Índice

[Software 2](#_Toc66467235)

[Resumo Executivo 2](#_Toc66467236)

[Operação 2](#_Toc66467237)

[Entidades envolvidas 3](#_Toc66467238)

[Sistemas / Interfaces 3](#_Toc66467239)

[Tecnologias 3](#_Toc66467240)

[Documentação 4](#_Toc66467241)

[Legado (2018- 2021) 4](#_Toc66467242)

[Atualização (2021 em diante) 4](#_Toc66467243)

[Nomes de arquivos 4](#_Toc66467244)

[Organização 5](#_Toc66467245)

[Projeto de Software 6](#_Toc66467246)

[Diretórios para os sistemas 6](#_Toc66467247)

[FrontEnd (web2) 7](#_Toc66467248)

[Backend (api2) 8](#_Toc66467249)

[Backend (api2) - testes internos 9](#_Toc66467250)

|  |  |  |
| --- | --- | --- |
| Data | Versão | Responsável |
| 12/03/2021 | 1.0 – Esboço | R. Groff |
|  |  |  |

# Software

## Resumo Executivo

A ConveyNET é uma **processadora de cartões convênio**, que une uma entidade emissora de cartões (físicos, em plástico) de seus associados a uma lista de estabelecimentos qualificada – como o próprio nome diz, uma rede conveniada.

O papel da ConveyNET é registrar as transações dos portadores e os lojistas, para posteriormente, num evento determinado como **fechamento**, termos os valores que deverão ser creditados pela emissora ao lojista a cada mês, fechando as parcelas das compras que vencem no dia determinado pela emissora.

**A emissora** lucra com um percentual sobre as vendas, chamado de repasse. Além disso, a emissora de cartões definirá a quantidade máxima de parcelas que as compras poderão ser feitas, e através do salário de seu associado, determinar um valor para seus limites financeiros de endividamento.

## Operação

Os portadores de cartão físico (embossados) podem ter:

* Limite mensal – definido pela emissora;
* Limite total – normalmente 20% de seu salário;
* Dependentes de cartão, que usam o limite do cartão titular;

Ações ainda sobre os cartões:

* Bloqueio;
* Desbloqueio;
* Segunda-via;
* Cota-extra;
* Dados bancários dos portadores;

## Entidades envolvidas

Seguem todas as partes envolvidas no processo:

* **CNET** – Processadora de convênios, que autoriza e gerencia os cartões;
* **Emissoras** – Associações (prefeituras, sindicatos) que possuem usuários que se beneficiarão dos cartões convênio;
* **Portadores** – Associados que farão as compras com o cartão na rede de convênios;
* **Lojistas** – Estabelecimentos comerciais que viabilizarão suas vendas (com parcelamento inclusive) em diversas emissoras;

## Sistemas / Interfaces

A ConveyNET, processadora de cartões convênio, tem por suas necessidades, diversos sistemas de atendimento e visões de seus processos. Seguem os mesmos:

* **Processamento ISO**: recebe, via sockets, estruturas de dados em formato ISO6563 para transações de vendas, estornos (etc) da plataforma SOFTWARE EXPRESS;
* **Mobile App (\*)**: para uso dos portadores dos cartões visualizarem seus extratos, parcelamentos e rede credenciada de estabelecimentos comerciais;
* **Site responsivo**: para uso de todas as partes interessadas na manutenção do software e de seus entregáveis;
* **API**: processamento de servidor para todas as transações via REST;

(\*) Um aplicativo registrado no google play e na loja da Apple apenas abre um site responsivo dentro do celular, e desta maneira não tem manutenções diretas.

## Tecnologias

Tecnologias empregadas:

* **Legado**: construído em 2018, migrou a empresa de WEB FORMS (2009 – 2018) para API / Angular / React e banco SQL Server;
* **Atualização**: para homologação até o inicio de 2022, teremos uma evolução para .NET Core 5 para API / Mobile e site progressivo usando ES6 / Node, além da conversão do banco de dados MS para PostgreSQL;

# Documentação

## Legado (2018- 2021)

Apenas manutenção do software, sem nenhuma documentação processual ou de artefatos entregáveis de código / projeto.

## Atualização (2021 em diante)

Teremos um novo processo de software respeitando as seguintes diretivas:

1. Atenção sempre pela prioridade da necessidade da ConveyNET;
2. Tickets de atendimento registrados via email / whatsapp;
3. **Projeto de software em abstração de algoritmos e interfaces;**
4. Aprovação / revisão antes de execução;
5. Codificação / testes unitários e de integração;
6. Prototipação em homologação;
7. Revisão final, e ajustes sugeridos, voltando para etapa [3];
8. **Demanda aprovada em homologação;**
9. **Entrega final em produção;**

## Nomes de arquivos

Todos os artefatos e documentos serão passíveis de ajustes. Até mesmo novos entendimentos em processos poderão ser revistos e registrados conforme o andamento das tarefas e performance da equipe.

Assim, para cada versão dos arquivos, teremos um padrão de nomenclatura em “\_revXX.doc”, onde XX de mais alto valor será a versão atual do documento.

## Organização

A construção do software compreende-se entre o registro de diversas features e ajustes do projeto e a atual documentação de cada subsistema de informação dentro da plataforma de serviços da ConveyNET.

Como toda a documentação estará em forma de arquivos em um diretório virtual, teremos a seguinte orientação para a estrutura de arquivos:

|  |  |
| --- | --- |
| ./docs | Neste diretório teremos todos os arquivos de documentação.  Software\_revXX.docx  Neste arquivo teremos todos os detalhes de como conduzir a construção de software. |
| ./docs/features | **Features\_revXX.docx**  Neste arquivo teremos o registro das features, tal como um resumo, data de sua criação, prioridade e importância na ConveyNET. |
| ./docs/features/AA | Cada requisito (feature) precisa de um diretório para registro dos requisitos de software a serem atendidos.  **“AA”** será um nome curto dado à feature para identificá-la, e também será o **nome de um diretório**. |

# Projeto de Software

## Diretórios para os sistemas

A documentação será feita apenas para software da atualização – o software legado será descontinuado sem alterações nas decisões de mantê-lo sem gerência de projeto.

Os arquivos onde teremos o projeto de software seguem como:

|  |  |
| --- | --- |
| ./docs/front | Diretório onde temos sistemas de interface com o usuário. |
| ./docs/front/web2 | O projeto **web2** é a nova interface mobile first, progressive app, ES6 / node, para os usuários da ConveyNET e todas as visões do software. |
| ./docs/back | Diretório **back** onde temos todas as descrições dos processos automatizados em nivel de servidor da empresa. |
| ./docs/back/api2 | O projeto **api2** é o novo aplicativo .NET 5 para API, e é responsável pelo processamento REST de todas as regras de negócio da ConveyNET. |
| ./docs/back/iso2 | O projeto **iso2** é o aplicativo .NET para processamento de sockets e ISO8563 das transações de venda. |
| ./docs/back/fech | O projeto **fech** é um aplicativo .NET para executar fechamento da empresa, disparando tarefas em schedulamento diárias. |

## FrontEnd (web2)

A interface para aplicativos mobile e desktop web será a mesma, pois escolhemos uma arquitetura progressiva de baixo custo.

A tecnologia escolhida será ES6 puro e NODE para hospedagem do site.

Os arquivos onde teremos o projeto de software seguem como:

|  |  |
| --- | --- |
| ./docs/front/web2/infra | Diretório onde residem os documentos gerais do aplicativo. |
| ./docs/front/web2/components | Diretório onde residem os documentos gerais do aplicativo. |
| ./docs/front/web2/pages | Diretório onde estão os documentos finais específicos sobre cada rota pública / privada do aplicativo. |

## Backend (api2)

A plataforma da ConveyNET tem uma arquitetura mesclada de SOA com DDD, construída em ambiente .NET Core.

**SOA** (Service Oriented Archictecture) – é uma prática utilizada para uma abstração de classes levando em conta “serviços”, onde um contexto de entrada resulta em um processamento que entrega um contexto de sáida.

**DDD** (Domain Driven Design) – é um metodo de agrupar serviços e classes que resulta em código mais limpo e força a criação de módulos e reaproveitamento de código; além disso, reforça o conceito de “single responsability” que é uma boa prática de desenvolvimento.

Os arquivos onde teremos o projeto de software seguem como:

|  |  |
| --- | --- |
| ./docs/back/api2/Controller | Diretório para recebimento REST da API. |
| ./docs/back /api2/Data | Diretório de definições das classes DTO. |
| ./docs/back /api2/Infra | Classes base da API |
| ./docs/back /api2/Repository | Classes para acesso ao banco de dados. |
| ./docs/back /api2/Services | Serviços para atendimento das requisições à API. |

## Backend (api2) - testes internos

Além das documentações de projeto, teremos dentro do processo de codificação as duas etapas abaixo:

* Testes **unitários** (usando mocks dos repositórios);
* Testes de **integração** (usando um ambiente de teste local real);

Estes processos acima serão executados pelo time de desenvolvimento em apoio pelo seu líder técnico – e não possuem processo definido.