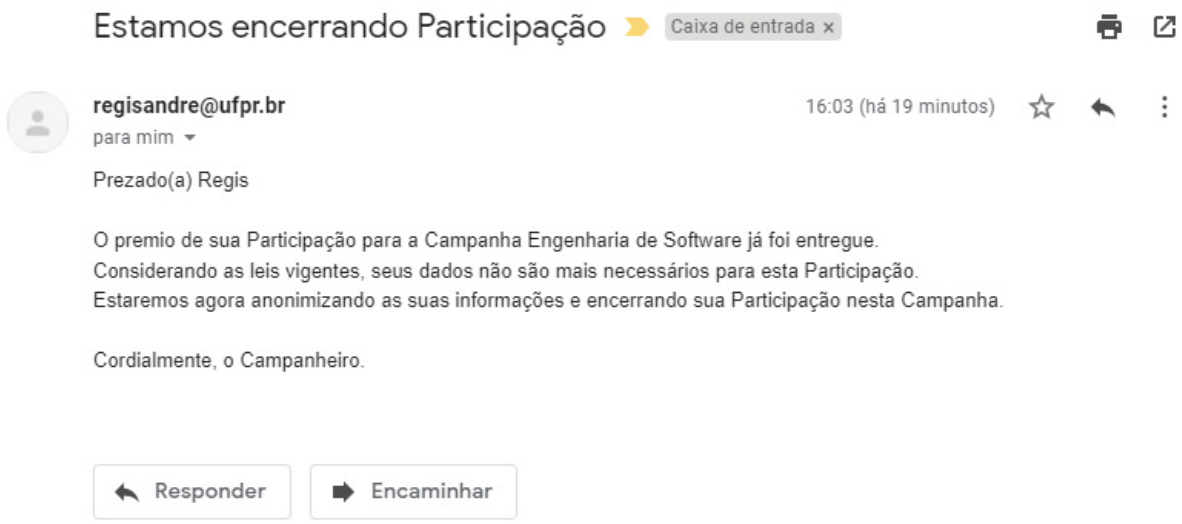
Figura 27 - Imagem do agente automático para encerramento de Participações pagas (UC41)

```
2022-01-23 16:03:35.420  INFO 38316 : Running scheduled search for Paid Participations.
2022-01-23 16:03:35.430  WARN 38316 : HHH10001002: Using Hibernate built-in connection pool (not for production use!)
2022-01-23 16:03:35.431  INFO 38316 : HHH10001005: using driver [com.mysql.cj.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/campaigneer_db
2022-01-23 16:03:35.431  INFO 38316 : HHH10001001: Connection properties: {password=****, user=root}
2022-01-23 16:03:35.431  INFO 38316 : HHH10001003: Autocommit mode: false
2022-01-23 16:03:35.431  INFO 38316 : HHH000115: Hibernate connection pool size: 20 (min=1)
2022-01-23 16:03:35.442  INFO 38316 : HHH000400: Using dialect: org.hibernate.dialect.MySQL8Dialect
2022-01-23 16:03:35.444  WARN 38316 : HHH000503: A class should not be annotated with both @Inheritance and @MappedSuperclass. @Inheritance w
2022-01-23 16:03:35.496  INFO 38316 : HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.Jdbc
2022-01-23 16:03:35.623  WARN 38316 : HHH10001002: Using Hibernate built-in connection pool (not for production use!)
2022-01-23 16:03:35.623  INFO 38316 : HHH10001005: using driver [com.mysql.cj.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/campaigneer_db
2022-01-23 16:03:35.623  INFO 38316 : HHH10001001: Connection properties: {password=****, user=root}
2022-01-23 16:03:35.623  INFO 38316 : HHH10001003: Autocommit mode: false
2022-01-23 16:03:35.623  INFO 38316 : HHH000115: Hibernate connection pool size: 20 (min=1)
2022-01-23 16:03:35.635  INFO 38316 : HHH000400: Using dialect: org.hibernate.dialect.MySQL8Dialect
2022-01-23 16:03:35.637  WARN 38316 : HHH000503: A class should not be annotated with both @Inheritance and @MappedSuperclass. @Inheritance w
2022-01-23 16:03:35.691  INFO 38316 : HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.Jdbc
2022-01-23 16:03:35.814  INFO 38316 : Found 1 Participations to be closed.
2022-01-23 16:03:35.814  INFO 38316 : Closing and anonimizing Participation with id: 2
```

Fonte: O Autor (2022)

Quando a Participação é encerrada, um novo e-mail é enviado informando o participante, e os dados do processamento são anonimizados na base. O e-mail do encerramento dos dados é representado pela Figura 28 - E-mail enviado ao participante notificando-o do encerramento de sua Participação.

Figura 28 - E-mail enviado ao participante notificando-o do encerramento de sua Participação (UC40)



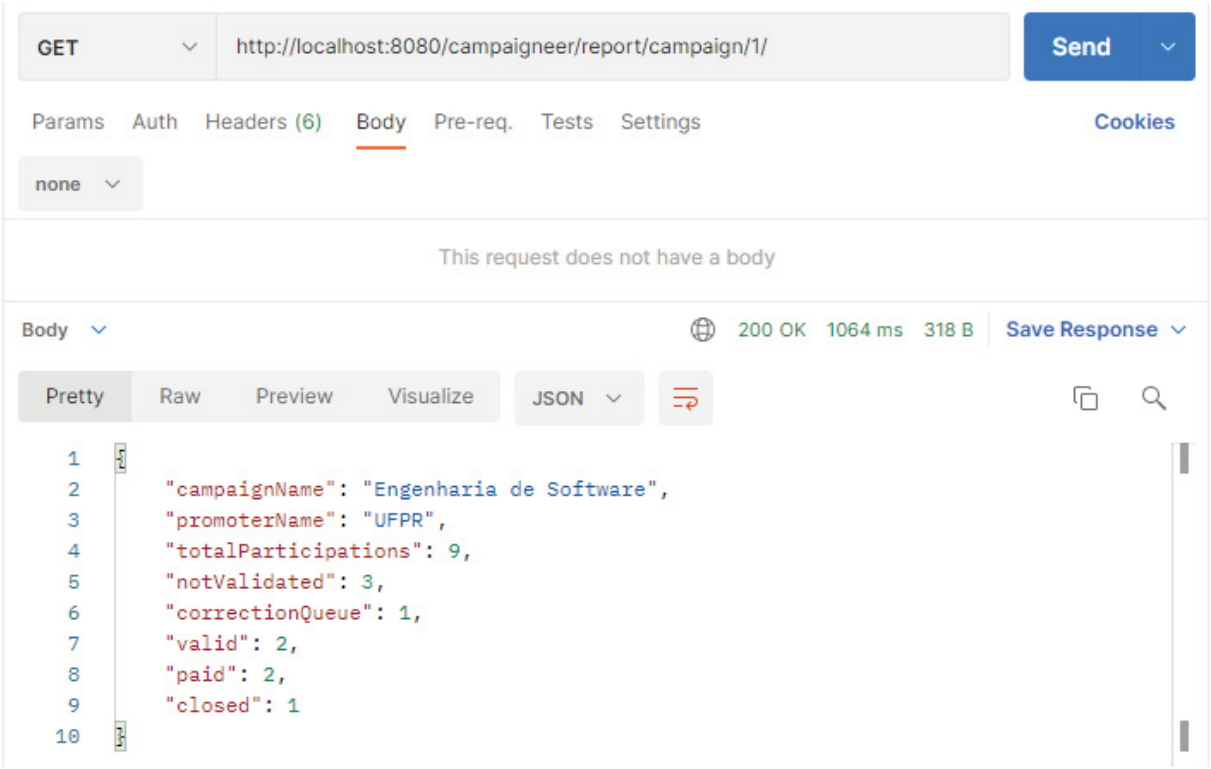
Fonte: O Autor (2022)

Alternativamente ao processamento como exemplificado até agora, uma Participação pode ficar indeterminadamente em verificação, caso sempre haja um novo erro identificado.

Existe a possibilidade para que uma Participação passe do estado de "*VALIDATION_QUEUE"* para "*FINAL_INVALID*", isto quer dizer que a Participação não será mais elegível para verificação manual e será encerrada quando o período da Campanha terminar.

Durante o período da Campanha, é possível gerar relatórios sobre as suas Participações. A Figura 29 - Chamada para a API de extração de relatório de participações por Campanha ilustra uma requisição para a extração do relatório de participação de uma Campanha.

Figura 29 - Chamada para a API de extração de relatório de participações por Campanha



Fonte: O Autor (2022)

5 CONSIDERAÇÕES FINAIS

Com a conclusão deste projeto, o sistema produzido foi capaz de ilustrar a construção e utilização de um motor de processamento para campanhas promocionais, no qual os promotores podem decidir os detalhes de suas campanhas promocionais e os participantes podem fazer o envio de seus dados de forma rápida e prática.

O sistema produzido tem capacidade de identificar qual campanha promocional é adequada a cada participação que recebe, resultando em uma avaliação automática e reduzindo a burocracia que é também uma barreira para a adesão para este tipo de promoção.

O sistema foi desenvolvido utilizando linguagem de programação Java, banco de dados MySQL, Spring Boot e outras ferramentas de desenvolvimento das quais o autor já estava familiarizado no início do desenvolvimento. A aplicação de práticas e conceitos da Engenharia de Software viabilizaram o desenvolvimento organizado e ágil do sistema proposto.

5.1 RECOMENDAÇÕES PARA TRABALHOS FUTUROS

O escopo do sistema desenvolvido nesse projeto visa desenvolver e demonstrar como o *back end* de um motor de processamento pode funcionar, mas pode ser evoluído e elaborado em várias áreas, até que se torne um sistema completo, tanto para o usuário consumidor, quanto para quem opera o sistema a fim de criar e configurar campanhas.

O sistema pode ser melhorado e polido ainda no formato que está. As respostas dadas pelo servidor podem ser alteradas para conter apenas as informações relevantes a cada API, ao invés de apresentar todos os dados relacionados, como os dados da Marca Promotora, da Campanha em si, entre outros.

Dentre os núcleos ou ferramentas que podem complementar um sistema como o desenvolvido nesse projeto, pode-se citar:

- Interface para configuração das campanhas e validação das participações por parte dos operadores do sistema;

- Interface para a conferência das configurações das campanhas e das estatísticas de participação por parte dos contratantes do sistema;
- Interface para o cadastro dos dados pessoais por parte dos consumidores participantes, com o intuito de agilizar o processo de registro em novas campanhas;
- Desenvolvimento de uma ferramenta para gerenciar os aceites de “acordos” dos contratos de termos e condições de cada campanha e acordos de armazenamento e processamento de dados pessoais relativos à Lei Geral de Proteção de Dados de cada região;
- Adição de um núcleo de inteligência artificial para a verificação dos arquivos de Nota Fiscal submetidos pelos participantes;
- Conversão de ferramentas existentes e novas adições no formato de micro serviços em adequação à evolução das tecnologias presentes no mercado de TI;
- Ajuste nos retornos do sistema para utilização dos Status Code HTTP apropriados e omissão de atributos não relevantes à chamada.

Esses são alguns dos tópicos que podem ser abordados na elaboração de um sistema para processamento de participações em campanhas promocionais mais completo e robusto.

REFERÊNCIAS

BRUEGGE, Bernd; DUTOIT, Allen H. **Object-Oriented Software Engineering**: Using UML, Patterns and Java. 3. ed. [*S. l.*]: Pearson, 2010.

CURRIE, Shane K. **An Analysis of Buyer Behaviour in Response to Rebate Promotions**. 2014. 228 p. Trabalho de conclusão de curso (Doutor em Filosofia) - Univeristy of Western Australia, Perth, 2014.

FIELDING, Roy Thomas. **Architectural Styles and the Design of Network-based Software Architectures**. Orientador: Prof. Dr. Richard N. Taylor. 2000. 180f. Trabalho de conclusão de curso (Doutor em Filosofia) – University of California, Irvine, 2000.

FOWLER, Martin. **UML Distilled**: A Brief Guide to the Standard Object Modeling Language. 3. ed. [*S. l.*]: Pearson, 2018.

GERALDO, Graciela C.; MAINARDES, Emerson W. Estudo sobre os fatores que afetam a intenção de compras online. **REGE - Revista de Gestão**, [*s. l.*], v. 24, ed. 2, p. 181-194, 2017.

KABANGO, Christian M.; ASA, Romeo. Factors influencing e-commerce development: Implications for the developing countries. **International Journal of Innovation and Economic Development**, Wuhan, v. 1, ed. 1, p. 64-72, 2015.

LAWRENCE, Japhet E.; TAR, Usman A. Barriers to ecommerce in developing countries. **Information, Society and Justice**, [*s. l.*], v. 3, ed. 1, p. 23-35, 2010.

MASSÉ, Mark. **REST API design rulebook**: designing consistent RESTful web service interfaces. [*S. l.*]: O'reilly, 2012.

MAXIMINI, Dominik. **The Scrum Culture**: Introducing Agile Methods in Organizations. 2. ed. Perth: Springer, 2018.

MENDES, Laura Z. R. **Comercio Eletrônico**: Quem impulsiona o Comercio Eletrônico no Brasil?. Orientador: Prof. Dr. Eveline Barbosa Silva Carvalho. 2015. 37 f. Trabalho de conclusão de curso (Bacharel) - Universidade Federal do Ceará, Fortaleza, 2015.

MENDES, Laura Z. R. **E-COMMERCE**: Origem, Desenvolvimento e Perspectiva. Orientador: Prof. Dr. Cássio da Silva Calvete. 2013. 64 f. Trabalho de conclusão de curso (Bacharel) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2013.

SCWABER, Ken; SUTHERLAND, Jeff. **The Scrum Guide**. 2020. Relatório Técnico.

SOMAN, Dilip; GOURVILLE, J. T. **The Consumer Psychology of Mail-In Rebates**: A Model of Anchoring and Adjustment (December 10, 2005). HBS Marketing Research Paper No. 06-02

SOMMERVILLE, Ian. **Software Engineering**. 9. ed. [*S. l.*]: Pearson, 2011.

TAT, Peter K.; LEE, Monle; Motivational Aspects of Rebate Redemption. **Journal of Marketing Theory and Practice**, v. 1, ed 2, p. 52-63, 1993.

VLIET, Hans Van. **Software Engineering**: Principles and Practice. [*S. l.*]: Wiley, 2007. 552 p.

APÊNDICE A – DOCUMENTO DE VISÃO

O Campanheiro é um sistema que visa receber e processar dados de participações em campanhas promocionais de diferentes modalidades. Este sistema disponibiliza um conjunto de APIs para que os clientes possam submeter dados para suas participações. Estes dados são processados pelo sistema automaticamente, para se obter uma resposta favorável ou negativa acerca da validade de cada participação.

O sistema possui dois núcleos, voltados a diferentes usuários: um refere-se à criação e configuração da estrutura das campanhas promocionais, utilizado por administradores e operadores do sistema; outro voltado a submissão de dados para criação de participações, utilizado por meio de quaisquer sistemas web ou mobile para a coleta dos dados do usuário final, o participante.

As participações são processadas inúmeras vezes até que se obtenha uma resposta definitiva. O sistema identifica violações e comunica o usuário final, por e-mail, a respeito das atualizações de sua participação, possibilitando correções dos dados enviados sem que seja necessário criar uma conta.

O sistema é voltado às marcas, pois automatiza o processamento de participações de suas campanhas promocionais e oferece um conjunto de APIs que abrange diferentes tipos de promoções. Cada promoção é customizável, de acordo com as regras e condições de cada marca.

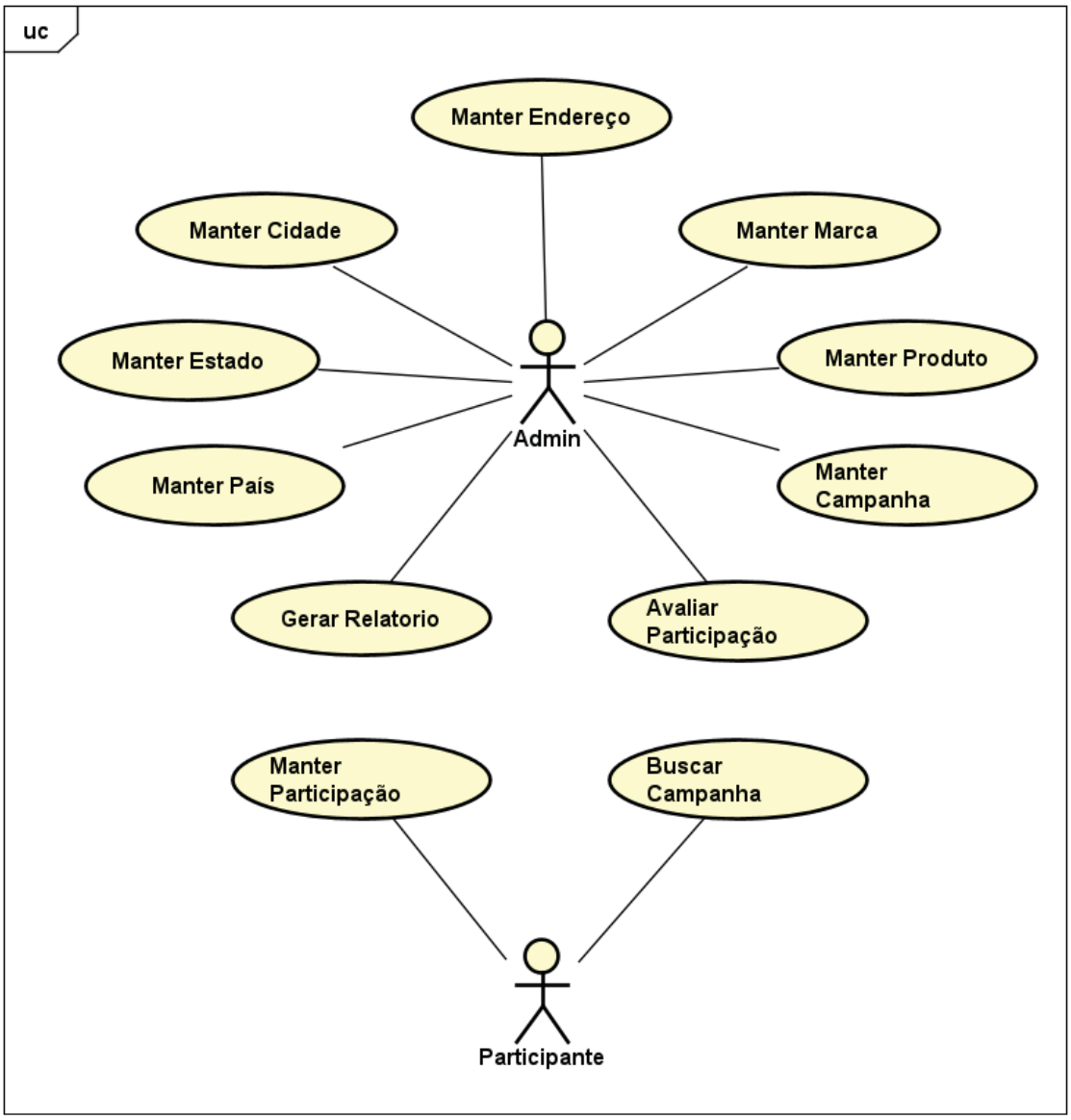
APÊNDICE B – CASOS DE USO NEGOCIAIS

Na Figura 30 - Diagrama de Casos de Uso Simplificado, são apresentadas de forma simplificada as ações disponíveis a cada usuário do sistema.

O usuário administrador é responsável por criar e configurar as estruturas necessárias para que se disponibilize uma campanha promocional, conforme especificado pela marca promotora.

O usuário participante é responsável por inserir os dados no sistema, de acordo com a sua participação candidata ao prêmio oferecido por cada campanha.

Figura 29 - Diagrama de Casos de Uso Simplificado



Fonte: O Autor (2022)

APÊNDICE C – GLOSSÁRIO

Marca - Marca é um nome, termo, desenho, símbolo ou qualquer outro meio pelo qual uma companhia, produto ou pessoa possa ser identificado.

Produto - Produto é um objeto, sistema, serviço ou qualquer objeto ou conceito que pode ser oferecido ao consumidor para satisfazer uma necessidade ou desejo.

Campanha - Campanha é um esforço para promover uma marca e seus produtos, e divulgar um benefício atrelado à compra desses produtos.

Participante - Participante é o indivíduo que submete seus dados em acordo com as regras propostas pela campanha com o intuito de obter o benefício oferecido.

Participação - Participação é o conjunto de dados enviados pelo participante e o processo feito para cada participante até que o prêmio oferecido pela campanha seja concedido ou negado ao solicitante.

Prêmio - Prêmio é o objeto oferecido pela campanha, pode ser um produto adicional, certificado de garantia estendida, devolução de parte do valor da compra, prestação de serviços adicionais etc.

APÊNDICE D – REGRAS DE NEGÓCIO

Uma Marca deve ser cadastrada no banco de dados possuindo um endereço composto por País, Estado, Cidade e Endereço. A Marca terá uma matriz, podendo possuir várias filiais em endereços diferentes.

A Campanha será válida no País da Marca promotora, ou nos endereços explicitamente definidos como válidos para a Campanha.

Para processar Participações automaticamente, o sistema precisa conhecer as regras de cada Campanha, tais como: Marca promotora, Produtos participantes, datas de validade para participação e para compra.

Uma validação manual dos dados é necessária para que uma Participação seja avaliada positiva ou negativamente.

Após avaliação negativa, seja automática ou manual, o sistema irá gerar um código para correção de dados. Correções das informações de Participações negativamente avaliadas só serão possíveis através desse código.

Um participante pode criar várias Participações para a mesma Campanha, o sistema deve identificar “Participações Gêmeas” e marcar essas ocorrências no banco de dados como duplicadas (*estado “DUPLICATED”)*.

Alguns casos de uso não observam regras de negócio, pois são funcionalidades necessárias para a manutenção dos dados já existentes no sistema. Abaixo, são descritas as regras de negócio para os casos de uso que observam esse tipo de regra, ordenados por caso de uso.

1. UC01: Criar País
    a. Não será possível criar dois Países com o mesmo código.
2. UC05: Criar Estado
    a. Não será possível criar dois Estados com o mesmo código no mesmo País.
    b. O Estado deve ser atrelado a um País.
3. UC09: Criar Cidade
    a. Não será possível criar duas Cidades com o mesmo nome no mesmo Estado.
    b. A Cidade deve ser atrelada a um Estado.
4. UC13: Criar Endereço

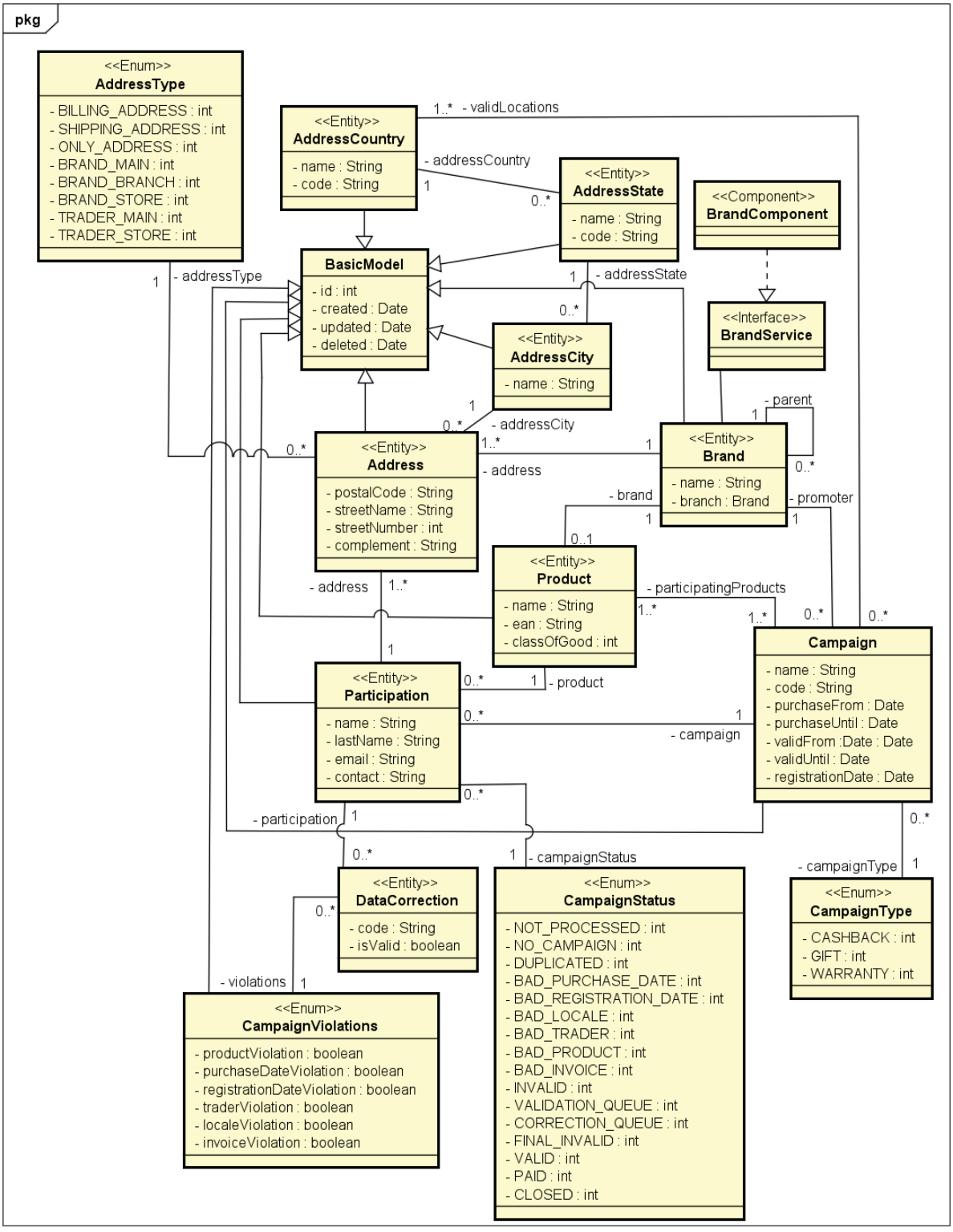
   a. O Endereço deve possuir Código Postal e Número.
   b. Vários Endereços podem possuir a mesma combinação de Código Postal e Número;
   c. O Endereço deve ser atrelado à uma Cidade.
5. UC17: Criar Marca
   a. A Marca deve possuir um Endereço identificado como “Matriz”.
   b. A Marca pode possuir Endereços alternativos, identificados como “Filial”.
   c. O Endereço da Marca – seja Matriz ou Filial – rege a área de validade da Campanha.
6. UC21: Criar Produto
   a. O Produto deve possuir um EAN único.
   b. O Produto deve ser atrelado à uma Marca.
7. UC25: Criar Campanha
   a. A Campanha deve possuir um Código único.
   b. A Campanha deve possuir um Nome.
   c. A Campanha deve ser atrelada à uma Marca.
   d. A Campanha deve possuir Data Início e Data Fim para sua validade.
   e. A Campanha deve possuir Data Início e Data Fim para a compra dos Produtos Participantes.
   f. A Campanha deve possuir uma lista de Produtos Participantes.
   g. A Campanha deve possuir um arquivo de termos e condições para a Participação.
   h. A Campanha deve ter um Tipo definido (*Cashback, Gift, Warranty, etc).*
8. UC27: Atualizar Campanha
   a. A Campanha não pode ser atualizada dentro do seu prazo de validade (entre sua Data Início e Data Fim).
   b. A Campanha pode ter seu nome alterado.
   c. A Campanha pode ter sua lista de Produtos alterada (UC29).
   d. A Campanha pode ter seus locais de validade alterados.
   e. A Campanha pode ter suas datas de validade alteradas.
9. UC30: Listar Participações Verificáveis

   a. Deverá listar somente Participações com o *estado* "*VALIDATION_QUEUE*".
10. UC31: Verificar Participação
   a. Deverá receber quais violações foram manualmente identificadas para a Participação.
   b. A ausência de violações implica que a Participação foi aprovada na etapa de validação manual.
11. UC32: Pagar Participação
   a. Somente são elegíveis ao pagamento as Participações com *estado "VALID"*.
12. UC34: Criar Participação
   a. O participante deve informar seus dados pessoais (Nome, Sobrenome, E-mail, Telefone, Endereço).
   b. O participante deve informar os dados da compra (Data da Nota Fiscal, Nota Fiscal e Produto).
13. UC36: Corrigir Informações
   a. O participante deve informar o Código de Correção juntamente aos novos dados da Participação.
   b. O Código de Correção deve ser válido.
   c. Ao final da correção, o *estado* da Participação será "*VALIDATION_QUEUE*".
14. UC39: Reprocessar Participação
   a. O Reprocessamento da Participação enviará um novo e-mail de notificação ao Participante.
   b. O Reprocessamento não implica na alteração do *estado* da Participação.
   c. Participações encerradas não podem ser reprocessadas.
15. UC41 Encerrar Participação
   a. Somente Participações com *estado* "*PAID*" ou "*FINAL_INVALID*" podem ser encerradas.
   b. Participações encerradas terão os dados do participante anonimizados.
16. UC42: Preparar Correção de dados

a. Participações que forem reprovadas – seja na validação automática ou manual – são elegíveis para correção de dados.
b. A correção de dados gera um Código de Correção único.
c. O Código de Correção utilizado será marcado como invalido no banco de dados para impossibilitar que o mesmo código seja utilizado mais que uma vez.
d. A correção de dados será disponibilizada para a Participação três vezes, caso a Participação seja reprovada após a utilização do terceiro código, receberá o *estado* "*FINAL_INVALID*".

APÊNDICE E – DIAGRAMA DE CLASSES DOS OBJETOS RELACIONAIS

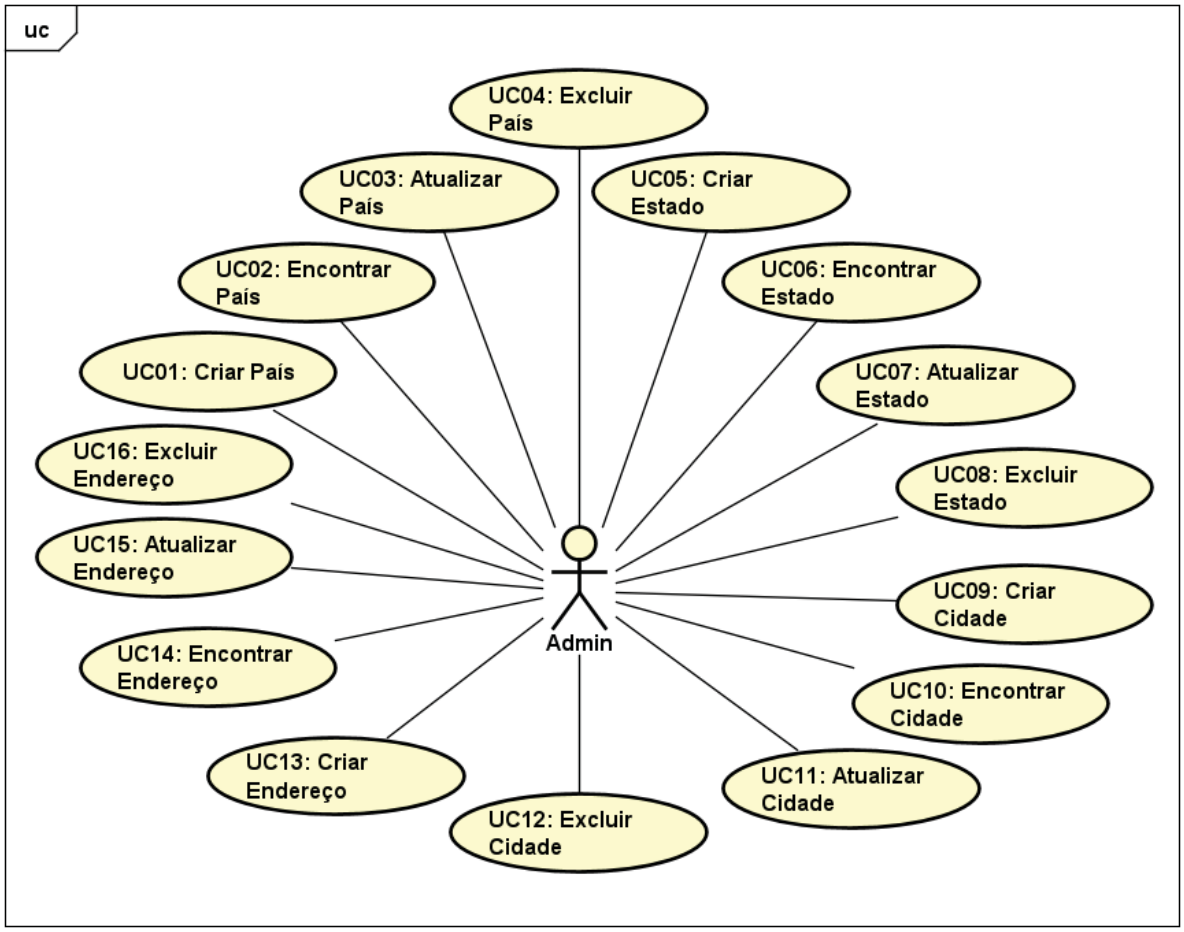
Figura 31 - Diagrama de Classes Relacionais



Fonte: O Autor (2022)

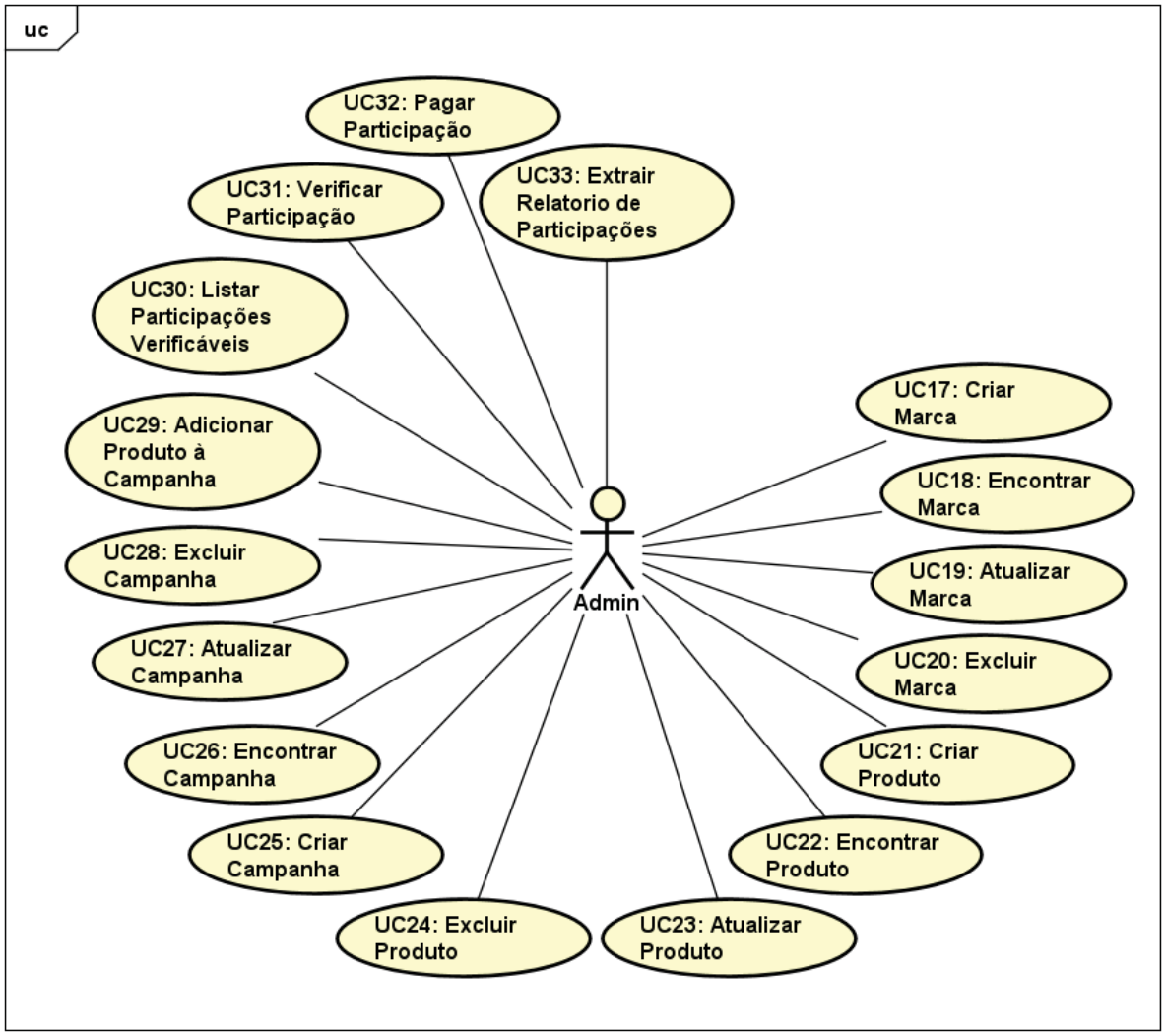
APÊNDICE F – DIAGRAMA DE CASOS DE USO DETALHADO

Figura 32 - Casos de Uso do Administrador



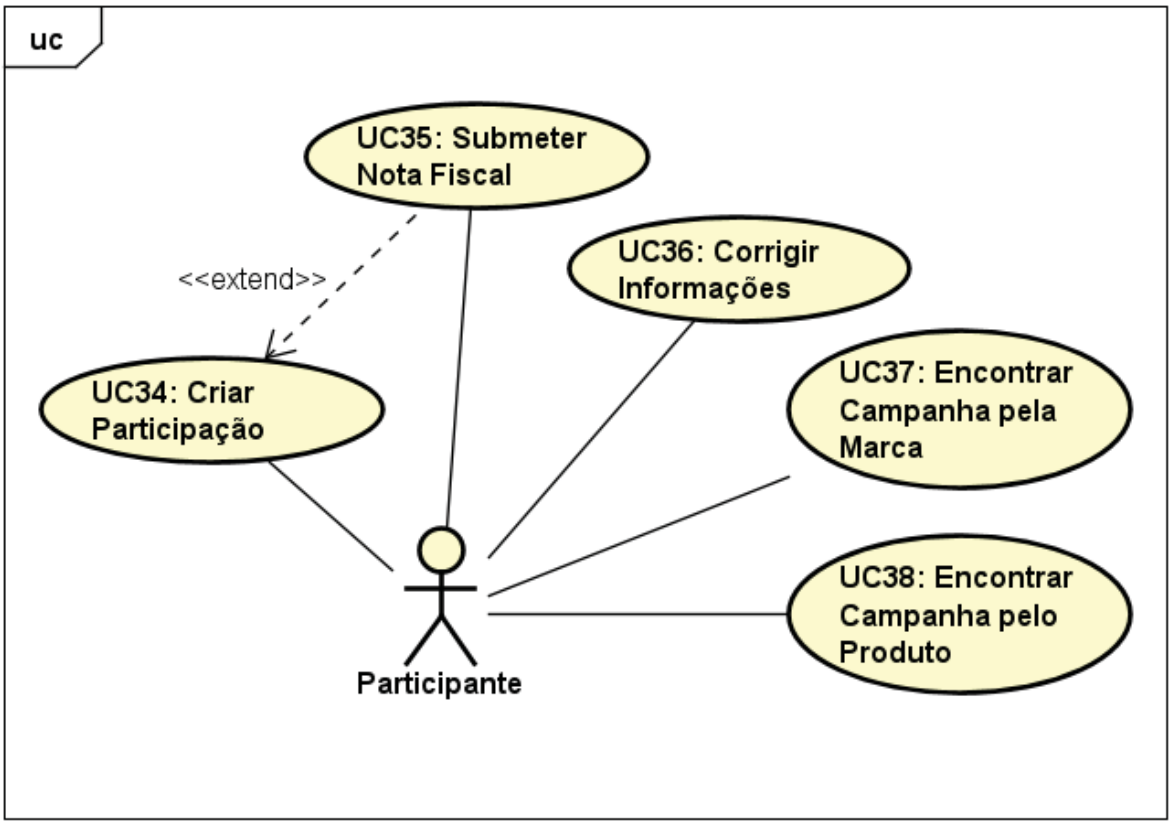
Fonte: O Autor (2022)

Figura 33 - Casos de Uso do Administrador (Parte 2)



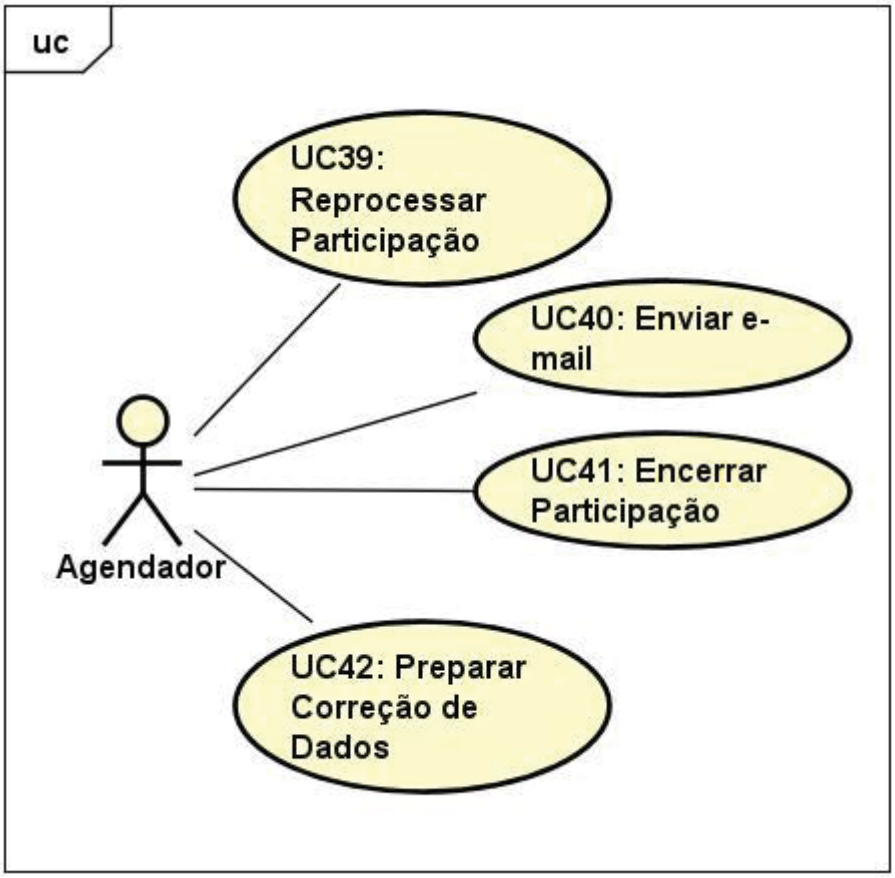
Fonte: O Autor (2022)

Figura 30 - Casos de Uso do Participante



Fonte: O Autor (2022)

Figura 31 - Casos de Uso do Sistema



Fonte: O Autor (2022)

APÊNDICE G – ESPECIFICAÇÃO DOS CASOS DE USO

UC01: Criar País

Descrição

Eu, como Administrador, quero criar um País para possibilitar a configuração de um Endereço.

Pré-Condições

Não há.

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para criação do País contendo Nome e Código.
2. O sistema persiste o formulário no banco de dados.
3. O sistema responde a requisição com os dados do País criado contendo seu ID.

Fluxo de Exceção

1. O usuário envia a requisição contendo o formulário para criação do País com Código que já existe no banco de dados.
2. O sistema responde a requisição informando que já existe um País com aquele código.

UC02: Encontrar País

Descrição

Eu, como Administrador, quero encontrar um País para verificar seus dados.

Pré-Condições

Não há.

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para pesquisar por um País dado seu Código.
2. O sistema responde a requisição com os dados do País encontrado.

Fluxo de Exceção

1. O usuário envia a requisição contendo o formulário para pesquisa do País com Código que não existe no banco de dados.
2. O sistema responde a requisição informando que nenhum País com aquele código foi encontrado.

UC03: Atualizar País

Descrição

Eu, como Administrador, quero atualizar um País para manter seus dados corretos.

Pré-Condições

O País deve existir no banco de dados e ser encontrado (UC02).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para atualização do País contendo ID, Nome e Código.
2. O sistema persiste o formulário no banco de dados.
3. O sistema responde a requisição com os dados do País atualizado.

Fluxo de Exceção

1. O usuário envia a requisição contendo o formulário para pesquisa do País com ID que não existe no banco de dados.
2. O sistema responde a requisição informando que nenhum País com aquele código foi encontrado.

UC04: Excluir País

Descrição

Eu, como Administrador, quero excluir um País para manter a base consistente.

Pré-Condições

O País deve existir no banco de dados e ser encontrado (UC02).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para exclusão do País contendo ID.

2. O sistema atualiza o banco de dados com a data de exclusão do País.
3. O sistema responde a requisição com os dados do País atualizado com a data de exclusão.

Fluxo de Exceção

E1: País não existente

1. O usuário envia a requisição contendo o formulário para pesquisa do País com ID que não existe no banco de dados.
2. O sistema responde a requisição informando que nenhum País com aquele código foi encontrado.

E2: País com Estado atrelado

1. O usuário envia a requisição contendo o formulário para exclusão de País que possui um Estado atrelado.
2. O sistema responde a requisição informando que não é possível excluir País que possui Estados atrelados.

UC05: Criar Estado

Descrição

Eu, como Administrador, quero criar um Estado para possibilitar a configuração de um Endereço.

Pré-Condições

O Estado deve ser atrelado a um País previamente criado (UC01).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para criação do Estado contendo Nome, Código e Código do País.
2. O sistema persiste o formulário no banco de dados.
3. O sistema responde a requisição com os dados do Estado criado contendo seu ID.

Fluxo de Exceção

1. O usuário envia a requisição contendo o formulário para criação do Estado combinando Código e Código do País que já existe no banco de dados.
2. O sistema responde a requisição informando que já existe um Estado com aquele código com aquele País.

UC06: Encontrar Estado

Descrição

Eu, como Administrador, quero encontrar um Estado para verificar seus dados.

Pré-Condições

Não há.

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para pesquisar por um Estado dado seu Código e Código do País.
2. O sistema responde a requisição com os dados do Estado encontrado.

Fluxo de Exceção

1. O usuário envia a requisição contendo o formulário para pesquisa do com combinação de Código e Código do País que não existem no banco de dados.
2. O sistema responde a requisição informando que nenhum Estado com aquele código foi encontrado.

UC07: Atualizar Estado

Descrição

Eu, como Administrador, quero atualizar um Estado para manter seus dados corretos.

Pré-Condições

O Estado deve existir no banco de dados e ser encontrado (UC06).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para atualização do Estado contendo ID, Nome, Código e Código do País.
2. O sistema persiste o formulário no banco de dados.
3. O sistema responde a requisição com os dados do Estado atualizado.

Fluxo de Exceção

1. O usuário envia a requisição contendo o formulário para pesquisa do Estado com ID que não existe no banco de dados.
2. O sistema responde a requisição informando que nenhum Estado com aquele ID foi encontrado.

UC08: Excluir Estado

Descrição

Eu, como Administrador, quero excluir um Estado para manter a base consistente.

Pré-Condições

O Estado deve existir no banco de dados e ser encontrado (UC06).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para exclusão do Estado contendo ID.
2. O sistema atualiza o banco de dados com a data de exclusão do Estado.
3. O sistema responde a requisição com os dados do Estado atualizado com a data de exclusão.

Fluxo de Exceção

E1: Estado não existente

1. O usuário envia a requisição contendo o formulário para pesquisa do Estado com ID que não existe no banco de dados.
2. O sistema responde a requisição informando que nenhum Estado com aquele ID foi encontrado.

E2: Estado com Cidade atrelada

3. O usuário envia a requisição contendo o formulário para exclusão de País que possui um Estado atrelado.
4. O sistema responde a requisição informando que não é possível excluir País que possui Estados atrelados.
5.

UC09: Criar Cidade

Descrição

Eu, como Administrador, quero criar uma Cidade para possibilitar a configuração de um Endereço.

Pré-Condições

A Cidade deve ser atrelada a um Estado previamente criado (UC05).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para criação da Cidade contendo Nome e Código do Estado.
2. O sistema persiste o formulário no banco de dados.
3. O sistema responde a requisição com os dados da Cidade criada contendo seu ID.

Fluxo de Exceção

1. O usuário envia a requisição contendo o formulário para criação da Cidade combinando Nome e Código do Estado que já existe no banco de dados.
2. O sistema responde a requisição informando que já existe uma Cidade com aquele Nome e Estado.

UC10: Buscar Cidade

Descrição

Eu, como Administrador, quero buscar uma Cidade para verificar seus dados.

Pré-Condições

Não há.

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para pesquisar por uma Cidade dado seu Nome e Código do Estado.
2. O sistema responde a requisição com os dados da Cidade encontrada.

Fluxo de Exceção

1. O usuário envia a requisição contendo o formulário para pesquisa da Cidade com combinação de Nome e Código do Estado que não existe no banco de dados.
2. O sistema responde a requisição informando que nenhuma Cidade com aquele Nome e Código do estado foi encontrada.

UC11: Atualizar Cidade

Descrição

Eu, como Administrador, quero atualizar uma Cidade para manter seus dados corretos.

Pré-Condições

A Cidade deve existir no banco de dados e ser encontrada (UC10).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para atualização da Cidade contendo ID, Nome e Código do Estado.
2. O sistema persiste o formulário no banco de dados.
3. O sistema responde a requisição com os dados da Cidade atualizada.

Fluxo de Exceção

1. O usuário envia a requisição contendo o formulário para pesquisa da Cidade com ID que não existe no banco de dados.
2. O sistema responde a requisição informando que nenhuma Cidade com aquele ID foi encontrada.

UC12: Excluir Cidade

Descrição

Eu, como Administrador, quero excluir uma Cidade para manter a base consistente.

Pré-Condições

A Cidade deve existir no banco de dados e ser encontrada (UC10).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para exclusão da Cidade contendo ID.
2. O sistema atualiza o banco de dados com a data de exclusão da Cidade.
3. O sistema responde a requisição com os dados da Cidade atualizada com a data de exclusão.

Fluxo de Exceção

E1: Cidade não existente

1. O usuário envia a requisição contendo o formulário para pesquisa da Cidade com ID que não existe no banco de dados.
2. O sistema responde a requisição informando que nenhuma Cidade com aquele ID foi encontrada.

E2: Cidade com Endereço atrelado

1. O usuário envia a requisição contendo o formulário para exclusão de Cidade que possui um Endereço atrelado.
2. O sistema responde a requisição informando que não é possível excluir Cidade que possui Endereços atrelados.

UC13: Criar Endereço

Descrição

Eu, como Administrador, quero criar um Endereço para possibilitar a configuração de Marcas, Campanhas e Participações.

Pré-Condições

O Endereço deve ser atrelado a uma Cidade previamente criada (UC09).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para criação do Endereço contendo Nome da rua, Número, Complemento, CEP, Tipo do Endereço, Nome da Cidade e Código do Estado.
2. O sistema persiste o formulário no banco de dados.
3. O sistema responde a requisição com os dados do Endereço criado contendo seu ID.

Fluxo de Exceção

E1: Endereço não existente

1. O usuário envia a requisição contendo o formulário para do Endereço combinando Nome e Código do Estado que já existe no banco de dados.
2. O sistema responde a requisição informando que já existe uma Cidade com aquele Nome e Estado.

UC14: Encontrar Endereço

Descrição

Eu, como Administrador, quero encontrar um Endereço para verificar seus dados.

Pré-Condições

Não há.

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para pesquisar por um Endereço dado seu Código Postal e Número.
2. O sistema responde a requisição com os dados do Endereço encontrado.

Fluxo de Exceção

1. O usuário envia a requisição contendo o formulário para pesquisa do Endereço com combinação de Código Postal e Número que não existe no banco de dados.
2. O sistema responde a requisição informando que nenhum Endereço com aquele Código Postal e Número foi encontrado.

UC15: Atualizar Endereço

Descrição

Eu, como Administrador, quero atualizar um Endereço para manter seus dados corretos.

Pré-Condições

O Endereço deve existir no banco de dados e ser encontrado (UC14).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para atualização do Endereço contendo Nome da rua, Número, Complemento, CEP, Tipo do Endereço, Nome da Cidade e Código do Estado.
2. O sistema persiste o formulário no banco de dados.
3. O sistema responde a requisição com os dados do Endereço atualizado.

Fluxo de Exceção

1. O usuário envia a requisição contendo o formulário para pesquisa do Endereço com ID que não existe no banco de dados.
2. O sistema responde a requisição informando que nenhum Endereço com aquele ID foi encontrada.

UC16: Excluir Endereço

Descrição

Eu, como Administrador, quero excluir um Endereço para manter a base consistente.

Pré-Condições

O Endereço deve existir no banco de dados e ser encontrado (UC13).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para exclusão do Endereço contendo ID.
2. O sistema atualiza o banco de dados com a data de exclusão do Endereço.
3. O sistema responde a requisição com os dados do Endereço atualizado com a data de exclusão.

Fluxo de Exceção

E1: Cidade não existente

1. O usuário envia a requisição contendo o formulário para pesquisa do Endereço com ID que não existe no banco de dados.
2. O sistema responde a requisição informando que nenhum Endereço com aquele ID foi encontrado.

E2: Cidade com Endereço atrelado

1. O usuário envia a requisição contendo o formulário para pesquisa de Endereço com ID que possui um Participação, Campanha ou Marca atrelado.

2. O sistema responde a requisição informando que não é possível excluir um Endereço que possui um Participação, Campanha ou Marca atrelado.

UC17: Criar Marca

Descrição

Eu, como Administrador, quero criar uma Marca para possibilitar a configuração de Produtos e Campanhas.

Pré-Condições

A Marca deve ser atrelada a pelo menos um Endereço previamente criado (UC13).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para criação da Marca contendo Nome e Endereço.
1. O sistema persiste o formulário no banco de dados.
2. O sistema responde a requisição com os dados da Marca criada contendo seu ID.

Fluxo de Exceção

E1: Endereço não existente

1. O usuário envia a requisição contendo o formulário para do Endereço combinando Nome e Código do Estado que já existe no banco de dados.
2. O sistema responde a requisição informando que já existe uma Cidade com aquele Nome e Estado.

UC18: Encontrar Marca

Descrição

Eu, como Administrador, quero encontrar uma Marca para verificar seus dados.

Pré-Condições

Não há.

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para pesquisar por uma Marca dado seu Nome.
2. O sistema responde a requisição com os dados da Marca encontrada.

Fluxo de Exceção

1. O usuário envia a requisição contendo o formulário para pesquisa da Marca com um Nome que não existe no banco de dados.
2. O sistema responde a requisição informando que nenhuma Marca com aquele Nome foi encontrada.

UC19: Atualizar Marca

Descrição

Eu, como Administrador, quero atualizar uma Marca para manter seus dados corretos.

Pré-Condições

A Marca deve existir no banco de dados e ser encontrada (UC18).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para atualização da Marca contendo Nome e Endereço.
2. O sistema persiste o formulário no banco de dados.
3. O sistema responde a requisição com os dados da Marca atualizada.

Fluxo de Exceção

1. O usuário envia a requisição contendo o formulário para atualização da Marca com ID que não existe no banco de dados.
2. O sistema responde a requisição informando que nenhuma Marca com aquele ID foi encontrada.

UC20: Excluir Marca

Descrição

Eu, como Administrador, quero excluir uma Marca para manter a base consistente.

Pré-Condições

A Marca deve existir no banco de dados e ser encontrado (UC18).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para exclusão da Marca contendo ID.
2. O sistema atualiza o banco de dados com a data de exclusão da Marca.
3. O sistema responde a requisição com os dados da Marca atualizada com a data de exclusão.

Fluxo de Exceção

E1: Cidade não existente

1. O usuário envia a requisição contendo o formulário para pesquisa da Marca com ID que não existe no banco de dados.
2. O sistema responde a requisição informando que nenhuma Marca com aquele ID foi encontrada.

E2: Marca com Produto atrelado

1. O usuário envia a requisição contendo o formulário para pesquisa da Marca com ID que possui um Produtos ou Campanhas atreladas.
2. O sistema responde a requisição informando que não é possível excluir uma Marca que possui um Produtos ou Campanhas atreladas.

UC21: Criar Produto

Descrição

Eu, como Administrador, quero criar um Produto para possibilitar a configuração de Campanhas e Participações.

Pré-Condições

O Produto deve ser atrelado a uma Marca previamente criada (UC17).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para criação do Produto contendo Nome, EAN, *ClassOfGood* e *Manufacturer* (Marca).
2. O sistema persiste o formulário no banco de dados.
3. O sistema responde a requisição com os dados do Produto criado contendo seu ID.

Fluxo de Exceção

E1: Produto não existente

1. O usuário envia a requisição contendo o formulário para do Produto com EAN que já existe no banco de dados.
2. O sistema responde a requisição informando que já existe um Produto com aquele EAN na base.

UC22: Encontrar Produto

Descrição

Eu, como Administrador, quero encontrar um Produto para verificar seus dados.

Pré-Condições

Não há.

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para pesquisar por um Produto dado seu EAN.
2. O sistema responde a requisição com os dados do Produto encontrado.

Fluxo de Exceção

1. O usuário envia a requisição contendo o formulário para pesquisa do Produto com EAN que não existe no banco de dados.
2. O sistema responde a requisição informando que nenhum Produto com aquele EAN foi encontrado.

UC23: Atualizar Produto

Descrição

Eu, como Administrador, quero atualizar um Produto para manter seus dados corretos.

Pré-Condições

O Produto deve existir no banco de dados e ser encontrado (UC22).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para atualização do Produto contendo Nome, EAN, *ClassOfGood* e *Manufacturer* (Marca).
2. O sistema persiste o formulário no banco de dados.
3. O sistema responde a requisição com os dados do Produto atualizado.

Fluxo de Exceção

1. O usuário envia a requisição contendo o formulário para pesquisa do Produto com EAN que não existe no banco de dados.
2. O sistema responde a requisição informando que nenhum Produto com aquele EAN foi encontrada.

UC24: Excluir Produto

Descrição

Eu, como Administrador, quero excluir um Produto para manter a base consistente.

Pré-Condições

O Produto deve existir no banco de dados e ser encontrado (UC22).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para exclusão do Produto contendo ID.
2. O sistema atualiza o banco de dados com a data de exclusão do Produto.
3. O sistema responde a requisição com os dados do Produto atualizado com a data de exclusão.

Fluxo de Exceção

E1: Cidade não existente

1. O usuário envia a requisição contendo o formulário para pesquisa do Produto com ID que não existe no banco de dados.
2. O sistema responde a requisição informando que nenhum Produto com aquele ID foi encontrado.

E2: Cidade com Endereço atrelado

1. O usuário envia a requisição contendo o formulário para pesquisa de Produto com ID que possui um Participação ou Campanha.

2. O sistema responde a requisição informando que não é possível excluir um Produto que possui uma Participação ou Campanha atrelado.

UC25: Criar Campanha

Descrição

Eu, como Administrador, quero criar uma Campanha para possibilitar a configuração da mesma e a criação de Participações.

Pré-Condições

A Campanha deve ser atrelada a pelo menos uma Marca previamente criada (UC17).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para criação da Campanha contendo pelo menos o Nome, Código, Promotor (Marca), Tipo da Campanha, Data de Início da Validade, Data de Final da Validade, Data de Início da Compra, Data final da Compra.
2. O sistema persiste o formulário no banco de dados.
3. O sistema responde a requisição com os dados da Campanha criada contendo seu ID.

Fluxo de Exceção

E1: Código Duplicado

1. O usuário envia a requisição contendo o formulário para criação da Campanha com um código que já existe na base.
2. O sistema responde a requisição informando que já existe uma Campanha com aquele Código.

UC26: Encontrar Campanha

Descrição

Eu, como Administrador, quero encontrar uma Campanha para verificar seus dados.

Pré-Condições

Não há.

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para pesquisar por uma Campanha dado seu Código.
2. O sistema responde a requisição com os dados da Campanha encontrada.

Fluxo de Exceção

1. O usuário envia a requisição contendo o formulário para pesquisa da Campanha com um Código que não existe no banco de dados.
2. O sistema responde a requisição informando que nenhuma Campanha com aquele Código foi encontrada.

UC27: Atualizar Campanha

Descrição

Eu, como Administrador, quero atualizar uma Campanha para manter seus dados corretos.

Pré-Condições

A Campanha deve existir no banco de dados e ser encontrada (UC26).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para atualização da Campanha contendo as correções ou novas informações da mesma.
2. O sistema persiste o formulário no banco de dados.
3. O sistema responde a requisição com os dados da Campanha atualizada.

Fluxo de Exceção

1. O usuário envia a requisição contendo o formulário para atualização da Campanha com ID que não existe no banco de dados.
2. O sistema responde a requisição informando que nenhuma Campanha com aquele ID foi encontrada.

UC28: Excluir Campanha

Descrição

Eu, como Administrador, quero excluir uma Campanha para manter a base consistente.

Pré-Condições

A Campanha deve existir no banco de dados e ser encontrado (UC26).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para exclusão da Campanha contendo ID.
2. O sistema atualiza o banco de dados com a data de exclusão da Campanha.
3. O sistema responde a requisição com os dados da Campanha atualizada com a data de exclusão.

Fluxo de Exceção

E1: Campanha não existente

3. O usuário envia a requisição contendo o formulário para pesquisa da Campanha com ID que não existe no banco de dados.
4. O sistema responde a requisição informando que nenhuma Campanha com aquele ID foi encontrada.

E2: Campanha com Participação atrelada

3. O usuário envia a requisição contendo o formulário para pesquisa da Campanha com ID que possui um Participações atreladas.
4. O sistema responde a requisição informando que não é possível excluir uma Campanha que possui Participações atreladas.

UC29: Adicionar Produtos à Campanha

Descrição

Eu, como Administrador, quero adicionar Produtos à uma Campanha para que atenda as exigências do contratante (Marca) e possibilitar a criação de Participações.

Pré-Condições

A Campanha deve existir no banco de dados e ser encontrada (UC26).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para atualização da Campanha contendo o ID da Campanha e os IDs dos Produtos que a compõe.
2. O sistema atualiza o banco de atrelando aqueles Produtos àquela Campanha.
3. O sistema responde a requisição com a data e os dados da Campanha atualizada.

Fluxo de Exceção

E1: Campanha não existente

1. O usuário envia a requisição contendo o formulário para atualização da Campanha com ID que não existe no banco de dados.
2. O sistema responde a requisição informando que nenhuma Campanha com aquele ID foi encontrada.

E2: Produto não existente

1. O usuário envia a requisição contendo o formulário para atualização da Campanha com Ids de Produtos que não existem no banco de dados.
2. O sistema responde a requisição informando que não é possível encontrar os Produtos para adicioná-los à Campanha.

UC30: Listar Participações Verificáveis

Descrição

Eu, como Administrador, quero encontrar Participações verificáveis para conferir se os dados informados pelo participante estão corretos.

Pré-Condições

A Participação deve existir no banco de dados e ser encontrada (UC34).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para busca das Participações verificáveis (estado "*VALIDATION_QUEUE*") dado o ID da Campanha.
2. O sistema responde a requisição com a próxima Participação com o estado solicitado, ordenado por ID da Participação.

Fluxo de Exceção

E1: Campanha não existente

1. O usuário envia a requisição contendo o formulário busca de Participações da Campanha com ID da Campanha que não existe no banco de dados.
2. O sistema responde a requisição informando que nenhuma Campanha com aquele ID foi encontrada.

E2: Participação não existente

1. O usuário envia a requisição contendo o formulário busca de Participações da Campanha com ID da Campanha que não existe no banco de dados.
2. O sistema responde a requisição informando que não há nenhuma Participação com o estado solicitado.

UC31: Verificar Participação

Descrição

Eu, como Administrador, quero informar ao sistema as violações encontradas durante a verificação manual de uma Participação.

Pré-Condições

A Participação deve existir no banco de dados e ser encontrada (UC34).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia o contendo as violações encontradas para uma Participação dado seu ID.
2. O sistema responde a requisição com o *HTTP* estado 200.

Fluxo de Exceção

E1: Participação não existente

1. O usuário envia a requisição contendo o formulário de violações para uma Participações com ID que não existe no banco de dados.
2. O sistema responde a requisição informando que não há nenhuma Participação com o ID informado.

UC32: Pagar Participação

Descrição

Eu, como Administrador, quero informar ao sistema que o pagamento da Participação foi efetuado.

Pré-Condições

A Participação deve existir no banco de dados e ser encontrada (UC34).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia formulário para o sistema contendo o ID da Participação que foi paga.
2. O sistema responde com os dados atualizados da Participação.

Fluxo de Exceção

E1: Participação não existente

1. O usuário envia a requisição contendo o formulário para pagamento de uma Participação com ID que não existe no banco de dados.
2. O sistema responde a requisição informando que não há nenhuma Participação com o ID informado.

UC33: Extrair Relatório de Participação por Campanha

Descrição

Eu, como Administrador, quero extrair o Relatório de Participação por Campanha para informar os dados ao Contratante (Marca).

Pré-Condições

A Campanha deve existir no banco de dados e ser encontrada (UC26).

Ator primário

Administrador

Fluxo Principal

1. O usuário envia formulário para o sistema contendo o ID da Campanha para extração do relatório.
2. O sistema responde com os dados e estatísticas da Participação.

Fluxo de Exceção

E1: Campanha não existente

1. O usuário envia a requisição contendo o formulário para extração do Relatório de Participações da Campanha com ID da Campanha que não existe no banco de dados.
2. O sistema responde a requisição informando que nenhuma Campanha com aquele ID foi encontrada.

UC34: Criar Participação

Descrição

Eu, como Participante, quero informar ao sistema meus dados de contato, endereço e dados do Produto para que minha Participação seja processada.

Pré-Condições

Não há.

Ator primário

Participante

Fluxo Principal

1. O usuário envia a requisição contendo o formulário contendo os dados de contato, Endereço e Produto da sua Participação.
2. O sistema persiste os dados no banco de dados.
3. O sistema responde a requisição com a Participação criada e seu ID, com estado "*NOT_PROCESSED"*.

Fluxo de Exceção

E1: Participação não existente

1. O usuário envia a requisição contendo o formulário de violações para uma Participações com ID que não existe no banco de dados.

O sistema responde a requisição informando que não há nenhuma Participação com o ID informado.

UC35: Submeter Arquivo da Nota Fiscal

Descrição

Eu, como Participante, quero submeter o arquivo da Nota Fiscal para que minha Participação seja processada.

Pré-Condições

A Participação deve existir (UC34).

Ator primário

Participante

Fluxo Principal

1. O usuário envia o arquivo (imagem) da Nota Fiscal do Produto informado em sua Participação.
2. O sistema persiste os dados no banco de dados.

3. O sistema processa a Participação (UC39) e responde a requisição com os dados da Participação, com estado “*VALIDATION_QUEUE”*.
4. O sistema prepara o envio do e-mail de notificação ao participante (UC40).

Fluxo Alternativo

A1: Uma Violação Identificada

1. O usuário envia o arquivo (imagem) da Nota Fiscal do Produto informado em sua Participação.
2. O sistema persiste os dados no banco de dados.
3. O sistema processa a Participação (UC41) e responde a requisição com os dados da Participação com estado da violação identificada.
4. O sistema prepara a Participação para a correção de dados (UC42).
5. O sistema prepara o envio do e-mail de notificação ao participante (UC40).

A2: Várias Violações Identificadas

1. O usuário envia o arquivo (imagem) da Nota Fiscal do Produto informado em sua Participação.
2. O sistema persiste os dados no banco de dados.
3. O sistema processa a Participação (UC41) e responde a requisição com os dados da Participação com estado “*INVALID”*.
4. O sistema prepara a Participação para a correção de dados (UC42).
5. O sistema prepara o envio do e-mail de notificação ao participante (UC40).

A3: Nenhuma Campanha Identificada

1. O usuário envia o arquivo (imagem) da Nota Fiscal do Produto informado em sua Participação.
2. O sistema persiste os dados no banco de dados.
3. O sistema processa a Participação (UC39) e responde a requisição com os dados da Participação com estado “*NO_CAMPAIGN”*.
4. O sistema prepara o envio do e-mail de notificação ao participante (UC40).

Fluxo de Exceção

E1: Participação não existente

1. O usuário envia o arquivo uma Participações com ID que não existe no banco de dados.
2. O sistema informa o usuário que não foi possível submeter um arquivo para uma Participação não encontrada.

UC36: Corrigir Participação

Descrição

Eu, como Participante, quero reenviar os dados de contato, endereço ou dados do Produto para que minha Participação seja processada.

Pré-Condições

A Participação deve existir (UC34).

Ator primário

Participante

Fluxo Principal

1. O usuário envia a requisição contendo o formulário contendo os dados de contato, Endereço ou Produto da sua Participação com o Código de Correção gerado pelo sistema.
2. O sistema verifica a validade do Código de Correção informado.
3. O sistema persiste os dados no banco de dados.
4. O sistema processa a Participação (UC39) e responde a requisição com os dados da Participação, com estado "*VALIDATION_QUEUE"*.
5. O sistema prepara o envio do e-mail de notificação ao participante (UC40).

Fluxo Alternativo

A1: Arquivo da Nota Fiscal

1. O usuário envia a requisição contendo o arquivo da nota fiscal da sua Participação com o Código de Correção gerado pelo sistema.
2. O sistema verifica a validade do Código de Correção informado.
3. O sistema persiste os dados no banco de dados.
4. O sistema processa a Participação (UC39) e responde a requisição com os dados da Participação, com estado "*VALIDATION_QUEUE"*.
5. O sistema prepara o envio do e-mail de notificação ao participante (UC40).

A1: Uma Violação Identificada

1. O usuário envia a requisição contendo o formulário contendo os dados de contato, Endereço ou Produto da sua Participação com o Código de Correção gerado pelo sistema.
2. O sistema verifica a validade do Código de Correção informado.
3. O sistema persiste os dados no banco de dados.

4. O sistema processa a Participação (UC31) e responde a requisição com os dados da Participação com estado da violação identificada.
5. O sistema prepara a Participação para a correção de dados (UC42).
6. O sistema prepara o envio do e-mail de notificação ao participante (UC40).

A2: Várias Violações Identificadas

1. O usuário envia a requisição contendo o formulário contendo os dados de contato, Endereço ou Produto da sua Participação com o Código de Correção gerado pelo sistema.
2. O sistema verifica a validade do Código de Correção informado.
3. O sistema persiste os dados no banco de dados.
4. O sistema processa a Participação (UC39) e responde a requisição com os dados da Participação com estado "*INVALID*".
5. O sistema prepara a Participação para a correção de dados (UC42).
6. O sistema prepara o envio do e-mail de notificação ao participante (UC40).

Fluxo de Exceção

E1: Código de Correção não existente

1. O usuário envia a requisição contendo o formulário contendo os dados de contato, Endereço ou Produto da sua Participação com o Código de Correção gerado pelo sistema.
2. O sistema verifica a validade do Código de Correção informado.
3. O sistema responde a requisição informando que o Código de Correção é inválido.

UC37: Encontrar Campanha Pela Marca

Descrição

Eu, como Participante, quero encontrar as Campanha promovidas por uma Marca para saber se posso participar de alguma.

Pré-Condições

Não há.

Ator primário

Participante

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para pesquisar por uma Campanha o nome da Marca promotora.

2. O sistema responde a requisição com os dados de todas as Campanhas promovidas pela Marca informada encontradas.

Fluxo de Exceção

Não há

UC38: Encontrar Campanha Pelo Produto

Descrição

Eu, como Participante, quero encontrar as Campanha que contém um Produto específico para saber se posso participar de alguma.

Pré-Condições

Não há.

Ator primário

Participante

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para pesquisar por uma Campanha o EAN do Produto participante.
2. O sistema responde a requisição com os dados de todas as Campanhas promovidas pela Marca informada encontradas.

Fluxo de Exceção

Não há

UC39: Reprocessar Participação

Descrição

Eu, como Agendador de Tarefas, quero reprocessar uma Participação para definir seu estado atual.

Pré-Condições

A Participação deve existir (UC34).

Ator primário

Agendador de Tarefas

Fluxo Principal

1. O usuário envia a requisição contendo o formulário para reprocessar uma Participação dado seu ID.
2. O Agendador de Tarefas recupera os dados da Participação presentes na base de dados.

3. O Agendador de Tarefas verifica o conjunto de datas, produto, para encontrar uma Campanha que seja adequada à Participação.
4. O Agendador de Tarefas verifica se há inconformidades de data, produto, localidade para os dados informados e a Campanha identificada.
5. O Agendador de Tarefas verifica se a Participação está com o estado "*VALID*".
6. O Agendador de Tarefas verifica se a Participação está com o estado "*PAID*".
7. O Agendador de Tarefas verifica se a Participação está com o estado "*CLOSED*".
8. O Agendador de Tarefas adiciona a Participação na fila para verificação ("*VALIDATION_QUEUE*").
9. O Agendador de Tarefas prepara o envio do e-mail de notificação ao participante (UC40).

Fluxo Alternativo

A1: Nenhuma Campanha Identificada

1. O usuário envia a requisição contendo o formulário para reprocessar uma Participação dado seu ID.
2. O Agendador de Tarefas recupera os dados da Participação presentes na base de dados.
3. O Agendador de Tarefas verifica se a Participação possui uma Campanha.
4. O Agendador de Tarefas atualiza a Participação com estado "*NO_CAMPAIGN*" no banco de dados.
5. O sistema prepara o envio do e-mail de notificação ao participante (UC40).

A2: Uma Violação Identificada

1. O usuário envia a requisição contendo o formulário para reprocessar uma Participação dado seu ID.
2. O Agendador de Tarefas recupera os dados da Participação presentes na base de dados.
3. O Agendador de Tarefas verifica se a Participação possui quaisquer Violações aplicáveis à Campanha, tais como violações de datas, produto, localidade, duplicidade.
4. O Agendador de Tarefas atualiza a Participação com o estado específico da Violação encontrada na base de dados ("DUPLICATED",

"BAD_PURCHASE_DATE", "BAD_REGISTRATION_DATE", "BAD_LOCALE", "BAD_TRADER", "BAD_PRODUCT", "BAD_INVOICE").

5. O Agendador de Tarefas prepara a Participação para a correção de dados (UC42).
6. O Agendador de Tarefas prepara o envio do e-mail de notificação ao participante (UC40).

A3: Várias Violações Identificadas

1. O usuário envia a requisição contendo o formulário para reprocessar uma Participação dado seu ID.
2. O Agendador de Tarefas recupera os dados da Participação presentes na base de dados.
3. O Agendador de Tarefas verifica se a Participação possui quaisquer Violações (participação duplicada, problemas nas datas, produto, endereço ou nota fiscal).
4. O Agendador de Tarefas atualiza a Participação com o estado "*INVALID*" na base de dados.
5. O Agendador de Tarefas prepara a Participação para a correção de dados (UC42).
6. O Agendador de Tarefas prepara o envio do e-mail de notificação ao participante (UC40).

A4: Nenhuma Violação Identificada

1. O usuário envia a requisição contendo o formulário para reprocessar uma Participação dado seu ID.
2. O Agendador de Tarefas recupera os dados da Participação presentes na base de dados.
3. O Agendador de Tarefas verifica que a Participação não possui quaisquer Violações (participação duplicada, problemas nas datas, produto, endereço ou nota fiscal).
4. O Agendador de Tarefas atualiza a Participação com o estado "*VALID*" na base de dados.
5. O Agendador de Tarefas prepara a Participação para a correção de dados (UC42).
6. O Agendador de Tarefas prepara o envio do e-mail de notificação ao participante (UC40).

A5: Participação Paga.

1. O usuário envia a requisição contendo o formulário para reprocessar uma Participação dado seu ID.
2. O Agendador de Tarefas recupera os dados da Participação presentes na base de dados.
3. O Agendador de Tarefas verifica se a participação está com o estado "*PAID*".
4. O Agendador de Tarefasatualiza a Participação com o estado "*PAID*" na base de dados.
5. O Agendador de Tarefas prepara a Participação para a correção de dados (UC42).
6. O Agendador de Tarefas prepara o envio do e-mail de notificação ao participante (UC40).

A6: Participação Encerrada.

1. O usuário envia a requisição contendo o formulário para reprocessar uma Participação dado seu ID.
2. O Agendador de Tarefas recupera os dados da Participação presentes na base de dados.
3. O Agendador de Tarefas verifica se a participação está encerrada.
4. O Agendador de Tarefas atualiza a Participação com o estado "*CLOSED*" na base de dados.
5. O Agendador de Tarefas prepara a Participação para a correção de dados (UC42).
6. O Agendador de Tarefas prepara o envio do e-mail de notificação ao participante (UC40).

Fluxo de Exceção

Não há

UC40: Enviar Notificação por E-mail

Descrição

Eu, como Agendador de Tarefas, quero enviar notificações por e-mail ao Participante com atualizações de suas Participações.

Pré-Condições

Não há.

Ator primário

Agendador de Tarefas.

Fluxo Principal

1. Quando o sistema atualiza uma Participação na base de dados.
2. O Agendador de Tarefas prepara e envia o e-mail de notificação baseado no estado atual da Participação atualizada.

Fluxo de Exceção

Não há.

UC41: Encerrar Participação

Descrição

Eu, como Agendador de Tarefas, quero encerrar uma Participação para que seus dados sejam anonimizados.

Pré-Condições

A Participação deve estar com o *estado* “PAID” (UC32).

Ator primário

Agendador de Tarefas.

Fluxo Principal

1. O Agendador de Tarefas encontra periodicamente Participações com *estado* “PAID”.
2. O Agendador de Tarefas anonimiza e persiste os dados do participante.
3. O Agendador de Tarefas prepara e envia o e-mail de notificação baseado no estado atual da Participação atualizada (UC40).

Fluxo de Exceção

Não há.

UC42: Preparar Correção de Dados

Descrição

Eu, como Agendador de Tarefas, quero gerar o Código de Correção para uma Participação avaliada negativamente e enviá-lo ao participante.

Pré-Condições

A Participação deve estar com o *estado* “*INVALID*” (UC31).

Ator primário

Agendador de Tarefas.

Fluxo Principal

1. O Sistema gera um Código de Correção único para correção.
2. O Sistema atrela o Código de Correção à Participação no banco de dados.
3. O sistema prepara e envia o e-mail de notificação contendo o Código de Correção da Participação (UC40).

Fluxo Alternativo

A1: Máximo de tentativas atingido

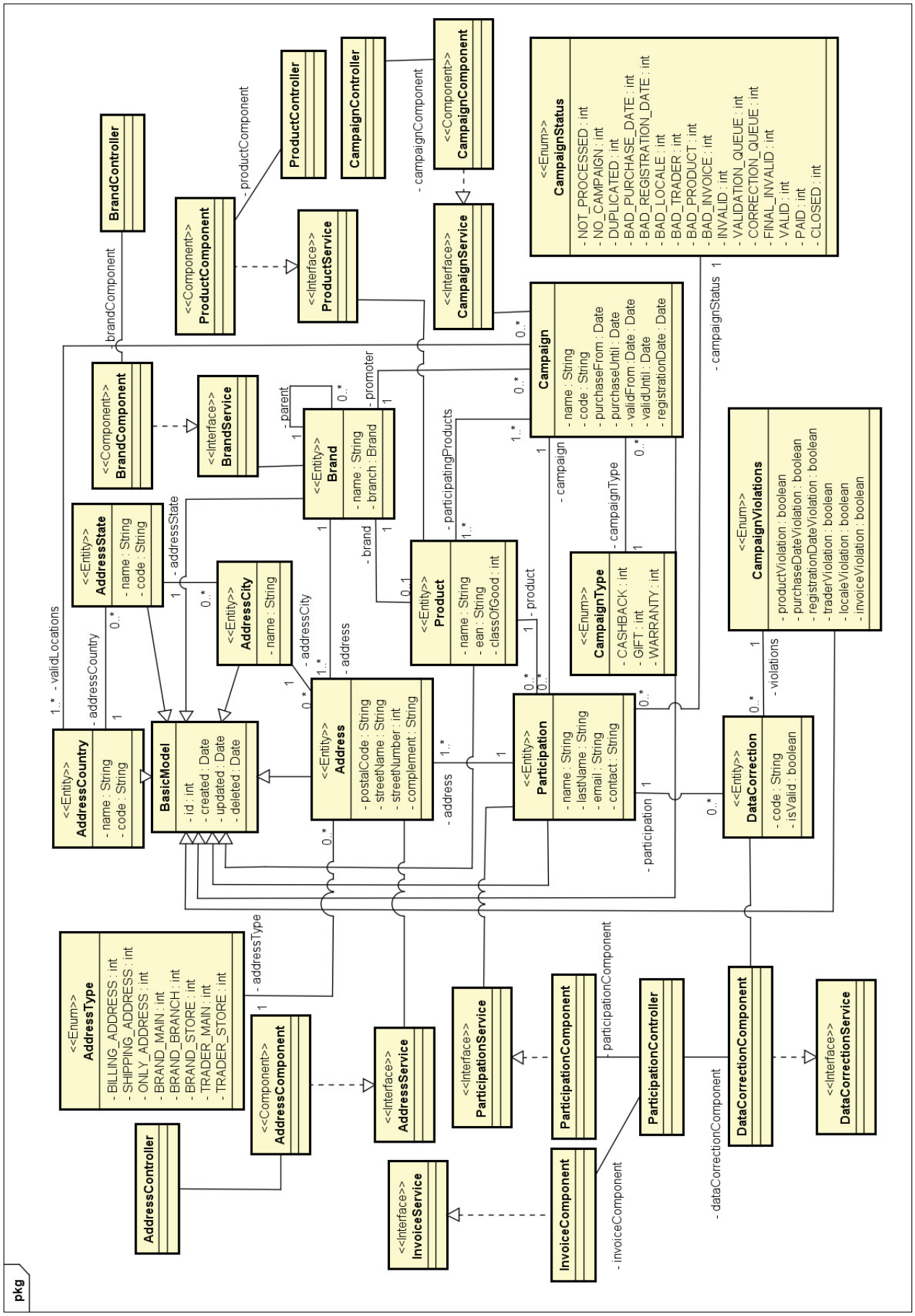
1. O Agendador de Tarefas identifica que a Participação já possui três Códigos de Correção atrelados no banco de dados.
2. O Agendador de Tarefas atualiza a Participação com o *estado* "*FINAL_INVALID*" no banco de dados.
3. O Agendador de Tarefas prepara e envia o e-mail de notificação baseado no estado atual da Participação atualizada (UC40).

Fluxo de Exceção

Não há.

APÊNDICE H – DIAGRAMA DE CLASSES

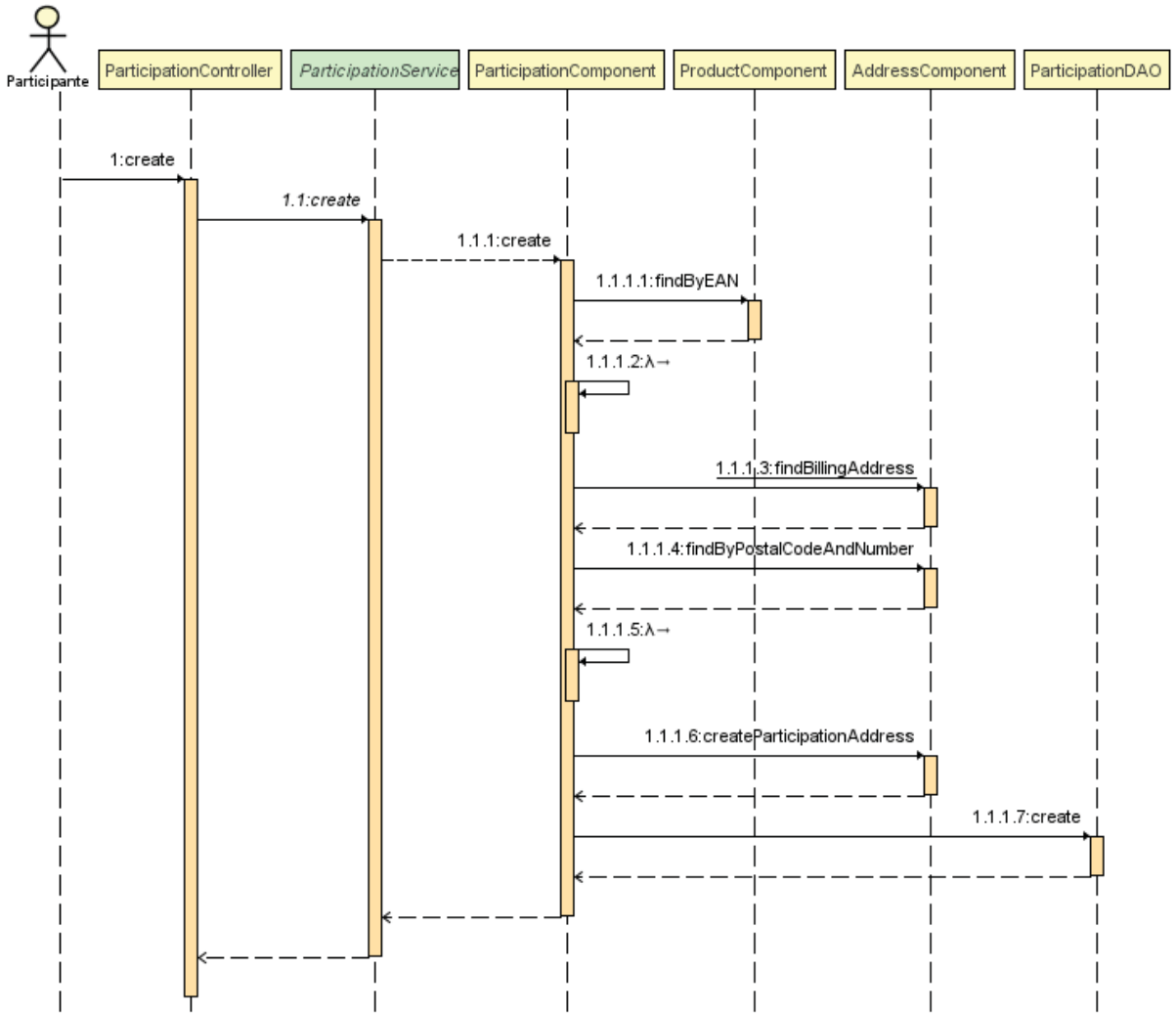
Figura 32 - Diagrama de Classes e Serviços



Fonte: O Autor (2022)

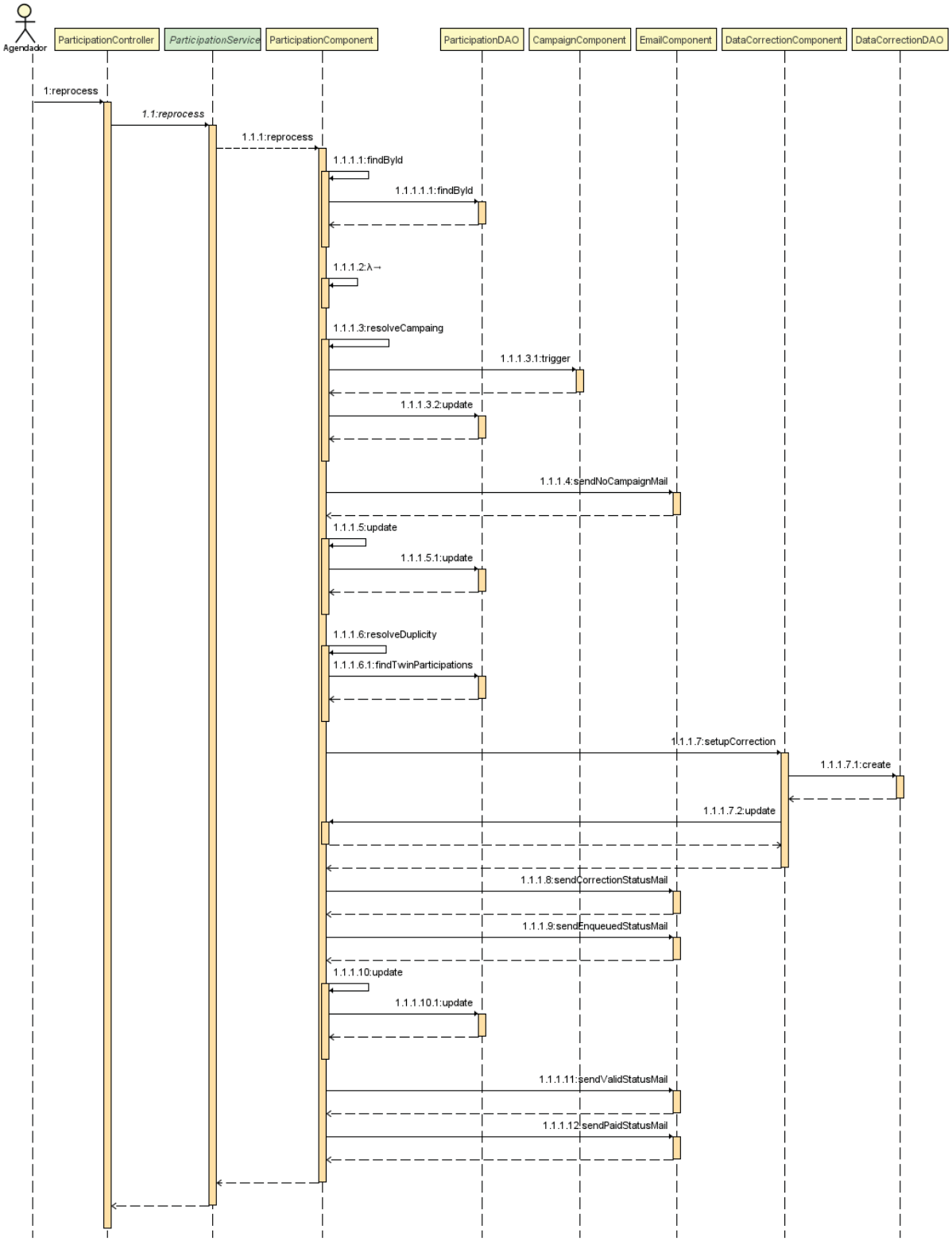
APÊNDICE I – DIAGRAMAS DE SEQUÊNCIA

Figura 33 - Diagrama de sequência para UC34: Criar Participação



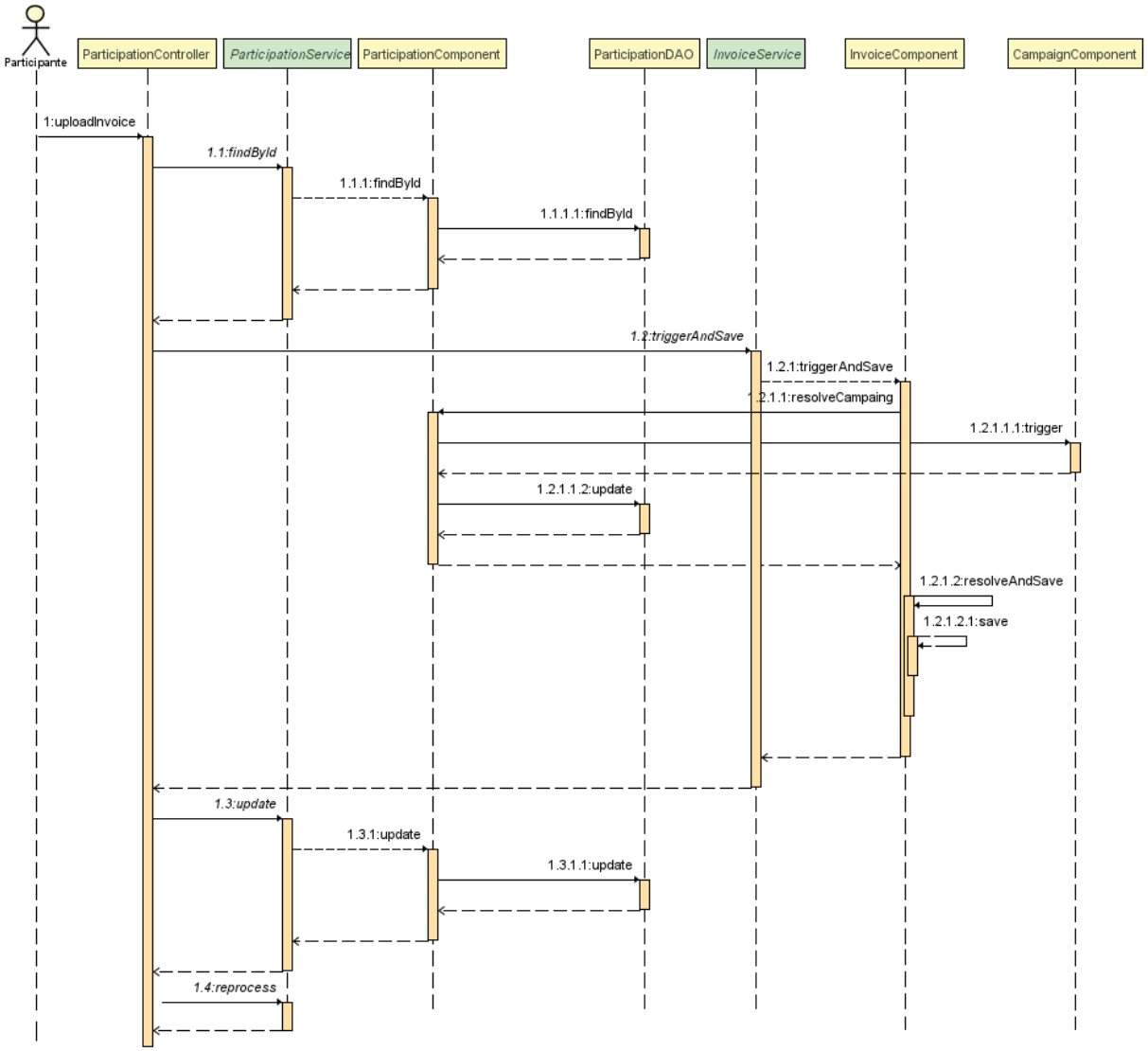
Fonte: O Autor (2022)

Figura 34 - Diagrama de sequência para UC39: Reprocessar Participação



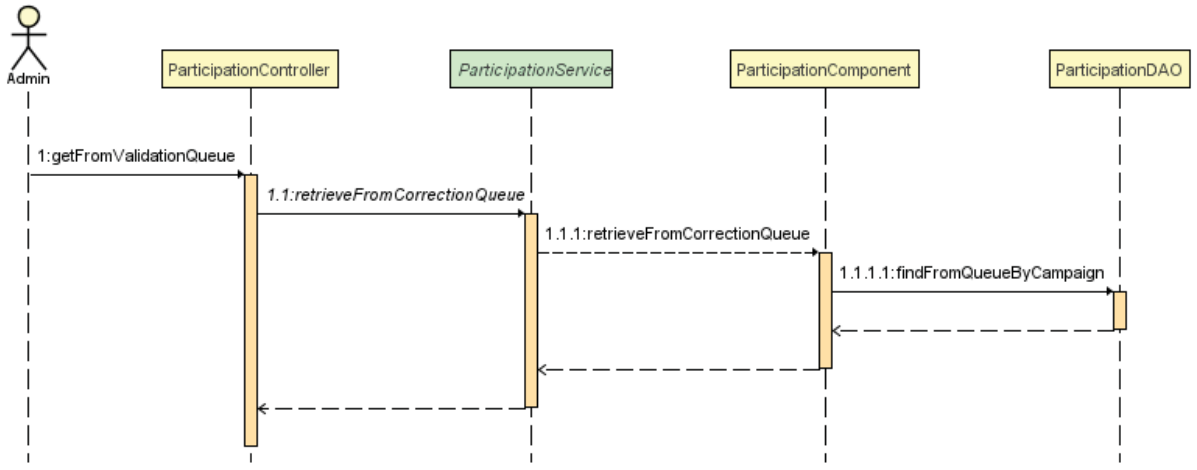
Fonte: O Autor (2022)

Figura 39 - Diagrama de sequência para UC35: Submeter Nota Fiscal



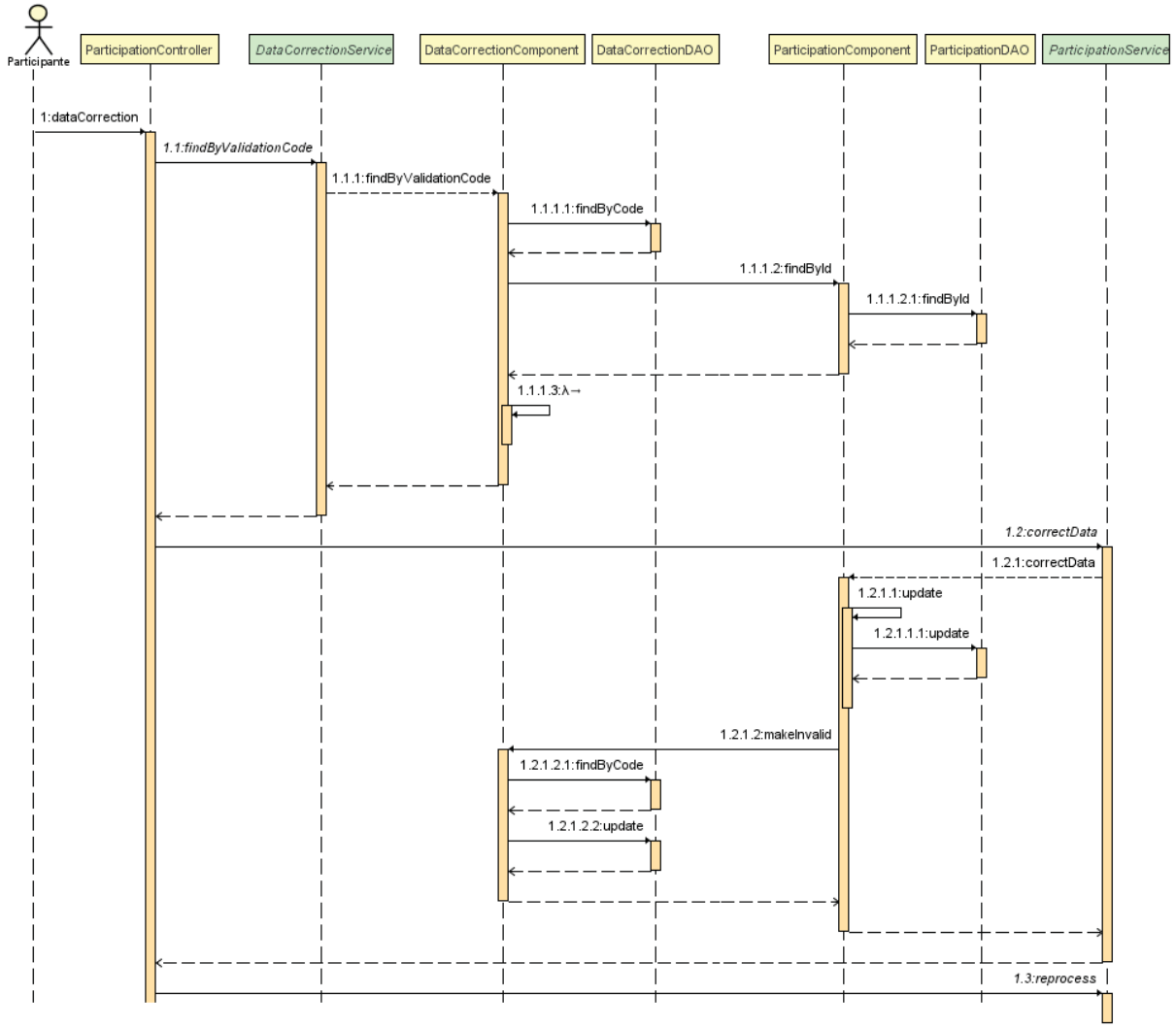
Fonte: O Autor (2022)

Figura 35 - Diagrama de sequência para UC30: Listar Participações Verificáveis



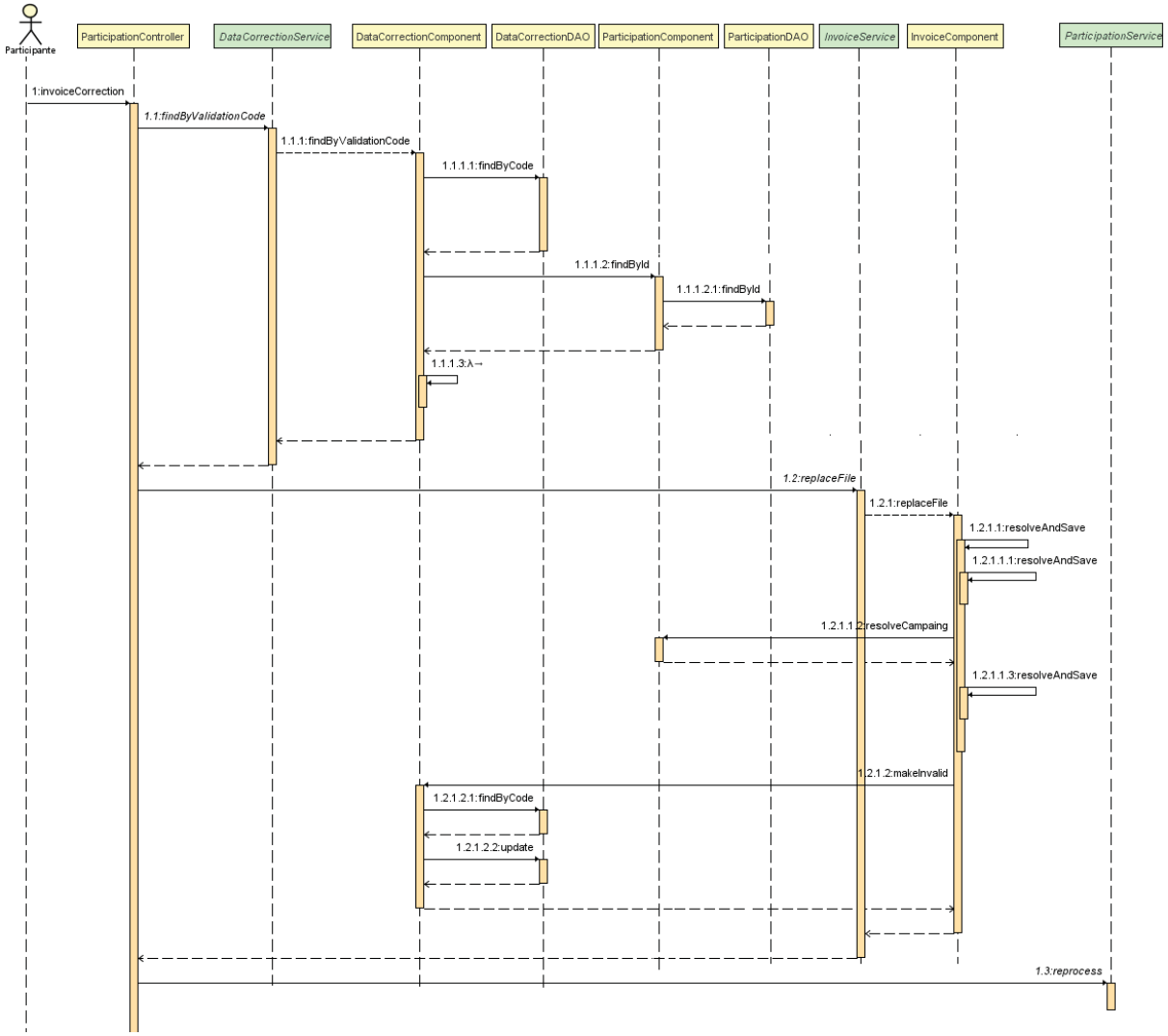
Fonte: O Autor (2022)

Figura 36 - Diagrama de sequência para UC36: Corrigir Participação



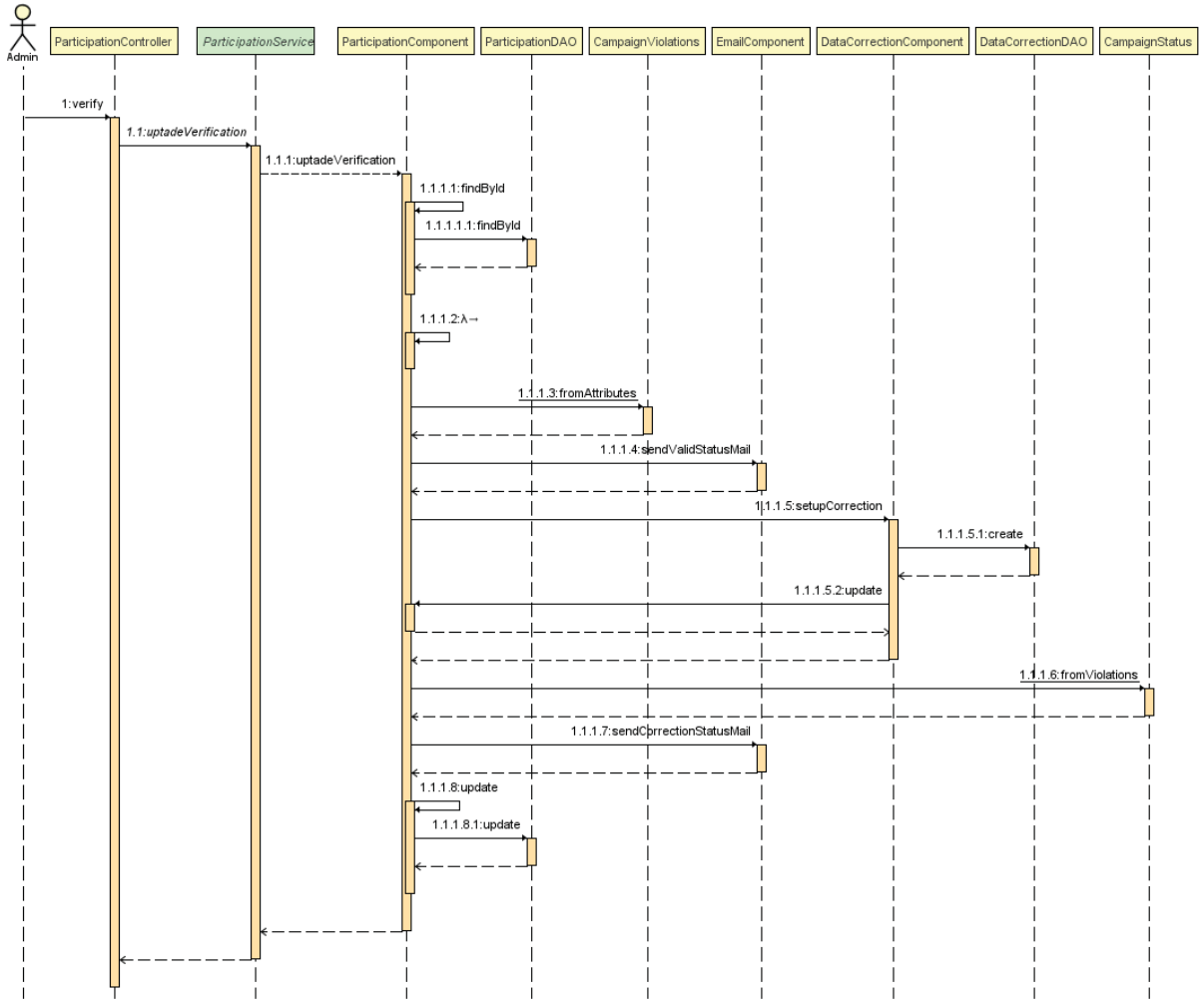
Fonte: O Autor (2022)

Figura 37 - Diagrama de sequência para UC36: Corrigir Participação (Nota Fiscal)



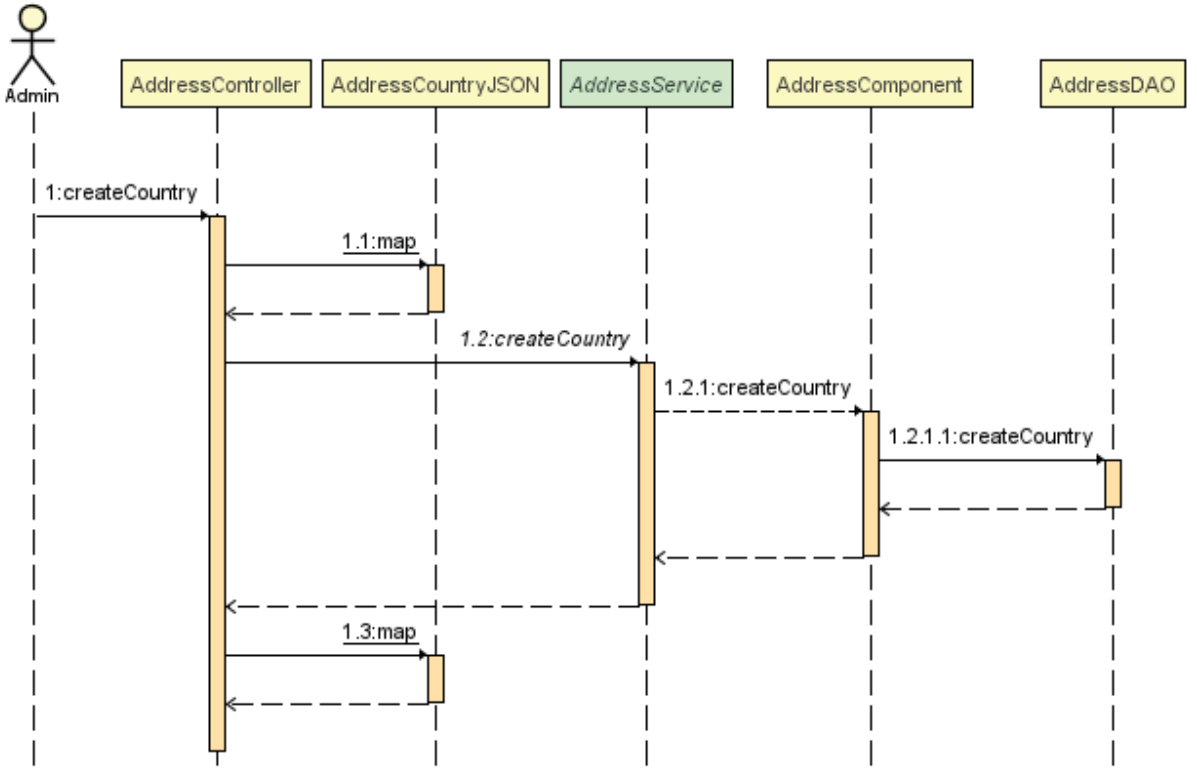
Fonte: O Autor (2022)

Figura 38 - Diagrama de sequência para UC31: Verificar Participação



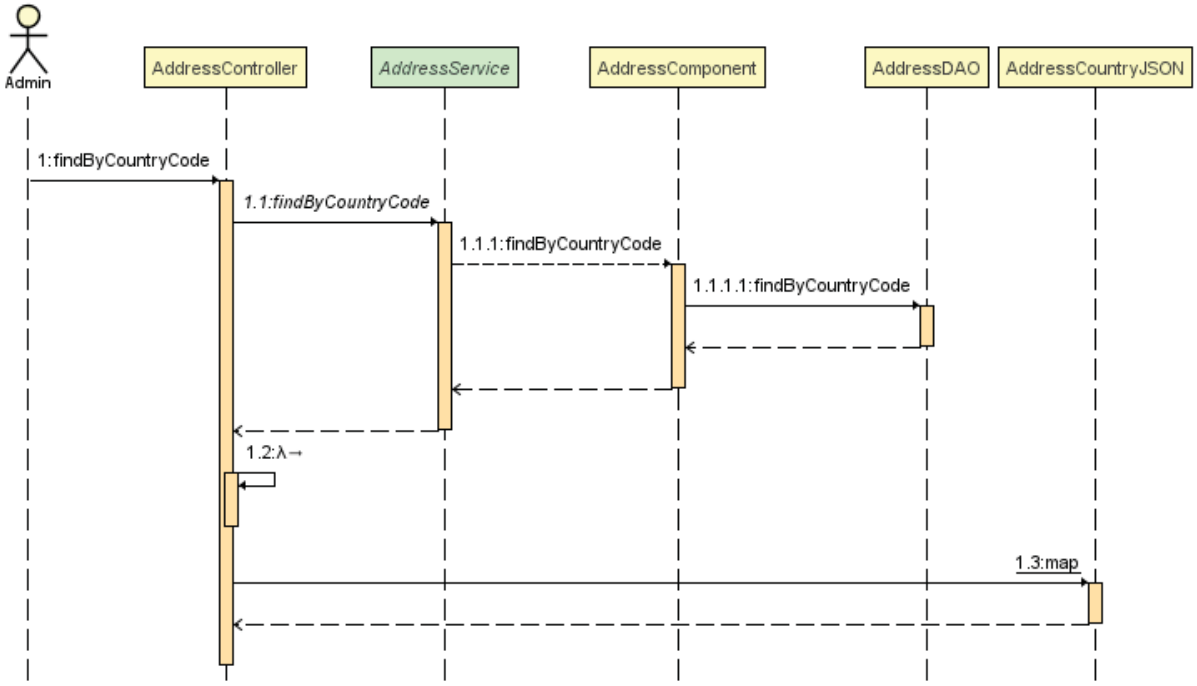
Fonte: O Autor (2022)

Figura 39 - Diagrama de sequência para UC01: Criar País



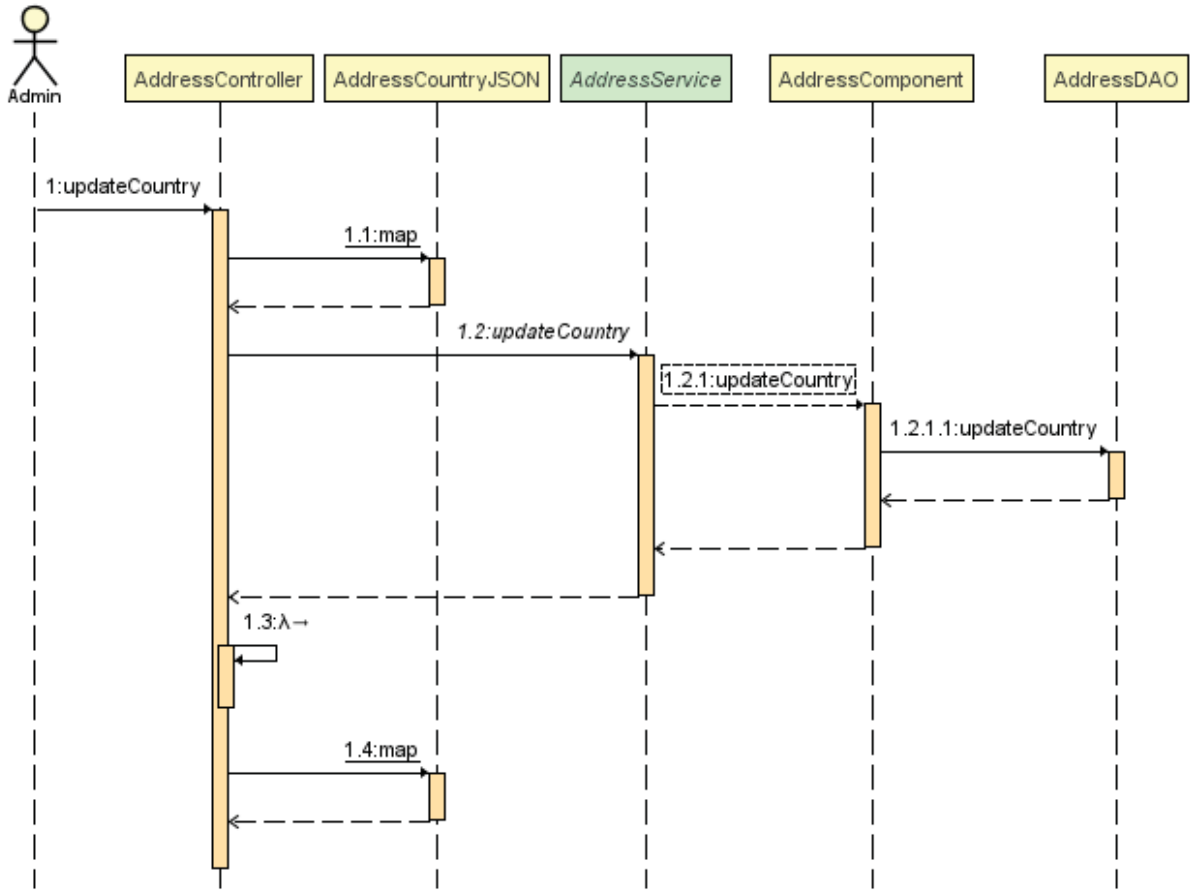
Fonte: O Autor (2022)

Figura 40 - Diagrama de sequência para UC02: Encontrar País



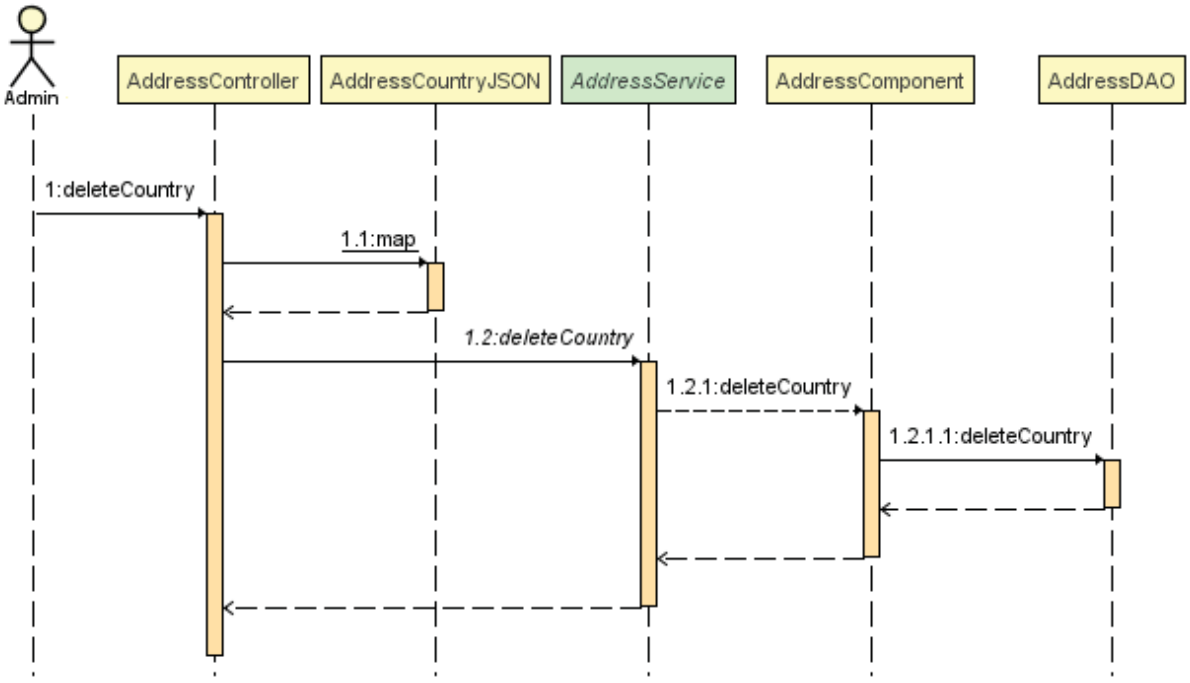
Fonte: O Autor (2022)

Figura 41 - Diagrama de sequência para UC03: Atualizar País



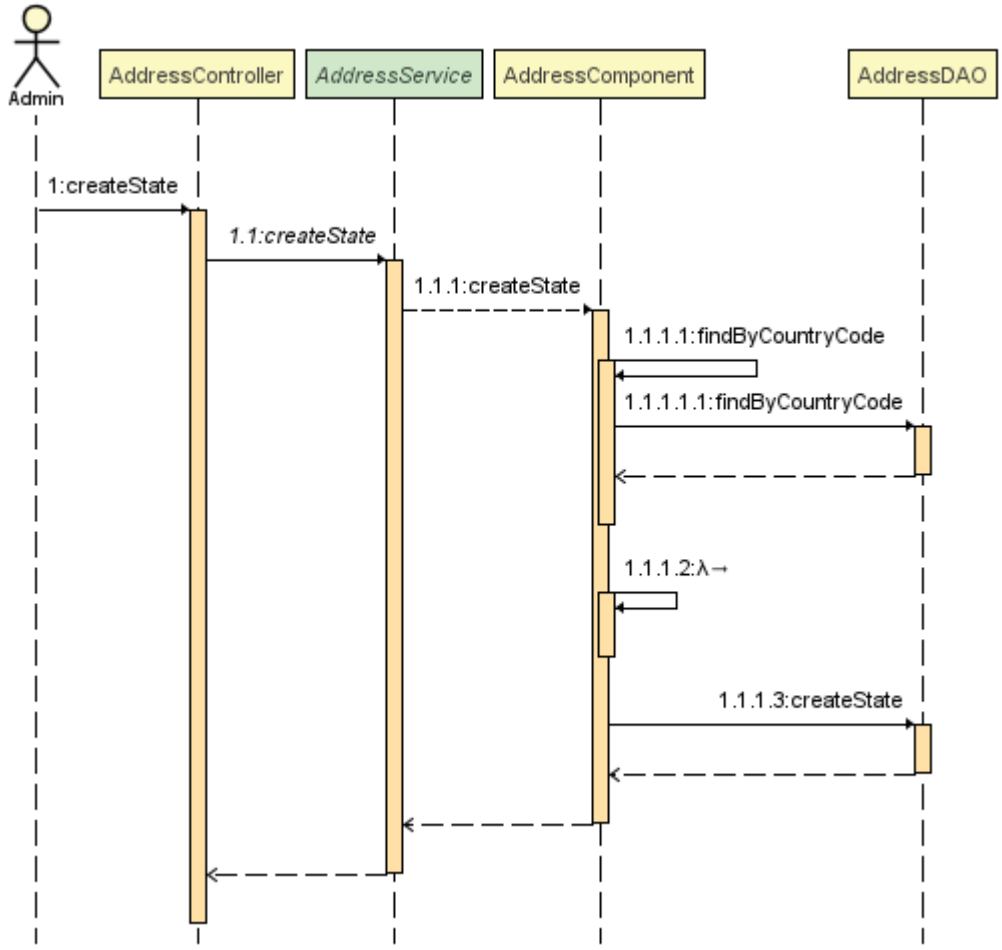
Fonte: O Autor (2022)

Figura 42 - Diagrama de sequência para a UC04: Excluir País



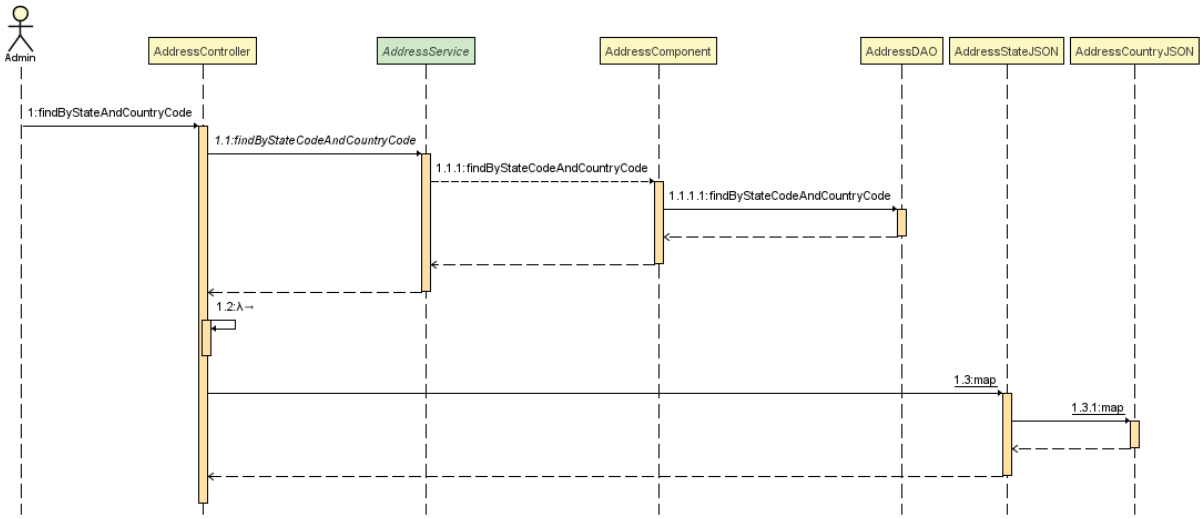
Fonte: O Autor (2022)

Figura 43 - Diagrama de sequência para UC05: Criar Estado



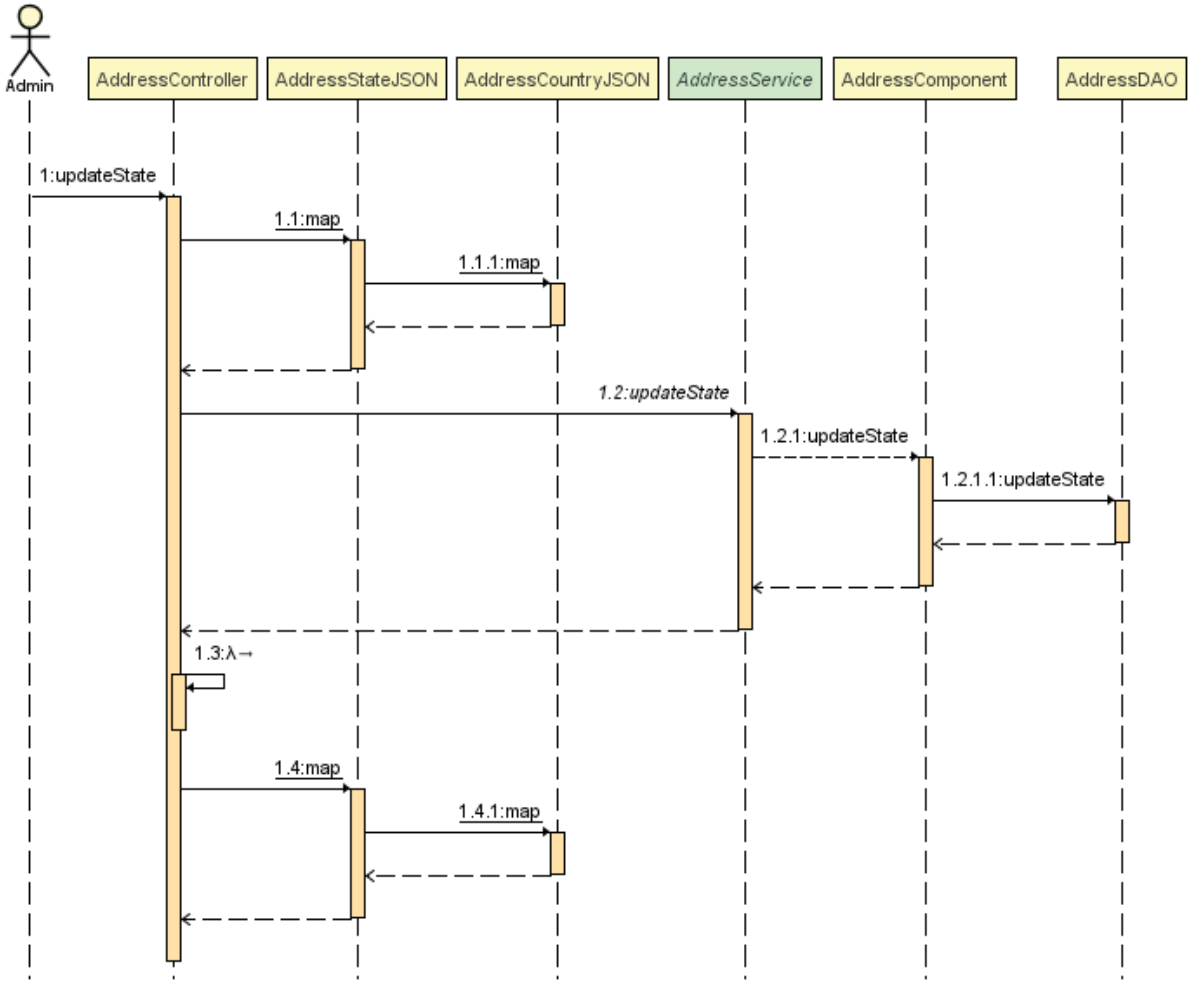
Fonte: O Autor (2022)

Figura 49 - Diagrama de sequência para UC06: Encontrar Estado



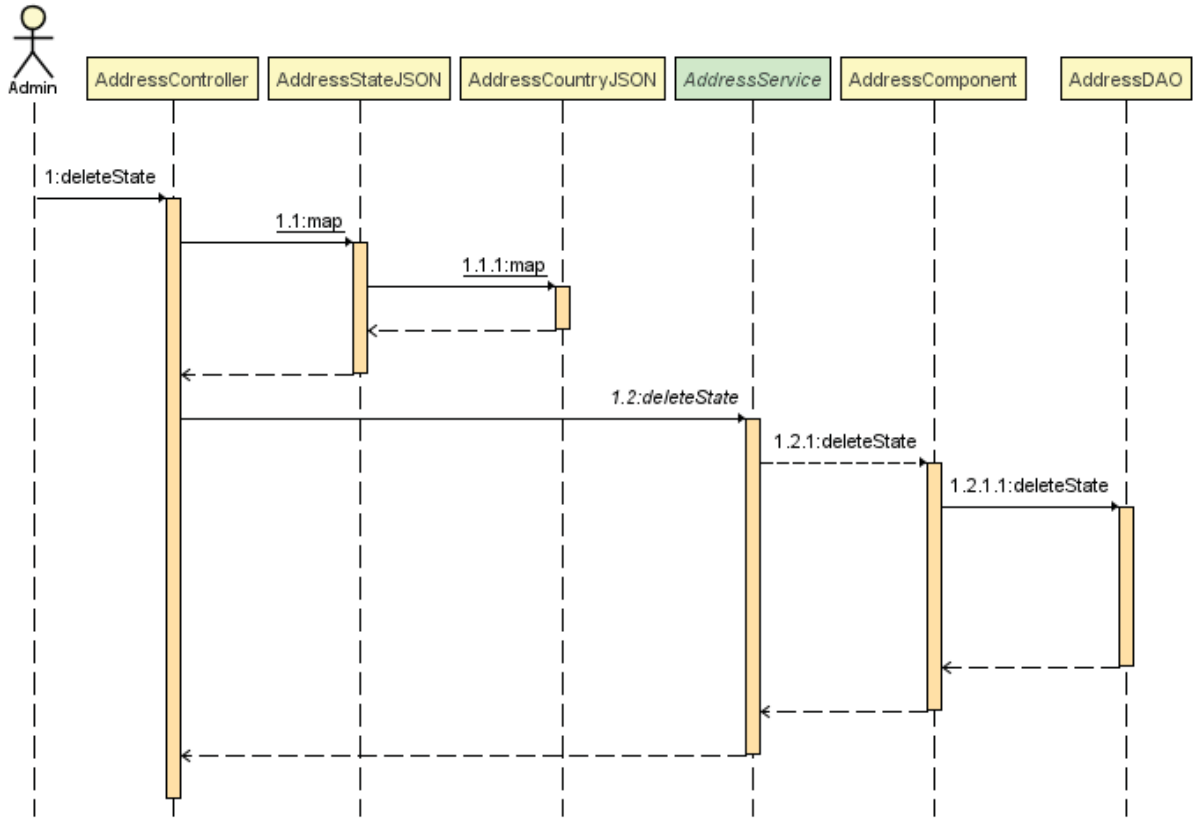
Fonte: O Autor (2022)

Figura 44 - Diagrama de sequência para UC07: Atualizar Estado



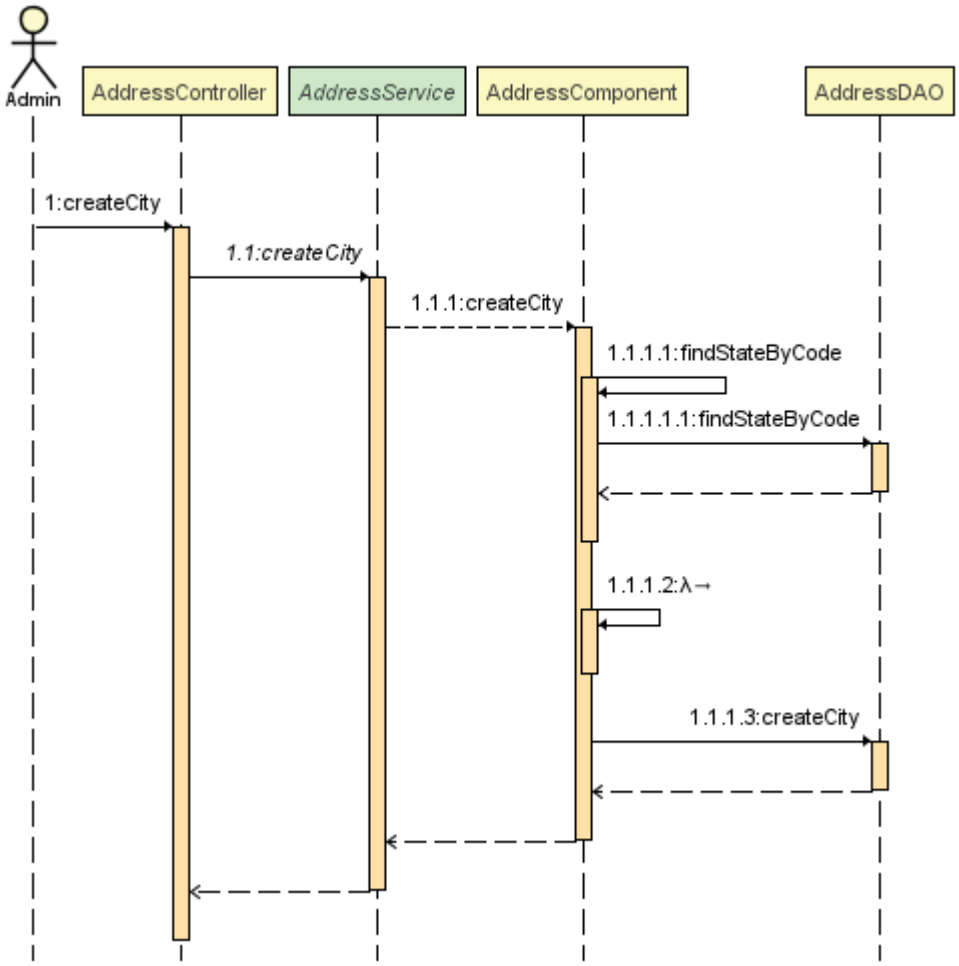
Fonte: O Autor (2022)

Figura 45 - Diagrama de sequência para UC08: Excluir Estado



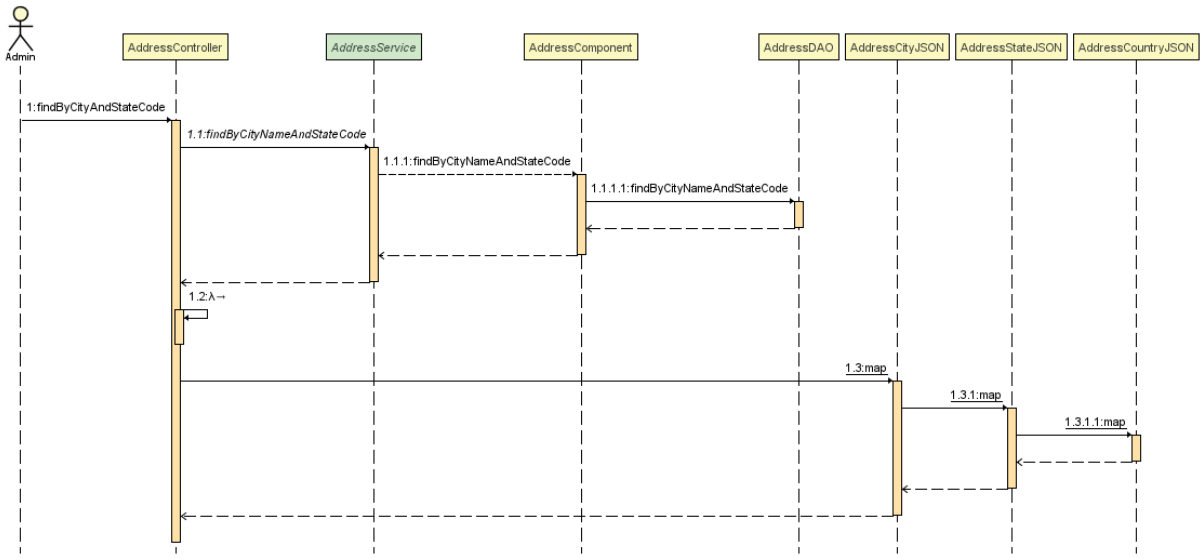
Fonte: O Autor (2022)

Figura 46 - Diagrama de sequência para UC09: Criar Cidade



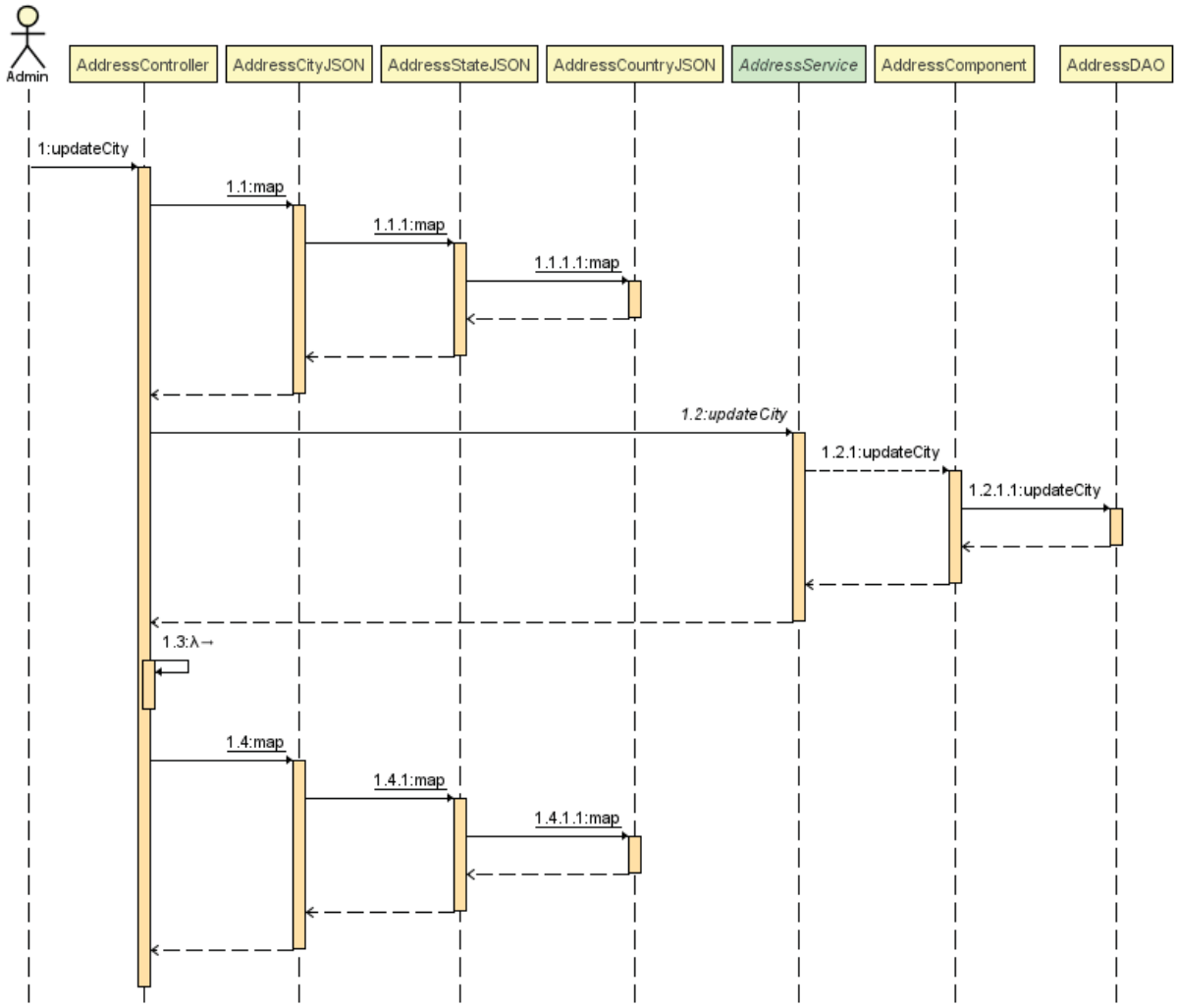
Fonte: O Autor (2022)

Figura 47 - Diagrama de sequência para UC10: Encontrar Cidade



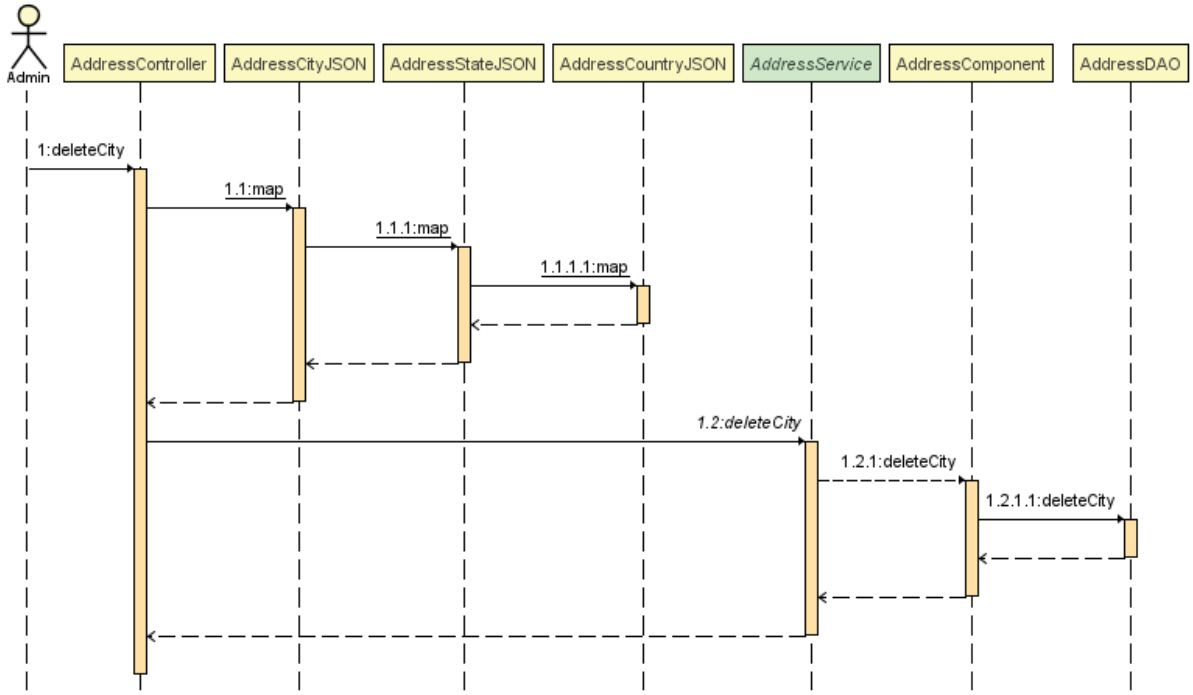
Fonte: O Autor (2022)

Figura 48 - Diagrama de sequência para UC11: Atualizar Cidade



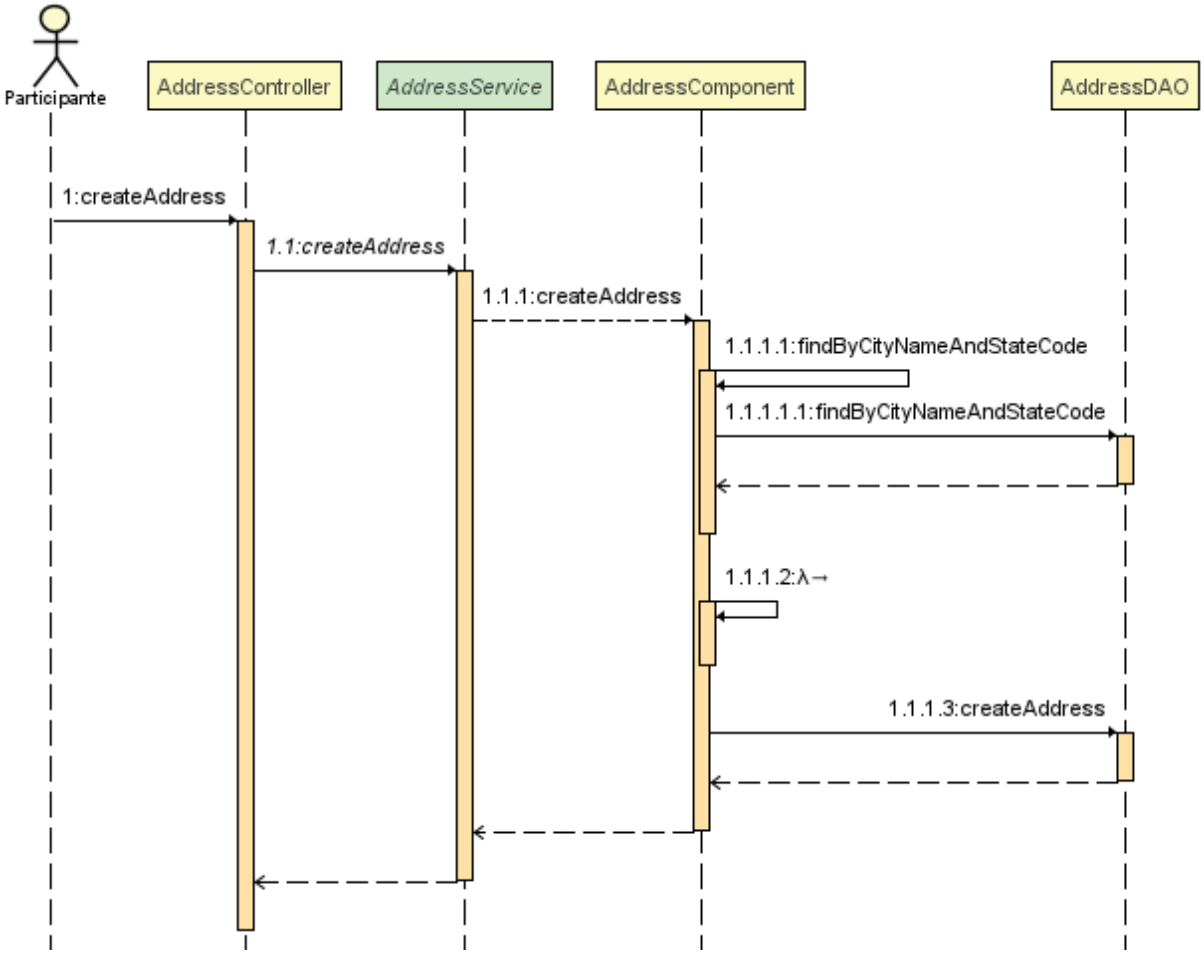
Fonte: O Autor (2022)

Figura 49 - Diagrama de sequência para UC12: Excluir Cidade



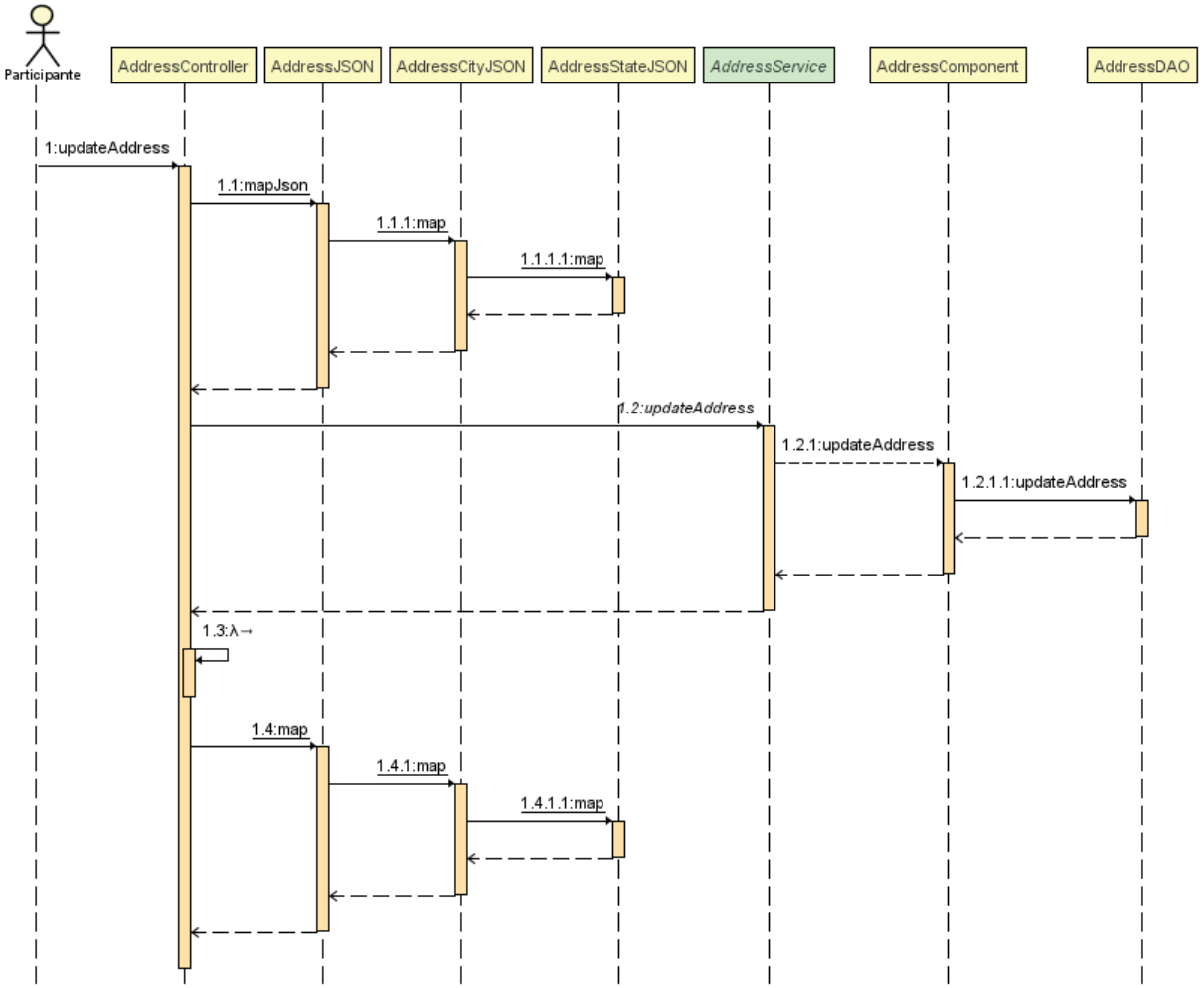
Fonte: O Autor (2022)

Figura 50 - Diagrama de sequência para UC13: Criar Endereço



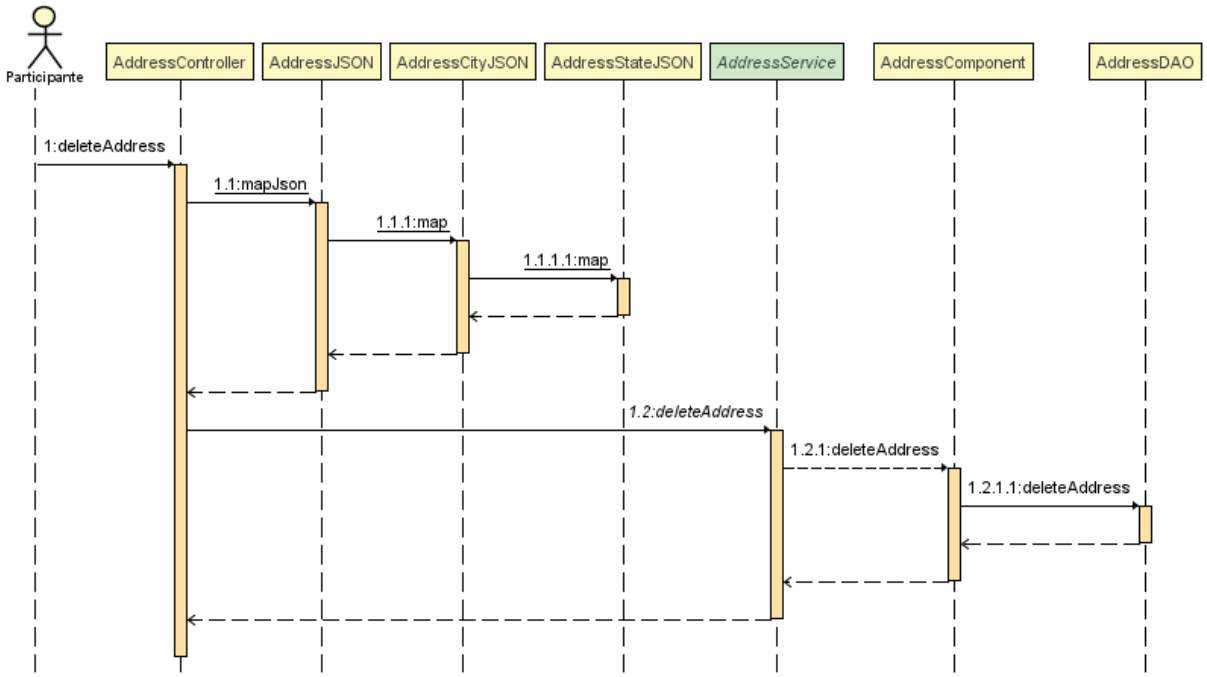
Fonte: O Autor (2022)

Figura 51 - Diagrama de sequência para UC15: Atualizar Endereço



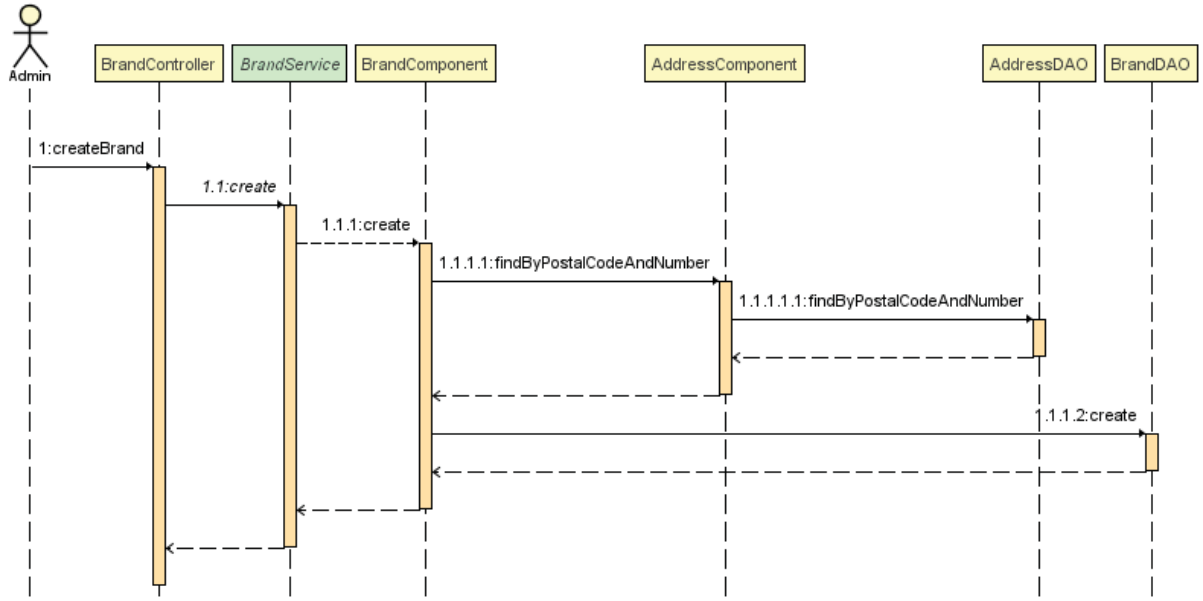
Fonte: O Autor (2022)

Figura 52 - Diagrama de sequência para UC16: Excluir Endereço



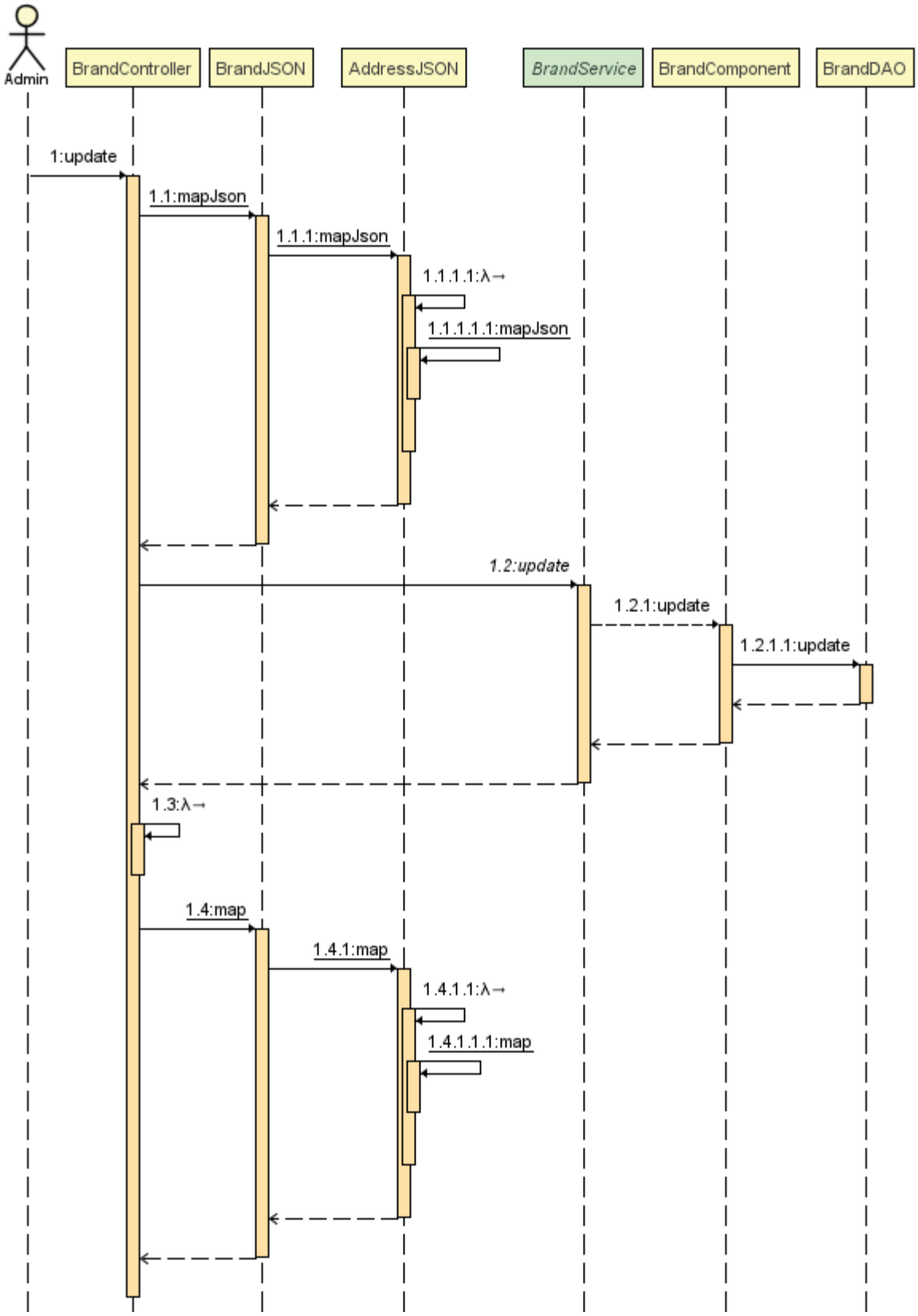
Fonte: O Autor (2022)

Figura 59 - Diagrama de sequência UC 17: Criar Marca



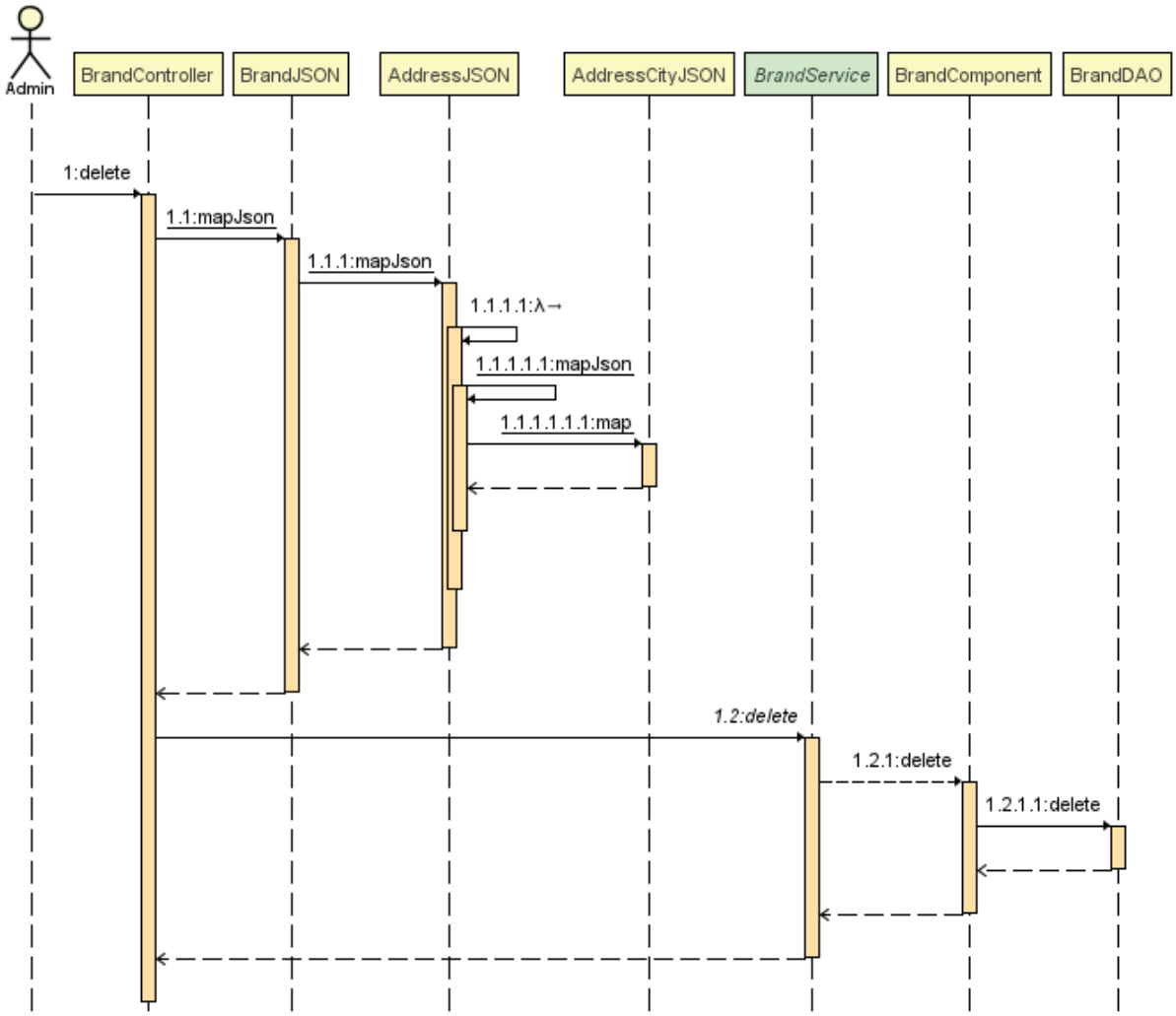
Fonte: O Autor (2022)

Figura 53 - Diagrama de sequência para UC19: Atualizar Marca



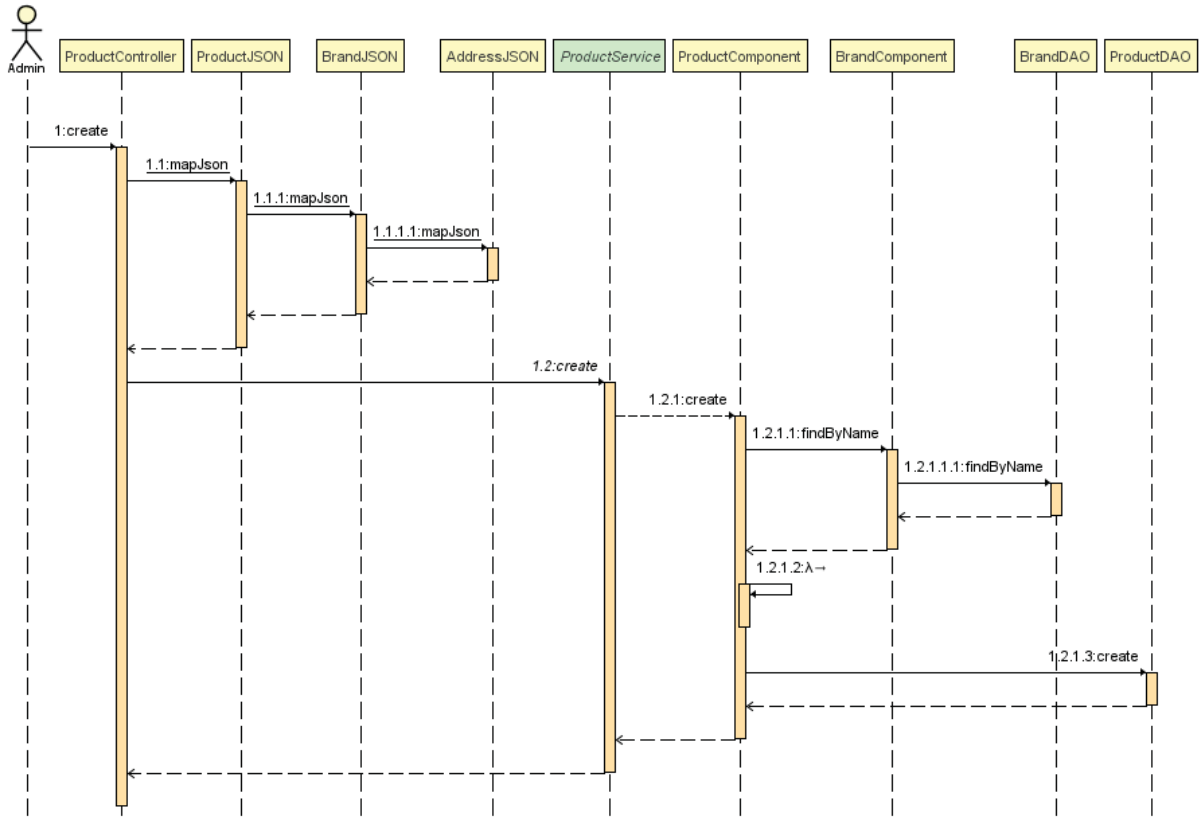
Fonte: O Autor (2022)

Figura 54 - Diagrama de sequência para UC20: Excluir Marca



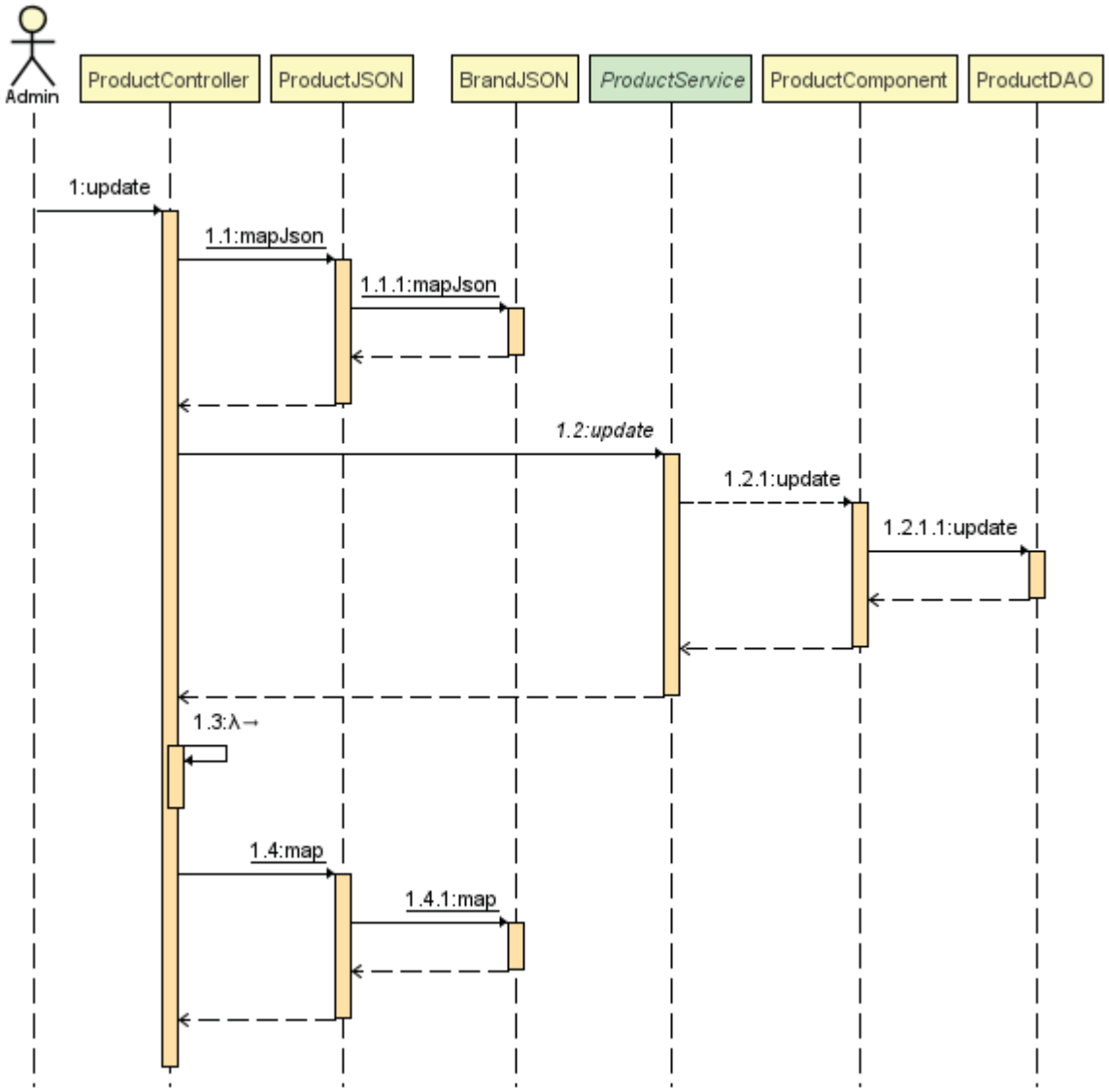
Fonte: O Autor (2022)

Figura 55 - Diagrama de sequência para UC21: Criar Produto



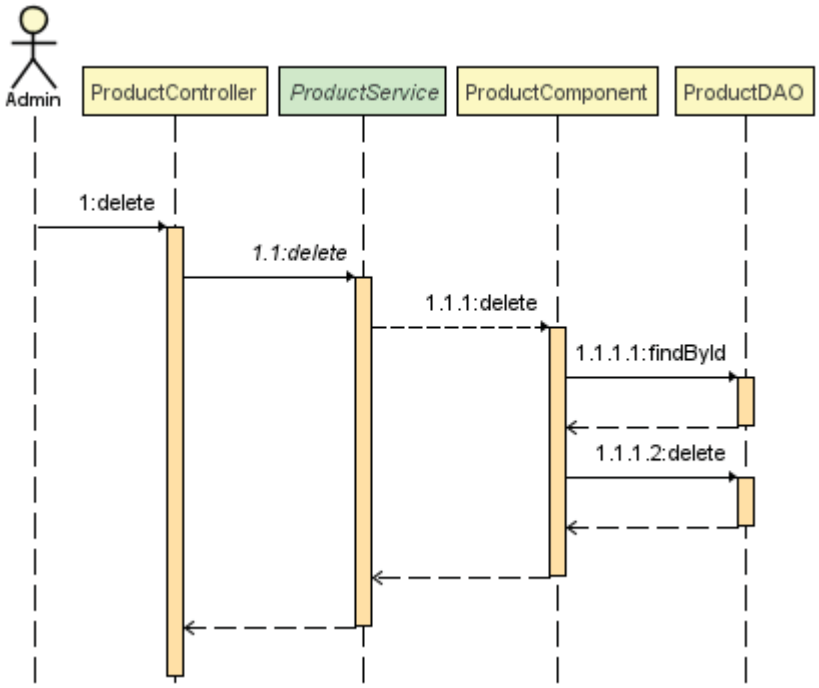
Fonte: O Autor (2022)

Figura 56 - Diagrama de sequência para UC23: Atualizar Produto



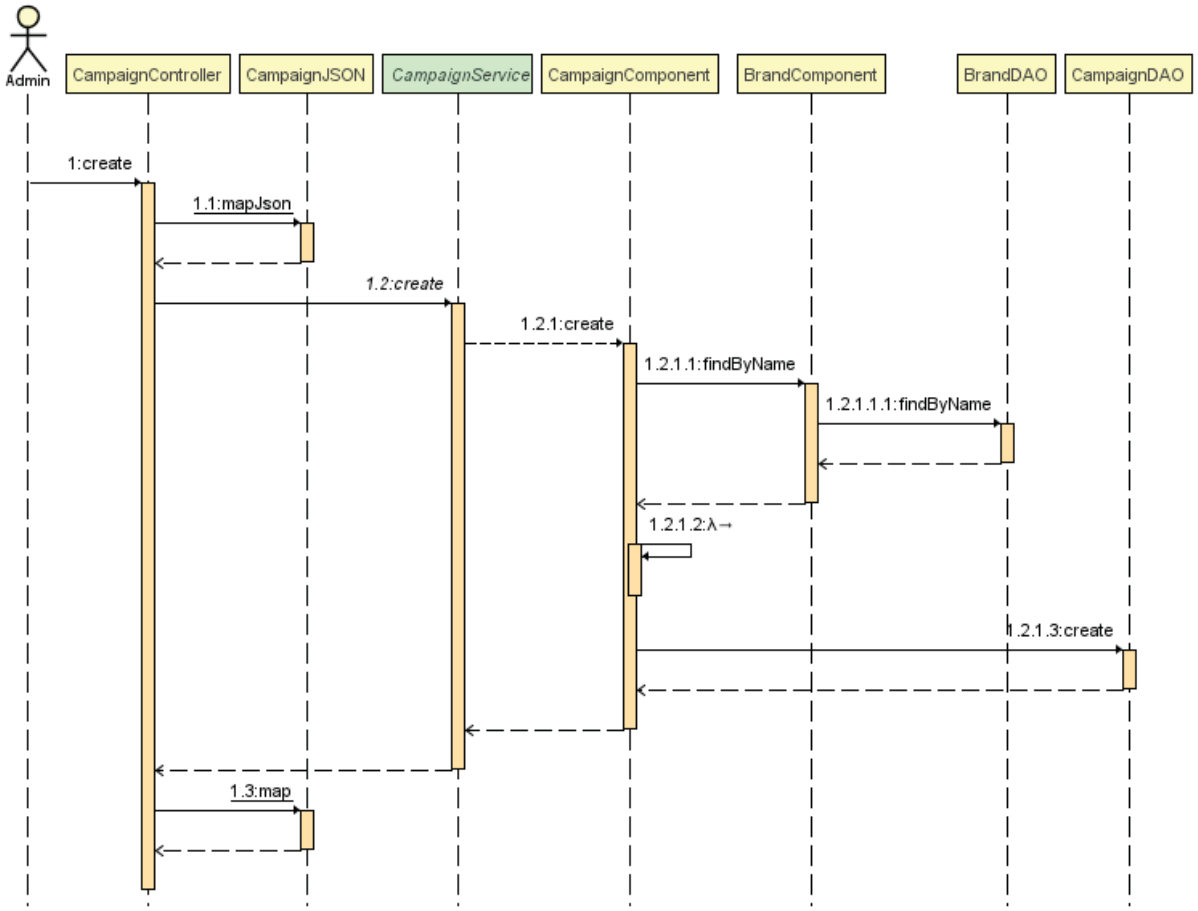
Fonte: O Autor (2022)

Figura 57 - Diagrama de sequência para UC24: Excluir Produto



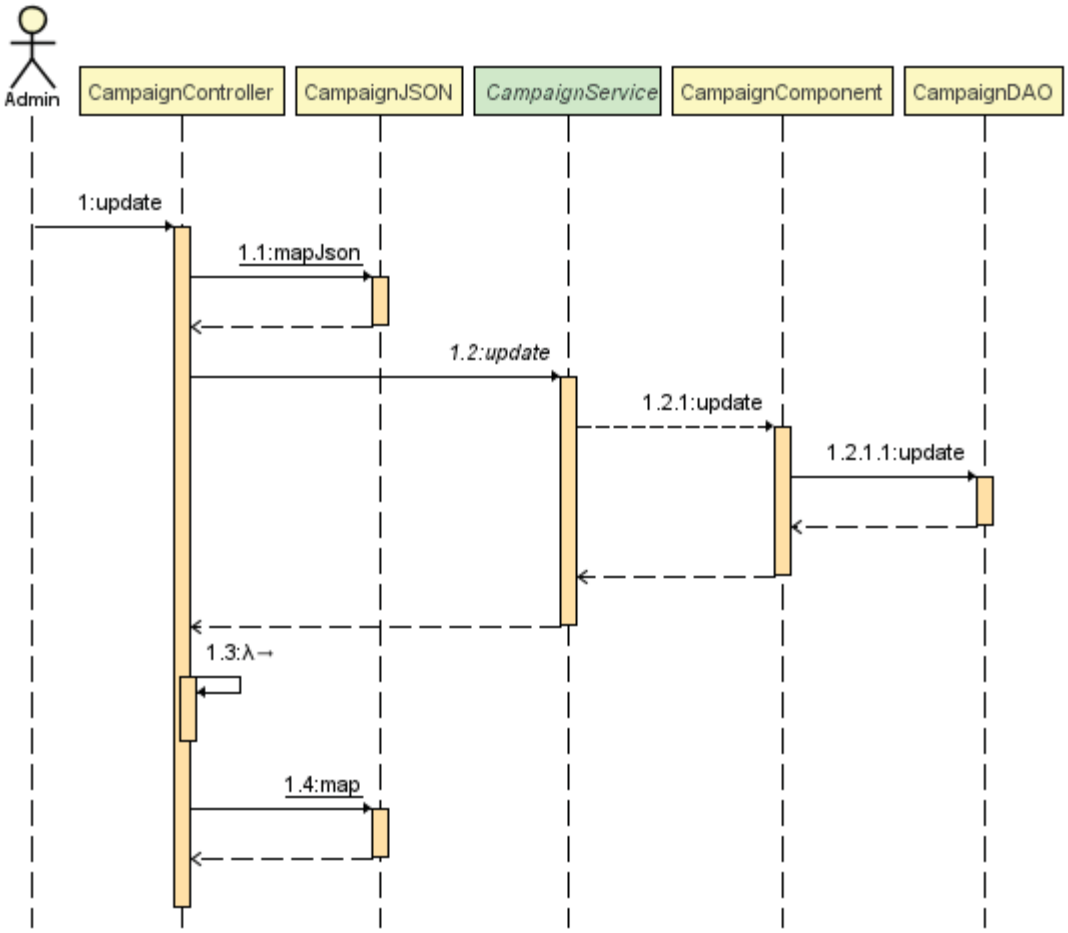
Fonte: O Autor (2022)

Figura 58 - Diagrama de sequência para UC25: Criar Campanha



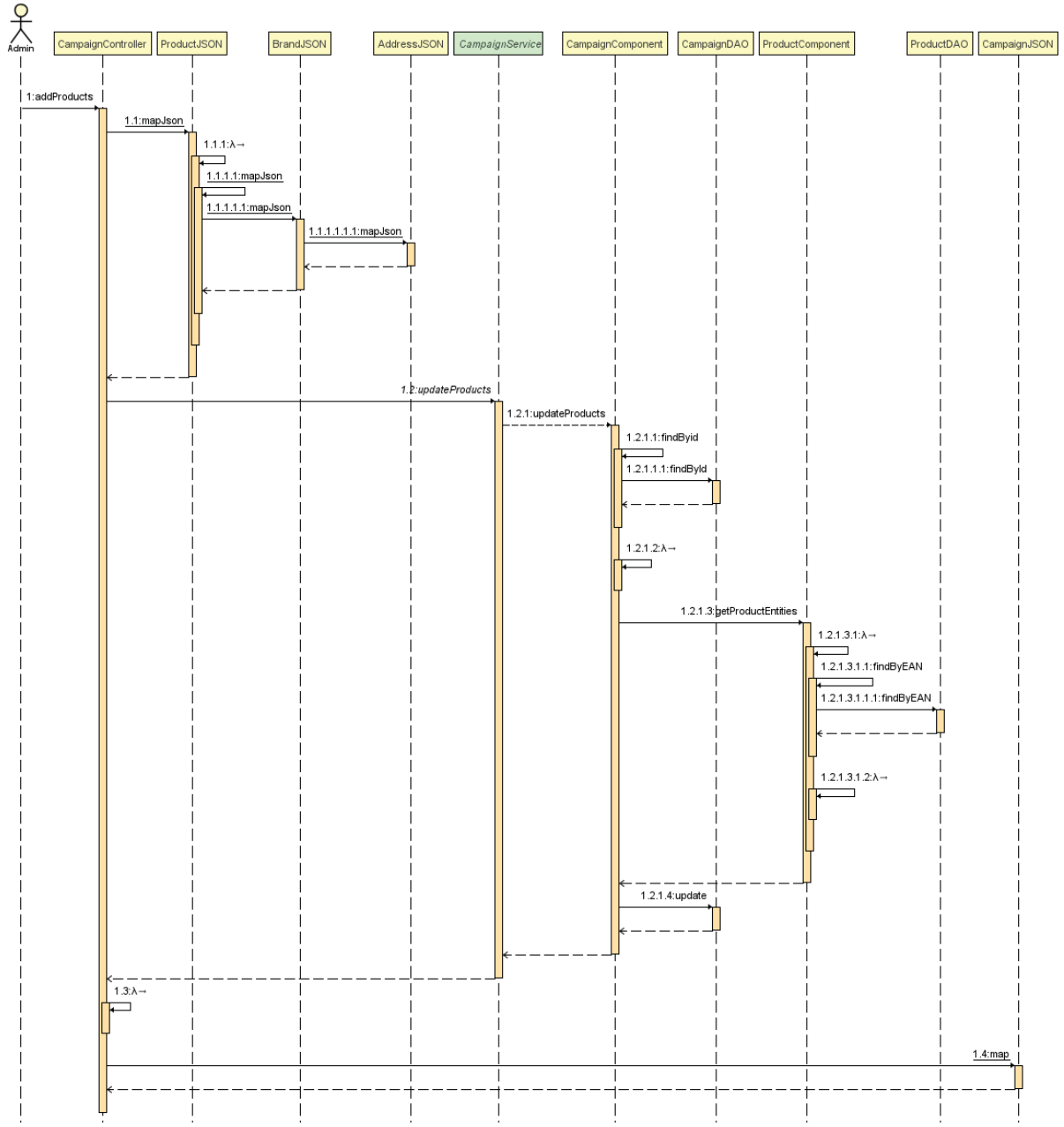
Fonte: O Autor (2022)

Figura 59 - Diagrama de sequência para UC27: Atualizar Campanha



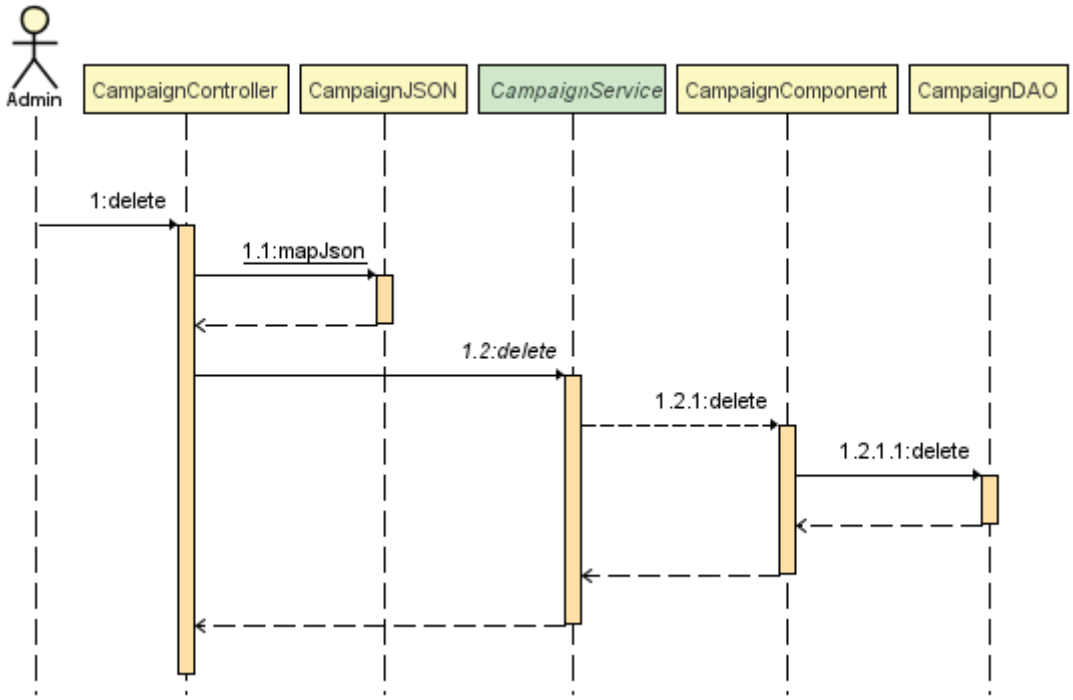
Fonte: O Autor (2022)

Figura 60 - Diagrama de sequência para UC29: Adicionar Produtos à Campanha



Fonte: O Autor (2022)

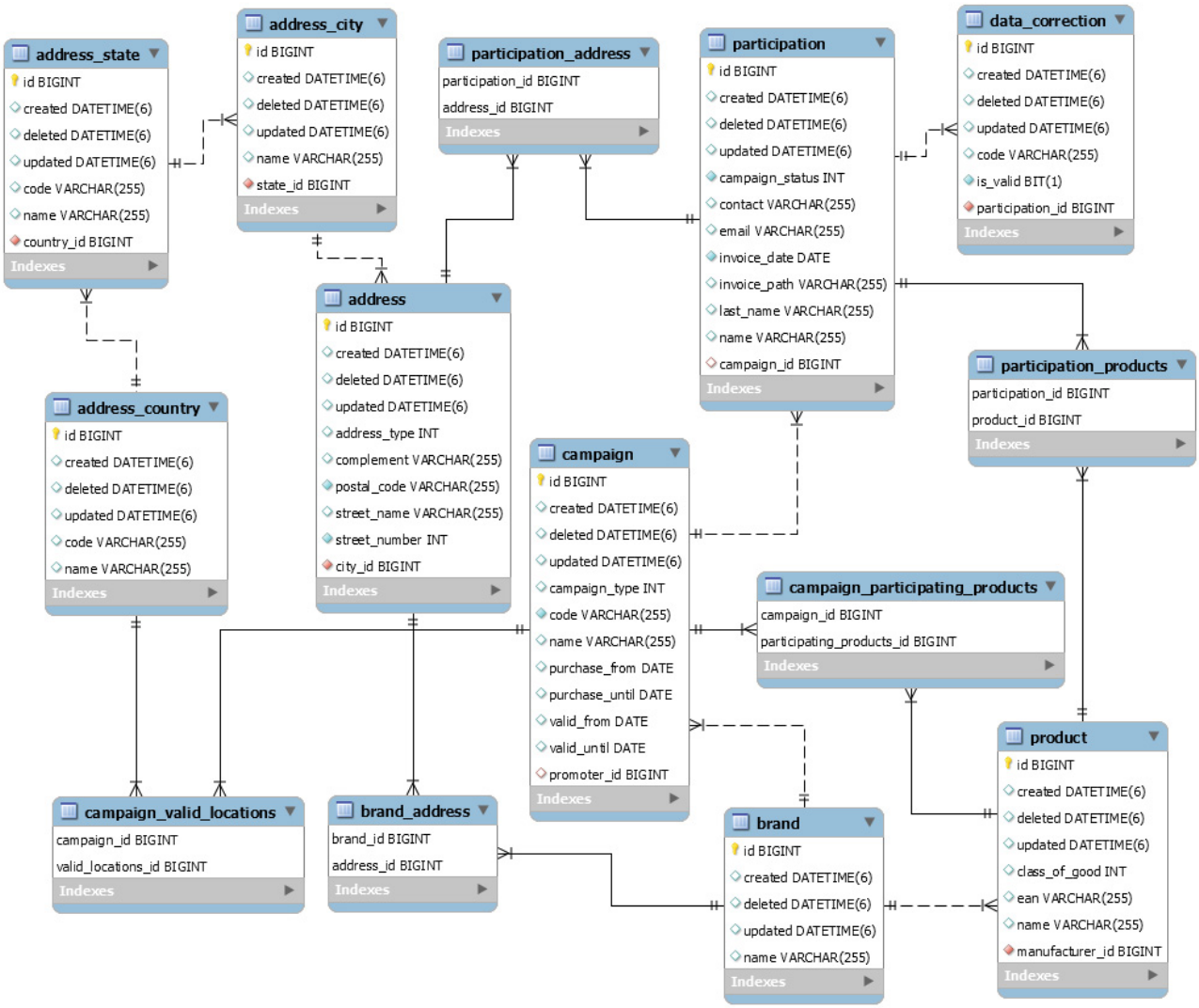
Figura 70 - Diagrama de sequência UC28: Excluir Campanha



Fonte: O Autor (2022)

APÊNDICE J – MODELO FISICO DE DADOS

Figura 61 - Modelo físico do banco de dados



Fonte: O Autor (2022)

APÊNDICE H – CASOS DE TESTE

Os testes deste projeto foram divididos em duas etapas. Inicialmente, os testes unitários de validação foram automatizados com o JUnit; esses testes não consideram o recebimento de uma requisição por parte do servidor, apenas as funcionalidades “internas” do sistema.

Os testes foram automatizados com o JUnit para as entidades relativas ao Endereço (País, Estado, Cidade e Endereço), e para as entidades da Marca, Produto, Campanha e Participação.

Os testes restantes foram automatizados pelo Postman, que permite que validações sejam feitas nos dados que são retornados do servidor, quando uma requisição é concluída. Esses testes verificam a funcionalidade das APIs e como respondem às requisições externas.

Os casos de teste estão descritos a seguir.

UC01: Criar País

Tabela 1 - Plano de Testes para o Caso de Uso 01: Criar País

Teste nº	Pré-Condição	Entrada	Saída esperada
1	Não há	Nome: “Brasil”, Código: “BR”	ID do País criado, juntamente aos dados de entrada.
2	Existir País com código “BR”	Nome: “Brasil”, Código: “BR”	Erro: Violação de *constraint* de código único.

Fonte: O Autor (2022)

UC05: Criar Estado

Tabela 2 - Plano de Testes para o Caso de Uso 05: Criar Estado

Teste nº	Pré-Condição	Entrada	Saída esperada
1	Existir País com código “BR”	Nome: “Paraná”, Código: “PR”, Código País: “BR”	ID do Estado criado, juntamente aos dados de entrada.
2	Existir Estado com código	Nome: “Paraná”, Código: “PR”,	Erro: Violação de *constraint* de código único.

	“PR”	Código País: “BR”	
3	Não existir País com código “BR”	Nome: “Paraná”, Código: “PR”, Código País: “BR”	Erro: Violação de *constraint* de código inexistente.

Fonte: O Autor (2022)

UC09: Criar Cidade

Tabela 3 - Plano de Testes para o Caso de Uso 09: Criar Cidade

Teste nº	Pré-Condição	Entrada	Saída esperada
1	Existir Estado com código “PR”	Nome: “Curitiba”, Código Estado: “PR”	ID da Cidade criada, juntamente aos dados de entrada.
2	Não existir País com código “BR”	Nome: “Paraná”, Código: “PR”, Código País: “BR”	Erro: Violação de *constraint* de código inexistente.

Fonte: O Autor (2022)

UC13: Criar Endereço

Tabela 4 - Plano de Testes para o Caso de Uso 09: Criar Endereço

Teste nº	Pré-Condição	Entrada	Saída esperada
1	Existir Cidade com nome “Curitiba”	Nome: “Curitiba”, Código Estado: “PR”, Zipcode: “00000-000”, Número: 0	ID do Endereço criado, juntamente aos dados de entrada.

Fonte: O Autor (2022)

UC03: Atualizar País

Tabela 5 - Plano de Testes para o Caso de Uso 03: Atualizar País

Teste nº	Pré-Condição	Entrada	Saída esperada
1	Existir País com código	ID do País, Nome: “República do	ID do País atualizado, juntamente aos dados de entrada.

	“BR”	Brasil”, Código: “BR”	
2	Não existir País com código “BR”	ID do País, Nome: “República do Brasil”, Código: “BR”	Erro: Violação de *constraint* de código inexistente.

Fonte: O Autor (2022)

UC07: Atualizar Estado

Tabela 6 - Plano de Testes para o Caso de Uso 07: Atualizar Estado

Teste nº	Pré-Condição	Entrada	Saída esperada
1	Existir Estado com código “PR”	ID do Estado, Nome: “São Paulo”, Código: “SP”	ID do Estado atualizado, juntamente aos dados de entrada.
2	Não existir Estado com código “PR”	ID do Estado, Nome: “São Paulo”, Código: “SP”	Erro: Violação de *constraint* de código inexistente.

Fonte: O Autor (2022)

UC11: Atualizar Cidade

Tabela 7 - Plano de Testes para o Caso de Uso 11: Atualizar Cidade

Teste nº	Pré-Condição	Entrada	Saída esperada
1	Existir Cidade com Nome “Curitiba” e Estado com Código “PR”	ID da Cidade, Nome: “Pinhais”, Código: “PR”	ID da Cidade atualizada, juntamente aos dados de entrada.
2	Existir Cidade com Nome “Pinhais” e Estado com Código “PR”	ID da Cidade, Nome: “São Paulo”, Código: “SP”	ID do Estado atualizado, juntamente aos dados de entrada.
3	Não existir Estado com	Nome: “Florianopolis”,	Erro: Violação de *constraint* de código inexistente.

	código “SC”	Código: “SC”	

Fonte: O Autor (2022)

UC15: Atualizar Endereço

Tabela 8 - Plano de Testes para o Caso de Uso 15: Atualizar Endereço

Teste nº	Pré-Condição	Entrada	Saída esperada
1	Existir Endereço com Cidade “Pinhais”, Zipcode “00000-000” e Número “0”	ID do Endereço, Nome: “Pinhais”, Código Estado: “PR”, Zipcode: “00000-000”, Número: 0”, Complemento: “ap300”	ID do Endereço atualizado, juntamente aos dados de entrada.
2	Não existir Endereço com Cidade “Pinhais”, Zipcode “00000-000” e Número “0”	ID do Endereço, Nome: “Pinhais”, Código Estado: “PR”, Zipcode: “00000-000”, Número: 0”, Complemento: “ap300”	Erro: Violação de *constraint* de código inexistente.

Fonte: O Autor (2022)

UC04: Excluir País

Tabela 9 - Plano de Testes para o Caso de Uso 04: Excluir País

Teste nº	Pré-Condição	Entrada	Saída esperada
1	Existir País com Código “BR”	ID do País, Código: “BR”	ID do País excluído.
2	Existir País com Código “BR” e o País possuir um Estado Atrelado	ID do País, Código: “BR”	Erro: Violação de *constraint* de entidades afiliadas.

Fonte: O Autor (2022)

UC08: Excluir Estado

Tabela 10 - Plano de Testes para o Caso de Uso 08: Excluir Estado

Teste nº	Pré-Condição	Entrada	Saída esperada
1	Existir Estado com Código “PR”	ID do Estado, Código: “PR”	ID do Estado excluído.
2	Existir Estado com Código “PR” e o Estado possuir uma Cidade Atrelada	ID do Estado, Código: “PR”	Erro: Violação de *constraint* de entidades afiliadas.

Fonte: O Autor (2022)

UC12: Excluir Cidade

Tabela 11 - Plano de Testes para o Caso de Uso 12: Excluir Cidade

Teste nº	Pré-Condição	Entrada	Saída esperada
1	Existir Cidade com Nome “Pinhais” e Código do estado “PR”	ID da Cidade, Nome: “Pinhais”, Código do Estado: “PR”	ID da Cidade excluída.
2	Existir Cidade com Nome “Pinhais” e Código do estado “PR” e a Cidade possuir um Endereço Atrelado	ID da Cidade, Nome: “Pinhais”, Código do Estado: “PR”	Erro: Violação de *constraint* de entidades afiliadas.

Fonte: O Autor (2022)

UC16: Excluir Endereço

Tabela 12 - Plano de Testes para o Caso de Uso 16: Excluir Endereço

Teste nº	Pré-Condição	Entrada	Saída esperada
1	Existir Endereço	ID da Cidade,	ID do Endereço Excluído.

	com Cidade “Pinhais”, Zipcode “00000-000” e Número “0”	Cidade: “Pinhais”, Zipcode: “00000-000”, Número: “0”	
2	Existir Endereço com Cidade “Pinhais”, Zipcode “00000-000” e Número “0” e o Endereço possuir uma Entidade (Marca, Campanha, Participação) Atrelada	Cidade: “Pinhais”, Zipcode: “00000-000”, Número: “0”	Erro: Violação de *constraint* de entidades afiliadas.

Fonte: O Autor (2022)

UC17: Criar Marca

Tabela 13 - Plano de Testes para o Caso de Uso 17: Criar Marca

Teste nº	Pré-Condição	Entrada	Saída esperada
1	Existir um Endereço com Zipcode “00000-000” e Número “0”	Nome: “UFPR”, Zipcode do Endereço: “00000-000”, Número do Endereço: “0”, Tipo do Endereço: “1”	ID da Marca criada, juntamente aos dados de entrada.
2	Não há	Nome: “UFPR”, Nome da rua: “Dr. A. Arcoverde”, Número: “1225”, Zipcode do Endereço: “00000-000”, Complemento: “SEPT”, Tipo do Endereço: “1”	ID da Marca criada, juntamente aos dados de entrada.

Fonte: O Autor (2022)

UC21: Criar Produto

Tabela 14 - Plano de Testes para o Caso de Uso 21: Criar Produto

Teste nº	Pré-Condição	Entrada	Saída esperada
1	Existir Marca com nome “UFPR”	Nome: “TADS”, EAN: “UFPRSEPTTADS”, Nome da Marca: “UFPR”, Tipo do Produto: “1”	ID do Produto criado, juntamente aos dados de entrada.
2	Não existir Marca com nome “UFPR”	Nome: “TADS”, EAN: “UFPRSEPTTADS”, Nome da Marca: “UFPR”, Tipo do Produto: “1”	Erro: Violação de *constraint* de código inexistente.

Fonte: O Autor (2022)

UC25: Criar Campanha

Tabela 15 - Plano de Testes para o Caso de Uso 25: Criar Campanha

Teste nº	Pré-Condição	Entrada	Saída esperada
1	Existir uma Marca com nome “UFPR”	Nome: “Engenharia de Software”, Código: “Eng_2020”, Promotor: “UFPR”, ParticipaçãoInicio: “2020-01-01” ParticipaçãoFim: “2021-31-12”, ComprasInicio: “2020-01-01”, ComprasFim: “2021-31-10”	ID da Campanha criada, juntamente aos dados de entrada.

2	Existir uma Marca com nome “UFPR” e um produto com EAN “UFPRSEPTTADS”	Nome: “Engenharia de Software”, Código: “Eng_2020”, Promotor: “UFPR”, ParticipaçãoInicio: “2020-01-01” ParticipaçãoFim: “2021-31-12”, ComprasInicio: “2020-01-01”, ComprasFim: “2021-31-10”, ProdutosParticipantes: “UFPRSEPTTADS”	ID da Campanha criada, juntamente aos dados de entrada.

Fonte: O Autor (2022)

UC19: Atualizar Marca

Tabela 16 - Plano de Testes para o Caso de Uso 19: Atualizar Marca

Teste nº	Pré-Condição	Entrada	Saída esperada
1	Existir uma Marca com nome “UFPR”	ID da Marca, Nome: “UF do Paraná”, Tipo do Endereço: “1”	ID da Marca atualizada, juntamente aos dados de entrada.

Fonte: O Autor (2022)

UC23: Atualizar Produto

Tabela 17 - Plano de Testes para o Caso de Uso 23: Atualizar Produto

Teste nº	Pré-Condição	Entrada	Saída esperada
1	Existir um Produto com EAN: “UFPRSEPTTADS”	ID do Produto, Nome: “TADS”, EAN: “UFPRTADS2020”, Nome da Marca: “UFPR”, Tipo do	ID do Produto atualizado, juntamente aos dados de entrada.

		Produto: “1”	

Fonte: O Autor (2022)

UC27: Atualizar Campanha

Tabela 18 - Plano de Testes para o Caso de Uso 27: Atualizar Campanha

Teste nº	Pré-Condição	Entrada	Saída esperada
1	Existir uma Campanha com Código “Eng_2020”	ID da Campanha, Nome: “Engenharia de Software”, Código: “UFPR_Eng_Software”, Promotor: “UFPR”, ParticipaçãoInicio: “2020-01-01” ParticipaçãoFim: “2021-31-12”, ComprasInicio: “2020-01-01”, ComprasFim: “2021-31-10”	ID da Campanha atualizada, juntamente aos dados de entrada.

Fonte: O Autor (2022)

UC20: Excluir Marca

Tabela 19 - Plano de Testes para o Caso de Uso 20: Excluir Marca

Teste nº	Pré-Condição	Entrada	Saída esperada
1	Existir Marca com Nome “UFPR”	ID da Marca	ID da Marca excluída.

Fonte: O Autor (2022)

UC24: Excluir Produto

Tabela 18 - Plano de Testes para o Caso de Uso 24: Excluir Produto

Teste nº	Pré-Condição	Entrada	Saída esperada
1	Existir Produto com EAN	ID do Produto	ID do Produto excluído.

	“UFPRTADS2020”		

Fonte: O Autor (2022)

UC34: Criar Participação

Tabela 19 - Plano de Testes para o Caso de Uso 34: Criar Participação

Teste nº	Pré-Condição	Entrada	Saída esperada
1	Não há.	Nome: “Regis”, Sobrenome: “Gaboardi”, Email: “regisandre@ufpr.br”, Contato: “988230000”, Data da Nota Fiscal: “2021-04-01”, EAN do produto: “UFPRTADS2020”, Zipcode do Endereço: “88088-999”, Número do Endereço: “10”, Tipo do Endereço: “2”	ID da Participação criada, juntamente aos dados de entrada.

Fonte: O Autor (2022)

UC35: Submeter Nota Fiscal

Tabela 20 - Plano de Testes para o Caso de Uso 35: Submeter Nota Fiscal

Teste nº	Pré-Condição	Entrada	Saída esperada
1	Existir uma Participação com o ID informado	ID da Participação, Arquivo da Nota Fiscal	ID da Participação.

Fonte: O Autor (2022)

UC36: Corrigir Participação

Tabela 21 - Plano de Testes para o Caso de Uso 36: Corrigir Participação

Teste nº	Pré-Condição	Entrada	Saída esperada

1	Existir uma Participação atrelada ao código de correção informado	Código de Correção, Dados corrigidos (Endereço, Pessoais, Produto ou Nota Fiscal)	ID da Participação atualizada, juntamente aos dados da Participação.
2	Código informado inválido	Código de Correção, Dados corrigidos (Endereço, Pessoais, Produto ou Nota Fiscal)	Erro: Código de correção informado já utilizado.

Fonte: O Autor (2022)

UC30: Listar Participações Verificáveis

Tabela 22 - Plano de Testes para o Caso de Uso 30: Listar Participações Verificáveis

1	Existir uma Campanha	Código da Campanha	A Participação mais antiga com estado "*VALIDATION_QUEUE*" da Campanha informada.
2	Existir uma Campanha, mas sem Participações verificáveis (estado "*VALIDATION_QUEUE*")	Código da Campanha	Resultado vazio

Fonte: O Autor (2022)

UC31: Verificar Participação

Tabela 23 - Plano de Testes para o Caso de Uso 31: Verificar Participação

1	Existir uma Participação retornada pelo caso de uso 30: Listar Participações Verificáveis	Avaliação manual dos dados apresentados	ID da Participação atualizada.

Fonte: O Autor (2022)