

# MPE - Uma ferramenta voltada à compra de créditos de Meia Passagem Estudantil da cidade de Salvador/BA

Filipe Brito  
Instituto Federal de Educação, Ciência e  
Tecnologia da Bahia  
Rua Emídio dos Santos, s/n, Barbalho,  
Salvador-Bahia  
filipebrito@ifba.edu.br

Antônio Carlos Souza  
Instituto Federal de Educação, Ciência e  
Tecnologia da Bahia  
Rua Emídio dos Santos, s/n, Barbalho,  
Salvador-Bahia  
antoniocarlos@ifba.edu.br

## RESUMO

Usuários dos mais variados serviços de diferentes setores econômicos contam atualmente com uma alta gama de aplicações tecnológicas que lhe permitem ações cada vez mais móveis, independente de onde e quando realizadas.

Embora exista uma aplicação tecnológica que permita a compra *on-line* de créditos de Meia Passagem Estudantil da cidade de Salvador/BA, não há um aplicativo móvel para tal, além de que a solução disponibilizada é ineficiente por não funcionar na maioria dos navegadores. Desta forma o presente trabalho expõe o MPE, um aplicativo móvel que provê alta disponibilidade e volume de utilização para atender a demanda dos estudantes da capital baiana na compra dos referidos créditos urbanos estudantis. Além do aplicativo móvel há também um sistema *web* para gerenciamento de informações relevantes.

São expostas as etapas de análise, projeto e desenvolvimento da ferramenta MPE, juntamente com suas tecnologias adotadas, como: o uso de Banco de Dados Não-Relacional, Computação Baseada em Nuvem e Programação Reativa, além dos *feedbacks* de alguns dos milhares de usuários que a utilizaram, permitindo assim a validação da proposta apresentada.

## Palavras-chave

Aplicativos, Computação Baseada em Nuvem, Bancos de Dados Não-Relacionais, Programação Reativa.

## 1. INTRODUÇÃO

Estão previstos 2,32 bilhões de usuários de *smartphones* em 2017. Até 2020 este número representará um aumento de 23,70% em relação ao ano anterior, significando 2,87 bilhões de pessoas com acesso à tais dispositivos móveis [15]. A Figura 1 ilustra o número de acesso aos *smartphones* por usuários ao redor do mundo a partir do ano de 2014 com projeções até o ano de 2020.

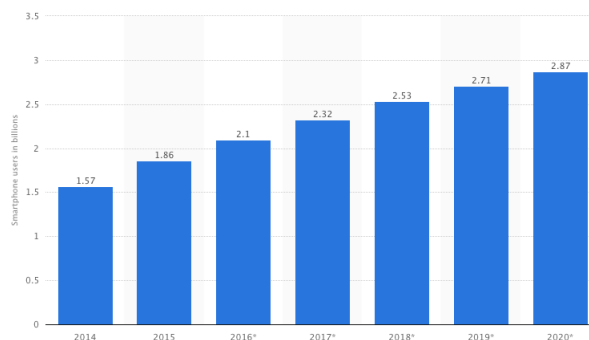


Figura 1: Número de usuários de *smartphones* ao redor do mundo a partir do ano de 2014 com projeções até 2020 [15].

Em contrapartida, não seguindo tendência de crescimento, o número de acesso à internet através dos computadores pessoais passou a ser cada vez menor [7], isto pois, os dispositivos móveis tornaram-se mais acessíveis e, porque não, mais pessoais. No Brasil, por exemplo, dados referentes ao ano de 2014 obtidos pelo Instituto Brasileiro de Geografia e Estatística (IBGE) indicam que os *smartphones* ultrapassaram e passaram a ser os aparelhos mais preferidos na hora de conectar-se à internet (Figura 2) [11]. A quantidade de *smartphones* supera até mesmo a população nacional do país [2].

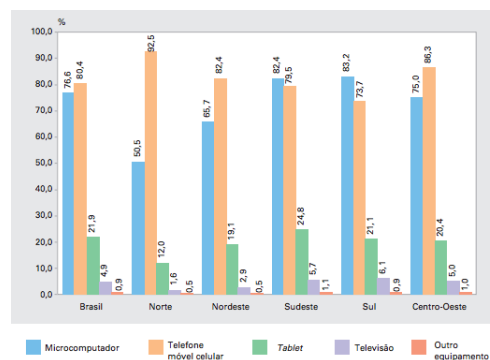
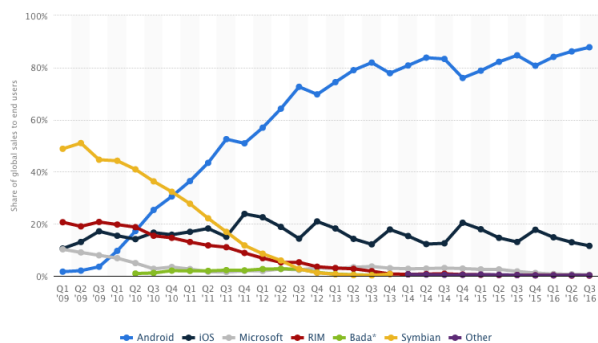


Figura 2: Percentual de domicílios com utilização da internet, por tipo de equipamento utilizado para acessar a internet [11].

Em relação aos sistemas operacionais para *smartphones* o Android™ é o mais utilizado por usuários finais (Figura 3), sendo que no Brasil a tendência é que o sistema operacional continue a dominar uma grande fatia de mercado - com representação de 92,40% [28].



**Figura 3: Participação de mercado dos sistemas operacionais móveis em vendas para usuários finais [14].**

Com acesso cada vez mais justo, disponível e abrangente à internet e o fortalecimento do *e-Commerce*, a Câmara Municipal de Salvador, baseada por estes fatores e pressionada principalmente pelos estudantes da cidade, decretou e sancionou a lei 8.457/2013.

A lei 8.457/2013 passou a garantir uma nova forma para a compra de créditos de Meia Passagem Estudantil - benefício utilizado por cerca de 300 mil estudantes [8] e que faz parte do catálogo de produtos do sistema de bilhetagem eletrônica de Salvador, primeira capital, entre todas as demais, a disponibilizar em 1996 a tecnologia aos seus habitantes [9]: via plataforma virtual.

O sistema de bilhetagem eletrônica de Salvador, que está implantado em uma frota de 2700 ônibus e 300 veículos do Subsistema de Transporte Complementar [9], possibilita o pagamento das tarifas dos coletivos urbanos de forma eletrônica, utilizando para tanto um cartão inteligente<sup>1</sup>.

O usuário beneficiário do serviço possuía duas formas de pagamento nos coletivos municipais antes da aplicação da lei supracitada: em espécie, sendo o menos seguro porém mais facilitado (no âmbito a seguir) e o realizado através do sistema de bilhetagem eletrônica, sendo o mais seguro porém com necessidade de deslocamento até os postos de venda (onde comumente se vê extensas filas de espera) para que se possa comprar os créditos eletrônicos (mediante pagamento em espécie) a serem inseridos em seu cartão inteligente. Com a lei 8.457/2013 uma nova possibilidade de compra passa a existir: através da plataforma virtual *MPE OnLine*<sup>2</sup>.

Embora benéfico, o *MPE OnLine* peca desde princípio em não fornecer meios alternativos de pagamento, pois os usuários necessitam pagar um boleto gerado a cada compra dese-

<sup>1</sup>Conhecido também como *Smart Card*, um cartão inteligente assemelha-se à um cartão de crédito (suas dimensões são normalizadas pela ISO 7816), é munido com um chip que pode possuir ou não um microprocessador [21].

<sup>2</sup>Acessível em: <https://estudante.mpeonline.com.br>

jada - trazendo à tona a questão do deslocamento, uma vez que seja necessário pagá-lo em caixas eletrônicas ou agências bancárias - e por sua plataforma ser ineficiente, já que em seus requisitos de uso o site é compatível somente com o *Internet Explorer 6*<sup>3</sup> ou superior.

Dessa forma é proposto o desenvolvimento de uma nova solução tecnológica que obtenha vantagem do amplo uso dos *smartphones*, internet e a revolução social, cultural e comercial causada pelo *e-Commerce*. A solução denominada MPE provê alta disponibilidade e suporte à múltiplas requisições, além de dispor meios alternativos de pagamento, entregando aos seus usuários uma forma mais prática e rápida entre as existentes para a compra de créditos de Meia Passagem Estudantil. Para tanto, utiliza a Computação Baseada em Nuvem, através de técnicas de balanceamento de carga e replicação de dados em múltiplos servidores e Banco de Dados Não-Relacional, tecnologias altamente escaláveis, além da Programação Reativa e o pagamento realizado através de cartões de crédito.

O aplicativo móvel está disponível na loja *Google Play*<sup>3</sup> e pode-se afirmar que seus objetivos foram alcançados através de indicadores de crescimento e interações obtidas com seus usuários.

Além desta Introdução, este trabalho está organizado como se segue. A seção 2 apresenta o Referencial Teórico. A seção 3 apresenta os Trabalhos Relacionados além de um comparativo entre eles. A seção 4 descreve a visão geral e as etapas de análise, projeto e desenvolvimento do sistema MPE. A seção 5 apresenta as estatísticas de uso do sistema (os resultados por ele obtidos). Por fim, a seção 6 conclui este trabalho, destacando suas limitações e trabalhos futuros.

## 2. REFERENCIAL TEÓRICO

Esta seção apresenta os principais assuntos relacionados à este trabalho. A subseção 2.1 apresenta conceitos relacionados à Computação Baseada em Nuvem. A subseção 2.2 discute o tópico Bancos de Dados Não-Relacionais. Por fim, na subseção 2.3 é abordado o assunto Programação Reativa.

### 2.1 Computação Baseada em Nuvem

#### 2.1.1 Definição

A Computação Baseada em Nuvem é um modelo que permite a localização ubíqua, conveniente e com acesso sob demanda à um conjunto de recursos compartilhados (como redes, servidores, aplicações e demais serviços) que podem ser rapidamente provisionados e liberados com o mínimo de esforço de gerenciamento ou interação com quem o provém [24].

O termo, como conhecemos hoje, é a materialização do que há mais de meio século cunhou Douglas Parkhill em seu livro *"The Challenge of the Computer Utility"* [29] e que, anos mais tarde, em 1969, propagou Leonard Kleinrock [18], um dos cientistas-chefe da Rede de Agência de Projetos de Pesquisa Avançada (*ARPANET*).

<sup>3</sup><https://play.google.com/store/apps/details?id=br.com.recargasalvadorcard>

### 2.1.2 Características

De acordo com o *NIST* (Instituto Nacional de Padrões e Tecnologia dos Estados Unidos da América e órgão do Departamento de Comércio do país) [24], a Computação Baseada em Nuvem é composta por cinco características essenciais:

- Serviços sob demanda

Um consumidor pode unilateralmente, conforme necessário, fornecer capacidades de computação (como o tempo do servidor e o armazenamento da rede) sem necessidade de interação humana com o provedor do serviço.

- Amplo acesso à rede

As capacidades estão disponíveis através da rede e são acessadas por meio de mecanismos padrões que promovem o uso por plataformas heterogêneas dos clientes (*smartphones*, *tablets*, *laptops*, estações de trabalho, ademais).

- Agrupamento de recursos

Os recursos de computação do provedor são agrupados para servir múltiplos consumidores através de diferentes recursos físicos e virtuais, dinamicamente atribuídos (e reatribuídos) de acordo com a demanda de cada consumidor.

O agrupamento de recursos resulta em uma sensação de independência de localização no sentido de que o cliente geralmente não tem controle ou conhecimento sobre a localização exata dos recursos fornecidos, embora possa especificar a localização em um nível mais alto de abstração (como, por exemplo, por país, estado ou *datacenter*).

- Elasticidade rápida.

As capacidades podem ser elasticamente provisionadas e liberadas, em alguns casos automaticamente, proporcionais à demanda, o que leva ao seu consumidor a pensar que os recursos disponíveis para fornecimento são ilimitados e que podem ser apropriados em qualquer quantidade e a qualquer momento.

- Serviço Mensurável

Os sistemas em nuvem controlam e otimizam automaticamente o uso de recursos por medição em um nível de abstração apropriado ao tipo de serviço (por exemplo, pelo armazenamento, processamento e largura de banda utilizados ou por contas de usuários ativos), o que proporciona transparência tanto para o provedor quanto para o consumidor (ambos monitoram, controlam e o relatam).

### 2.1.3 Mobilidade e Computação Baseada em Nuvem

Há algum tempo que a indústria *mobile* vem crescendo, com mais *smartphones* sendo vendidos para os consumidores do que *PC's*, portanto é de se imaginar que desenvolver aplicações móveis que suportem essa demanda e sejam tão eficientes quanto as aplicações *desktop* é um desafio, já que os

recursos encontrados em tais *devices* (como espaço de armazenamento e capacidade computacional) são inferiores, o que torna a Computação Baseada em Nuvem um aliado para superar tais limitações, já que os recursos computacionais são fornecidos como um serviço através de uma rede ou internet.

Vista como uma das tecnologias mais proeminentes entre os profissionais de TI (muito por causa de alguns dos motivos discutidos na subseção anterior), a Computação Baseada em Nuvem permite que os dispositivos móveis executem muitas das tarefas semelhantes às quais poderiam ser executadas em supercomputadores, pois estas tarefas são processadas por servidores em segundo plano.

A Figura 4 ilustra como alguns dos (variados) recursos se encontram disponíveis através da Computação Baseada em Nuvem, evidenciando, como discutiu [3], que “todo o armazenamento de dados e o poder computacional são movidos para dentro da nuvem”, distantes dos dispositivos móveis.

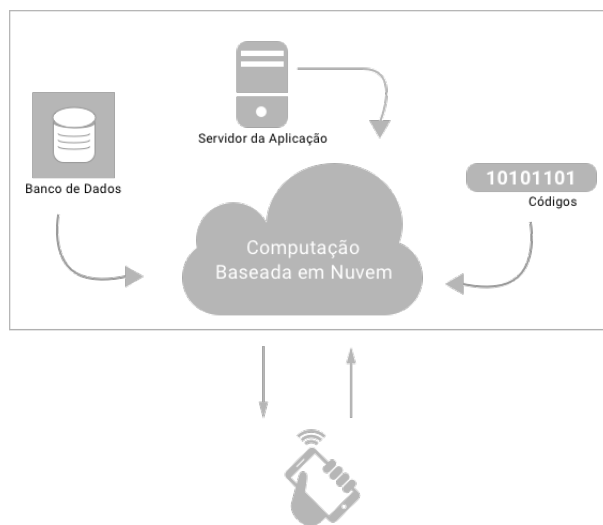


Figura 4: A Computação Baseada em Nuvem e alguns dos seus serviços.

### 2.1.4 Vantagens

A principal vantagem de utilizar os recursos da Computação Baseada em Nuvem (principalmente para negócios iniciantes) é que esta tecnologia reduz os custos de implementação de *hardware*, *software* e suas licenças, proporcionando também a redução de energia e o rápido desenvolvimento de soluções.

Outros benefícios obtidos, segundo [19], incluem:

- Confiabilidade, Usabilidade e Extensibilidade

A Computação Baseada em Nuvem provê um modo seguro para o armazenamento de dados, não precisando seus usuários se preocuparem com questões como: atualização de *software*, ataques de vírus e perda destes dados.

- Ampla escala

Por possuir a capacidade de supercomputação e armazenamento em massa, um sistema de Computação Baseada em Nuvem normalmente é composto por milhares de servidores e *PC's*.

Para [12], existem outras importantes vantagens que a Computação Baseada em Nuvem proporciona, desta vez no contexto *mobile*. São elas:

- Aumento da duração da bateria dos dispositivos

A bateria é uma das principais preocupações nos dispositivos móveis e mesmo que muitas soluções tenham sido propostas para melhorarem a performance da *CPU* e o gerenciamento dos seus discos de maneira mais inteligente, seus resultados requerem mudanças nas suas estruturas ou um novo *hardware*, como efeito, um aumento de custo e a não viabilidade em sua totalidade são esperados.

Através do uso dos recursos da Computação Baseada em Nuvem, grandes cálculos e processamentos complexos são migrados dos dispositivos com recursos limitados para máquinas engenhosas, evitando um longo tempo de execução nos primeiros, o que repercutiria em grandes quantidades de consumo de energia.

- Melhora da capacidade de armazenamento de dados

Capacidade de armazenamento também é uma restrição nos dispositivos móveis e a Computação Baseada em Nuvem no contexto *mobile* habilita que usuários destes dispositivos acessem e armazenem grandes quantidades de dados na nuvem através de redes sem fio.

Alguns serviços de compartilhamento de fotos para celular, por exemplo, permitem que os usuários façam *upload* de imagens para a nuvem imediatamente após uma captura, o que permite que eles economizem quantidade considerável de espaço de armazenamento nos seus dispositivos.

### 2.1.5 Desvantagens

Um dos pontos críticos da utilização dos recursos disponíveis em um sistema de Computação Baseada em Nuvem diz respeito à privacidade, pois, mesmo que a maioria dos seus servidores possuam certificados digitais, levando a crer que todas as informações estão em segurança, todos os dados lá encontrados são de posse também do seu provedor de serviço. Tal realidade causa espanto devido a possibilidade de funcionários terem acesso à dados sensíveis ou até mesmo na utilização destes pela companhia (como na venda de informações privilegiadas ou no repasse para órgãos governamentais). Tal criticidade leva a outra adversidade: A de que o controle dos recursos, dados e informações não é 100% pertencente ao cliente.

Outro questionamento, de acordo com [19], é que os provedores de serviço não projetam plataformas para companhias específicas, ou seja, não há uma relação fiel às suas práticas de negócio.

### 2.1.6 Classificação dos serviços de Computação Baseada em Nuvem

Serviços de Computação Baseada em Nuvem são classificados de acordo com três modelos essenciais:

- *IaaS (Infrastructure-as-a-Service)*

É a entrega de infraestrutura informática como um serviço e, além da maior flexibilidade, um dos seus principais benefícios é que seu esquema de pagamento é baseado em uso, permitindo que os clientes paguem à medida que crescem. Outra importante vantagem é que os clientes estão sempre utilizando as mais recentes tecnologias.

- *PaaS (Platform-as-a-Service)*

Destina a lidar com os diferentes estágios do ciclo de vida de uma aplicação, bem como ocultar a complexidade dos mecanismos para suportar a implementação e a execução desta.

- *SaaS (Software-as-a-Service)*

Baseia-se em *software* como um serviço que é executado em computadores remotos na nuvem. Estes *softwares* são operados e pertencentes à uma organização e a internet é utilizada para permitir conectividade com os computadores dos usuários finais.

Exemplos de aplicações para os modelos acima relacionados incluem:

1. *Amazon EC2*: Modelo *IaaS*, permite que aplicações de clientes sejam executadas em sua plataforma proporcionando capacidade computacional segura e redimensionável na nuvem. Uma clássica utilização é disponibilizar um servidor *web*, tal como o *Apache HTTP Server*, com o mesmo.
2. *Amazon EMR*: Exemplo *PaaS*, permite executar e escalar facilmente várias estruturas de *Big Data* (como o *Hadoop* e o *HBase*) através de sua plataforma.
3. *SalesForce*: Pioneira na prestação de um serviço entregue como modelo *SaaS*, caracteriza-se por permitir que não seja necessário que uma equipe de especialistas de TI configure ou gerencie sua plataforma de *CRM (Customer Relationship Management)* como comumente visto em serviços semelhantes.

## 2.2 Bancos de Dados Não-Relacionais

Com o desenvolvimento da internet e da Computação Baseada em Nuvem, os bancos de dados precisam ser capazes de armazenar e processar grandes quantidades de dados de forma eficaz, dado a demanda por alto desempenho nas operações de leitura e escrita. Especialmente em aplicações de grande escala e alta concorrência, como os motores de busca, usar bancos de dados relacionais para armazenar e consultar dados dinâmicos parece ser cada vez mais inadequado. Neste caso, os Bancos de Dados Não-Relacionais foram criados [16].

### 2.2.1 Definição

O termo *NoSQL* (*Not Only SQL*) foi criado pela primeira vez em 1998 como o nome de um banco de dados relacional baseado no *Unix Shell*, concebido para oferecer uma melhor flexibilidade e otimização do uso dos recursos em relação aos bancos de dados relacionais existentes. Anos depois, em 2009, o termo foi revivido com a ascensão da Computação Baseada em Nuvem e com a apresentação do *Google Bigtable*<sup>4</sup> e, desde então, tem sido generalizado para descrever bases de dados que modelam, armazenam e recuperam dados de uma maneira diferente das tradicionais bases [1], ou seja, referem-se ao modelo de dados lógicos que não seguem a álgebra relacional.

### 2.2.2 Características

Uma das principais características dos Bancos de Dados Não-Relacionais é que seus esquemas<sup>5</sup> são livres, portanto, não possuem *constraints*, usadas para particularizar regras em uma tabela de um banco de dados. Esta liberdade facilita que a estrutura destes mesmos dados seja facilmente e rapidamente modificada sem a necessidade de, por exemplo, reescrever novas tabelas.

Outras importantes características por [25] incluem:

- Forte Consistência

Todos os clientes veem a mesma versão dos dados, mesmo em atualizações simultâneas em um mesmo conjunto de dados.

- Alta disponibilidade

Todos os clientes sempre podem encontrar pelo menos uma cópia dos dados solicitados mesmo se algumas das máquinas de um *cluster*<sup>6</sup> no qual estes estavam estiverem desativadas.

- Tolerância de partição

O sistema total mantém sua característica mesmo quando sendo implantado em servidores diferentes, sendo portanto transparente para o cliente.

As três características acima são conhecidas como Teorema *CAP* (*Strong Consistency*, *High Availability* e *Partition-tolerance*). O Teorema *CAP* postula que apenas dois dos três diferentes aspectos de escala podem ser alcançados completamente ao mesmo tempo (Figura 5).

<sup>4</sup>*Google Bigtable* é um serviço de banco de dados de *BigData NoSQL* do Google<sup>TM</sup> projetado para lidar com cargas de trabalho maciças em baixa latência e é o mesmo banco de dados que fornece muito dos seus principais serviços, como as ferramentas de pesquisa e mapas.

<sup>5</sup>Um esquema de banco de dados é a sua estrutura descrita em linguagem formal, referindo-se a forma como seus dados são organizados.

<sup>6</sup>*Cluster* (ou *clustering*) é o nome dado a um sistema que relaciona dois ou mais computadores para que estes trabalhem de maneira conjunta no intuito de processar tarefas, dividindo entre si as atividades de processamento e executando-as de maneira simultânea.

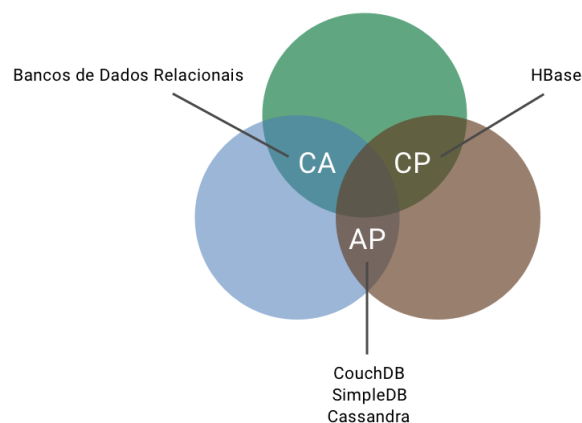


Figura 5: Teorema CAP.

### 2.2.3 Vantagens

Como descrito na subseção anterior, por serem *schema-free*, os Bancos de Dados Não-Relacionais permitem que a estrutura dos dados armazenados seja facilmente modificada sem nenhum esforço extra comparado aos bancos de dados tradicionais. Como exemplo, suponha que uma tabela *T* possua como atributos *N* e *M*, sendo *N* declarado como um tipo *varchar(250)* e *M* também; alterar algumas destas propriedades incluem ações diferentemente aplicadas em ambos os bancos.

Outras vantagens em relação aos bancos de dados relacionais, segundo [26], incluem:

- Alta escalabilidade

Alguns Bancos de Dados Não-Relacionais podem funcionar em uma configuração distribuída. É o caso do *Hbase*, que oferece capacidades semelhantes ao *Google BigTable* pelo uso do *Hadoop*.

- O papel do *DBA* (*Database Administrator*) geralmente não é requerido para os clientes

Com a introdução do *DBaaS* (*Database-as-a-service*) a administração de muitos Bancos de Dados Não-Relacionais é executada pelo provedor de serviços e não por seus clientes.

- Lidam com falhas de *hardware*

Alguns Bancos de Dados Não-Relacionais (como o *Cassandra*) são programados para lidarem com falhas de *hardware* e continuarem a operar de maneira necessária, mesmo se estas falhas forem ocasionais, transitórias ou permanentes, de componentes internos e/ou de módulos.

- Mais rápidos, eficientes e flexíveis

Nos bancos de dados relacionais, todos os dados armazenados estão sujeitos ao conjunto *ACID* (*Atomicidade*, *Consistência*, *Isolação* e *Durabilidade*). Por seguir estas restrições (garantidos paralelamente) em cada

parte destes dados, os bancos de dados relacionais normalmente são mais lentos em suas operações do que os não-relacionais.

- Em constante evolução

Por serem cada vez mais necessários às organizações modernas, os Bancos de Dados Não-Relacionais incorporam muitos recursos para satisfazerem as exigências do mercado, tornando-os em grande competitividade tecnológica.

#### 2.2.4 Desvantagens

Algumas desvantagens ao adotar o uso de Bancos de Dados Não-Relacionais são:

- Ausência da *Standard Query Language*

Bancos de Dados Não-Relacionais não funcionam com a *Standard Query Language*, comumente usada nos Bancos de Dados Relacionais. Dado a ausência, é exigido a programação das consultas diferentemente das utilizadas, o que torna-se mais rápido para algumas tarefas, mas pode demandar tempo de aprendizado para os usuários.

- Confiabilidade

Como os Bancos de Dados Não-Relacionais não suportam nativamente o conjunto *ACID*, os mesmos não fornecem o grau de confiabilidade garantido pelo conjunto.

- Administração

Jenny Richards, em sua página na internet [30], deixa claro que embora o objetivo final do projeto de banco de dados *NoSQL* seja oferecer uma solução que não exige nenhuma administração, a realidade é que os Bancos de Dados Não-Relacionais exigem muita habilidade técnica tanto com a sua instalação quanto com a sua manutenção - muito principalmente para os provedores *DBaaS*.

#### 2.2.5 Classificação de Banco de Dados Não-Relacionais

Segundo [20], Bancos de Dados Não-Relacionais podem ser classificados em três tipos: os baseados em armazenamento chave-valor, os orientados por coluna e os baseados em armazenamento de documentos.

- Baseados em armazenamento chave-valor

Um armazenamento chave-valor é um sistema que armazena valores indexados para recuperação por chaves. Estes sistemas podem conter dados estruturados ou não estruturados.

- Orientados por coluna

Em vez de armazenar conjuntos de informações em uma tabela de colunas-linhas com campos uniformes

para cada registro (como ocorre nos bancos de dados relacionais), bancos de dados orientados por coluna contém uma coluna extensível de dados estreitamente relacionados.

- Baseados em documentos

São os bancos de dados que armazenam e organizam dados como coleções de documentos e não como tabelas estruturadas com campos uniformes para cada registro. Com estes bancos de dados os usuários podem adicionar qualquer número de campos de qualquer comprimento à um documento.

Exemplos de aplicações para os itens anteriormente citados incluem:

1. *SimpleDB*: Baseado em armazenamento chave-valor, o banco de dados *SimpleDB* minimiza o trabalho da administração do banco de dados, onde o armazenamento e a consulta de itens de dados são feitos por meio de solicitações de serviços da *web*.
2. *Cassandra* e *Hbase*: Banco de dados orientados por coluna, *Cassandra* é adequado nas situações onde deseje-se escalabilidade e alta disponibilidade sem comprometer o desempenho, enquanto o *Hbase* adequa-se onde o acesso aleatório, em tempo real, nas operações de leitura e escrita de um *BigData* é esperado. O *Hbase* é modelado com base no *Google BigTable* e, como o *Cassandra*, é disponibilizado aos usuários com código-fonte aberto.
3. *CouchDB*: Baseado em documentos, o *CouchDB* é um Banco de Dados Não-Relacional de código-fonte aberto, escalável, escrito em *Erlang* e acessível a partir de qualquer navegador. O *CouchDB* concentra-se na facilidade de uso, tendo uma arquitetura que “abraça” completamente a *Web*.

Cada uma destas aplicações garantem características explicitamente definidas pelo Teorema *CAP* (Figura 5).

### 2.3 Programação Reativa

Devido ao crescimento da internet e a demanda por tempo real a programação precisa ser dinâmica, ou seja, diferente da forma tradicional de desenvolvimento. A forma tradicional de programar/desenvolver cria várias tarefas que se comunicam em tempos determinados, com respostas determinadas; são rígidas e seguem regras diretas. Embora funcionasse bem e mesmo sendo utilizada até hoje, esta lógica não é mais compatível com a necessidade atual [33].

Seu apelo, para adoção do paradigma, é o contínuo crescimento da internet nos seus mais diversos aspectos, como: A maior quantidade de usuários, de requisições por cada vez mais dados e que, conseqüentemente, exige maior performance e tempo de resposta, além da intolerância à *downtime* e maior quantidade de nós de servidores por aplicação necessários para atender a demanda corrente. [13].

### 2.3.1 Definição

Termo concebido há um longo tempo, a Programação Reativa diverge de outros dois tipos de programas de computador: os Programas de Transformação e os Programas Interativos [5].

- Programas de Transformação

São aqueles que computam os resultados de um dado conjunto de entradas, exemplos típicos são: Compiladores ou programas de computação numérica.

- Programas Interativos

São aqueles que interagem na sua própria velocidade com os usuários ou com outros programas, exemplos típicos são: Programas de *e-mail* e de fuso horário.

Embora os Programas Reativos também mantenham uma interação contínua com seu ambiente (assim como os Programas Interativos), sua principal diferença é que esta interação é determinada pelo ambiente e não pelo próprio programa, além disso, enquanto Programas Interativos funcionam em seu próprio ritmo, os Programas Reativos funcionam apenas em resposta às demandas externas e lidam, principalmente, com o tratamento preciso de interrupções.

### 2.3.2 Características

Sistemas criados como Reativos são muito mais flexíveis, desacoplados e escaláveis e, segundo seu manifesto [6], todos os Sistemas Reativos são caracterizados por quatro pilares bem definidos. Conforme a Figura 6, os Sistemas Reativos são:

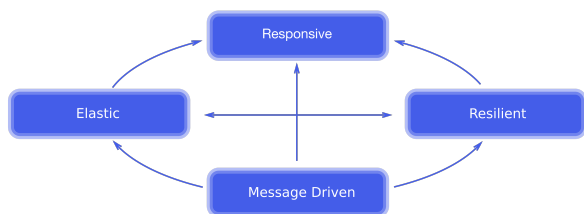


Figura 6: Características dos Sistemas Reativos [6].

- Responsivos

O sistema responde em tempo hábil se possível. Ser responsivo é fundamental para usabilidade e utilidade, mais do que isso, responsividade significa que problemas podem ser detectados rapidamente e tratados com a máxima eficácia. Sistemas Responsivos são focados em fornecer tempos de resposta rápidos e consistentes, estabelecendo limites superiores de confiança para que eles entreguem qualidade de modo consistente. Este comportamento consiste em simplificar o tratamento de erro, reforçar a confiança do usuário final e incentivar futuras interações.

- Resilientes

O sistema continua respondendo em caso de falha. Isto é aplicável não apenas para sistemas de missão crítica ou para alta disponibilidade — qualquer sistema que não é resiliente estará inutilizável depois de uma falha. Resiliência é alcançada por replicação, contenção, isolamento e delegação. A contingência à falhas é feita dentro de cada componente, isolando-os uns dos outros e assim garantindo que partes do sistema possam falhar e se recuperar sem comprometer o sistema como um todo. A recuperação de cada componente é feita por outro componente (externo) e alta disponibilidade é garantida por replicação quando necessário. Os clientes de cada componente não são sobrecarregados com o tratamento de falhas.

- Elásticos

O sistema continua responsivo mesmo sob variações de demanda. Sistemas Reativos podem reagir à mudanças na taxa de entrada através do aumento ou diminuição dos recursos alocados para lidar com estas entradas. Isso requer projetos que não tenham pontos de contenção ou gargalos centrais, resultando na habilidade de dividir ou replicar componentes e distribuir a demanda entre eles. Sistemas Reativos suportam algoritmos de escalonamento preditivos, assim como reativos, provendo métricas relevantes e em tempo real.

- Orientados à Mensagens

Sistemas Reativos usam passagem de mensagens assíncronas para estabelecer fronteiras entre os componentes e garantir baixo acoplamento, isolamento, transparência na localização e proveem meios para delegar o tratamento de erros através de mensagens. Empregar explicitamente a passagem de mensagens, modular e monitorar as filas do sistema e aplicar contra-pressão quando necessário, permite gerenciamento de demanda, elasticidade e controle de fluxo. Mensagens com transparência na localização como um meio de comunicação tornam possível o gerenciamento de falhas da mesma maneira seja em um *cluster* ou em um único *host*. Comunicação não bloqueante permite que destinatários consumam recursos quando ativos, evitando desgaste do sistema.

### 2.3.3 Vantagens

A Programação Reativa proporciona as seguintes vantagens:

- Alta manutenção

Desenvolvedores aproveitam a concorrência de uma aplicação sem as preocupações típicas de segurança e sincronização de *threads*, permitindo, devido ao controle de simultaneidade, buscar melhorias de desempenho do sistema sem medo de “quebrar” o código do lado cliente [10].

- Eficiência

Quando dados mudam, componentes são atualizados automaticamente de forma muito eficiente, somente quando necessário, resultando que todo o trabalho de

anexar e desanexar manipuladores de eventos não é mais preciso de ser assegurado pelos desenvolvedores [31].

- Robustez e Adaptabilidade

Por estar no cerne de muitos sistemas concorrentes e de alto desempenho a Programação Reativa torna-se essencial para o desenvolvimento de qualquer tipo de serviço *web*.

- Potencial

Em 2013 a empresa Gartner, líder mundial em pesquisa e consultoria em tecnologia da informação, incluiu, através da sua publicação “*Hype Cycle for Application Development*” [17], a Programação Reativa no rol de tecnologias emergentes.

O *Hype Cycle* é uma forma de representar o surgimento, adoção, maturidade e impacto em aplicações de tecnologias específicas.

### 2.3.4 Desvantagens

Para [22], devido ao desencadeamento assíncrono de eventos, torna-se difícil rastrear e entender todo o controle de fluxo de um sistema caso sua abordagem seja baseada unicamente no Padrão de Projeto *Observer*.

Segundo [23], apesar deste paradigma estar recebendo atenção crescente, definir a semântica precisa e as garantias de consistência para a Programação Reativa em ambientes distribuídos é ainda um problema de pesquisa em aberto.

Por fim, seguindo a mesma argumentação de [23], [4] cita que a manipulação de falhas de rede é uma questão ainda não completamente resolvida.

## 3. TRABALHOS RELACIONADOS

Embora não haja nenhum aplicativo móvel para a compra de passagens dos coletivos urbanos da cidade de Salvador/BA, esta seção apresenta três aplicativos de outras localidades que se inserem no contexto do tema central deste trabalho. Nesta seção são apresentadas as principais características de cada um dos aplicativos no funcionamento de suas respectivas versões para a compra de passagens dos coletivos urbanos de suas localidades.

### 3.1 Aplicativos móveis para compra de passagens de coletivos urbanos

#### 3.1.1 Ponto Certo Bilhete Único

A Rede Ponto Certo é pioneira na implantação e gestão de uma rede de recarga para transporte público e está há mais de uma década no mercado oferecendo soluções para o gerenciamento do transporte urbano. Atendendo as cidades de São Paulo/SP, Ribeirão Preto/SP, Recife/PE, Cuiabá/MT e Maceió/AL, a Rede Ponto Certo disponibiliza um aplicativo que permite aos seus usuários a compra e recarga dos créditos dos coletivos urbanos destas localidades através de seus *smartphones*.

Entre os seis aplicativos móveis existentes, são eles: dois que atendem a cidade de São Paulo/SP (Ponto Certo Bi-

lhete Único e Ponto Certo BOM), Ponto Certo Nosso (Ribeirão Preto/SP), Ponto Certo VEM (Recife/PE), Ponto Certo Bem Legal (Maceió/AL) e Ponto Certo TEM (Cuiabá/MT), um é destaque por possuir o maior número de *downloads* entre os demais: o aplicativo Ponto Certo Bilhete Único.

Com mais de 1 milhão de instalações em *smartphones* com o sistema operacional Android™, o aplicativo Ponto Certo Bilhete Único permite a compra de créditos, recarga e consulta de saldo do cartão Bilhete Único da cidade de São Paulo/SP, tendo como destaque a validação da recarga via tecnologia *NFC*<sup>7</sup> embutida em alguns dispositivos móveis.

A Figura 7 exhibe a interface de usuário do aplicativo Ponto Certo Bilhete Único com sua listagem de opções em dispositivos com o sistema operacional Android™.



Figura 7: Interface de usuário do aplicativo Ponto Certo Bilhete Único com a listagem de suas opções exibida em *smartphones* com o sistema operacional Android™.

#### 3.1.2 Moovit

Aplicativo líder em transporte público no mundo, utilizado por mais de 50 milhões de usuários em mais de 1200 cidades, o Moovit permite que usuários do cartão TEU! (Porto Alegre/RS) recarreguem seus cartões de transporte através de seus *smartphones*.

Atendendo mais de 55 mil usuários na cidade, a ferramenta auxilia a mobilidade urbana, facilitando o dia a dia de quem depende do transporte público em Cuiabá e busca agilidade não só nos momentos de recarga do cartão, mas também durante seu deslocamento diário - uma vez que o aplicativo móvel permite consultar a localização exata dos coletivos urbanos através de um mapa interativo.

<sup>7</sup>A tecnologia *NFC* (*Near Field Communication*) funciona com o uso de um campo magnético de indução, operando em um campo de ondas de rádio de 13.56 MHz [32]. Quando um outro dispositivo *NFC* fica próximo o suficiente para entrar em contato com o campo, o acoplamento magnético indutivo transfere energia e dados de um dispositivo para o outro. O uso de acoplamento magnético é a principal diferença entre o *NFC* e tecnologias como *Bluetooth* e *Wi-Fi* [27].



A interface de usuário do processo de compra de créditos do cartão TEU! é ilustrada na Figura 8.

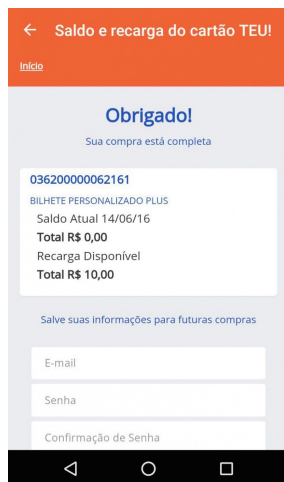


Figura 8: Interface de usuário do aplicativo Moovit para compra de créditos do cartão TEU! exibida em smartphones com o sistema operacional Android™.

### 3.1.3 Banco do Brasil

Ainda que seja um aplicativo móvel voltado principalmente à finanças, o Banco do Brasil merece destaque por pertencer à instituição bancueira, de mesmo nome, mais antiga do Brasil, fundada em 1808.

Com mais de 10 milhões de instalações em smartphones com o sistema operacional Android™, o aplicativo Banco do Brasil permite a compra de créditos do cartão Bilhete Único (São Paulo/SP) de forma bem prática, como exibido na Figura 9. O destaque fica com a possibilidade de comprar em uma das organizações mais confiáveis do mundo e que em 2016 apresentou a marca de R\$ 1,445 trilhão em ativos, líder absoluto no ranking dos maiores bancos brasileiros e latinos.

O banco é o primeiro a ser credenciado para a recarga do cartão Bilhete Único da SPTrans (Secretaria Municipal de Transportes de São Paulo), permitindo que além do aplicativo móvel, a recarga seja efetuada nos seus mais de 42 mil pontos de terminais de autoatendimento.

## 3.2 Comparativo entre os aplicativos móveis

A Tabela 1 apresenta uma análise comparativa entre os aplicativos móveis estudados.

Conforme visto na Tabela 1, diversas soluções móveis para a compra de passagens de coletivos urbanos estão disponíveis, porém, o que mais se evidencia é que dentre as soluções existentes nenhuma permite a compra de passagens dos coletivos urbanos da cidade de Salvador/BA. Este trabalho propõe então uma solução para suprir esta lacuna, justificando assim a sua escolha. Destacam-se no aplicativo MPE a possibilidade de, obviamente, adquirir passagens para os cartões Meia Passagem Estudantil, o aceite de múltiplos cartões de crédito para pagamento e a não necessidade de prévio cadastramento para que seus usuários comecem a utilizá-lo.

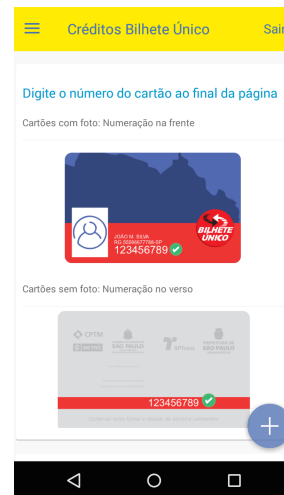


Figura 9: Interface de usuário do aplicativo Banco do Brasil durante processo de compra de créditos do cartão Bilhete Único exibida em smartphones com o sistema operacional Android™.

## 4. O SISTEMA MPE

O MPE é uma ferramenta que permite a compra de créditos de Meia Passagem Estudantil da cidade de Salvador/BA. Para a construção do seu sistema foi necessário seguir as etapas inerentes ao processo de desenvolvimento de software (e.g., levantamento e análise de requisitos, projeto e implementação). Nas subseções a seguir são apresentados os detalhes da solução desenvolvida.

### 4.1 Requisitos Funcionais

Um requisito funcional descreve o que um sistema permite fazer, seu conjunto é portanto suas funcionalidades.

São apresentados os requisitos funcionais para o sistema *mobile* e *web*, respectivamente, nas Tabelas 3 e 4, localizados no Apêndice A deste trabalho.

### 4.2 Requisitos Não-Funcionais

Um requisito não-funcional define propriedades e restrições de um sistema, além disso, inclui atributos de qualidade para um produto.

São apresentados os requisitos não-funcionais para o sistema *mobile* e *web*, respectivamente, nas Tabelas 5 e 6, localizados no Apêndice A deste trabalho.

### 4.3 Projeto Arquitetural

A arquitetura do sistema *mobile* do MPE baseia-se no MVP (*Model-View-Presenter*), uma variação do padrão arquitetural MVC (*Model-View-Controller*), onde ambos objetivam a separação de *concerns* entre a interface de usuário e a camada de negócio de uma aplicação, promovendo independência de desenvolvimento.

A Figura 10 apresenta a visão comportamental do padrão MVP do sistema *mobile* MPE.

Na visão de implantação do sistema MPE, apresentada na

	MPE	Ponto Certo Bilhete Único	Moovit	Banco do Brasil
Compra de passagens para cartões MPE	x			
Aceitação de múltiplos cartões de crédito	x	x	x	
Aceitação de múltiplos cartões de débito		x		
Suporte ao usuário através do aplicativo	x	x	x	x
Prévio cadastramento para utilização		x		x
Acompanhamento do pedido de recarga	x	x	x	

Tabela 1: Análise comparativa entre os trabalhos relacionados.

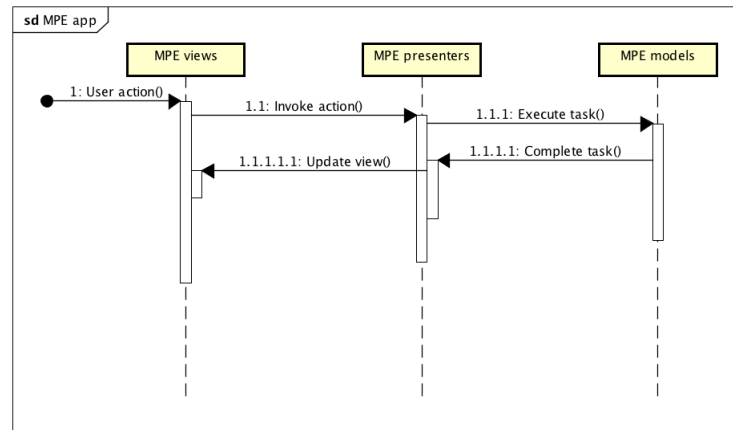


Figura 10: Visão comportamental do padrão MVP do sistema mobile MPE.

Figura 11, é possível constatar o uso do *RESTful*. O *RESTful* permite implementar serviços web baseado no *REST* (*Representational State Transfer*), um estilo arquitetural que busca minimizar a latência nas aplicações que comunicam-se via rede, além de manter a interoperabilidade entre diferentes soluções tecnológicas desenvolvidas.

#### 4.4 Projeto Detalhado

Os requisitos funcionais descritos na subseção 4.1 estão representados pelos Diagramas de Casos de Uso que o contém. Estes diagramas são composto por atores, casos de uso e seus relacionamentos. Os Diagramas de Casos de Uso do sistema *mobile* e *web* do MPE são apresentados, respectivamente, nas figuras 12 e 13.

#### 4.5 Tecnologias utilizadas

Para que o projeto MPE fosse desenvolvido fizeram-se necessárias as utilizações das seguintes principais tecnologias:

- *Amazon Simple Storage Service*

*Amazon Simple Storage Service* (*Amazon S3*) permite o armazenamento de objetos simples e amplamente escaláveis através de uma interface de *web service* para o armazenamento e recuperação de qualquer volume de dados, sendo projetado para oferecer 99,999999999% de durabilidade e possibilidade de escalar mais de 1 trilhão de objetos. Seus principais recursos, bem como suas características, incluem:

- Simplicidade: *Amazon S3* possui um *console* que permite seu fácil gerenciamento, além de entregar *APIs*

*REST* e *SDKs* (*Software Development Kit*) completos.

- Segurança: *Amazon S3* oferece suporte à transferência de dados usando *SSL* (*Secure Sockets Layer*) e criptografia automática de dados após *uploads*.

- Disponibilidade: *Amazon S3* é concebido para proporcionar até 99,99% de disponibilidade de objetos, podendo seus usuários escolherem a região onde seus recursos ficarão armazenados para otimizar a latência, minimizar os custos ou cumprir requisitos normativos.

Uma das utilizações do *Amazon S3* pelo sistema MPE é o armazenamento seguro de simples arquivos de configuração, como exemplo, é possível ativar ou desativar pedidos de novas recargas da aplicação *mobile* sem que seja necessário publicar uma nova versão a cada uma destas modificações, além disso, sua plataforma permite o armazenamento conveniente dos comprovantes de pagamento gerados a cada transação dos pedidos de recarga dos usuários.

- *Amazon DynamoDB*

*Amazon DynamoDB* é um serviço de banco de dados *NoSQL* rápido e flexível para aplicações que precisam de latência constante abaixo de 10 milissegundos em qualquer escala. Algumas das suas características incluem:

- Suporte à dupla estrutura de dados: É possível utilizar o *Amazon DynamoDB* tanto com a estrutura chave-

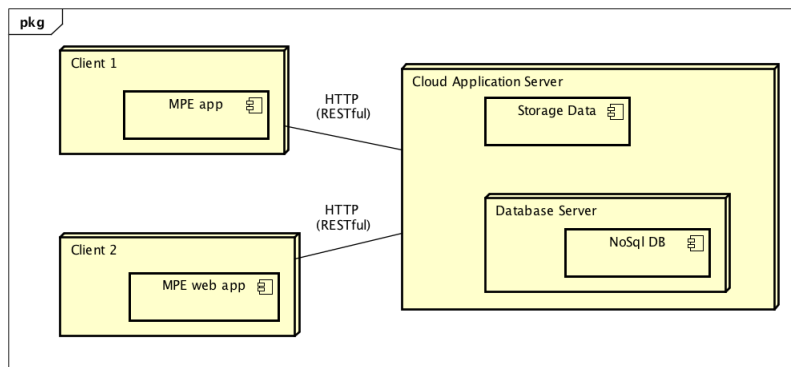


Figura 11: Diagrama de Implantação do sistema MPE.

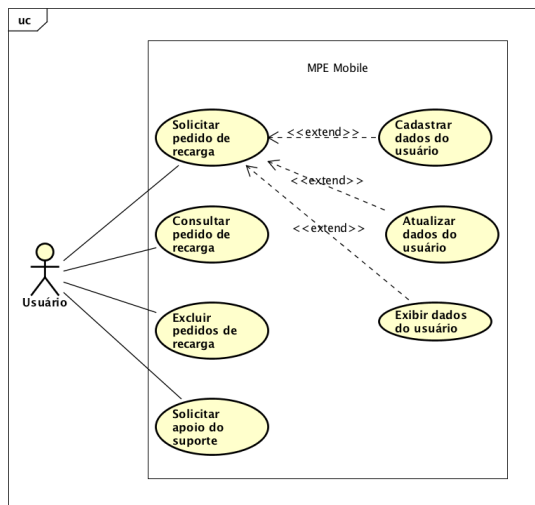


Figura 12: Diagrama de Casos de Uso do sistema mobile MPE.

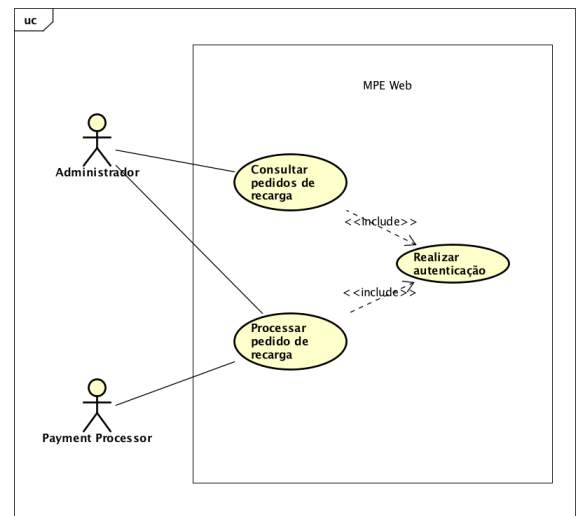


Figura 13: Diagrama de Casos de Uso do sistema web MPE.

valor quanto com a estrutura de documentos.

- Segurança: *Amazon DynamoDB* permite atribuir credenciais de segurança únicas e controlar o acesso aos serviços e recursos do banco de dados para cada usuário.

- Programação orientada à eventos: *Amazon DynamoDB* oferece *triggers* que permitem definir a arquitetura de aplicações que reagem automaticamente à alterações de dados.

Devido ao seu modelo de dados flexível e desempenho confiável o *Amazon DynamoDB* torna-se a escolha adequada para aplicativos móveis e da web, ou seja, o sistema MPE juntamente com todos os seus dados relacionados.

- *ReactiveX*

*ReactiveX* é uma API (*Application Programming Interface*) para programação assíncrona com fluxos observáveis. *ReactiveX* suporta sequências de dados e/ou

eventos e adiciona operadores que permitem compor estas sequências de forma declarativa, abstraindo preocupações sobre questões como sincronização, *thread-safety*, estruturas de dados concorrentes e E/S não-bloqueantes. Outras das suas principais características incluem:

- Redução de linhas de código: Os operadores disponibilizados pelo *ReactiveX* muitas vezes reduzem o que antes era um elaborado desafio em poucas linhas de código.

- Manipulação de erros: *ReactiveX* está equipado com mecanismos adequados para lidar com erros, já que outros tradicionais mecanismos (como *try\catch*) não são suficientes para erros em computações assíncronas.

*ReactiveX* é muito útil na versão *mobile* do sistema MPE, pois, em versões do Sistema Operacional Android™, por exemplo, garante o estado de uma transação independentemente do ciclo de vida da aplicação, sendo benéfico por evitar a duplicidade de transações

e por induzir na fácil implementação do código, garantindo o correto tratamento de erros no caso específico.

## 4.6 Interfaces de usuário

Ao informar a escolha do valor que deseja ser recarregado no cartão estudantil (Figura 14) e concluir o pedido de recarga, o usuário do aplicativo *mobile* MPE acompanha o pedido em uma seção específica no aplicativo (Figura 15). Acompanhar o pedido significa que o usuário poderá conferir em qualquer momento todas as informações relevantes da sua compra, bem como ter a exata noção de até quando a recarga estará disponível no seu cartão estudantil e ser notificado sobre questões pertinentes (Figura 16).



Figura 14: Interface de usuário do aplicativo MPE com opções de valores a serem recarregados nos cartões estudantis, exibida em *smartphones* com o sistema operacional Android™.



Figura 15: Interface de usuário do aplicativo MPE com a listagem dos pedidos de recargas, exibida em *smartphones* com o sistema operacional Android™.

Uma vez que o pedido de recarga é solicitado por um usuário através da versão *mobile* do MPE, o mesmo é gerenciado

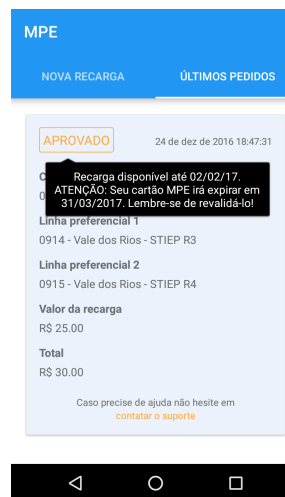


Figura 16: Interface de usuário do aplicativo MPE com notificações pertinentes, exibida em *smartphones* com o sistema operacional Android™.

através da sua versão *web*. O sistema *web* do MPE é disponibilizado sob os principais navegadores *web* do mercado e, conforme visto nas Figuras 17 e 18, é acessível independente da resolução do dispositivo que seu administrador estiver utilizando, o que ajuda-o a gerir o sistema, quer ele esteja em um escritório, utilizando um computador comum, ou quer ele esteja utilizando apenas um dispositivo móvel.

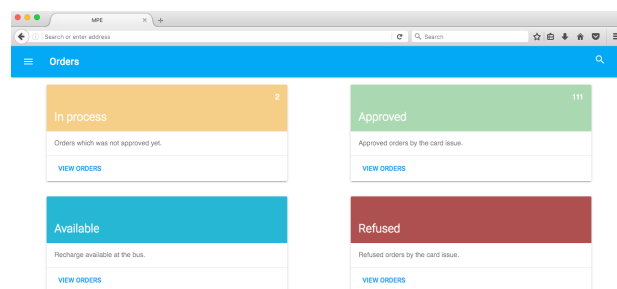


Figura 17: Interface de usuário do sistema *web* do MPE exibida em *PC's*.

## 5. ESTATÍSTICAS DE USO

Esta seção apresenta o estudo conduzido que teve como objetivo: *i*) Coletar os *feedbacks* dos usuários em relação à aplicação *mobile* desenvolvida; *ii*) Avaliar, do ponto de vista do pesquisador, a solução *mobile* considerando dados reais produzidos; e, *iii*) Analisar métricas reais que atestam a confiabilidade, disponibilidade e performance da solução MPE.

### 5.1 Análise dos *feedbacks* dos usuários

Resenhas e avaliações obtidos através da página da aplicação no *Google Play*<sup>8</sup> e em outros canais oficiais da ferramenta,

<sup>8</sup><https://play.google.com/store/apps/details?id=br.com.recargasalvadorcard>

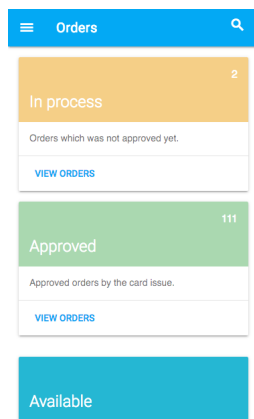


Figura 18: Interface de usuário do sistema *web* do MPE exibida em *smartphones*.

apresentados na Tabela 2, atestam o nível de satisfação dos usuários em torno da tecnologia desenvolvida.

Conforme notado, um dos maiores questionamentos dos usuários é referente ao prazo de 72 horas para que os pedidos de recarga sejam disponibilizados em seus cartões estudantis. Entre alguns, há os que não concordaram com os valores cobrados, uma vez que é necessário debitar um montante repassado para as administradoras de cartões de crédito. Entretanto, para ampla maioria, e de modo geral, a solução é vista como um ótimo utilitário, funcionando muito bem e de modo bastante prático.

Comentário
“Raramente faço comentários sobre aplicativos, mas esse merece uma exceção. Muito prático. Funciona perfeitamente bem. O único problema é o prazo de 72h, necessitando assim de uma programação prévia do usuário. Fiz o pedido no dia 18/11 e a recarga caiu [sic] ontem, dia 22/11, devido ao final de semana.”
“Taxa? Que história é essa de solicitar R\$ 30,00 de recarga e ser cobrado R\$ 35,50? Não estou entendendo.”
“Olá, boa tarde. Achei o aplicativo muito interessante e útil [...]”
“[...] O app é bastante útil. [...] Amei.”
“O prazo não condiz com a praticidade necessária. Se me deslocando é imediato via App deveria ser algo próximo disto.... 72horas é inviável esperar diante da tecnologia atual.”
“Muito bom. Vai facilitar muito a minha vida.”
“Massa! App muito bom para recarregar meu cartão. Não preciso pegar mais fila! Muito bom o serviço.”
“Muito bom! Cumpre bem o que promete! De forma rápida eu posso solicitar uma recarga para meu cartão. Quando a recarga cair [sic] no mesmo dia será muito melhor! Mas já salva [sic] meu dia.”
“Amei *-* É só não deixar para recarregar de última hora por causa do prazo.”

Tabela 2: Resenhas sobre a aplicação *mobile* obtidas através dos seus canais oficiais.

## 5.2 Análise da solução *mobile*

A solução *mobile* foi analisada através de estatísticas que permitiram obter indicadores precisos sobre a mesma. Embora a solução não tenha sido amplamente divulgada, foi possível obter o total de mais de 1460 instalações em dispositivos móveis durante seu período de lançamento até o momento da finalização deste trabalho, conforme visto na Figura 19.



Figura 19: Número de instalações por usuário do sistema *mobile* MPE.

Outro indicador valioso refere-se ao percentual de versões do sistema operacional Android<sup>TM</sup> em que a solução esteve implantada. Conforme visto na Figura 20, a solução *mobile* cumpriu o requisito de estar disponível nas mais modernas versões dos sistemas móveis operacionais presentes nos dispositivos dos usuários, adequando-se ao proposto no requisito não-funcional Compatibilidade (Tabela 5). É possível verificar também, através da mesma figura, o mesmo percentual em relação à outras soluções mundialmente disponíveis, no segmento Compras.

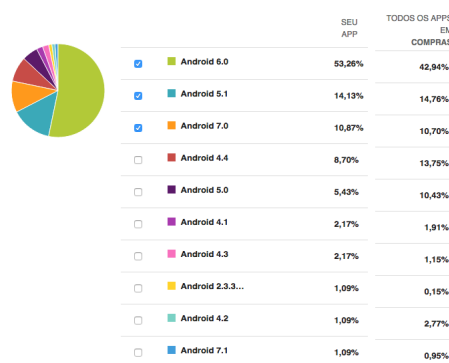


Figura 20: O sistema *mobile* MPE e seu percentual de instalações por versões do sistema operacional Android<sup>TM</sup>.

### 5.3 Métricas providas

Para atender as necessidades de *throughput* do sistema MPE como um todo, garantindo desempenho consistente e de baixa latência, fizeram-se necessário o rastreamento e a coleta de métricas dos seus recursos. Conforme visto na Figura 21, durante o período de 10 meses, compreendido entre os dias 31/05/2016 à 31/03/2017, sua capacidade de escrita manteve-se em conformidade ao que foi proposto no requisito não-funcional Volume de utilização (Tabela 5). Através da figura, é possível visualizar, por um gráfico, o quanto de capacidade de escrita foi provisionada e consumida durante o período, atestando sua normalidade.

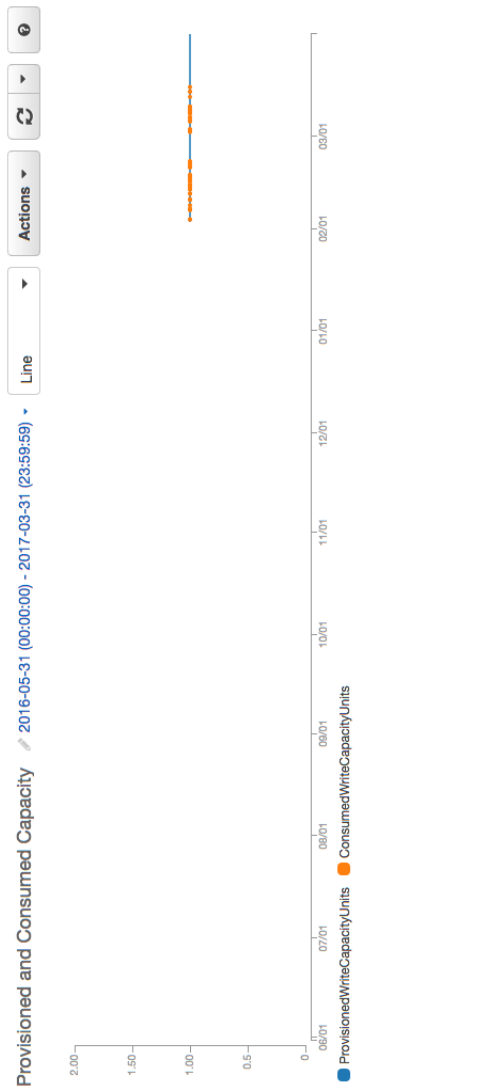


Figura 21: Capacidade de escrita provisionada e consumida do sistema MPE, durante período de 10 meses.

No mesmo período compreendido ao anteriormente citado, foi possível constatar a anormalidade da relação entre a capacidade de leitura dos recursos que foram provisionados e consumidos (Figura 22), entretanto, com ajustes manuais e automáticos, foi possível a normalização do mesmo, conforme visto na Figura 23.

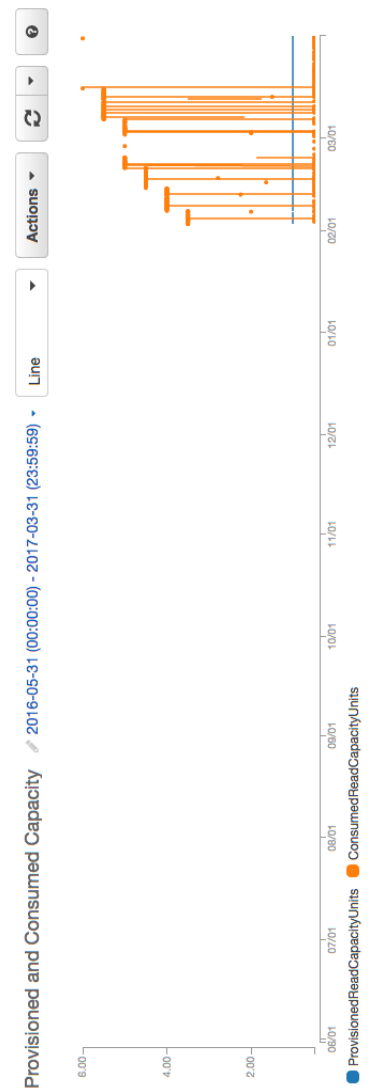


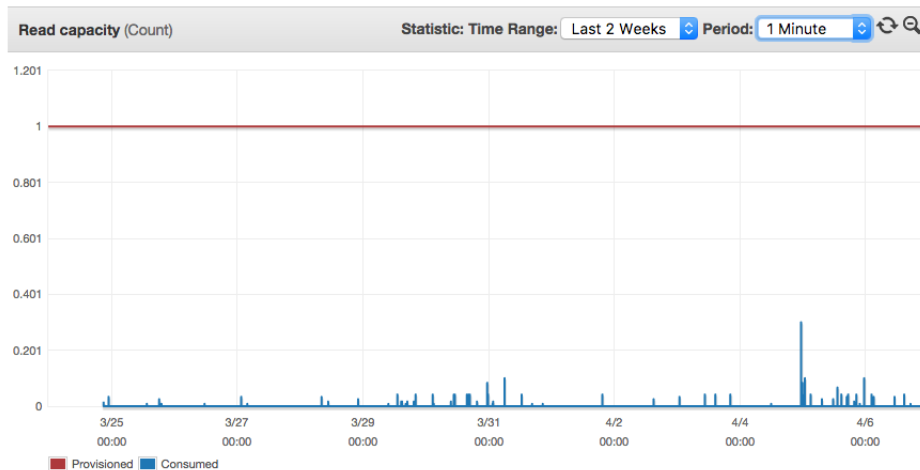
Figura 22: Capacidade de leitura provisionada e consumida do sistema MPE, durante período de 10 meses.

## 6. CONCLUSÃO

Apesar da pouca publicidade acerca da solução desenvolvida - como ferramentas utilitárias comumente possuem, como visto em Trabalhos Relacionados -, a ferramenta MPE demonstrou um elevado grau de eficiência e utilidade ao cumprir seus objetivos esperados. Este resultado sugere que as aplicações móveis tornaram-se cada vez mais essenciais (e por causa disso, cada vez mais onipresentes) na vida humana.

O sistema MPE contribui no dia-a-dia dos milhares de estudantes da capital soteropolitana, entregando uma forma mais prática, fácil e tecnologicamente mais rápida na compra dos créditos de Meia Passagem Estudantil.

Pôde-se observar que, dado a tecnologia atual, poucas são as restrições que impedem o desenvolvimento e progresso de soluções tradicionais para soluções móveis.



**Figura 23:** Capacidade de leitura provisionada e consumida do sistema MPE após ajustes manuais e automáticos, durante período compreendido entre 25/03/2017 à 06/04/2017.

Foi possível notar também que, apesar da disponibilidade tecnológica existente, barreiras burocráticas podem retardar o uso de soluções inovadoras - uma vez que para dar continuidade ao projeto MPE foi necessário adequar-se à questões existentes do órgão público responsável por controlar e gerir o sistema de transporte urbano da localidade.

Como trabalhos futuros podem ser elencados as seguintes gerências: aceitação de cartões de débito, melhoria na interface de usuário para *tablets*, expansão da ferramenta para outros sistemas móveis operacionais e integração com *web services* do órgão controlador do sistema de transporte urbano de Salvador/BA, como meio à melhorar a validação dos cartões estudantis dos usuários do sistema MPE.

## 7. REFERÊNCIAS

- [1] M. Abourezq and A. Idrissi. Database-as-a-service for big data: An overview. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 7(1), 2016.
- [2] C. I. Agency. The world factbook - brazil. <https://www.cia.gov/library/publications/the-world-factbook/geos/br.html>. Acessado: 13/11/2016.
- [3] A. Alzahrani, N. Alalwan, and M. Sarrab. Mobile cloud computing: Advantage, disadvantage and open challenge. In *Proceedings of the 7th Euro American Conference on Telematics and Information Systems, EATIS '14*, pages 21:1–21:4, New York, NY, USA, 2014. ACM.
- [4] E. Bainomugisha, A. L. Carreton, T. v. Cutsem, S. Mostinckx, and W. d. Meuter. A survey on reactive programming. *ACM Comput. Surv.*, 45(4):52:1–52:34, Aug. 2013.
- [5] G. Berry. Real time programming : special purpose or general purpose languages. Research Report RR-1065, INRIA, 1989.
- [6] J. Bonér, D. Farley, R. Kuhn, and M. Thompson. The reactive manifesto. <http://www.reactivemanifesto.org/>. Acessado: 10/03/2017.
- [7] J. Brackebush. How mobile is overtaking desktop for global media consumption, in 5 charts. <http://digiday.com/publishers/mobile-overtaking-desktops-around-world-5-charts>. Acessado: 13/11/2016.
- [8] S. Card. Estudantes já podem fazer recarga no salvadorcard pela internet. <http://www.salvadorcard.com.br/noticias/estudantes-ja-podem-fazer-recarga-no-salvadorcard-pela-internet/>. Acessado: 15/01/2017.
- [9] S. Card. Institucional. <http://www.salvadorcard.com.br/institucional/>. Acessado: 11/01/2017.
- [10] B. Christensen and J. Husain. Reactive programming in the netflix api with rxjava. <http://techblog.netflix.com/2013/02/rxjava-netflix-api.html>. Acessado: 11/03/2017.
- [11] I. C. de Trabalho e Rendimento. *Acesso à internet e à televisão e posse de telefone móvel celular para uso pessoal : 2014*. Coleção Ibgeana. IBGE, Rio de Janeiro, 1 edition, 2016. Acima do título: Pesquisa Nacional por Amostra de Domicílios. Bibliografia: [61]-64. Inclui glossário.
- [12] H. T. Dinh, C. Lee, D. Niyato, and P. Wang. A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless communications and mobile computing*, 13(18):1587–1611, 2013.
- [13] A. Faria. Programação reativa (reactive programming). <http://blog.andrefaria.com/programacao-reativa-reactive-programming>. Acessado: 15/01/2017.
- [14] S. GmbH. Global market share held by the leading smartphone operating systems in sales to end users from 1st quarter 2009 to 3rd quarter 2016. <https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems>. Acessado: 30/11/2016.

- [15] S. GmbH. Number of smartphone users worldwide from 2014 to 2020 (in millions). <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide>. Acessado: 12/11/2016.
- [16] J. Han, E. Haihong, G. Le, and J. Du. Survey on nosql database. In *Pervasive computing and applications (ICPCA), 2011 6th international conference on*, pages 363–366. IEEE, 2011.
- [17] G. Inc. Hype cycle for application development, 2013. <https://www.gartner.com/doc/2560015/hype-cycle-application-development->, July 2013. Acessado: 11/03/2017.
- [18] L. Kleinrock. A vision for the internet. *ST Journal of Research*, 2(1):4–5, 2005.
- [19] V. Kottari, V. Kamath, L. P. Saldanha, and C. Mohan. A survey on mobile cloud computing: Concept, applications and challenges. *International Journal of Advanced and Innovative Research*, 2(3):487–492, 2013.
- [20] N. Leavitt. Will nosql databases live up to their promise? *Computer*, 43(2), 2010.
- [21] P. S. T. Magalhães and H. D. d. Santos. Biometria e autenticação. *Actas da 4ª Conferência da Associação Portuguesa de Sistemas de Informação*, pages 3–5, 2003. CONFERÊNCIA DA ASSOCIAÇÃO PORTUGUESA DE SISTEMAS DE INFORMAÇÃO, 4, Porto, 2003 - "CAPSI 2003 : actas" [CD-ROM]. [S.l. : APSI, 2002]. ISBN 972-9354-42-1.
- [22] I. Maier and M. Odersky. Deprecating the observer pattern with scala. react. Technical report, react. Technical report, 2012.
- [23] A. Margara and G. Salvaneschi. We have a dream: Distributed reactive programming with consistency guarantees. In *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*, DEBS '14, pages 142–153, New York, NY, USA, 2014. ACM.
- [24] P. Mell, T. Grance, et al. The nist definition of cloud computing. 2011.
- [25] A. Moniruzzaman and S. A. Hossain. Nosql database: New era of databases for big data analytics-classification, characteristics and comparison. *arXiv preprint arXiv:1307.0191*, 2013.
- [26] A. Nayak, A. Poriya, and D. Poojary. Type of nosql databases and its comparison with relational databases. *International Journal of Applied Information Systems*, 5(4):16–19, 2013.
- [27] S. Ortiz. Is near-field communication close to success? *Computer*, 39(3):18–20, 2006.
- [28] K. W. Panel. Smartphone os sales market share. <http://www.kantarworldpanel.com/global/smartphone-os-market-share>. Acessado: 13/11/2016.
- [29] D. Parkhill. *The Challenge of the Computer Utility*. Addison-Wesley Publishing Company, 1 edition, 1966.
- [30] J. Richards. Advantages and disadvantages of nosql databases – what you should know. <http://www.hadoop360.com/blog/advantages-and-disadvantages-of-nosql-databases-what-you-should-k>. Acessado: 01/03/2017.
- [31] S. Stefanov. Remarkable react. <http://www.phpied.com/remarkable-react/>, Aug. 2013. Acessado: 11/03/2017.
- [32] R. Want. Near field communication. *IEEE Pervasive Computing*, 10(3):4–7, 2011.
- [33] Yonathan. O que é reactive programming (programação reativa)? <http://pt.stackoverflow.com/a/55380/41350>. Acessado: 10/03/2017.

## APÊNDICE

### A. ELICITAÇÃO DE REQUISITOS

As tabelas a seguir expõem os requisitos funcionais e não funcionais do sistema MPE.



Requisito funcional	Descrição
Solicitar pedido de recarga	O usuário do sistema <i>mobile</i> deverá ser capaz de solicitar um pedido de recarga de quaisquer valores disponíveis, para tanto, deverá informar a escolha do valor a ser recarregado no cartão estudantil, o número do cartão estudantil, as linhas de ônibus preferenciais nas quais a recarga aprovada será disponibilizada e os dados para o pagamento da compra (número, <i>CVV</i> , data de expiração e nome do titular do cartão de crédito). Um pedido de recarga não aprovado ocorrerá nas seguintes circunstâncias: quando o número do cartão estudantil informado for inválido, quando o cartão estudantil estiver expirado ou quando o banco emissor do cartão de crédito negar o pagamento da compra.
Consultar pedido de recarga	O usuário do sistema <i>mobile</i> poderá consultar em um histórico seus pedidos de recarga, incluindo aqueles pedidos que foram aprovados, não aprovados ou que ainda aguardam processamento.
Excluir pedidos de recarga	O usuário do sistema <i>mobile</i> , através do histórico que lista seus pedidos de recarga, poderá excluir todos os pedidos de recarga que foram solicitados através do seu dispositivo móvel. Esta ação excluirá apenas os pedidos de recarga armazenados no dispositivo móvel do usuário.
Solicitar apoio do suporte	O usuário do sistema <i>mobile</i> poderá solicitar apoio do suporte MPE a partir de qualquer pedido de recarga listado no histórico que o contém.
Cadastrar dados do usuário	Todos os dados que foram informados após e somente na primeira solicitação de um pedido de recarga (número do cartão estudantil, linhas de ônibus preferenciais, nome do titular do cartão de crédito e número do cartão de crédito) deverão ser cadastrados no dispositivo móvel do usuário.
Atualizar dados do usuário	Dados do usuário (número do cartão estudantil, linhas de ônibus preferenciais, nome do titular do cartão de crédito ou número do cartão de crédito) que previamente estejam armazenados no dispositivo móvel do usuário e que forem modificados em um pedido de recarga subsequente deverão ser atualizados.
Exibir dados do usuário	Dados do usuário (número do cartão estudantil, linhas de ônibus preferenciais, nome do titular do cartão de crédito e número do cartão de crédito) deverão ser exibidos durante e a partir da segunda solicitação de um pedido de recarga caso estes mesmos dados estejam cadastrados no dispositivo móvel. Com exceção do número do cartão de crédito, que deverá ser exibido com seus primeiros 12 dígitos ocultados, todos os outros dados deverão ser exibidos normalmente, não ocultados, por completo.

**Tabela 3: Requisitos funcionais do sistema *mobile* MPE.**

Requisito funcional	Descrição
Realizar autenticação	Para que o administrador do sistema <i>web</i> possa ter acesso à aplicação, o sistema deverá ser capaz de validar as credenciais informadas (nome de usuário e senha) durante processo de <i>login</i> .
Consultar pedidos de recarga	O administrador do sistema <i>web</i> , devidamente autenticado, deverá ser capaz de consultar todos os pedidos de recarga efetuados através do sistema <i>mobile</i> . A consulta será realizada através da listagem dos pedidos aprovados, não aprovados ou que ainda não foram processados. Filtros deverão ser disponibilizados para permitir a consulta através do identificador do pedido de recarga, do nome do usuário titular do cartão de crédito ou do número do cartão estudantil.
Processar pedido de recarga	O administrador do sistema <i>web</i> , devidamente autenticado, deverá ser capaz de processar os pedidos de recarga que ainda aguardam processamento. Durante o processamento de um pedido, o administrador poderá não aprová-lo de acordo com as seguintes circunstâncias: quando o número do cartão estudantil informado for inválido, quando o cartão estudantil estiver expirado ou quando o banco emissor do cartão de crédito negar o pagamento da compra.

**Tabela 4: Requisitos funcionais do sistema *web* MPE.**

Requisito não-funcional	Descrição
Volume de utilização	O sistema <i>mobile</i> deverá suportar uma carga mínima de 5 usuários simultâneos por segundo durante o processamento de um pedido de recarga.
Disponibilidade	O sistema <i>mobile</i> deverá estar disponível durante 90% do tempo em dias úteis da semana e 80% do tempo em finais de semana ou feriados.
Compatibilidade	O sistema <i>mobile</i> deverá ser compatível com ao menos as últimas 4 versões disponibilizadas pelo sistema móvel operacional no qual estiver implantado.
Usabilidade	O sistema <i>mobile</i> deverá seguir as últimas diretrizes de <i>design</i> do sistema móvel operacional no qual estiver implantado. O sistema também deverá permitir que ações recorrentemente utilizadas sejam facilmente realizadas, por exemplo: o usuário deverá ser capaz de visualizar o último número do cartão estudantil anteriormente informado em um pedido de recarga. Durante uma nova solicitação de pedido de recarga, caso o número do cartão estudantil a ser informado seja o mesmo, não haverá necessidade de sua inclusão neste novo pedido e sim, apenas de sua confirmação. O mesmo deverá ocorrer para os seguintes dados: linhas de ônibus preferenciais, número do cartão de crédito (com seus dígitos ocultados - com exceção dos últimos quatro) e nome do titular do cartão de crédito. A soma destas decisões irá refletir diretamente na melhor <i>UX</i> (experiência do usuário) e <i>UI</i> (interface do usuário) do sistema <i>mobile</i> .
Segurança	O sistema <i>mobile</i> utilizará recursos externos somente via protocolo <i>HTTPS</i> ( <i>Hyper Text Transfer Protocol Secure</i> ), além disso, nenhuma senha para acesso a tais recursos deverá ser implementada <i>hard-coded</i> . Ademais, todas informações cadastradas no sistema deverão estar privadas, ou seja, apenas o sistema MPE os visualiza, altera ou acessa.
Internacionalização	O sistema <i>mobile</i> deverá ser disponibilizado em inglês e português mas de forma que estas variações linguísticas sejam definidas automaticamente de acordo com o idioma do dispositivo móvel do usuário.
Manutenibilidade	Desenvolvedores deverão ser capazes de alterar facilmente a estrutura do sistema <i>mobile</i> para suportar mudanças. A capacidade de ser ágil na mudança do sistema <i>mobile</i> se deve à sua própria natureza - novas versões geralmente são lançadas frequentemente.

**Tabela 5: Requisitos não-funcionais do sistema *mobile* MPE.**

Requisito não-funcional	Descrição
Disponibilidade	O sistema <i>web</i> deverá estar 95% disponível em dias úteis da semana, entre às 08:00h e 17:00h.
Integridade	O sistema <i>web</i> somente poderá ser acessado por usuários autenticados.
Interoperabilidade	O sistema <i>web</i> deverá ser capaz de funcionar com quaisquer sistemas processadores de pagamento.
Software	O sistema <i>web</i> deverá funcionar ao menos com as últimas versões dos seguintes navegadores: Google Chrome <sup>TM</sup> , Safari <sup>TM</sup> e Firefox <sup>TM</sup> .
Usabilidade	A interface do sistema <i>web</i> deverá funcionar adequadamente independente de como se dará seu acesso – via <i>browser</i> , <i>smartphone</i> ou <i>tablet</i> .

**Tabela 6: Requisitos não-funcionais do sistema *web* MPE.**