ATLÂNTICA
UNIVERSITY HIGHER INSTITUTION

Licenciatura em Sistemas e Tecnologias de Informação

Blockchain nas organizações de ajuda humanitária

Projeto Final de Licenciatura

Elaborado por: José Neves

Nº: 20172195

Orientador Professor Doutor José Braga de Vasconcelos

Barcarena

Junho 2019

Licenciatura em Sistemas e Tecnologias de Informação

Blockchain nas organizações de ajuda humanitária

Projeto Final de Licenciatura

Elaborado por: José Neves

Nº: 20172195

Orientador Professor Doutor José Braga de Vasconcelos

Barcarena

Junho 2019

O autor é o único responsável pelas ideias expressas neste relatório

Lista de Abreviaturas e siglas

ACL – Access Control List

AML – Anti-Money Laundry

API – Application Programing Interface

CA – Certification Authority

CERF – Central Emergency Response Fund

CPU – Central Processing Unit

DFDI – Department For International Development

DLT – Distributed Ledger Technology

DSR – Design Science Research

EEE – Espaço Económico Europeu

ECOSOC – Economic and Social Council

FCA – Financial Conduct Authority

FTL – Frontier Technology Livestream

GDPR – General Data Protection Regulation

HPG – Humanitarian Policy Group

HTTP – Hypertext Transfer Protocol

IFRC – International Federation of Red Cross and Red Crescent Societies

KRCS – Kenya Red Cross society

KYC – Know Your Customer

MSP – Membership Service Provider

NGO – Non Governmental Organization

OCHA –Office for the Coordination of Humanitarian Affairs

ONU – Organização das Nações Unidas

P2P – Peer-to-peer

PKI – Public Key infrastructure

POS – Point Of Sale

RAM – Random Access Memory

REST – Representational State Transfer

SDK – Software Developer Kit

UN – United Nations

UNDP – United Nations Development Programme

UNHCR – United Nations High Commissioner for Refugees

UNICEF - United Nations children's Fund

USAID – United States Agency for International Development

WFP – World Food Programme

WVI – World Vision International

ABSTRACT

The increasing needs in the humanitarian sector, due to recent refugee's crisis and other natural disasters, and the lack of funding to address these needs have led to a paradigm shift in the sector. Humanitarian organizations are increasingly needing of focusing on innovation and emerging technologies to meet humanitarian needs in a better and more efficient way, also increasing transparency and trust.

As several literature acknowledges, the Blockchain technology has the ability to increase transparency and enhance efficiency, standardizing processes, increase collaboration and reduce expenditures, such as transaction costs, and people's time across industry. One of the most ambitious application of the Blockchain technology is smart contracts, but there are other applications of this technology as well that can benefit the organizations using this technology.

The works was conducted following the Design Science Research methodology, designing, implementing, testing and gather evidences to demonstrate that Blockchain is able to bring transparency and efficiency to organizations working in humanitarian aid.

During this work a Humanitarian aid Blockchain business network was modeled and built, using the HyperLedger Platform. The purpose was to prove that Blockchain is able to provide greater benefits on transparency and efficiency, to non-governmental organizations working in humanitarian aid. A set of test use cases were conducted, using specific conditions to validate how the HyperLedger Blockchain platform and the implemented business network can address transparency and efficiency issues of the organizations working in the humanitarian aid sector. Other benefits regarding traceability, accountability, audit, security and privacy when accessing private data were also validated.

The conclusion from this works was that Blockchain technology can effectively bring transparency and efficiency to organizations working in the humanitarian aid sector.

Keywords: Blockchain; HyperLedger; Transparency; Efficiency; humanitarian aid; Non-Government Organizations; Smart Contracts;

RESUMO

As necessidades crescentes no setor humanitário, devido à recente crise de refugiados e alguns desastres naturais, e a falta de financiamento para atender a todas essas necessidades levaram a uma mudança de paradigma no setor. As organizações humanitárias estão a necessitar cada vez mais de colocar mais foco na inovação e nas tecnologias emergentes para endereçar melhor e de forma mais eficiente as necessidades humanitárias, aumentando também a transparência e a confiança.

Como várias publicações reconhecem (pela revisão a literatura), a tecnologia *Blockchain* tem a capacidade de aumentar a transparência e melhorar a eficiência, padronizando processos, aumentando a colaboração e também na redução custos (como os custos associados a transações monetárias e o tempo gasto por pessoas), independente da indústria. Uma das aplicações mais ambiciosas da tecnologia *Blockchain* está relacionada com os *Smart Contracts*, mas também existem outras aplicações desta tecnologia que podem beneficiar as organizações quando usam a tecnologia *Blockchain*.

Este trabalho foi conduzido seguindo a metodologia da *Design Science Research*, no desenho, implementação, testes e na recolha de evidências para demonstrar que a *Blockchain* traz benefícios de transparência e eficiência nas organizações que trabalham na ajuda humanitária.

Durante este trabalho, foi modelada e implementada uma rede de negócios *Blockchain* de ajuda humanitária, usando a Plataforma *HyperLedger*. O objetivo visou assim provar que o *Blockchain* é capaz de adicionar benefícios na transparência e eficiência das organizações não-governamentais que trabalham na ajuda humanitária. Foi realizado um conjunto de teste com casos de uso, usando condições específicas para validar, como a plataforma *Blockchain* da *HyperLedger* e a rede de negócios implementada, podem abordar as questões de transparência e eficiência das organizações que trabalham no setor de ajuda humanitária. Outros benefícios relacionados com a rastreabilidade, responsabilidade, auditoria, segurança e privacidade no acesso a dados privados também foram validados.

A conclusão deste trabalho foi que a tecnologia *Blockchain* pode efetivamente trazer transparência e eficiência nas organizações que trabalham no setor de ajuda humanitária.

Palavras-chave: *Blockchain*; *HyperLedger*; Transparência; Eficiência; Ajuda Humanitária; Organizações Não Governamentais; Contratos Inteligentes;

ÍNDICE

ÍNDICE DE TABELAS

1 INTRODUÇÂO

1.1 Contexto

Estamos a travessar os primeiros estágios de uma transformação económica e social impulsionada pela tecnologia de *Distribuited Ledger Tecnology* (DLT), nomeadamente no *Blockchain*. Uma multitude de indústrias e consumidores finais já estão a participar activamente em plataformas *Blockchain* e a explorar o uso desta tecnologia. Vários casos de uso da tecnologia *Blockchain* estão a ser explorados nas áreas do imobiliário, registos clínicos, *Supply Chain*, registos legais, reportes financeiros, gestão de activos financeiros entre outros.

Este trabalho pretende explorar a utilização da tecnologia de *Blockchain* aplicada a casos de usos da ajuda humanitária, nomeadamente como é que alguns problemas de transparência e eficiência podem ser ultrapassados pela utilização desta tecnologia, na distribuição adequada de fundos ou bens às pessoas carenciadas e instituições.

Alguns exemplos de potencial utilização do *Blockchain* no sector humanitário são, segundo Vanessa Ko e Andrej Verity (KO & VERITY, 2016):

- **Identificação e documentação:** O *Blockchain* permite que uma entidade individual/particular ou organização prove a sua existência e identidade por via do DLT. Disponibilizando um sistema de gestão dos dados pessoais detido e gerido pelos próprios.
- **Recolha e partilha de informação e dados:** A segregação de informação em silos ou falta de confiança na mesma, pode ser endereçada por mecanismos de partilha de informação via *Blockchain*. Podendo tornar o acesso à informação, que pode ser partilhada aos participantes, de forma segura e garantindo a sua integridade.
- **Rastreamento de bens e transparência na atribuição:** O *Blockchain* pode ser utilizado como plataforma de dados para o rastreamento da proveniência, distribuição, atribuição e utilização de bens de ajuda humanitária.
- **Financiamento:** Transparência e eficiência na gestão de financiamentos, com a utilização do *Blockchain* para a coordenação, atribuição e distribuição de fundos entre os diferentes atores humanitários.
- ***Crowdfunding*:** O *Blockchain* pode servir de plataforma que assegura a rápida atribuição de fundos de forma transparente e com baixo custo por transação, ao mesmo tempo que assegura o cumprimento das regulamentações aplicáveis.

1.2 O Problema

As organizações envolvidas em ajuda humanitária, têm imperativos éticos e princípios humanitários nos quais se baseiam para orientar a sua acção, são estes que ajudam a construir a confiança e aceitação das organizações. Para construir e manter essa confiança as organizações também precisam de ser eficientes e transparentes nos processos, nas decisões e na comunicação, quando isto não acontece as organizações ficam mais permeáveis à fraude e à corrupção ou a uma percepção de que a sua acção possa ser injusta ou tendenciosa.

As Nações Unidas em 2012 estimavam que cerca de 30% da ajuda humanitária vai parar a pessoas e entidades corruptas (United Nations, 2012).

Em vários inquéritos e estudos, que tem como fonte as pessoas em situações de crise, a corrupção é destacada consistentemente como um dos principais impedimentos para receber ajuda (chsalliance, 2015). Tendo sido identificadas as seguintes principais consequências:

- Limita a quantidade ajuda que chega às pessoas que desesperadamente dela necessitam;
- Fator limitativo num financiamento eficiente e eficaz às organizações humanitárias nos países em desenvolvimento: No Reino Unido 53% dos inquiridos concorda que devem ser efectuados cortes na ajuda governamental a outros países (Bond, 2015);
- A percepção de corrupção mina o apoio dos potenciais doadores: No Reino Unido 59% dos inquiridos indicam que faz pouco sentido ajudar países pobres devido à corrupção (Bond, 2015).

Outros relatórios produzidos por associações de *stakeholders* na ajuda humanitária descrevem que as pessoas em contextos de crise humanitária, também percepcionam racionais pouco claros, injustos ou incoerentes na distribuição de ajuda (ALNAP, 2018).

Quando os fundos de ajuda são mal administrados e os incidentes e riscos relativos a corrupção não são devidamente endereçados, destinatários dos fundos acabam por sair lesados, a credibilidade dos doadores é colocada em causa assim com a eficácia dos esquemas de doação.

Anualmente tem-se verificado que as necessidades de ajuda humanitária estão a aumentar em todo o mundo, o financiamento humanitário internacional não está a crescer ao ritmo necessário para acompanhar as demandas crescentes. Assim a forma como a ajuda humanitária é disponibilizada (em muitos casos passando por numa cadeia extensa de doadores, organizações internacionais, organizações nacionais, parceiros, entidades governamentais, voluntários até chegar aos beneficiários finais), deve portanto tornar-se mais eficaz e eficiente pois só assim consegue enfrentar os novos desafios e atender às necessidades das populações carenciadas. Foram as conclusões do principais organizações de doadores que assinaram o *Grand Bargain*, em Maio de 2016 em seguimento da *World Humanitarian Summit* (Inter-Agency standing committee, 2016).

Estão assim identificados dois problemas relacionados entre si. O problema da transparência na ajuda humanitária, que facilita más práticas e abusos com consequências na eficiência e eficácia da ajuda humanitária, para além de minar a confiança em todo o sistema humanitário. E o problema da eficiência na ajuda humanitária, em que as lacunas na existência ou cumprimento de processos standardizados e automatizados acaba por gerar falta de transparência e pouca eficácia.

Com todas estas preocupações em mente, a Islândia, o Liechtenstein e a Noruega fizeram uma parceria com a *Transparency International* para avaliar e mitigar os potenciais riscos de corrupção no Espaço Económico Europeu (EEE) (Transparency International, 2015), deste trabalho conclui-se que há necessidade de intervenção nos seguintes pontos:

- **Transparência:** Assegurar total cumprimento da regulamentação e alto nível de transparência no processo de atribuição de fundos e ajuda. Informação acessível e

facilmente entendível são um factor preponderante no combate à corrupção, pois permite o escrutínio pelo cidadão comum.

- **Responsabilização:** Garantir que estão definidos os mecanismos de monitorização que garantem que cada beneficiário e interveniente no processo de decisão possam ser responsabilizados por práticas que tenham colocado em causa a adequada distribuição de fundos.
- **Segurança e Privacidade:** Assegurar que os diferentes atores desempenhem suas funções independentemente uns dos outros (essencial para evitar possíveis conflitos de interesse). Salvaguardando sempre questões de segurança e privacidade no acesso à informação.
- **Supervisão:** Monitorização exaustiva das áreas de maior risco, definição projetos e programas para implementação de processos mais eficientes. Embora exista uma motivação para introduzir a estandardização de processos do ponto de vista da eficiência, existe o risco de favorecimento ou desfavorecimento, que pode ser mitigado por meio de escrutínio mais minuciosas nos próprios processos e melhoria contínua decorrente das atividades de supervisão.

Tendo em conta os pontos anteriores e os problemas de investigação referidos colocam-se a seguinte questão, a ser potencialmente endereçada no decorrer deste projeto.

- Pode a tecnologia de *Blockchain* providenciar benefícios às organizações de ajuda humanitária para endereçar os problemas de transparência e eficiência?

1.3 Objectivos

Para além das questões e problemas inerentes ao sector humanitário apresentadas, e de um aumento nos conflitos mundiais e situações de desastres naturais, têm vindo a surgir novas tecnologias, parceiros e conceitos que permitem aos atores humanitários resolver problemas de forma mais rápida e eficaz. Tendo em conta esta necessidade do sector humanitário ter de passar responder com abordagens diferentes e mais inovadores surgiu o conceito de "*Humanitarian Innovation*". (BETTS & BLOOM, 2014)

Também nesta área existem princípios que regem qualquer acção que visa trazer inovação ao sector humanitário. Várias entidades reuniram-se em 2015 na conferência *World Humanitarian Summit* organizada pelo *Humanitarian Innovation Project* onde foram estabelecidos os 7 princípios base pelos quais se devem reger projetos de inovação no sector humanitário (Humanitarian Innovation Project, 2015):

1. **Propósito Humanitário:** A Inovação tem um propósito humanitário.
2. **Relacionamento Primário:** A principal preocupação nas relações da inovação humanitária deve ser com a relação doador/Beneficiário.
3. **Autonomia:** Toda a inovação humanitária deve ser conduzida com o objetivo de promover os direitos, a dignidade as capacidades da população beneficiária.
4. **Beneficência:** A inovação deve basear-se no princípio de "não prejudicar".
5. **Experimentação**: A realização de experiências, pilotos e ou ensaios devem ser realizados em conformidade com padrões éticos internacionalmente reconhecidos.
6. **Justiça**: Equidade e justiça devem sustentar a distribuição dos benefícios, custos e riscos resultantes da inovação.

7. **Responsabilização**: A inovação constitui uma obrigação em garantir a responsabilização e prestar contas às populações beneficiárias, incluindo a disponibilização de processos de reclamações e contrapartidas relacionadas com consequências imprevistas e maleficência.

Assim pretende-se com este trabalho avaliar e demonstrar como a tecnologia *Blockchain* pode endereçar os problemas de transparência e eficiência identificados para as organizações de ajuda humanitária, trazendo benefícios claros.

Adicionalmente sendo uma tecnologia inovadora, garantir que a aplicação desta tecnologia nas organizações de ajuda humanitária promove os princípios definidos em cima para a inovação no sector.

1.4 Metodologia de Investigação

A realização deste trabalho foi desenvolvida de acordo com o método de *Design Science Research* (DSR). Tendo sido seguido o processo de 6 fases proposto por Ken Peffers e Tuure Tuunanen (Peffers, et al., 2006) que se passam a descrever:

1. **Identificação do problema** - Definição do problema de pesquisa específico que será então a base para o desenvolvimento de uma solução efectiva e eficaz.
2. **Objetivos da solução** - Inferir os objetivos da solução para o problema definido. Os objetivos podem ser quantitativos, por exemplo, termos em que a solução será melhor do que os atuais, ou qualitativos, por exemplo, que benefícios são esperados para a resolução do problema.
3. **Desenho e desenvolvimento** - Nesta atividade são identificadas os requisitos e as funcionalidades pretendidas, o desenho da arquitetura e a criação do artefato real.
4. **Demonstração** - Demonstrar a eficácia do artefato para resolver o problema. Esta atividade envolve testes, simulação de casos de uso e provas.
5. **Avaliação** - Observar e medir quão bem o artefato suporta uma solução para o problema. Essa atividade envolve comparar os objetivos da solução com os resultados reais observados demonstração da utilização do artefato.
6. **Comunicação** - Comunicação do problema e a sua importância, o artefato e sua utilidade e eficácia para pesquisadores e outros públicos relevantes, quando apropriado.

FIGURA 1 - PROCESSO DO DESIGN SCIENCE RESEACH ADAPTADO DE (PEFFERS, ET AL., 2006)

No decurso do projeto houve uma revisão de literatura de base tecnológica e também relativa ao problema em causa, com estudo de alguns casos onde se realizou a aplicação da tecnologia *Blockchain* em projecto-piloto.

1.5 Estrutura do Documento

Este documento é composto, essencialmente, por 5 capítulos.

O presente capítulo, o primeiro, apresenta o contexto e enquadramento geral do projeto, com a identificação do problema, os objetivos e método de investigação utilizado.

O segundo capítulo apresenta um resumo da revisão de literatura efectuada, apresentando uma visão sobre o estado actual do sector e organizações de ajuda humanitária, da tecnologia *Blockchain*, e da sua aplicabilidade em contexto de ajuda humanitária.

O terceiro capítulo detalha a produção do artefato, focando no processo técnico e nos aspectos de desenho arquitetura e implementação.

O quarto capítulo apresenta a análise de resultados dos testes na utilização da solução implementada, sobre casos de uso concretos e relevantes para responder às questões de investigação.

O quinto capítulo apresenta as conclusões do trabalho tendo em conta a revisão da literatura, o processo de implementação e os resultados anteriormente apresentados

2 Revisão da Literatura

Esta secção visa dar um enquadramento teórico da tecnologia e conceitos base. Apresenta-se o estado actual e contexto do sector da ajuda humanitária e das organizações de ajuda humanitária, explorando também alguns casos de uso e pilotos realizados com a aplicação da tecnologia *Blockchain*.

Estabelece-se também uma base comum de suporte tecnológico d o *Blockchain*, descrevendo e detalhando a plataforma na sua globalidade, as diferentes componentes e cada um dos conceitos subjacentes.

2.1 Ajuda Humanitária e Organizações Não Governamentais (NGO)

A ajuda humanitária tem como objetivo salvar vidas, aliviar o sofrimento e manter a dignidade humana durante e após crises provocadas pelo homem e/ou desastres naturais, bem como prevenir e reforçar a preparação para quando situações dessa natureza possam ocorrem. A assistência humanitária deve ser sempre governada pelos princípios humanitários fundamentais: humanidade; imparcialidade; neutralidade e independência.

Teve lugar em São Francisco, no dia 25 de Abril de 1945, a Conferência das Nações Unidas sobre as organizações internacionais, tendo como resultado a Carta das Nações Unidas para a introdução uma nova ordem mundial baseada nos ideais de paz e segurança internacional (no sentido de prevenir que não voltassem a acorrer alguns eventos, de memória recente, em consequência da Segunda Guerra Mundial). As NGO's foram designadas, dessa forma, pela primeira vez no artigo 71 da Carta das Nações Unidas, contudo foram ainda necessárias algumas emendas, a essa mesma carta, para que existisse um reconhecimento mais consensual sobre o papel das organizações de ajuda humanitária e NGO's.

As NGO reconhecidas são então inscritas no cartório da Organização das Nações Unidas (ONU), podendo assim ter acesso a documentos, reuniões do assembleia do Conselho Económico e Social das Nações Unidas (ECOSOC) e dos vários órgãos subsidiários assim como dos seus programas. Este estatuto consultivo permite às NGOs, enquanto atores de utilidade internacional, desenvolver um papel político e ter uma interacção directa no mesmo contexto institucional dos outros atores governamentais internacionais e inter-governamentais.

O *World Bank* define as NGOs da seguinte forma: "organizações privadas que desenvolvem atividades para aliviar o sofrimento, promover os interesses dos desfavorecidos, proteger o ambiente, providenciar serviços sociais básicos ou trabalhar no desenvolvimento das comunidades. O termo NGO pode ainda ser aplicado a qualquer organização não lucrativa que seja independente do governo. As NGOs são tipicamente organizações baseadas em valores altruístas, dependendo total o parcialmente da caridade, donativos e serviço de voluntariado. Embora o sector esteja, nas duas últimas décadas, a tornar-se gradualmente mais profissionalizado, os princípios altruístas e voluntariado continuam a ser chave na sua definição" (World Bank, 1995).

2.1.1 O Sistema Humanitário

Num desastre humanitário, há sempre necessidade de coordenação, com o objetivo de maximizar a eficiência e a eficácia do esforço humanitário para atender às necessidades das comunidades afetadas. Dai a importância de definição de um sistema em que todos os intervenientes tenha bem definido o modelo de governo.

Os atores humanitários pertencem a uma ampla rede de organizações, agências e instituições que colaboram para permitir que a assistência humanitária internacional seja canalizada para os lugares e pessoas que dela necessitam. Incluem agências da ONU, a Cruz Vermelha e o Crescente Vermelho, organizações não-governamentais (NGOs), assim como outras agências e em coligação humanitária como: instituições militares e outras instituições do governo local.

A coordenação é fundamental para a criar um ambiente favorável em que organizações independentes possam colaborar na melhoria da qualidade e na ampliação do âmbito e o impacto de suas intervenções. Nas emergências humanitárias em que os governos nacionais não estão dispostos e/ou são incapazes de oferecer ajuda humanitária às populações afetadas, é necessário apoio internacional. A Organização das Nações Unidas (ONU) está mandatada para apoiar a coordenação de agências humanitárias internacionais, a fim de prestar a assistência de forma coesa e eficaz no salvamento de vidas e alívio do sofrimento. Em algumas situações, o sistema das Nações Unidas pode ter que assumir a liderança na supervisão da resposta humanitária; em outras, e sempre que os governos nacionais estão dispostos a cumprir sua responsabilidade de apoiar as populações afetadas pela crise, o grau de atividade internacional pode ser limitado ao apoio de doadores bilaterais. Assim todos os atores humanitários envolvidos têm a responsabilidade de coordenar com outras organizações na partilha de informações ou recursos que contribuem para as prioridades de uma resposta humanitária, particularmente pela importância e criticidade das situações de desastre.

Uma abordagem coordenada de todo o sistema da ONU para a ajuda humanitária é essencial para prestar assistência rápida e eficiente aos necessitados, assim a ONU conta com as seguintes entidades responsáveis (United Nations, s.d.):

- **Departamento de Coordenação de Assuntos Humanitários (OCHA)** - é responsável por coordenar as respostas às emergências. Recorre ao Comité Permanente Interagências, cujos membros incluem as entidades do sistema das Nações Unidas responsáveis por fornecer ajuda em situações de emergência.
- **Fundo Central de Resposta a Emergências (CERF)** - administrado pelo OCHA, é uma das formas mais rápidas e eficazes de apoiar uma resposta humanitária rápida a pessoas afetadas por desastres naturais e conflitos armados. O CERF recebe contribuições voluntárias durante todo o ano para fornecer financiamento imediato nas ações humanitárias que salvam vidas em qualquer parte do mundo.

A ONU tem ainda entidades responsáveis pelo desenvolvimento de atividades de ajuda humanitária e prestar assistência no terreno: Programa das Nações Unidas para o Desenvolvimento (UNDP); Agência das Nações Unidas para Refugiados (UNHCR); Fundo das Nações Unidas para a Infância (UNICEF) e Programa Alimentar Mundial (WFP). O UNDP é

também a agência responsável pelas atividades operacionais de mitigação, prevenção e preparação para desastres naturais. Quando ocorre uma situação de emergência, os Coordenadores Residentes do UNDP coordenam os esforços de assistência e reabilitação ao nível nacional.

2.2 Blockchain e o Hyper Leger Fabric

Um *Blockchain* é uma tecnologia de registo imutável e distribuído de transações (*Distributed Ledger technologies ou DLT*), baseado em protocolos criptográficos e geridas dentro de uma rede distribuída de nós em ponto a ponto (*peer-to-peer* - P2P).

No *Blockchain*, todas as transações são registadas, incluindo informações sobre a data, hora, participantes e quantidade de cada transação. Cada nó da rede possui uma cópia completa do *Blockchain*. Com base em modelos matemáticos complexos, as transações são verificadas pelos outros participantes no *Blockchain* por um protocolo de consenso, sendo as mesmas agrupadas em blocos que por via de encriptação (*hash)*, que liga cada bloco ao seu bloco predecessor (dai a origem do nome *Blockchain* – cadeia de blocos). Estes modelos matemáticos também garantem que esses nós concordem automática e continuamente sobre o estado actual do *Ledger* e todas as transações nelas contidas.

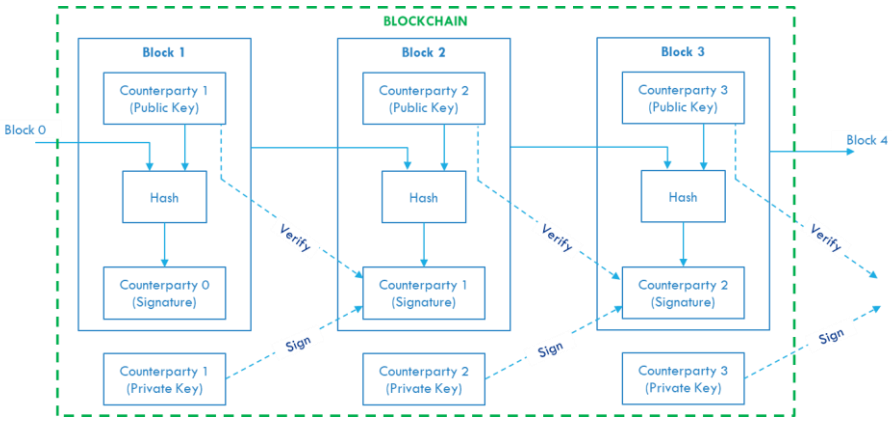

FIGURA 2- BLOCOS NO BLOCKCHAIN

Se alguém tentar corromper uma transação, os nós não chegarão a um consenso e, assim, recusarão a incorporação da transação no *Blockchain*. Cada transação é acessível e milhares de nós concordam unanimemente que uma transação ocorreu na data X no horário Y. É como se houvesse um notário presente em cada transação. Desta forma, todos têm acesso a uma única fonte de verdade compartilhada, o que torna o *Blockchain* inviolável.

A figura em baixo apresenta de forma simplificada o processo descrito em cima para o encadeamento dos blocos no *Blockchain*.

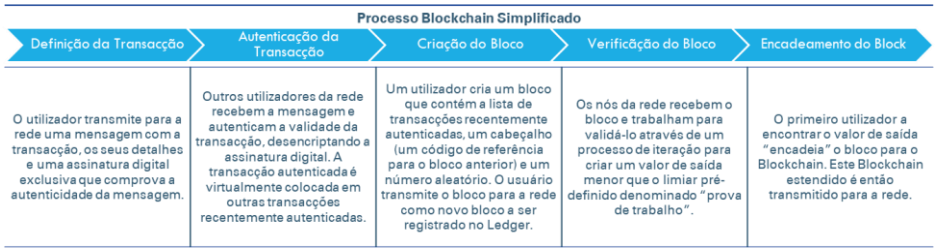

FIGURA 3 - O PROCESSO BLOCKCHAIN

O *Bitcoin* é uma criptomoeda e tendo sido a primeira e é a mais reconhecida aplicação do *Blockchain*. Outras surgiram entretanto tendo como base o *Ethereum*, um *Blockchain* alternativo, que introduziu a funcionalidade dos contratos inteligentes (*Smart Contracts*) para criar uma plataforma para aplicações distribuídas.

As criptomoedas tem um controlo descentralizado e desregulado, por oposição às moedas geridas pelos bancos centrais dos países, esta classe de *Blockchain* são definidos como abertos ou permissivos (*Permissionless*), pois estão abertos a qualquer pessoa e os participantes interagem anonimamente.

No entanto, a necessidade do uso corporativo da tecnologia de *Blockchain* exige características de desempenho que as plataformas de *Blockchain* de classe aberta não podem fornecer. Além disso, maioritariamente no uso corporativo e em redes de negócio reguladas, conhecer a identidade dos participantes é um requisito, tal como nos casos que são abrangidos por regulamentação de *Know-Your-Customer* (KYC), Anti-branqueamento de capitais e financiamento do terrorismo (AML).

Surgiram assim as classes de *Blockchain* não-permissivos (*Permissioned*) ou privados, focados nos casos de uso corporativo ou comercial, onde só participantes autorizados podem aceder e emitir transações num circuito fechado.

Um exemplo de uma plataforma de tecnologia de *Distributed Ledger* (DLT) *Open-source* de classe privada é o *HyperLedger Fabric*, projectado para utilização em contextos corporativos e fornecendo alguns fatores de diferenciação face a outras plataformas populares de *Blockchain*.

O *HyperLedger* foi criado sob a égide da Linux Foundation, reconhecida por fomentar projetos de *open-source* num ecossistema de comunidades. O *HyperLedger* é governado por um comité de direção técnica por um conjunto diversificado de organizações e o desenvolvimento do projeto *HyperLedger Fabric* conta com comunidade de desenvolvimento de quase 200 programadores e de mais de 35 organizações.

2.2.1 Arquitectura *HyperLedger*

A Figura em baixo representa a arquitetura de referência do *HyperLedger*, dividida nas suas 3 componentes principais: *Membership*, *Blockchain* e *Chaincode*. Estas três componentes representam estruturas lógicas e não se relacionam directamente com componentes físicas.

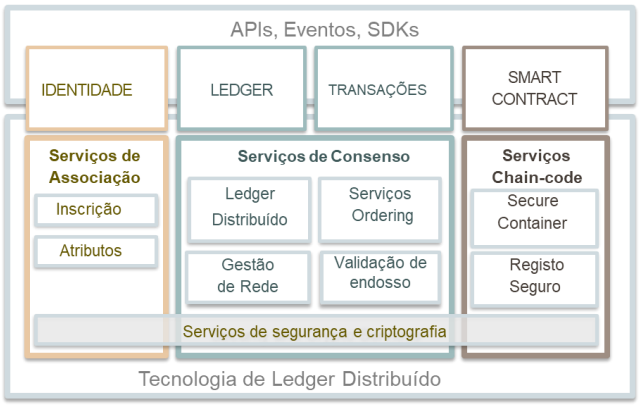

FIGURA 4- ARQUITECTURA DE REFERÊNCIA DO HYPERLEDGER

O Serviço de Associação (*Menbership service provider* - MSP) gere a identidade, privacidade e confidencialidade da rede. Os participantes inscrevem-se para obter identidades, o que permite a Autoridade de Certificação (*Certification Authortity* - CA) emita chaves de segurança para as transações. Este serviço também permite a auditores a visualização de transações pertencentes a um determinado participante, desde que ao auditor lhe tenha sido atribuído as autorizações de acesso pêlos participantes.

Os serviços de consenso englobam mais do que simplesmente concordar com a ordem das transações e sua distribuição pelos nós para validação, sendo o factor de diferenciação do *HyperLedger Fabric*, pelo papel fundamental que desempenha este serviço em todo o fluxo de transações, desde a proposta, o endosso, o pedido, a validação e o comprometimento. Em suma, o consenso é definido como a verificação do ciclo completo da validação da completude de um conjunto de transações, a que corresponde um único bloco.

Os serviços *Chaincode* é uma peça de *software* que define um ou vários *Assets* e as instruções para as transação poderem modificar *Assets*. Em resumo, é componente de lógica e regras de negócio. O *Chaincode* impõe as regras para efectuar leitura, alterações à base de dados de *World-State*, por meio da execução de uma transação. A execução de *Chaincode* resulta num conjunto de pares chave-valor (escritas – *write set*) que são enviadas para a rede *Blockchain* e aplicadas aos *Ledgers* distribuídos em todos os nós (*peer*).

2.2.2 HyperLedger Composer

O *HyperLedger Composer* é uma plataforma de desenvolvimento aberta, no âmbito do projeto *HyperLedger*, que inclui um conjunto de ferramentas, para facilitar a implementação de soluções em *Blockchain*. O objetivo principal é facilitar o desenvolvimento e a integração das soluções *Blockchain* com os sistemas de negócios existentes. O *HyperLedger Composer* permite o desenvolvimento rápido e intuitivo de casos de uso, permitindo a modelação de redes de negócio e a integração de sistemas existentes com novas soluções com base em *Blockchain*.

O *HyperLedger Composer* incorpora a infra-estrutura e *runtime* de execução de *Blockchain* do *HyperLedger Fabric*, suportando os protocolos de consenso de *Blockchain* garantido a

validação das transações de acordo com as políticas e também pelos participantes da rede de negócios definidos.

Com base nestas definições, podem ser desenvolvidas aplicações que interagem e consomem dados destas redes de negócios, disponibilizando aos utilizadores finais interfaces e pontos de controlo simples e intuitivos.

O *HyperLedger Composer* permite a modelação intuitiva e fácil de redes de negócios, definindo e caracterizando os seus *Assets* e as *Transactions* relacionadas com os mesmos; *Assests* são bens, serviços ou bens tangíveis ou intangíveis. Também como parte do modelo são definidas as *Transactions* que podem interagir com os *Assets*. Nestas redes de negócios também são definidos e caracterizados os seus participantes que irão interagir entre eles, cada um pode ser associado a uma identidade única e participar em várias redes de negócios.

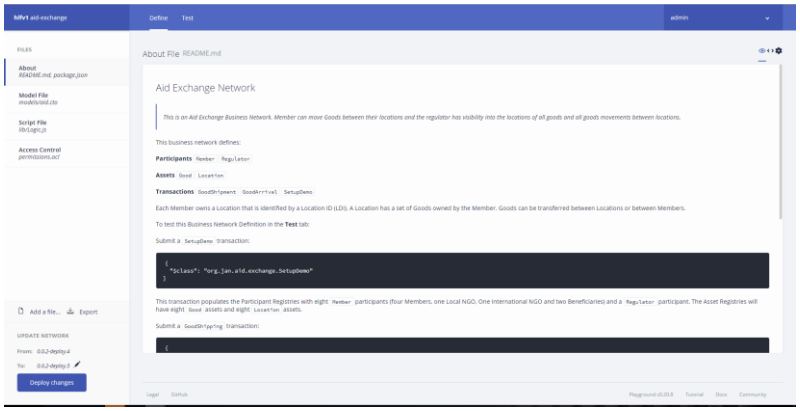

FIGURA 5 – HYPERLEDGER COMPOSER NA DEFINIÇÃO DA REDE DE NEGÓCIO

2.2.2.1 Conceitos

A figura seguinte representa uma rede *Blockchain* e as suas principais componentes e constituintes.

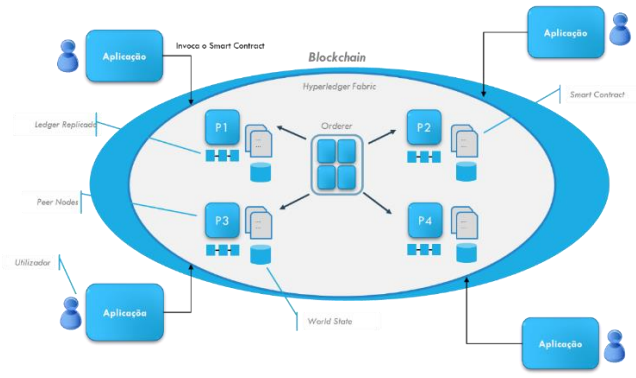

FIGURA 6 - VISÃO GERAL DE UMA REDE BLOCKCHAIN

Nesta rede P1,P2,P3 e P4 são os nós (*peer*) que formam a rede *Blockchain*, cada um contém uma versão do *Ledger* distribuído e do *World State*. Cada um deles disponibiliza uma API REST e SDK pelo qual aplicações clientes de negócio interagem com a rede, disponibilizando aos utilizadores uma interface própria e adequada às suas funções. Como parte central da rede

está o serviço de *ordering*, tipicamente existe pelo menos um nó (*peer*) designado para esta função.

2.2.2.1.1 Repositórios de Estado

O *Ledger* é um conceito primordial no *Blockchain*, ele guarda informação factual a respeito dos objectos de negócio; valor actual dos seus atributos, e toda a história de transacções, das quais resultou o valor actual. A estas duas componentes distintas ainda que relacionadas, dá-se o nome de *World State* e *Blockchain*.

- ***World State*** – Base de dados que mantêm uma cache dos valores actuais de estados dos registos. Desta forma é mais fácil o acesso directo ao valor actual do estado dos registos, não havendo necessidade de calculatória recorrentemente, percorrendo o *log* de transações na sua totalidade. Os estados do registo são, por defeito, expressos por pares chave-valor. O valor do *World State* muda frequentemente, de acordo com a criação de estados, actualizações e sua eliminação.
- ***Blockchain*** – É um log transaccional, num encadeamento de blocos, que regista todas as alterações que resultaram em actualização do *World State*. Tal como explicado anteriormente na secção 2.2 e detalhado na Figura 2, as transações são segregadas em blocos que são então anexados ao *Blockchain*, permitindo entender a história e linha temporal das alterações que deram origem ao *World State* actual. A estrutura de dados do *Blockchain* é muito diferente da do *World State* pois uma vez escrita não pode ser alterada, é imutável.

Assim todas as transações submetidas no âmbito de uma rede de negócios são persistidas no registo (*Ledger*) do *Blockchain*, o estado actual dos activos e os participantes são persistidos na base de dados de estado do *Blockchain* (também conhecida como *World State*).

O *Blockchain* mantem o *Ledger* e o *World State* distribuídos através de um conjunto de nós (*peers*) assegurando a actualização do *Ledger* e *World State* são consistentes entre todos os nós por via de algoritmos de consenso.

2.2.2.1.2 Assets

Os *Assets* (activos) correspondem aos bens a serem transaccionados na rede. A sua definição é o que permitem a transação e troca de bens na rede, os bens podem ser qualquer coisa com valor monetário, caracterizando-se em tangíveis (alimentos, carros antigos, imóveis e hardware) ou intangíveis (criptomoedas, contratos e propriedade intelectual).

Os *Assets* têm de ter um identificador único, contudo são também constituídos por um conjunto de propriedades ou atributos definidos em conformidade com os requisitos do negócio da solução a implementar. Os *Assets* podem relacionar-se com outros *Assets* ou com participantes.

2.2.2.1.3 Participantes

Os participantes (*Participants*) são membros de uma rede de negócio em *Blockchain*. Eles detêm *Assets* e submetem transações (*Transactions*). Os participantes também são modelados e definidas as suas configurações e características, tal como os Assets, devem ter um

identificador único e um conjunto de propriedades e atributos que os definem. Um Participante pode ser mapeado para uma ou várias identidades.

2.2.2.1.4 Identidade

Todos os atores numa rede *Blockchain*, podem ser nós (*peers*), organizadores, aplicações clientes, administradores e outros, sendo elementos activos dentro ou fora da rede com capacidade para consumir serviços, detêm uma identidade digital encapsulada num certificado digital.

Para uma identidade ser verificável, tem de provir de um *Membership Service Provider* (MSP). Mais especificamente, um MSP é um componente que define as regras que governam as identidades válidas para essa organização. A *Fabric* na implementação o MSP usa certificados X.509 como identidades, adotando um modelo tradicional de *Public Key Infrastructure* (PKI).

2.2.2.1.5 Membros

O MSP identifica quais *Root Certificate Authorities* (CA) e as *Intermediate Certificate Authorities* que são confiáveis para definir os membros de um domínio, por exemplo, numa organização, listando as identidades de seus membros ou identificando quais CAs que estão autorizadas a emitir identidades válidas para seus membros. A MSP também identifica os papéis específicos que um ator pode desempenhar dentro da organização (e.g. administrador ou como membros de um grupo de sub-organizações) e define privilégios de acesso de uma rede *Blockchain* e canal (e.g. administrador de canal, leitor ou escritor).

2.2.2.1.6 Peers

Os *Peers* (nós *peer*) são os elementos principais na composição de uma rede Blockchain, são neles que reside o *Ledger* distribuído e os *Smant Contracts*, ambos encapsulam os processos e a informação partilhada na rede. Esta particularidade de cada nó (*peer*) conter instâncias do Ledger e Chaincode é o que permite a redundância na rede, evitando assim um ponto único de falha. Os nós disponibilizam API's, que são o mecanismo para a interacção das aplicações com a rede (sempre através de um nó).

2.2.2.1.7 Smart Contract

Um *Smart Contract* é uma colecção de regras, em código executável, partilhada e validada colectivamente entre diferentes participantes na rede. As aplicações invocam um *Smart Contract* para gerar transações, que serão posteriormente registradas no Ledger. O *Smart Contract* implementa regras para qualquer objecto de negócio, estas são aplicadas automaticamente com a sua execução.

2.2.2.1.8 ChainCode

O *ChainCode* (Golang, Java ou javascript) é instalado e instanciado nos nós *peers* por um membro autorizado. O *ChainCode* contem a lógica de negócio e regras específicas para as transações definidas nos *Smart Contracts*.

As aplicações clientes que fazem a interface com os nós *peer*, invocam o *ChainCode* de acordo com as funcionalidades que pretendem oferecer ao utilizador final. É o *ChainCode* que propaga

uma transação na rede, contudo só após a validação, será agregada ao *Ledger* distribuído e desencadeia a alteração do *World State*.

2.2.2.1.9 Transacções

As transações são o mecanismo pelo qual os participantes podem interagir com os *Assets*. Transacções podem ser acções do tipo: participante licitar sobre uma peça ou lote num leilão; leiloeiro encerrar o leilão de uma peça ou lote, transferindo assim a propriedade automaticamente para a licitação mais alta.

As transações para serem válidas e propagadas no *Blockchain*, requerem um endosso (*Endorsement*) seguindo as políticas de endosso a aplicar ao *Smart Contract.* As políticas de endosso são muito importantes, uma vez que definem quem é que dentro da rede de negócio deve assinar a transação gerada pelo *Smart Contract* para que as mesmas possam ser declaradas válidas.

2.2.2.1.10 Aplicações

Uma aplicação cliente pode interagir com uma rede *Blockchain* enviando transações para um *Ledger* ou consultando o conteúdo do *World State*, sempre na interacção directa com um nó (peer) e com mecanismos de segurança baseados em certificados digitais.

Estas aplicações de negócio disponibilizam uma interface de utilizador e encapsulam lógica específica para um propósito de interacção com a rede *Blockchain*. Tipicamente existe uma destas aplicações por participante na rede ou categoria de participante, uma vez que cada um desempenha funções distintas na rede *Blockchain*.

2.2.3 HyperLedger Composer REST Server

O HyperLedger Composer pode gerar uma API REST à medida, tendo como base uma rede negócio previamente definida e *deployed*. Esta API REST é fundamental sempre que se pretende criar uma aplicação cliente para interagir com o Blockchain, a API REST fornece uma camada abstração muito útil e agnóstica à linguagem de programação escolhida para desenvolvimento da aplicação.

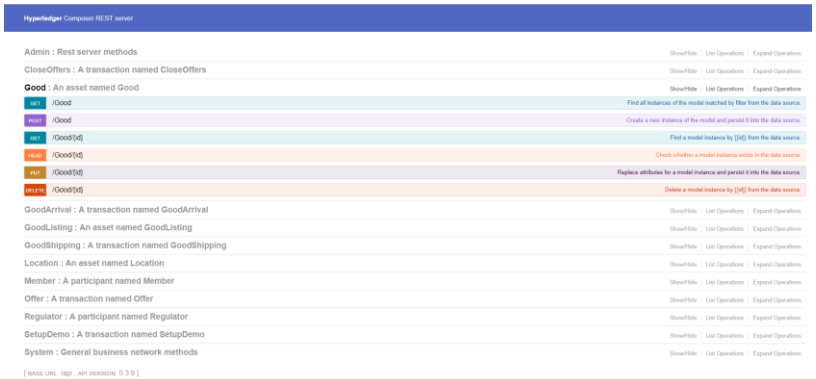

FIGURA 7 - API REST GERADA EM VISTA SWAGGER

O REST Server usa a *framework Loopback*, para gerar dinamicamente as APIs REST que permitem às aplicações clientes interagirem e cooperarem através do pacote da biblioteca *loopback-connector-composer* na disponibilização de serviços.

O *Loopback* utiliza um modelo de objectos de *Javacript* para representar as fontes de dados de *back-end*, nomeadamente a rede de negócios em *Blockchain*. O REST Server tira partido desta funcionalidade na consulta da rede de negócios, de onde cria um modelo customizado que é posteriormente utilizado na geração da API.

A correcta configuração do REST *Server* é fundamental para que consiga determinar a rede de negócios a que se deve ligar. O modelo auto-gerado contendo outros objectos específicos do sistema são também configurados antes de lançar a aplicação *Loopback*.

2.2.4 HyperLedger Composer Angular Web Application

Para interagir com uma rede de negócios já implementada, as aplicações Web tem de fazer chamadas à API REST, disponibilizada pelo *HyperLedger Composer REST Server*. O *HyperLedger Composer* já disponibiliza um modelo, que utilizando a framework Yeoman (https://yeoman.io/), pemite a geração automática de aplicações Web em Angular.

O gerador de aplicações Yeoman Angular é apropriado apenas, para gerar esqueletos de aplicações Web, com base nas definições do modelo de redes de negócios simples e básicas. Neste processo utiliza a definição da API REST, para a criação do seu modelo interno, identificando à partida os Assets e as transações.

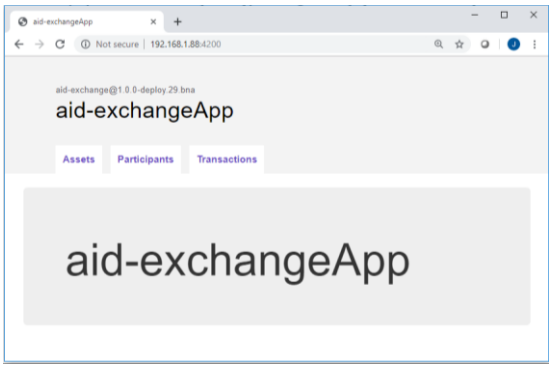

FIGURA 8 - EXEMPLO DA APLICAÇÃO ANGULAR GERADA PELO HYPERLEDER COMPOSER

2.2.5 HyperLedger Explorer

O *HyperLedger Explorer* foi inicialmente um contributo da IBM, Intel e DTCC, passando para a alçada da *The Linux Foundation*, no âmbito dos projetos *HyperLedger*. O *HyperLedger Explorer* é um módulo *Blockchain* que permite visualizar e consultar blocos, transações e os respectivos dados associados. Também permite consultar outras informações da rede de negócios (tais como o nome, status, lista de nós), os códigos da cadeia de blocos *Blockchain* e a famílias de transações, bem como quaisquer outras informações relevantes armazenadas no *Ledger*.

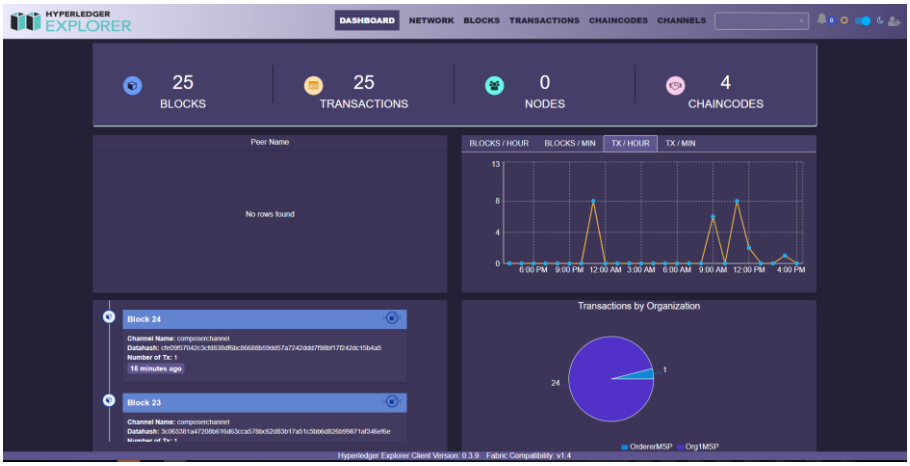

FIGURA 9 - EXEMPLO ILUSTRATIVO DO HYPERLEDGER EXPLORER

2.3 Redes de negócio

Em termos comerciais, uma rede de negócios pode ser definida como uma associações entre várias empresas e companhias, que trabalham em conjunto com o objetivo de criarem riqueza. Essa riqueza é gerada pelo fluxo transações de bens e serviços, através da rede, baseadas em contractos estabelecidos entre as partes (Wikipedia).

No *Blockchain* e no caso particular da ajuda humanitária, o conceito de rede de negócio é precisamente o mesmo, com excepção que a riqueza não necessita ser obrigatoriamente monetária, ou pelo menos com uma correlação monetária directa.

2.3.1 Rede de negócio no ecossistema humanitário

O ecossistema humanitário tem um alargado leque de atores, oferecendo assim um potencial elevado para diferentes ligações entre eles. A figura em baixo apresenta os diferentes atores e as potenciais ligações, estas ligações podem ser comuns (linhas contínuas) pouco comuns (linhas tracejado). O relatório da OCHA (Humanitarian Innovation Project, 2015) indica que este é um ecossistema com bastante potencial para a inovação. Ainda para mais no contexto desta tecnologia uma vez que é uma rede com muitas ligações, não existindo há partida modelos de governo instituídos que garantam automatismos.

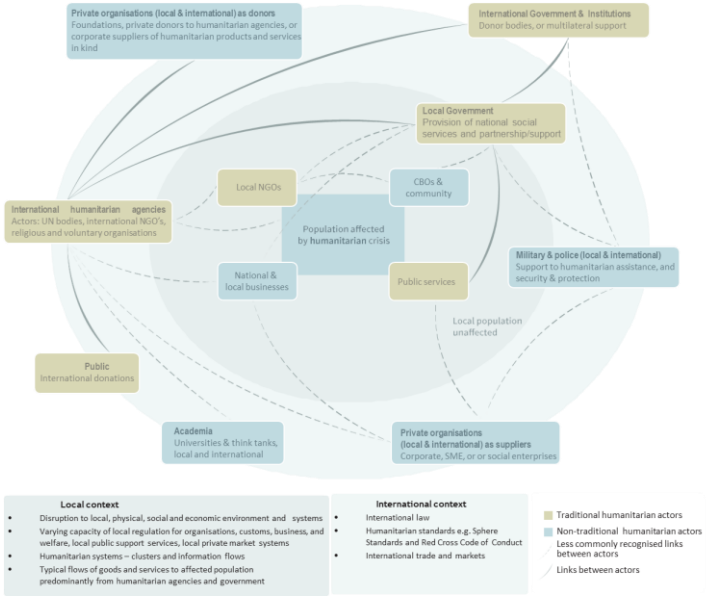

FIGURA 10 - ECOSSISTEMA HUMANITÁRIO PELA OCHA

2.4 *Blockchain* e a sua aplicabilidade no Sector Humanitário

A aplicabilidade de *Digital Ledger Technology* (DLT) ou de *Blockchain* no sector humanitário, não irá certamente ficar confinada à sua ideia e aplicabilidade inicial nas criptomoedas (*Blockchain premisionless*). Existe uma opinião consensual na indústria, de que as vantagens tecnológicas do *Blockchain*, podem vir a automatizar processos em vários sectores e industrias, tais como: garantir e verificar a identidade; seguir a proveniência e rastreamento de transações e bens; ou a imposição de "S*mart Contracts*" (*Blockchain permissioned*).

Alguns eventos e atividades criminosas ou de financiamento a terrorismo, envolvendo o *Blockchain premisionless*, nomeadamente nas criptomoedas, despoletaram várias críticas à plataforma e criaram alguma desconfiança na sua aplicação. Contudo no *Blockchain permissioned*, houve um esforço dos principais consórcios de desenvolvimento destas plataformas, na criação de funcionalidades e mecanismos que pudessem resolver e mitigar riscos da plataforma predecessora, permitindo assim às organizações ganharem cada vez mais confiança.

De acordo com os princípios éticos da inovação no sector humanitário, já referidos na secção 1.1, existem princípios que devem ser tidos em conta no desenho da arquitetura de soluções, fazendo assim parte das fundações e também dos pilares dos projetos nesta área, desde o início. A título de exemplo a organização *Blockchange*, definiu os princípios GENESIS, com base num estudo recente (Verhulst, 2018), constam do seguinte:

- **Legitimidade de governo:** é essencial a garantia de um processo transparente, responsável e participativo na criação de aplicações e plataformas legítimas e confiáveis. É essencial que os responsáveis pelos projetos desenvolvam e coloquem em prática, desde o início, políticas e princípios de protecção de dados. As decisões devem ser transparentes, seguindo processos claros e pré-definidos.

- **Ética:** A ética nos projetos deve estar sempre a ser considerada para garantir confiança e legitimidade. Questões éticas são particularmente importantes em aplicações baseadas em identidade, pois podem estar relacionadas com questões de acesso e direitos individuais.
- **Orientação a soluções para problemas reais e não para a tecnologia:** Como muitas novas tecnologias, *Blockchain* é muitas vezes tratado como o martelo em busca de um prego (tal como no proverbio). Embora a experimentação tenha um objetivo digno, uma compreensão clara e assertiva do problema é essencial para criar soluções duradouras. Os decisores devem permanecer abertos a soluções não-*Blockchain* e envidar todos os esforços para articular a proposta de valor real das abordagens propostas.
- **Pegada Ecológica:** A maioria dos *Blockchains* tem uma pegada ecológica enorme e crescente devido aos altos níveis de energia necessários para processar e validar atividades e transações, custos associados com tecnologia e profissionais. Isso exige soluções que contribuem para o desenvolvimento sustentado.
- **Alinhamento com as iniciativas existentes:** Há a necessidade de resistir à tentação de reinventar a roda. Já existe uma multiplicidade de plataformas, e muitas vezes há vantagens em juntar recursos e aprender com experiências anteriores.
- **Interoperabilidade e padrões abertos:** Muitas das iniciativas atuais estão a ser conduzidas pelo setor privado. Isso pode ter vantagens em termos de financiamento e inovação. Mas o governo e a sociedade civil devem evitar o *vendor-locking* a longo prazo e o patenteamento de protocolos essenciais. Assegurar a interoperabilidade dos diferentes sistemas e o desenvolvimento de normas técnicas abertas será fundamental para garantir a usabilidade e a acessibilidade a longo prazo.
- **Assegurar a precisão do primeiro bloco:** Embora os atributos de imutabilidade e integridade do *Blockchain* garantam a exatidão das informações na cadeia, o primeiro bloco da cadeia pode ser um importante ponto único de falha. Assim o *Blockchain* é tão bom quanto o bloco da génese. Os arquitectos de soluções devem garantir a exatidão desse primeiro bloco, buscando, por exemplo, processos e intermediários para validar e cruzar informação e estabelecer mecanismos de habilitação e controle de qualidade.

Recentemente, tendo em conta os princípios indicados, têm vindo a ser desenvolvidos alguns projetos e pilotos de utilização da tecnologia *Blockchain*, ainda que em ambientes muito controlados e circunscritos.

2.4.1 Exemplos de Casos de Uso Humanitário

Nesta secção serão detalhados alguns casos de uso, em situações reais no terreno, exemplificativos da aplicabilidade do *Blockchain* no sector humanitário. Para cada é apresentada uma visão geral do problema, a solução e abordagem aplicada e os resultados obtidos. Os casos de usos estão agrupados por tipologia, tal como identificadas na secção 1.1.

2.4.1.1 Financiamento – Programas de transferência em numerário

Este é uma tipologia específica para fazer chegar ajuda monetária às populações carenciadas, a ajuda monetária pode ser por via de numerário ou *voucher*. Como abordagem foram aplicadas duas, *closed loop*, em que a disponibilização de voucher ou numerário é efectuado apenas numa rede fechada de parceiros ou agentes comerciais (lojas). No open-loop os montantes são

disponibilizados num subsistema de pagamentos, por exemplo mobile Money, não estando fechado a rede de parceiros.

Giulio Coppi e Larissa Fast são autores de um relatório da *Humanitarian Policy Group* (HPG), tendo entrevistado mais de 30 elementos de NGO's e outras organizações e institutos relacionados, identificaram alguns casos de uso, maioritariamente em programas de financiamento, aplicados com sucesso, da aplicabilidade do *Blockchain* no sector humanitário (Coppi & Fast, 2019).

2.4.1.1.1 World Food Programme

O *World Food Programme* (WFP) está cada vez mais dependente de transferências em numerário, para promover a segurança alimentar e o combate à fome. O projeto *Building Blocks*, desta instituição, utiliza a tecnologia *Blockchain* (*Ethereum*) para tornar mais eficientes as suas transferências em numerárias baseadas em vouchers com o objetivo de melhorar a colaboração dentro do sistema humanitário.

A província de Sindh no Paquistão foi a base para a prova de conceito deste projeto, para testar as capacidades de autenticação e registo das transações dos beneficiários. Seguindo-se um piloto de maior envergadura na Jordânia, em dois campos de refugiados. Os registos em Setembro de 2018, já contabilizavam mais de 100.000 refugiados Sírios na Jordânia, que obtiveram ajuda através da WFP através da solução baseada em *Blockchain* (Correspondendo a mais de 1,1 milhão de transações em apenas 16 meses de projeto).

A WFP, graças a esta solução baseada na tecnologia *Blockchain*, passou a ter o registo de todas as transações em cada um dos seus retalhistas associados, consequentemente facilitando todo o processo de reconciliação e redução de custos com terceiros (deixaram de ter necessidade de intermediários financeiros). Adicionalmente a solução *Building Blocks* passou a integrar com a plataforma de autenticação biométrica da UNHCR já existente. Assim as transações são efectuadas e aprovadas com o *Scan* da iris nos dispositivos POS (*point-of-sale*), a solução não tem acesso à informação de identificação pessoal (*Personally-identifiable information* - PII) dos beneficiários, apenas valida um *hash* anonimizado.

Para o futuro a WFP pretende estabelecer novas parcerias que permitam às mulheres sírias, que participam no programa das Nações Unidas (UN) "*Women's Cash for Work*", levantarem dinheiro no supermercado do campo de refugiados ou efectuar compras. Como as transações passam a ser validadas no *Blockchain*, quer pelo nó da WFP quer pelo nó da UN, a segurança aumenta. É também intenção da WFP a exploração de outros casos de usos em *Blockchain*, nomeadamente nas áreas do rastreamento em *Supply Chain* e da gestão de Identidade Digital (World Food Programme, 2019).

2.4.1.1.2 Parceria Start Network e Disberse para projetos piloto

A rede Disberse é uma rede *peer-to-peer* de nós parceiros, tendo como base uma tecnologia de *Distribuited Ledger Technology* e recorrendo *Smart Contrats* para a automatização de transações.

A Disberse recebeu autorização da Financial Conduct Authority (FCA), como uma instituição de moeda electrónica, tornando-se uma das poucas *fintech* do Reino Unido a combinar tecnologia de DLT e gestão de dinheiro electrónico (Disberse, 2019).

A Disberse estabeleceu uma parceria com Start Network para a realização de alguns testes e provas de conceito. Para garantir a segurança, dos testes piloto os mesmos foram realizados utilizando os *Web Browsers* dos participantes, tendo a Disberse garantido a autenticação dos mesmos por via de mecanismos de autenticação de dois fatores.

Nota importante é que os montantes transferidos não correspondem em si a uma criptomoeda, mas sim a *e-money* emitido pela Disberse contra uma divisa FIAT (divisa sem valor monetário intrínseco, mas definida como moeda pelo governo) depositada pelo cliente (neste caso as organizações dos pilotos). Esta forma garante a estabilidade dos fundos uma vez que pode haver muita volatilidade nas criptomoedas.

Start Network estende-se por 42 NGO's e os seus 700 Parceiros, empregando mais de um quarto de milhão de pessoas em 200 países e territórios. O seu objetivo é fornecer uma ajuda eficaz, aproveitando o poder e o conhecimento da rede para tomar melhores e mais rápidas decisões na ajuda as pessoas afetadas por crises. Está focada em desenvolver formas mais eficazes de trabalharem juntos e novas abordagens que reduzirão a escala do sofrimento humano (Start Network, 17).

O papel da Start Network como um intermediário foi fundamental na angariação de parceiros e estabelecer a confiança necessária para os pilotos, Dorcas e Trocáire, respectivamente.

2.4.1.1.2.1 Dorcas

Dorcas é um dos membros da Start Network, e foi pioneira no uso da tecnologia Blockchain. Iniciou por transferir uma pequena quantia de financiamento de € 5.000, através da plataforma Disberse para um escritório internacional, sem a necessidade de um banco.

Um relatório da Start Network concluiu que os principais benefícios deste teste centravam-se na rastreabilidade dos fundos por meio da criação de um registro imutável de transações. Economia nos custos diretos – foi verificada uma poupança de 1,15% - o montante de financiamento envolvido foi baixo logo a poupança também, contudo em uma situação de crise com a transação de valores mais elevados, são também esperadas maiores poupanças nestes custos. O feedback dos participantes da Dorcas também observou a facilidade de uso da plataforma e sua aplicabilidade para transferências de ajuda humanitária (DORCAS, 2018).

2.4.1.1.2.2 Trócaire

Após o sucesso do piloto da Dorcas, a Trócaire ofereceu-se para participar num piloto maior, com o objetivo de testar uma cadeia com mais parceiros. Contudo nas operações comerciais regulares de transferência de fundos a Trócaire nunca tinha utilizado uma cadeia de parceiros tão longo. Na fase de concepção, a Trócaire Rwanda concordou em convidar a Caritas Ruanda como parceiro local para participar também neste piloto. O piloto teve também um objetivo adicional, que pretendia medir as implicações diretas da plataforma *Blockchain*, enviando duas

transferências financeiras paralelas, uma através do sistema bancário regular e outra usando a plataforma Disberse.

Todas as partes preferiram na execução do piloto a utilização de uma quantia relativamente pequena, e foi acordado que a Trócaire Ireland transferiria simultaneamente € 10.000, através da Trócaire Rwanda, para a Caritas Ruanda utilizando a plataforma Disberse, e € 10.000 através do seu canal bancário regular.

A transferência via Disberse não incorreu em encargos adicionais, embora a Disberse pretenda introduzir uma taxa de transação no futuro como parte seu modelo de negócios. A transferência paralela através do sistema bancário incorreu num custo adicional de € 35.

A Trócaire Ireland utilizou o canal bancário com o propósito único de testar a tecnologia, uma vez que não o utiliza como parte dos seus processos normais. O sistema bancário levou seis dias úteis, enquanto a transferência da Disberse levou cinco. Ambos os processos, no entanto, sofreram de atrasos causados por falta de comunicação e pelo fato dos testes terem ocorrido durante um fim-de-semana prolongado, devido a um feriado nacional do Ruwanda (Start Network, 2018).

2.4.1.1.3 Sikka

Sikka (significa "Moeda" em Nepalês) é uma plataforma digital de transferência de ativos digitais projetada para as populações marginalizadas e financeiramente marginalizada, fundada e implementada pelo World Vision International (WVI) Nepal Innovation Lab em Kathmandu.

Com a Sikka, os membros da comunidade podem receber transferências em dinheiro nos seus telefones, utilizando o serviços de mensagens de texto SMS, através de comerciantes locais ou de cooperativas financeiras locais na sua aldeia, permitindo assim que tanto os transferências bancárias (*Cash-based transfers* -CBTs) restritas e não restritas. O Sikka foi projetado de forma que pode fornecer CBTs não restritas através de cooperativas financeiras locais ou CBTs restritos e a distribuição de bens de ajuda humanitária através de uma rede de comerciantes locais. Desta forma, a plataforma de transferência de ativos da Sikka disponibiliza os benefícios de uma plataforma *Blockchain* para aqueles que não possuem conhecimento, nem tecnologia ou recursos exigidos na disponibilização de serviços semelhantes.

Os *tokens* da Sikka não são criptomoedas, mas é lhes atribuído um valor determinado pelas agências de ajuda e dos planos de pagamentos definidos para o ecossistema. Assim os *tokens* pode ser indexados a 1 Rúpia Nepalesa ou a 1 Kilograma de Arroz, Bastando apenas aos beneficiários saberem utilizar um telefone.

Os testes foram realizados pela World Vision International em Sindhupalchowk, para um programa de *Cash for Work* em que 105 habitantes de Balefi, Phulpingkot, foram contratados para a reabilitação de um canal de irrigação existente. Em colaboração com uma cooperativa local os detalhes dos beneficiários, incluindo números de telefone e os montantes a receber foram exportados e carregados na solução Sikka.10 dias após o início dos trabalhos, a Sikka foi

utilizada para distribuir o pagamento em dinheiro aos habitantes, que apenas necessitaram utilizar os seus telefones para receber os pagamentos.

Após ser verificada a identidade de cada um dos beneficiários, os mesmos recebem uma mensagem de SMS confirmando o número de Sikkas que irão receber. Com base numa breve troca de SMS, recebem da cooperativa local ou dos comerciantes o dinheiro ou bens que lhe são devidos. (World Vision, 2019)

2.4.1.1.4 IFRC – Cruz Vermelha Quénia

Em maio de 2018, a IFRC em colaboração com a Cruz Vermelha do Quénia (KRCS) implementou um projeto piloto para explorar como *Blockchain* poderia adicionar transparência e responsabilização no programa *open-loop* de transferência de dinheiro, na província de Isiolo no Quênia, auxiliando mais de dois mil domicílios afetados pela seca. A provincia de Isiolo faz parte das vinte e três terras áridas e semi-áridas do Quênia onde prevalecem condições de seca, com alta taxa de mortalidade na pecuária, o que conduz a insegurança na subsistência.

No projeto-piloto, os beneficiários já tinham sido registados num projeto anterior para a uma distribuição de dinheiro na província de Isiolo. Um conjunto de 2.000 famílias foram seleccionadas do grupo original e de acordo com critérios pré-estabelecidos. Informações pessoais relevantes, tais como, nome do beneficiário, número de telefone e número de identificação nacional foram então exportados para o novo projeto. Para permitir o registo de transações no *Blockchain* um conjunto de chaves públicas e privadas foram geradas para cada beneficiário.

Foi solicitada a implementação do sistema à Red Rose, um fornecedor de tecnologia para transferências de dinheiro no setor humanitário, e a integração do mesmo com a plataforma M-Pesa da empresa de telecomunicações Safaricom M-Pesa, para a construção de um *Blockchain* privado para registo das transações. Este *Blockchain* é composto por quatro nós que incluem IFRC, KRCS e Red Rose. Estas podem visualizar as transações por meio de uma interface de adequada. Assim que a Safaricom recebeu o pedido de pagamento e dinheiro desembolsado para as carteiras móveis dos destinatários, o transação é registrada na plataforma Red Rose e no *Blockchain*.

Alternativamente a IFRC e o KRCS estão a explorar ativamente o conceito de identidades digitais "*Self-Sovereign*", o que seria permitir que os indivíduos mantenham e controlem suas próprias informações pessoais, por oposição a ter a necessidade de envolver organizações humanitária ou fornecedores subcontratados para o fazer (ICHA, 2018).

2.4.1.2 Rastreamento de bens na cadeia de distribuição

Há vários registos de utilização do *Blockchain* como plataforma para o rastreamento da proveniência, distribuição, atribuição e utilização de bens em vários sectores da economia. No entanto no sector na ajuda humanitária apenas foi possível encontrar literatura sobre um piloto na área.

2.4.1.2.1 DFID

O Department of International Development (DFID) através do programa Frontier Technology Livestreaming (FTL), desenvolveu um projeto piloto denominado "*Blockchain* para o *Supply Chain* humanitário" para situações de catástrofe (Chrysochou, 2019).

O DFID pretende melhorar a forma como a sua equipa trabalha em todo o mundo usando novas tecnologias. Naturalmente, o *Blockchain* é uma dessas tecnologias, e o *Supply Chain* é uma áera de aplicabilidade da tecnologia elas seguintes três razões principais:

- **Transparência** – O *Supply Chain* pode beneficiar do uso das ferramentas certas para conseguir maior transparência de forma também mais segura. Mais transparência facilita também a colaboração entre organizações.
- **Eficiência** - Se todos os atores que trabalham em organizações do DFID ou em organizações similares (por exemplo, a USAID, o Programa Mundial de Alimentos da ONU, etc.) pudessem confiar mais na qualidade dos dados, poderiam concentrar-se em outros assuntos. Isso poderia contribuir para diminuir as perdas e, assim, melhorar a eficiência à medida que mais mercadorias forem entregues aos necessitados.
- **Colaboração** – Um sistema partilhado para a gestão de bens e remessas, garante a responsabilização, rastreabilidade e privilégios de acesso na leitura e na escrita. Permitindo assim criação de padrões e modelos a serem seguidos pelos financiadores (doadores) e aqueles que recebem financiamento (beneficiários).

O projeto consistiu em construir um sistema baseado em *Blockchain* para rastreamento de envio de *kits* de abrigos em plástico de um armazém remoto, utilizando vários operadores de logística para um país onde são necessários. Uma entidade local terá que assumir a responsabilidade do envio perante os serviços alfandegários do país. E posteriormente, os parceiros locais podem iniciar o transporte e distribuição dos *kits* pelo resto do país. Tudo isto deve ser rastreado usando um contracto inteligente.

A Datarella, uma empresa de soluções de *big data* e *Blockchain* baseada em Munique, na Alemanha, foi seleccionada como o parceiro de tecnológico de implementação.

O desafio para Datarella foi construir uma prova de tecnologia para um Cenário de Resposta Rápida (RR). Basicamente, um Cenário RR é uma situação em que bens de emergência (como medicamentos, tendas, lâmpadas solares, etc.) têm de ser distribuídos à população afetada após um evento catastrófico como um terremoto ou furacão. É essencial que esses itens cheguem ao destinatário em tempo útil, geralmente até 48 horas após o envio.

A integração e adoção de novas tecnologias nas organizações humanitárias pode ser difícil de conciliar com os procedimentos existentes e pode levar algum tempo antes de descolar.

Independentemente dos desafios enfrentados, estes projetos piloto inovadores são catalisadores para iniciativas maiores e potenciam melhorias nas vidas das pessoas a quem se destinam.

Apesar do piloto ainda estar a decorrer, foram identificados alguns sucessos:

- **Feedback rápido das partes interessadas -** A capacidade de solicitar e priorizar com rapidez e eficácia o feedback e reclamações. Ser capaz de reunir informações num relatório conciso permite um melhor planeamento e uma taxa de sucesso maior na satisfação. A Datarella (o parceiro de implementação) foi capaz de criar inquéritos *online*, concisos e de fácil acesso para envolver e as várias partes interessadas.
- **Colaboração** - O sucesso do piloto centra-se em grande parte no facto de ter um parceiro de implementação local envolvido na facilitação da entrega dos pacotes de ajuda entre os países.
- **Medição do sucesso** - Uma fase chave na medição de sucesso em num *Supply Chain* humanitário é a chamada entrega de "*last mile*" diretamente ao beneficiário, a rastreabilidade e a transparência na "*Last mile*" são tradicionalmente muito baixas. Neste piloto passaram a ser recolhidas estatísticas em tempo real, relativamente ao tempo que leva para o beneficiário a receber os pacotes de ajuda, se os itens foram entregues no local certo, a satisfação e a correcta utilização desses itens, e qual a sua contribuição para melhorar a confiança e a eficiência na *Supply Chain*.

2.4.1.3 Crowdfunding

O *Blockchain* pode servir de plataforma que assegura a rápida angariação de fundos e a sua atribuição de forma transparente e com baixo custo por transação, ao mesmo tempo que assegura o cumprimento das regulamentações aplicáveis. No entanto no sector na ajuda humanitária apenas foi possível encontrar literatura sobre um piloto na área.

2.4.1.3.1 Helperbit

Helperbit, é uma *startup* italiana, emergiu com o aumento da conscientização de ineficiências na gestão de fundos para ajuda em emergências humanitárias. Helperbit visa mudar práticas relacionados à resposta de emergência, focando-se na assistência humanitária, o setor de caridade e o sistema de seguros. A *startup* está a desenvolver duas soluções, ambas baseadas em redes públicas de *Bitcoin*: Um serviço de seguros paramétrico *peer-to-peer* para a fase pré-desastre e pós-desastre; Um sistema de donativos *peer-to-peer*.

No sistema paramétrico a liquidação do seguro não é com o valor actual da perda, mas em vez disso, é um montante específico pré-acordado que é pago em caso de ocorrência de um evento e condições definidas (e.g. o pagamento de 20.000€ para a perda total do carro, são pagos automaticamente havendo registo de um ciclone na zona de residência e pela foto do carro debaixo de uma arvore).

Na plataforma de donativos as pessoas podem fazer donativos directamente às NGO's para causas genéricas ou outras entidades (protecção civil, hospitais ou municípios) envolvidos na resposta de emergência.

Em dezembro de 2016, após um terremoto ter atingido a região centro de Itália, a Helperbit iniciou uma colaboração com Legambiente, uma NGO italiana, que se tornou a primeira grande organização sem fins lucrativos na Itália que aceita doações de *Bitcoin*. Eles receberam mais de 10 *Bitcoins* (cerca de 50.000 USD à data do terremoto e 70.092 USD ao valor actual) com aproximadamente 200 doações. Isso permitiu a conclusão da primeira cadeia doações

certificadas pelo *Blockchain* transparente. A *startup* lançou uma plataforma de doações transparentes em Novembro de 2017 no parlamento italiano. Cerca de 15 organizações sem fins lucrativos estão a angariar fundos usando este nova abordagem (Parker, 2017).

O projeto está atualmente focado em resposta a terremotos, mas planeia expandir a cobertura para todos os desastres. Actualmente está também a suportar donativos para a reconstrução da catederal de Notre Damme em Paris, tendo já angariado 10.000 USD pela Eidoo

3 Processo técnico

Com o objetivo de avaliar a tecnologia de *Blockchain* na sua resposta aos problemas levantados no sector da ajuda humanitária, foi implementada no decorrer do projeto uma rede *Blockchain*. Pretende-se assim simular num ambiente controlado alguns exemplos específicos da aplicabilidade da tecnologia, nomeadamente a transparência no rastreiro de bens transaccionados entre os diferentes intervenientes (com a garantia de confiabilidade e consenso), e avaliando também os benefícios em termos de eficiência.

A implementação deste artefato envolveu as seguintes fases:

- **Setup do ambiente de trabalho –** A fim de estabelecer as base do ambiente de trabalho foi criada uma máquina virtual com 2 cores e 8GB de memória RAM com o sistema operativo *Ubuntu 16.04 LTS*. Posteriormente foram instaladas as diferentes componentes do ambiente de desenvolvimento, *HyperLedger Fabric*, *Hyperleder Composer* e *HyperLedger Explorer*.
- **Definição e mapeamento da Rede de Negócio -** Em primeiro lugar foi necessário definir e mapear uma rede de negócio, para o caso de uso de trocas de ajuda humanitária ou solidária, com a definição dos seus intervenientes e os bens a serem transaccionados com recurso ao *HyperLedger Composer*.
- ***Deployment* da rede da Rede de Negócio –** Disponibilização da rede de negócio no *HyperLedger Fabric*, expondo API's e geração de aplicação para interacções com a rede.
- **Realização de testes à Rede de Negócio –** Utilizando a aplicação web de interacção com a Rede de Negócio *Blockchain*, foram efectuados testes para validação dos problemas elencados inicialmente. Os testes incidiram fundamentalmente sobre a criação e remoção de *Participants*, *Assets* e também na execução de transações que alteram a propriedade dos *Assets* ou seus atributos. Nesta fase foi utilizado o *HyperLedger Explorer* para validar os blocos e transações na rede nomeadamente os elementos criptográficos, relacionamento entre blocos, auditoria às transações e acessos. Actualizações aos *Smart Contracts* na rede e avaliação de impactos.

3.1 Arquitectura da solução implementada

O *HyperLedger Composer* permitiu a criação de forma rápida e intuitiva uma solução de *Blockchain* completa. Ou seja lógica de negócios executada no *runtime* de *Blockchain*, APIs REST que expõem a lógica *Blockchain* para as aplicações Web clientes, ou para integrar o *Blockchain* com outros sistemas e registos já existentes.

No diagrama lógico de arquitetura seguinte apresenta-se a forma com as diferentes componentes da solução foram configuradas de forma a suportarem *Blockchain* para uma rede de negócios de ajuda humanitária. Bem como o tipo de interacção com os utilizadores finais que vão utilizar a rede no dia-a-dia, quer um auditor ou especialista forense, bem como os analistas ou programadores nas fases de definição e mapeamento da rede ou posteriormente em manutenção evolutiva da mesma.

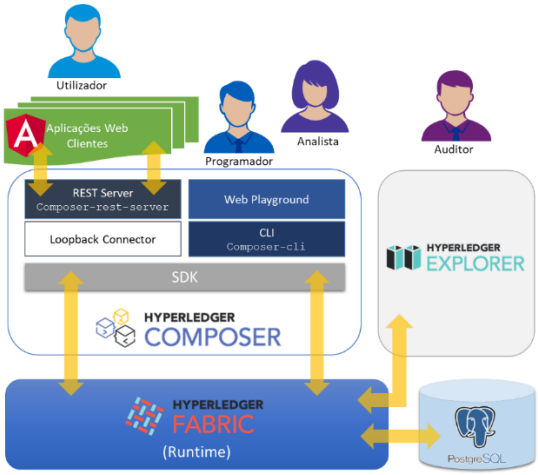

FIGURA 11 -DIAGRAMA LÓGICO DE ARUITECTURA DA SOLUÇÃO

A ligação do *HyperLedger Composer* com o *Runtime* é efectuada a partir de um perfil de conexão que está contido no *Business Network Card* (será explicado mais adiante). O perfil de conexão para o *HyperLedger Fabric Runtime* contem os endereços e portas TCP / IP para os nós (*peers*) da *Fabric*, assim como os certificados criptográficos, etc.

O SDK *JavaScript do HyperLedger Composer* é um conjunto de APIs em *Node.js* que permite aos programadores criarem aplicações para gerir e interagir com as redes de negócio implementadas.

Composer-client é utilizado para enviar transações para uma rede de negócios ou executar operações de criação, leitura, atualização e eliminação de *Assets* e participantes. No âmbito deste trabalho foi instalado e configurado com dependência do *HyperLedger Composer Playground*.

HyperLedger Composer Playground disponibiliza uma interface web que permitiu ao aluno efectuar a definição, mapeamento e testes da rede de negócio de ajuda humanitária. Permite que de forma rápida se consigam criar rapidamente provas de conceito e pilotos de redes de negócios podem ser executadas tanto no *HyperLedger Composer* como no *HyperLedger Fabric*.

O *HyperLedger Composer REST Server* gerou automaticamente uma API REST em *Open* API (*Swagger*) para a rede de negócios da ajuda humanitária. O REST *Server* (tendo como base o *LoopBack Connector*) converte o modelo do *HyperLedger Composer* da rede de negócios numa definição em *Open* API e, em *runtime*, implementa o suporte para *Create, Read, Update* e *Delete* (CRUD) para *Assets* e Participantes, adicionalmente também permite a execução de Transações.

Por fim o *HyperLedger Composer* com base na API REST disponibilizada pelo *HyperLedger Composer REST Server*, e com base no gerador de código *Open Source Yeoman* criou o esqueleto de um projeto de aplicação cliente web baseada na Angular framework.

A Base de dados Postgre SQL é usada pelo *HyperLedger Explorer* como *state database*, é lá que é armazenado o *World State* também é persistida informação relativamente aos blocos e transações, tendo sido instalada como dependência do *HyperLedger Explorer*. O *HyperLedger Explorer* é um módulo *Blockchain* que permite visualizar e consultar blocos, transações e os respectivos dados associados.

As componentes com interface HTTP para WEB ou API foram disponibilizadas nos seguintes portos:

- *HyperLedger Composer Playground:* http://localhots:9090
- *HyperLedger Composer REST Server:* http://localhost:3000
- Aplicação *Web* cliente gerada em *Angular:* http://localhost:4200
- *HyperLedger Explorer:* http://localhost:8080

3.2 Rede de Negócio implementada

A rede de negócio explorada neste trabalho é composta da seguinte forma:

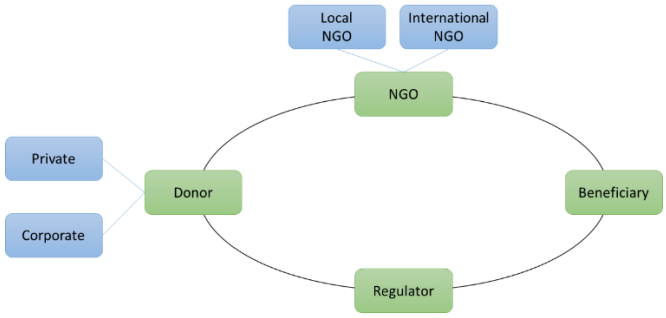

FIGURA 12 - DIAGRAMA DA REDE DE NEGÓCIO

A rede representada em cima define os participantes, os participantes podem ser dos seguintes tipos:

- ***Donor***: quem efectua a doação dos bens de ajuda humanitária ou de solidariedade.
- ***Beneficiary***: quem recebe os bens de ajuda humanitária ou de solidariedade
- ***NGO* Local:** Organização não-governamental, cuja área de intervenção é geograficamente próxima dos beneficiários, e que providência ou age como facilitador no circuito para fazer chegar os bens doados pelos "***Donors***" aos "***Beneficiaries***".
- ***NGO* Internacional:** Organização não-governamental, que actual de forma global e independente dos pais ou área, é normal utilizar NGO's locais como parceiros para que os donativos da sua rede de "***Donors***" chegue aos "***Beneficiaries***".
- ***Regulator*:** Entidade Governamental, ou não, que age como regulador na rede, tem com a responsabilidade a definição das regras que tem de ser observadas nas transferências de bens assim como auditar que as mesmas são cumpridas.

Os participantes vão poder transaccionar bens na rede de ajuda humanitária, denominados de *Assets*. Para este trabalho definimos os seguintes *Assets* para as trocas entre participantes:

- ***Good:*** Bens a serem transaccionados pelos participantes, podem ser: Comida; vestuário; materiais de construção; medicamentos; etc. Um G***ood***, tem sempre um proprietário, que é um dos participantes na rede.
- ***GoodListing:*** Lista de bens que representam um pedido de doação ou campanha de angariação de diferentes tipos de ***Goods***. O ***Goodlisting*** pode ser solicitado por um beneficiário ou NGO, sendo que ca um dos participantes pode doar ***Goods*** que correspondem a elementos da lista.
- ***Location:*** Representa a localização física do bem; um ***Donor*** ou ***Beneficiary*** apenas têm uma localização; NGOS podem ter várias, uma por armazém ou centro de recolha. Esta definição é fundamental para garantir a rastreamento dos bens durante o seu fluxo na cadeia.

Para a criação de *Assets* na rede, para a troca de propriedade dos mesmos ou para suportar o rastreamento da transferência destes bens de ajuda humanitária entre diferentes localizações, Armazéns, centros de recolha ou ajuda. São definidas transações, que por sua vez vão embeber lógica e regras de negócio específicas a cada uma das operações. As transações definidas são:

- ***Donation:*** transferência de proprietário do ***Good*** associando o mesmo a um ***Goodlisting***, inclui a actualização da ***Location*** do ***Good*** e da quantidade requerida no ***Goodlisting***.
- ***AssignListing:*** transferência de ***Good*** associados a ***GoodListing*** a beneficiários, por um participante NGO. Só pode acontecer quando o estado do ***GoodListing*** é "Closed" alterando a ***Location*** de acordo com o beneficiário e o estado do ***Good***.
- ***GoodArrival:*** alteração da ***Location*** do ***Good***, por um participante NGO, após o transporte.
- ***GoodShipping:*** alteração do estado do ***Good***, relativamente a uma ***Location*** de destino por um participante NGO, quando inicia o transporte**.**

Tendo em contas as definições apresentadas foi efectuado o mapeamento da rede no *HyperLedger Composer*, a próxima secção descreve esse processo de desenvolvimento.

3.3 Implementação no *Hyperleder Composer*

Como suporte deste projeto foi utilizado o *HyperLedger Composer* (descrito na secção 2.2.2). *O HyperLedger Composer Playground* fornece uma interface para configuração, implementação e teste de uma rede de negócios em *Blockchain*.

A utilização desta plataforma foi muito fácil e intuitiva para a modelação da rede negócios definida na secção anterior, tendo em conta as seguintes definições:

- ***Assets***: bens, listagens e localizações
- ***Participants***: doadores, beneficiários, NGOs locais, NGOs internacionais e Regulador
- ***Transactions***: doação, assignação de bem, envio de bem e recebimento de bem.

A *Business Network Definition* é um conceito chave no modelo de programação do *hyperLedger Composer*. Sendo composto pelas seguintes componentes, que assim que criados podem ser empacotados para *deploy* no *HyperLedger Fabric runtime*:

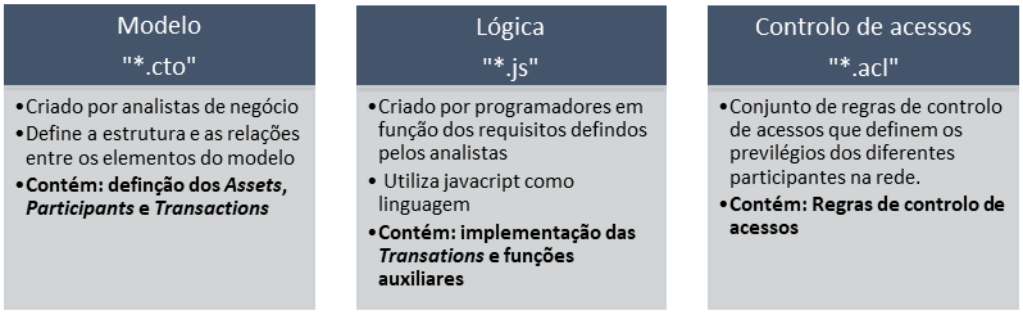

FIGURA 13- ARTEFACTOS DE PROGRAMAÇÃO UTILIZADOS NO HYPERLEDGER COMPOSER

Para efeitos de *deployment* o conjunto de ficheiros é agregado num só ficheiro de pacote com extensão "*.bna" (*Business Network Archive*). O *Deployment* é efectuado no *HyperLedger Fabric* por linha de comando.

Adicionalmente foi utilizado o *Visual Studio Code* como ferramenta de edição de código, utilizando uma extensão própria para o *HyperLedger Composer* que permite a validação da estrutura do modelo e efectua o *parsing* do código para validação de erros de sintaxe.

3.3.1 Modelo

Após a análise à rede de negócios foi possível efectuar a definição do modelo de dados do domínio para a mesma. O modelo de dados foi expresso usando o *Composer Modeling Language* e define a estrutura dos recursos que serão armazenados no *Ledger* ou processados pelas transações.

Assim tendo em conta os participantes na rede já identificados na secção 3.2, foram todos caracterizados como sendo da classe ***Member*** que por sua vez estende a classe abstracta ***User***, com excepção do participante ***Regulator*** que é da classe ***User***.

```
/**
 * Basic class to define a participant
 */
abstract participant User identified by vatNumber {
  o String vatNumber
  o String firstName
  o String lastName
  o String email
}

/**
 * A Member participant
 */
participant Member extends User {
    o String address1
    o String address2
    o String county
    o String postcode
    --> Location location optional
}

/**
 * A Regulator participant
 */
participant Regulator extends User {
}
```

FIGURA 14- DEFINIÇÂO DE PARTICIPANTES

Os *Assets* identificados anteriormente foram também definidos no modelo, com a seguinte representação.

```
/**
 * A Location asset which is owned by a Member, is related
 * to a list of warehouses and has a list of incoming goods.
 */
asset Location identified by lid {
     o String lid
     o String address1
     o String address2
     o String county
     o String postcode
     --> Member owner
     --> Good[] incomingGoods optional
}

/**
 * An Good asset, which can be stored in a Wharehouse
 */
asset Good identified by id {
  o String id
  o String description
  o GoodType type
  o Integer amount
  o MovementStatus movementStatus
  --> Location location optional
  --> Member owner
}
/**
 * Listing of goods that are in need
 */
asset GoodListing identified by listingId {
  o String listingId
  o String description
  o Integer quantity
  o ListingState state
  o Donation[] donation optional
  --> Member owner
}
```

FIGURA 15 - DEFINIÇÃO DOS ASSETS

Igualmente as transações também foram definidas no mesmo modelo. No modelo não se efectua implementação apenas a definição.

```
/**
 * A transaction type for an Good leaving a location
 */
transaction GoodShipping extends GoodMovement {
  --> Location fromLocation
}

/**
 * A transaction type for an Good arriving at a location
 */
transaction GoodArrival extends GoodMovement {
  --> Location arrivalLocation
}

/**
 * Member makes a Good donation to a specific Listing
 */
transaction Donation {
  --> Member member
  --> Good good
  --> GoodListing listing
}

/**
 * Listing owner closed the listing
 */
transaction CloseDonation {
  --> GoodListing listing
}

/**
 * Member assign a Listing to beneficiary
 */
transaction AssignListing {
   --> GoodListing listing
   --> Member beneficiary
}
```

FIGURA 16- DEFINIÇÃO DAS TRANSACÇÕES

3.3.2 Transacções

Com a definição do modelo de domínio concluída, definiram-se então os *Smart Contracts*, que contêm as regras e lógica de negócio escrita em linguagem *JavaScript*. Estes *Smart Contracts* que incluem as funções a serem processadas na execução das transações.

A lógica implementada em *JavaScript* nas transações foi um pouco extensa, assim apresenta-se apenas exemplo de duas das transações mais relevantes na próxima figura.

```
/**
 * Make an donation for a GoodListing
 * @param {org.jan.aid.exchange.Donation} donation - the donation
 * @transaction
 */
async function makeDonation(donation) {  // eslint-disable-line no-unused-vars
    let listing = donation.listing;
    if (listing.state !== 'IN_NEED') {
        throw new Error('Listing is not IN_NEED');
    }
 /*   if (listing.member !== donation.good.owner) {
        throw new Error('Not the owner!!!');
    }
 */    if (listing.quantity >= donation.good.amount) {
        listing.quantity -=  donation.good.amount;

        donation.good.owner = listing.owner;
        donation.good.location= listing.owner.location;

    } else {
        throw new Error('Good amount hihger than needed');
    }
    if (!listing.donation) {
        listing.donation = [];
    }
    listing.donation.push(donation);

    // save the Good listing
    const goodListingRegistry = await getAssetRegistry('org.jan.aid.exchange.GoodListing');
    await goodListingRegistry.update(listing);

    // save the Goods
    const goodRegistry = await getAssetRegistry('org.jan.aid.exchange.Good');
    await goodRegistry.update(donation.good);
}

/**
 * Make an assing for a GoodListing
 * @param {org.jan.aid.exchange.AssignListing} assignment - the assignment
 * @transaction
 */
async function makeAssignListing(assignment) {  // eslint-disable-line no-unused-vars
    const listing = assignment.listing;
    const listofGoods = [];
    if (listing.state !== 'CLOSED') {
        throw new Error('Listing is not CLOSED');
    }
    listing.donation.forEach(function(donation) {
        donation.good.owner = assignment.beneficiary;
        donation.good.location = assignment.beneficiary.location;
        donation.good.movementStatus = 'ASSIGNED';
        listofGoods.push(donation.good);
    });
    // save the Goods
    const goodRegistry = await getAssetRegistry('org.jan.aid.exchange.Good');
    await goodRegistry.updateAll(listofGoods);
}
```

FIGURA 17- TRANSACÇÕES DONATION E ASSIGNLISTING

Em anexo pode ser consultado o código fonte implementado para cada uma das transações definidas, incluídas no ficheiro de lógica de transações em *JavaScript*.

3.3.3 Controlo de acessos

O *HyperLedger Composer* inclui uma linguagem de controle de acesso (ACL) que fornece controle de acesso declarativo sobre os elementos do modelo de domínio definido anteriormente. Ao definir regras de ACL, os participantes podem ter acesso restrito para criar, ler, actualizar ou eliminar *Assets* e invocar transações com base no seu perfil como beneficiário, doador ou NGO.

```
/**
 * Access Control List for the Aid Exchange network.
 */
rule Regulator {
    description: "Allow the Regulator full access"
    participant: "org.jan.aid.exchange.Regulator"
    operation: ALL
    resource: "org.jan.aid.exchange.*"
    action: ALLOW
}
rule Member {
    description: "Allow the member Full access only when is the owner"
    participant: "org.jan.aid.exchange.Member"
    operation: READ
    resource: "org.jan.aid.exchange.*"
    action: ALLOW
}
rule GoodOwner {
    description: "Allow the owner of a Good total access"
    participant(m): "org.jan.aid.exchange.Member"
    operation: ALL
    resource(g): "org.jan.aid.exchange.Good"
    condition: (g.owner.getIdentifier() == m.getIdentifier())
    action: ALLOW
}
rule GoodListingOwner {
    description: "Allow the owner of a good total access to their goods listing"
    participant(m): "org.jan.aid.exchange.Member"
    operation: ALL
    resource(gl): "org.jan.aid.exchange.GoodListing"
    condition: (gl.good.owner.getIdentifier() == m.getIdentifier())
    action: ALLOW
}
rule SystemACL {
    description:  "System ACL to permit all access"
    participant: "org.hyperledger.composer.system.Participant"
    operation: ALL
    resource: "org.hyperledger.composer.system.**"
    action: ALLOW
}
rule NetworkAdminUser {
    description: "Grant business network administrators full access to user resources"
    participant: "org.hyperledger.composer.system.NetworkAdmin"
    operation: ALL
    resource: "**"
    action: ALLOW
}
rule NetworkAdminSystem {
    description: "Grant business network administrators full access to system resources"
    participant: "org.hyperledger.composer.system.NetworkAdmin"
    operation: ALL
    resource: "org.hyperledger.composer.system.**"
    action: ALLOW
}
```

FIGURA 18- CONTROLO DE ACESSOS

4 Análise de resultados

Neste capítulo descreve-se a análise ao resultados dos testes na utilização da solução implementada, sobre casos de uso concretos e relevantes para responder às questões de investigação.

Os testes incidiram essencialmente sobre as componentes de funcionalidades do *Blockchain* e da rede de negócios implementada, e como endereçam as questões de transparência e eficiência.

4.1 Segurança e privacidade no acesso à informação

Por forma a validar como se pode garantir a privacidade e o acesso a informação, quer no suporte tecnológico já oferecido pelo *Blockchain* quer na rede de negócios implementada, foram realizadas as algumas atividades e configurações paras as quais se descrevem as evidências encontradas.

Na criação de utilizadores para acesso à rede, foi possível mapeá-los com cada um dos participantes na rede (bastou criar um *user* por entidade participante, com a indicação que este pode criar outros *users* associados ao mesmo participante), o *Member Service Provider* (MSP) do *HyperLedger Fabric* gera uma transação e um certificado para cada um desses utilizadores. Este certificado é então utilizado para validação do registo e assinatura de transações e também para gerar um *Business Network Card* que vai permitir que uma aplicação de negócio implementada pela entidade participante possa conectar-se e autenticar-se na API disponibilizada para esta rede de ajuda humanitária denominada Aid-Exchange.

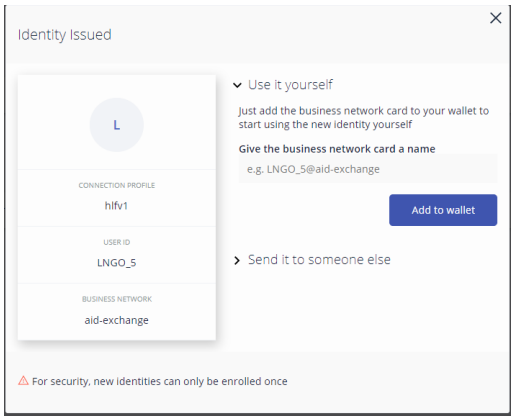

FIGURA 19 - BUSINESS NETWORK CARD DE UTILIZADOR LNGO_5

Para além da *Business Network Cards* para a identidade dos utilizadores, existem outros 3 tipos de cartões:

- ***Peer Admin*** – Cartão com o certificado que administra cada um dos nós na rede. Neste caso em concreto apenas foi gerado uma vez que só existia um nó (*peer*). Não é possível iniciar uma rede sem ele definido, contém o certificado que é a base para toda a criptografia relativa ao nó.
- ***Channel Admin*** – Cartão com certificado que administra um canal, o canal é o que permite definir redes dentro da rede, os nós estão associados a canais.

- ***Business Network Admin*** – Cartão com o certificado do administrador da rede de negócios. É fundamental para a criação da rede, não é possível fazer *deploy* da rede sem ele estar definido.

Na criação da identidade LNGO_5, utilizador para o participante (**Member**) na rede definido também pelo identificador LNGO_5, foi gerada uma transação no *Blockchain* para a criação da referida identidade. Em baixo apresenta-se o registo da referida transação, consultada no *HyperLedger Explorer* onde se pode observar a sua activação e o respectivo certificado emitido pelo *Member Service Provider* (MSP) do *HyperLedger Fabric*.

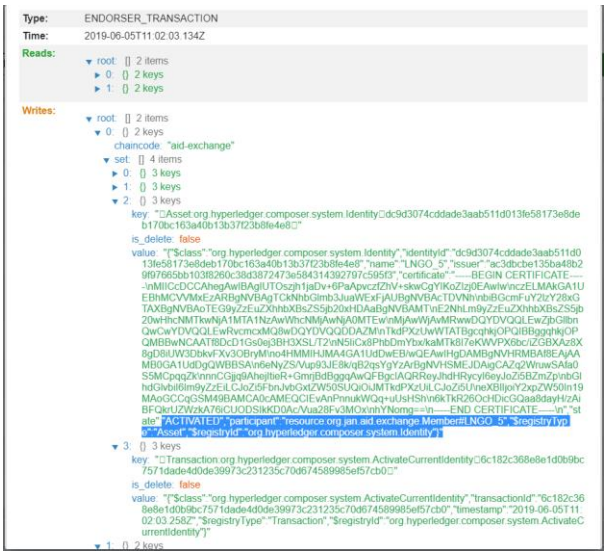

FIGURA 20 – TX DE CRIAÇÃO DE UMA IDENTIDADE PARTICIPANTE PARA O USER LNGO_5

Para demonstrar que é também possível declarativamente definir os privilégios de acesso à informação partilhada na rede de ajuda humanitária, efectuou-se uma alteração nas regras de acesso dos membros da rede (participantes) inicialmente definida. Assim a definição estabelecida inicialmente era que cada membro da rede poderia ter acesso exclusivo de leitura sobre toda a informação da rede. Estas definições tinham sido efectuadas no ficheiro de controlo de acesso "*.acl".

```
rule Member {
    description: "Allow the member read access"
    participant: "org.jan.aid.exchange.Member"
    operation: READ
    resource: "org.jan.aid.exchange.*"
    action: ALLOW
}
```

FIGURA 21- PERMISSÕES DOS MEMBROS INICIAL

Neste cenário o utilizador e participante LNGO_5 conseguia visualizar todos os *Assets* na rede, de acordo com as permissões definidas, mesmo daqueles dos quais não era proprietário.

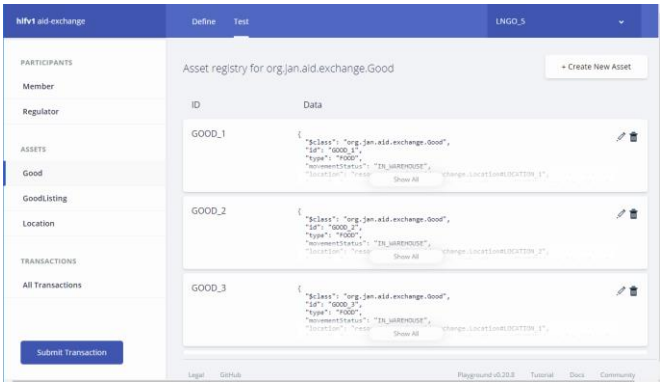

FIGURA 22- LISTAGEM DOS ASSETS DO TIPO "GOOD"

Assim sendo, alterou-se a regra para que cada membro apenas pode ter acesso exclusivo sobre os **"Goods"**, dos quais é proprietário. Alterando-se o ficheiro de permissões "*.acl".

```
rule Member {
    description: "Allow the member Full access only when is the owner"
    participant(m): "org.jan.aid.exchange.Member"
    operation: ALL
    resource(g): "org.jan.aid.exchange.Good"
    condition: (g.owner.getIdentifier() == m.getIdentifier())
    action: ALLOW
}
```

FIGURA 23 - PERMISSÕES DOS MEMBROS ACTUALIZADA

Assim que a nova definição da rede de negócios foi *deployed*, cada membro deixou de poder ver todos os ***"Goods"*** existentes na rede e passou a penas a poder ter acesso aos que são da sua exclusiva propriedade, tal com apresentado na próxima figura.

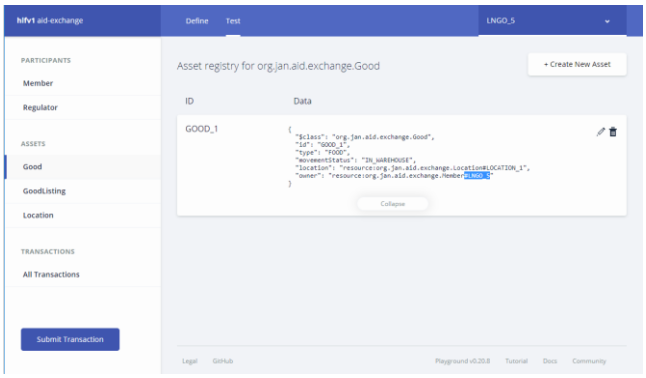

FIGURA 24 - MEMBRO LNGO_5 COM ACESSO RESTRITO AOS SEUS PROPRIOS ASSETS

Foi então possível demonstrar o *Blockchain* e a rede de ajuda humanitária implementada tem vários mecanismos que permitem salvaguardar a segurança e a privacidade no acesso aos dados. Estes mecanismos passam pela criação de utilizadores mapeados em identidades com certificados digitais gerados pelo MSP, são utilizados para a autenticação e garantia de identidade nas interacções com o *Blockchain*, seja no acesso directo ou por vai das API's REST. Os próprios módulos do *HyperLedger* necessitam que sejam criados *Business Network Cards* que serão então utilizados para a segurança na interoperabilidade entre *HyperLedger Fabric*, *HyperLedger Composer*, *HyperLedger Composer REST Server*, *HyperLedger Explorer* e na própria geração de aplicações clientes.

Na própria rede é possível definir permissões de acesso a todos os recursos modelados na rede, incluindo definir programaticamente regras de negócio para uma granularidade superior.

Assim verificou-se que as questões de segurança e privacidade no acesso à informação estão asseguradas para todos os utilizadores da rede. Embora seja importante que haja transparência na rede, conhecendo bem todos os participantes, existem dados cuja relevância obriga que exista algum tipo de restrição no acesso (Sexo, Religião, contactos pessoais, etc), pois um dos princípios fundamentais na inovação humanitária têm a ver com a impossibilidade da sua utilização para fins de maleficência e não descriminação.

A segurança e privacidade no acesso à informação, assegura que apenas sistemas e pessoas credenciados conseguem interagir na rede de ajuda humanitária, assim situações de abuso e usurpação de identidade no acesso a ajuda humanitária estão salvaguardadas. Também pela restrição de acessos há a garantia de que não existem condicionalismos na disponibilização e atribuição da ajuda humanitária aos beneficiários, no sentido de provocar prejuízo ou de dar benefíciar.

4.2 Rastreabilidade e eficiência nas trocas dentro da rede de negócio

Para demonstrar a rastreabilidade nas trocas dentro da rede de ajuda humanitária, foi criada por uma organização humanitária local denominada LNGO_5 uma solicitação de ajuda humanitária de 15 T-shirts, tendo sido criado um *Asset* do tipo ***GoodListing*** com o identificador "SHIRT_REQ".

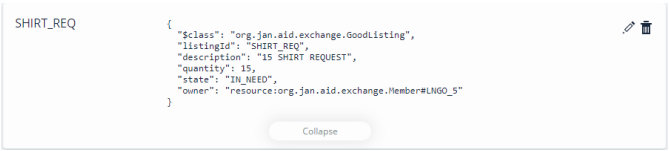

FIGURA 25- ASSET GOODLISTING (ESTADO INICIAL)

Os doadores DONOR_1 e DONOR_2 criaram dois *Assets* (***Good***) correspondentes às quantidades de 5 e 10 T-Shirts, identificados como SHIRT_1 e SHIRT_2 respectivamente.

FIGURA 26- ASSETS CRIADOS PELOS DONOR_1 E DONOR_2

Posteriormente cada um dos doadores efectuou transação (***Donation*)** dos seus bens (***Good***) para a requisição (***GoodListing***) de T-Shirts efectuada pelo membroLNGO_5.

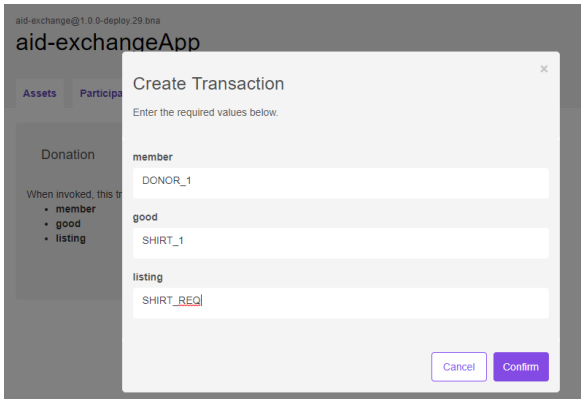

FIGURA 27- DONATION EFECTUADA VIA APLICAÇÃO WEB CLIENT

Com a execução da primeira transação pode comprovar-se foi adicionada ao ***GoodListing*** uma doação que inclui que para além dos parâmetros da transação inclui também o seu identificador único no *Ledger* e o *timestamp*. Para além disso com a execução do *Smart Contract* a quantidade da requisição foi subtraída de acordo com a quantidade doada. Também para o Asset (GOOD) SHIRT_1 foi alterado o proprietário automaticamente, no âmbito da mesma transação (execução de *Smart Contract*).

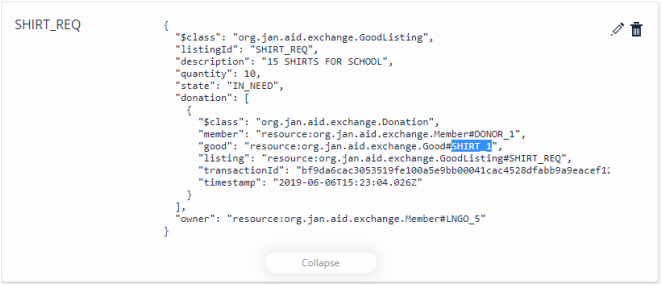

FIGURA 28- ASSET GOODLISTING (APÓS 1ª DOAÇÂO)

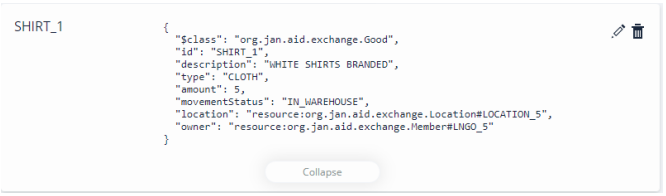

FIGURA 29- ASSET SHIRT_1 ALTEROU O PROPRIETÁRIO

Para o DONOR_2 executou-se a transação ***Donation*** utilizando directamente o *Hyperleger Composer playground* utilizando o respectivo código *JSON* com os parâmetros.

```
{
  "$class": "org.jan.aid.exchange.Donation",
  "member": "resource:org.jan.aid.exchange.Member#DONOR_2",
  "good": "resource:org.jan.aid.exchange.Good#SHIRT_2",
  "listing": "resource:org.jan.aid.exchange.GoodListing#SHIRT_REQ"
}
```

```
{
 "$class": "org.jan.aid.exchange.Donation",
 "member": "resource:org.jan.aid.exchange.Member#DONOR_2",
 "good": "resource:org.jan.aid.exchange.Good#SHIRT_2",
 "listing": "resource:org.jan.aid.exchange.GoodListing#SHIRT_REQ",
 "transactionId":
"a2fb1299701ab70819aa0d93de3a9baf06caeb27178ab76d0a54211bbd0e8a9a",
 "timestamp": "2019-06-06T15:24:45.918Z"
}
```

FIGURA 30- PEDIDO E RESPOSTA EM JSON DA EXECUÇÃO DA TRANSACÇÃO

Uma vez mais esta execução teve efeitos automáticos nos *Assests* ***Good*** e ***GoodListing*** envolvidos. Na solicitação SHIRT_REQ pode comprovar-se que tem agora duas ***Donations*** e a quantidade passo a zero, umas vez que foi totalmente suprida com esta segunda doação de quantidade 15. De referir que tal como a primeira transação também nesta ficou o registo do identificador e respectivo *timestamp*. O *Asset* ***Good*** doado mudou automaticamente de proprietário.

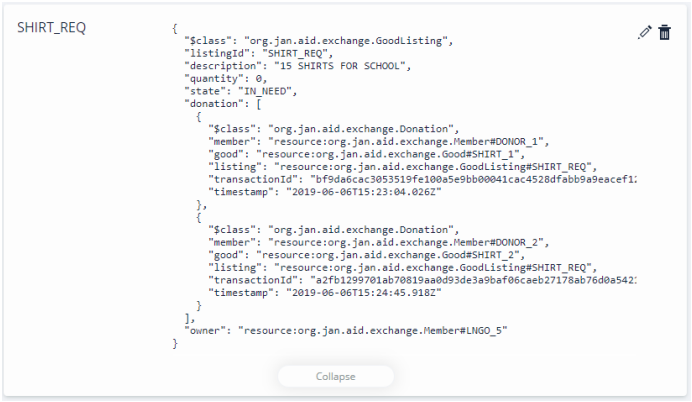

FIGURA 31- ASSET GOODLISTING (APÓS 2ª DOAÇÂO)

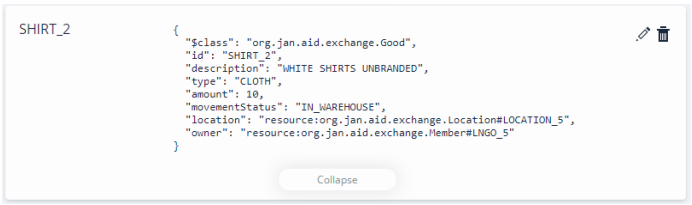

FIGURA 32-ASSET SHIRT_2 ALTEROU O PROPRIETÁRIO

Uma vez que a quantidade de T-Shirts requerida foi atingida, o membro LNGO_5 altera o estado do ***GoodListing*** de "IN_NEED" para "CLOSED". Pois existe uma regra no *Smart Contract* que impede que a ajuda seja atribuída a beneficiários sem que a requisição esteja concluída.

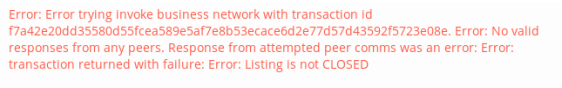

FIGURA 33- ERRO DE VALIDAÇÃO NO ASSIGNLISTING

FIGURA 34- ASSET GOODLISTING (APÓS ALTERAR ESTADO)

Finalmente LNGO_5 pode atribuir as 15 T-Shirt para que a escola da aldeia, BENEFECIARY_7, as distribua pelos meninos e meninas da turma do 1º Ano. Para este efeito Executa a transação ***AssignListing*** para o ***GoodListing*** SHIRT_REQ ao ***Member*** BENEFECIARY_7, que como efeito os ***Good*** SHIRT_1 e SHIRT_2 passam a ter como proprietário o BENEFECIARY_7 em vez do LNGO_5.

Para a execução desta transação foi utilizando directamente a REST API gerada para a rede de negócios pelo *Hyperleger Composer REST Server*.

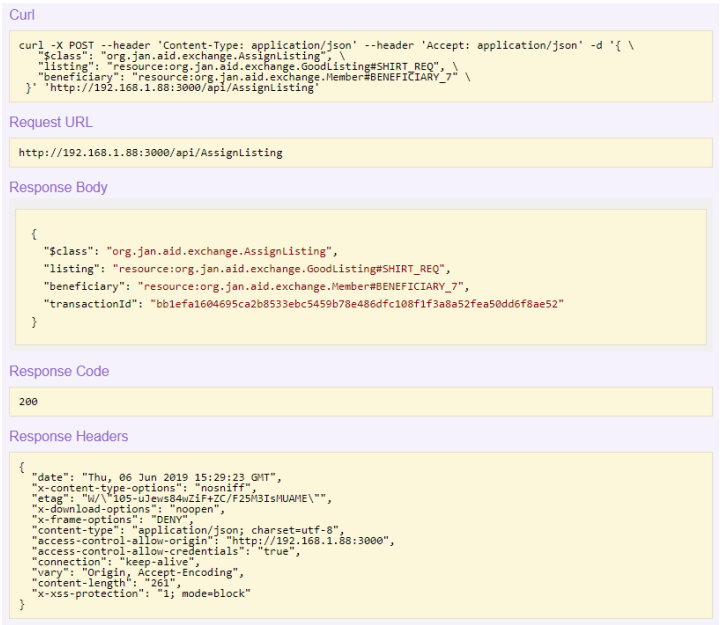

FIGURA 35- PEDIDO E RESPOSTA DA EXECUÇÃO DO ASSIGNLISTING

Pode-se verificar que ambos os ***Good***, alteraram o estado para ASSIGNED, de proprietário e para a localização associada ao novo proprietário.

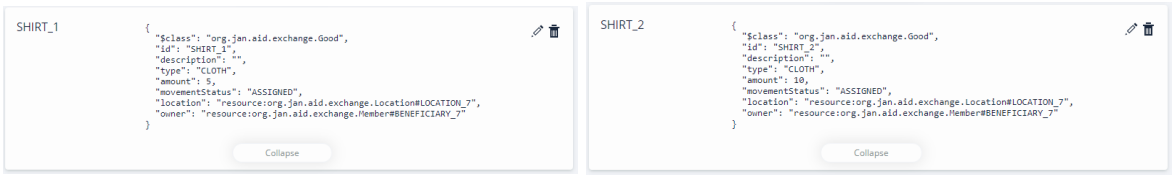

FIGURA 36- ESTADO FINAL DOS GOOD SHIRT_1 E SHIRT_2

O fluxo *end-to-end* de um processo de ajuda humanitária, desde o doador até ao beneficiário e com a articulação e logística das organizações de ajuda humanitária, ficou assim concluído neste exemplo.

Contudo é possível a um regulador ou auditor externo poder analisar em concreto as transações executadas, o *Blockchain* regista todas as transações executadas, embora no decorrer deste exemplo apenas se tenhamos dado enfase a três que envolveram a alteração de propriedade do bem transaccionado, cada uma com os seguintes códigos:

- Alteração de propriedade de SHIRT_1 na Doação: "bf9da6cac3053519fe100a5e9bb00041cac4528dfabb9a9eacef12c069823579"
- Alteração de propriedade de SHIRT_2 na Doação: "a2fb1299701ab70819aa0d93de3a9baf06caeb27178ab76d0a54211bbd0e8a9a"
- Alteração de propriedade de SHIRT_1 e SHIRT_2 na assignação ao beneficiário: "bb1efa1604695ca2b8533ebc5459b78e486dfc108f1f3a8a52fea50dd6f8ae52"

Utilizando o *Hyperleger Explorer* é possível observar as transações no *Ledger* e confirmar quem efectivamente executou o quê, quando e como. Para cada uma das 3 transação identificadas apresentam-se evidências do que efectivamente ficou registado no *Ledger*, cada transação está associada a um bloco no *Blockchain*.

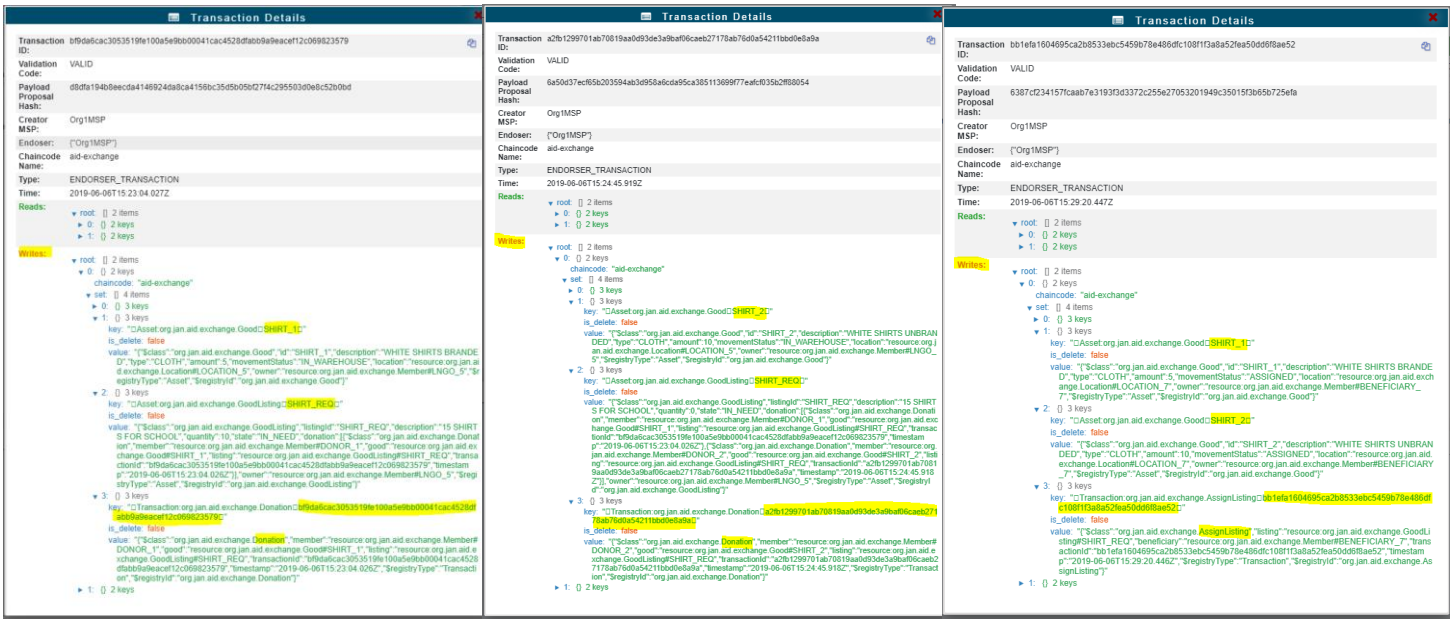

FIGURA 37 - EVIDÊNCIAS DA EXECUÇAO DAS TRANSACÔES DONATION E ASSIGNLISTING

Foi possível demonstrar que é possível garantir a rastreabilidade numa rede *Blockchain* para ajuda humanitária, quer no desenho e implementação da própria rede de negócio. Neste caso particular o ***GoodListing***, guarda o registo de todos os donativos efectuados e das respectivas transações associadas com o *timestamp*. Garantindo que os utilizadores da rede têm completa visibilidade sobre as trocas efectuadas por via da execução de *smart contracts* nas transações, quer pela consulta dos próprios *Assests* da rede.

Mas também pelas próprias funcionalidades na plataforma do *HyperLedger* que através de diferentes ferramentas: *HyperLedger Explorer* que permite aceder aos registos de tudo o que aconteceu na rede caso haja necessidade de auditoria; na integração com as aplicações o *HyperLedger Composer REST Server* disponibiliza API REST que para casa serviços REST invocado retorna sempre o identificador da transação correspondente, permitindo também Às aplicações clientes registarem e manterem um histórico dessas invocações.

A rastreabilidade e a transparência da rede de negócios para a ajuda humanitárias têm um papel fundamental para aumentar a confiança no sistema por parte dos seus utilizadores. Por outro lado e uma vez que é a rastreabilidade fica acessível a todos os participantes, há um desincentivo das más práticas e falta de critério na atribuição de ajuda ou da utilização abusiva das ajudas provenientes dos diferentes doadores e organizações.

Por outro lado também ficou claro que pelo facto de estarem a ser executados *Smart Contracts*, com aplicação das regras e lógica de negócio automaticamente nas transações, quer na alteração de propriedade, na actualização da localização dos bens, das quantidades necessárias, etc. Existem ganhos de eficiência nos processos de ajuda humanitária que consequentemente beneficiam todos os atores do sistema humanitário, mas mais importante os beneficiários directos da ajuda, uma vez que com os ganhos de eficiência decorre menos custos, logo mais ajuda para distribuir, e/ou menos tempo, logo os bens e a ajuda chegam mais rápido onde é mais necessária. A Automatização também liberta os agentes humanitários das tarefas burocráticas

podendo os mesmos concentrarem-se em atividades de valor acrescentado, na angariação de ajuda e fundos ou na assistência às populações carenciadas.

4.3 Consenso e imutabilidade

Explorando algumas das limitações, em termos de capacidade e performance, do ambiente de suporte ao projeto e também da assincronissidade das ferramentas de desenvolvimento baseadas em tecnologias WEB. Foi possível e provocar situações de pedidos concorrentes de transações ao *Blockchain*, nomeadamente na remoção/eliminação de *Assets* na rede, em que foi possível clicar para eliminar o *Asset*, mas uma vez que a transação não era imediata devido às limitações de performance do ambiente virtualizado, o *Web Browser* não removia o cartão imediatamente, sendo possível voltar a clicar para eliminar. Esta situação gera pedidos concorrentes de transações sobre o mesmo artefato, ao qual o *Blockchain* consegue validar, por via do *endorsement*, a transação que tem o *timestamp* mais baixo e consequentemente rejeitando (não validando) as subsequentes. Tal como o erro apresentado na figura em baixo.

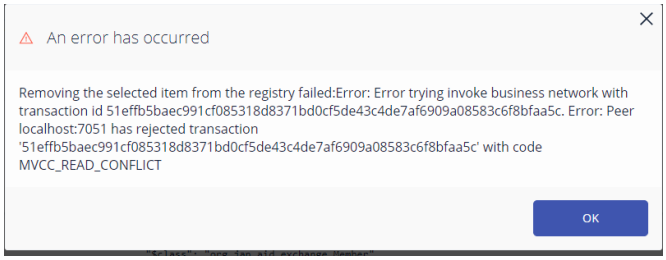

FIGURA 38- ERRO GERADO NO ACESSO CONCORRENTE AO MESMO RECURSO NO BLOCKCHAIN

4.4 Não Repudiação

Com o objetivo de validar a não repudiação das transações efectuadas na rede, eliminou-se um dos Assets da rede, neste caso particular um ***Good***. Esta operação foi efectuada com sucesso pelo utilizador LNGO_5, que é um membro (participante) e tem uma identidade criada na rede de negócio.

Utilizou-se o *HyperLedger Explorer* para consultar as transações executadas, directamente no *Ledger*, e verificou-se o que efectivamente ficou registado que a transação de eliminação referida ficou registada com tendo sido executada pela identidade do membro da rede LNGO_5, como se pode comprovar na figura em baixo.

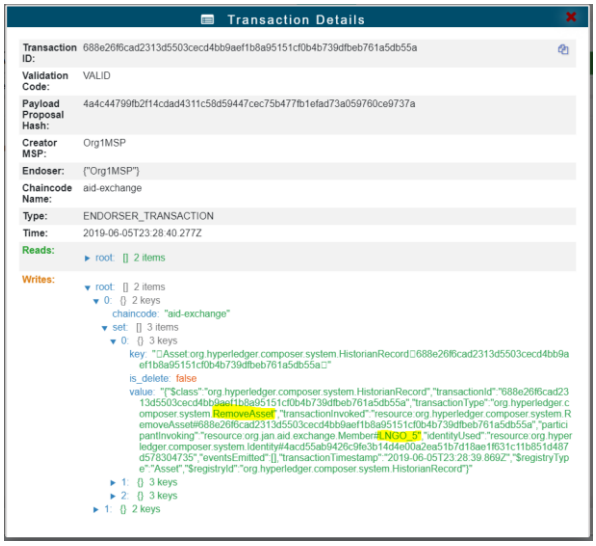

FIGURA 39 - TRANSACÇÃO DE ELIMINAÇÃO DE ASSET COM INDICAÇÃO DO EXECUTANTE

Este padrão verifica-se para toda e qualquer transação executada, desta forma está garantida a não repudiação na rede de negócios. A visibilidade sobre este tipo de operações reforça a transparência existente na plataforma.

5 Conclusões

A solução implementada e os testes realizados, em ambiente controlado na fase de análise de resultados (secção 4) pela exploração de casos práticos de utilização da solução, permitiu a consolidação dos conceitos base do *Blockchain* e a validação da sua adequabilidade para endereçar de forma cabal os problemas de negócio do sector humanitário. Assim passam-se a descrever as principais conclusões do trabalho executado tendo em conta alguns dos problemas e questões de investigação identificados na secção 1.2. Adicionalmente introduziu-se alguns comentários tendo em conta a análise efectuada aos casos de uso de projetos-piloto detalhados na secção 2.4.1

5.1 Transparência

Na solução implementada foi possível comprovar que todos os elementos da rede tem acesso às transações e *Assests*, de acordo com as permissões e acessos definidos. A interacção com os *Assets* é efectuada via *Chaincode,* de acordo com a lógica definida nos *smartcontracts* e executada nas transações. Cada um dos elementos da rede corresponde a um nó (*peer*), que consequentemente valida e confirma cada uma das transações e mantem um *Ledger* distribuído e *World State*. O *HyperLedger*, disponibiliza as funcionalidades core, tendo como base algoritmos e cifras, que suportam toda a segurança e imutabilidade do *Ledger*, garantindo assim a rastreabilidade sobre todas as transações e alterações de propriedade sobre os *Assets*. A transparência advém do facto da toda a rastreabilidade poder ser acessível a todo e qualquer elemento da rede, são os nós (peer) que validam as transacções e registos no *Blockchain*, todas as alterações ficam registadas e os registos são imutáveis, interligando o a transparência com o consenso.

Na solução implementada nomeadamente quando é efectuada uma doação de um ***Good*** e associada a uma requisição ***GoodListing,*** o mesmo passa a ter um registo de todos os ***Good*** associados. Assim como é possível verificar o proprietário de cada um dos ***Good*** disponíveis bem como a sua localização durante toda a cadeia de distribuição logística.

A transparência nos processos das organizações de ajuda humanitária, permitida pelo *Blockchain*, permite garantir a uniformidade de critérios na distribuição e atribuição de ajuda. Cria confiança no doadores, pois passam a ter visibilidade sobre a forma como as suas dádivas estão a ser distribuídas e o impacto que estão a ter nas vidas dos beneficiários. Com a transparência são também disponibilizados e registados mais dados pelo Blockchain na rede, dados esses que podem ser analisados à posteriori para implementar medidas que permitam trazer mais eficiências.

Na solução implementada, nas secções 4.3 e 4.4, foi possível verificar a rastreabilidade das transações utilizando o *HyperLedger Explorer* por um auditor, sendo também possível garantir a responsabilização (não-repudiação) do autor das transações que efectivamente realizou. Esta é também uma forma de transparência introduzida pelo *Blockchain*.

Adicionalmente solução não obriga à inclusão no *Blockchain* de dados pessoais dos membros da rede, sejam doadores, beneficiários ou NGO’s, dados pessoais devem ficar contidos nos repositórios locais às aplicações que interagem com os nós (*peer*). Desta forma é possível

endereçar requisitos de GDPR (*General Data Protection Regulation*) da Comunidade Europeia ou outras, quer pela introdução da pseudonimização, confidencialidade, encriptação, integridade e disponibilidade dentro da rede *Blockchain* do *HyperLeder*, quer pela garantia que existem diferentes instâncias das aplicações com repositórios locais nos países de origem que garantem a não portabilidade de dados para países terceiros.

As questões da segurança e privacidade no acesso à informação, são relevantes para a transparência no sentido em que, aumentam a confiança na rede e no *Blockchain*. Desta forma os participantes na rede tem a garantia que podem disponibilizar informação relevante para os processos de identificação, distribuição e atribuição de ajuda sem que a mesma seja comprometida.

Da revisão da literatura foi também possível verificar ganhos de Transparência em alguns dos projetos-piloto. Nomeadamente no projeto piloto *Open Loop* implementou um *Blockchain* usando a plataforma *Multichain* (https://www.multichain.com/) fazendo uso das funcionalidades de base de DLT já fornecidas pela plataforma. Subscreveram um serviço de 4 nós de *Blockchain* permitindo KRCS, IFRC, e RedRose a visualização das transações, promover assim a transparência e a co-responsabilização entre todos os envolvidos na rede.

E no caso da KRCS, os dados relativos à maioria dos projetos é armazenada nas aplicações, na interligação com *Blockchain* esta informação torna-se descentralizada e cada nó (*peer*) da cadeia sendo um garante da disponibilização da informação de acordo com os previlégios. Durante a implementação de projeto piloto nenhuma informação pessoal foi gravada no *Blockchain* (ICHA, 2018).

Assim conclui-se que de facto a introdução do *Blockchain* em redes de organizações de ajuda humanitária permite benefícios pela transparência.

5.2 Eficiência

Também na solução implementada para a ajuda humanitária foram encontrados benefícios de eficiência relacionados com o facto do *Blockchain* passar a ser um sistema centralizador para todos os atores humanitários envolvidos, deixando cada um individualmente interagir exclusivamente com o seu sistema proprietário. O Blockchain incentiva também a uma maior colaboração entre os atores humanitários, garantindo a gestão eficiente entre a oferta e a procura, a transacionalidade imediata na transferência dos bens, garantindo que os mesmos possam ser direccionados eficazmente para onde são realmente necessários num espaço de tempo mais reduzido.

Adicionalmente verificou-se que a utilização de *Smart Contracts* permite uma automação das regras e lógica de negócio permitindo que a execução das transações decorra de forma automática entre os atores na transferência dos bens. Esta automação trás por si só ganhos de eficiência na linearização dos processos com impacto directo na redução do tempo de duração dos mesmos.

No caso de uso de utilização definido na secção 4.2 verificou-se a aplicação automática de regras de negócio, que na actualização de proprietários, localização ou quantidades disponíveis. Este é um tipo de standardização que passa a ser uniforme em toda a rede, que para além de mais eficiência acresce também maior justiça e equidade no processo. Adicionalmente também maior eficácia sempre que as regras de negócio nos *Smart Contracts* permitam que a ajuda chegue mais cedo e seja melhor direccionada para os beneficiários em maior necessidade.

Os ganhos em eficiência, podem ser traduzidos em ganhos em tempo ou monetários, por economia de custos, ambos os ganhos podem ser redireccionados em prol dos beneficiários.

- Tempo - Passa a haver maior disponibilidade das pessoas para se focarem em atividades de valor acrescentado, em vez de atividades burocráticas, que na angariação de donativos, junto dos doadores, que na ajuda directa às populações carenciadas.
- Monetários – Passa a haver mais disponibilidade monetária para introduzir melhorias nos processos, com ganhos adicionais de eficiência. Ou pode aumentar o raio de acção da intervenção para chegar a mais pessoas.

Também da revisão da literatura se pode constatar alguns ganhos de eficiência, nomeadamente no projeto da Dorcas, em que foi possível ter ganhos nos custos com entidades terceiras, quer nas comissões das transferências monetárias e quer nas taxas do câmbio. No caso da Trocáire conseguiram reduzir o tempo total para fazer chegar os fundos ao destino, com 15% de ganhos em eficiência no *Blockchain* face ao canal tradicional no sistema bancário.

Assim conclui-se que de facto a introdução do *Blockchain* em redes de organizações de ajuda humanitária permite benefícios pela eficiência.

5.3 Outras considerações

A plataforma *HyperLedger* revelou-se de fácil instalação, os seus módulos *HyperLedger Composer* e *HyeperLedger Explorer* permitiram níveis de eficiência muito consideráveis no desenho e implementação da solução. A plataforma *HyperLedger* demonstrou estar madura e consolidada, disponibilizando bastante documentação e tutoriais de ajuda na iniciação.

Apesar do ambiente de desenvolvimento desta solução, não ter tido uma infra-estrutura dedicada, a sua execução numa máquina virtual com apenas 1 CPU e 8 GB de RAM acabou por não trazer limitações impeditivas à realização do projeto.

Pela revisão da literatura verificou-se que nos projetos-piloto executados têm tirado partido do facto da plataforma permitir níveis de interoperabilidade muito bons através de APIs REST, há registos de casos de integração com sistemas biométricos de identificação em POS, integração com sistemas de pagamento móveis e sistemas de *mobile Money*.

Ainda assim é um sistema desenhado e construído para a internet, requer alguma cautela na escolha dos casos de uso de implementação pois algumas das zonas mais carenciadas têm grandes constrangimentos em termos de conectividade.

6 Bibliografia

Accenture. (2017). *Blockchain for Good - 4 Guidelines for Transforming Social Innovation Organizations.* Accenture Labs. Bangalore: Accenture LAbs. Obtido de https://www.accenture.com/t20180102T200432Z__w__/us-en/_acnmedia/PDF-68/Accenture-808045-BlockchainPOV-RGB.pdf#zoom=50

ALNAP. (2018). *The state of the humanitarian system.* London: ALNAP. Obtido de https://www.alnap.org/system/files/content/resource/files/main/SOHS%202018%20Summary%20online_2.pdf

BETTS, A., & BLOOM, L. (2014). *Humanitarian Innovation: State of the Art.* United Nations Office for the Coordination of Humanitarian Affairs, Policy Development and Studies Branch (PDSB). New York: OCHA. Obtido de https://www.unocha.org/sites/unocha/files/Humanitarian%20Innovation%20The%20State%20of%20the%20Art_0.pdf

Bond. (2015). *uk public attitudes towards development.* https://www.bond.org.uk/. London: Bond. Obtido de https://www.bond.org.uk/sites/default/files/resource-documents/uk_public_attitudes_towards_development.pdf

Chrysochou, N. (27 de Março de 2019). *Some successes and challenges during humanitarian aid Blockchain pilot.* Obtido de Medium: https://medium.com/frontier-technology-livestreaming/some-successes-and-challenges-during-humanitarian-aid-Blockchain-pilot-a481115b3733

chsalliance. (12 de December de 2015). *Evidence on corruption and humanitarian aid*. Obtido de https://www.chsalliance.org: https://www.chsalliance.org/news/blog/evidence-on-corruption-and-humanitarian-aid

Coppi, G., & Fast, L. (2019). *Blockchain and distributed.* Overseas Development Institute, Humanitarian Policy group. London: Overseas Development Institute. Obtido de https://www.odi.org/sites/odi.org.uk/files/resource-documents/12605.pdf

Disberse. (2019). *About*. Obtido de Disberse: https://disberse.com/

DORCAS. (14 de Fevereiro de 2018). *Sucessful test Blockchain*. Obtido de Dorcas: https://www.dorcas.org/succesful-test-Blockchain/

Humanitarian Innovation Project. (2015). Principles for Ethical Humanitarian Innovation. Em U. o. Oxford (Ed.), *World Humanitarian Summit* (pp. 3-4). Oxford: Humanitarian Innovation Project. Obtido de https://europa.eu/capacity4dev/innov-aid/document/principles-ethical-humanitarian-innovation-draft-principles-based-joint-hip-whs-oxford-work

ICHA. (2018). *Blockchain technology in humanitarian programming.* International Center for Humanitarian Affairs. Nairobi: International Center for Humanitarian Affairs. Obtido

de https://www.cash-hub.org/-/media/cashhub-documents/resources/2018/Blockchain-technology_pilot-project-in-kenya_2018.pdf

Inter-Agency standing committee. (Maio de 2016). *The Grand Bargain*. Obtido de Inter-Agency Standing Committee (IASC): https://interagencystandingcommittee.org/grand-bargain-0

Kenny, C. (23 de Janeiro de 2017). *How Much Aid is Really Lost to Corruption?* Obtido de Center of Global Development: https://www.cgdev.org/blog/how-much-aid-really-lost-corruption

KO, V., & VERITY, A. (2016). *BLOCKCHAIN FOR THE HUMANITARIAN SECTOR: FUTURE OPPORTUNITIES.* UN Office for the Coordination of Humanitarian Affairs (OCHA), Digital Humanitarian Network. New York: Digital Humanitarian Network. Obtido de http://digitalhumanitarians.com/resource/Blockchain-humanitarian-sector-future-opportunities

Parker, L. (25 de Fevereiro de 2017). Bitcoin charity platform Helperbit completes first case study. *BRAVE NEWCOIN*, 1. Obtido de https://bravenewcoin.com/insights/helperbit-bitcoin-charity-platform-completes-first-case-study

Peffers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., Virtanen, V., & Bragge, J. (2006). *THE DESIGN SCIENCE RESEARCH PROCESS: A MODEL FOR PRODUCING AND PRESENTING INFORMATION SYSTEMS RESEARCH.* Obtido de https://pdfs.semanticscholar.org/e1fa/ec8846289113fdeb840ff3f32d102e46fbff.pdf

Start Network. (11 de Julho de 17). *Blockchain Experiment humanitarian Aid*. Obtido de Start Network: https://startnetwork.org/news-and-blogs/Blockchain-experiment-humanitarian-aid

Start Network. (25 de Outubro de 2018). *Blockchain pilot II- summary of lessons.* Obtido de https://startnetwork.org: https://startnetwork.org/resource/Blockchain-pilot-ii-summary-lessons

Transparency International. (2015). *ADDRESSING CORRUPTION RISK IN THE EEA AND NORWAY GRANTS.* Berlin: Transparency.org. Obtido de https://www.transparency.org/whatwedo/publication/addressing_corruption_risk_in_the_eea_and_norway_grants

United Nations. (2012). *Secretary-General's closing remarks at High-Level Panel on Accountability, Transparency and Sustainable Development.* New York: United Nations Secretary General. Obtido de https://www.un.org/sg/en/content/sg/statement/2012-07-09/secretary-generals-closing-remarks-high-level-panel-accountability

United Nations. (s.d.). *Deliver Humanitarian Aid*. Obtido de United Nations: https://www.un.org/en/sections/what-we-do/deliver-humanitarian-aid/

Verhulst, s. (19 de Dezembro de 2018). *Seven design principles for using Blockchain for social impact.* Obtido de https://apolitical.co: https://apolitical.co/solution_article/design-principles-Blockchain-for-social-impact/

Wikipedia. (s.d.). *Business network.* Obtido de Wikipedia: https://en.wikipedia.org/wiki/Business_network

World Bank. (1995). *Working with NGOs.* World Bank, Operations Policy Department. World Bank. Obtido de http://documents.worldbank.org/curated/en/814581468739240860/pdf/multi-page.pdf

World Food Programme. (2019). *Building Blocks*. Obtido de WFP Innovation Accelerator : https://innovation.wfp.org/project/building-blocks

World Vision. (2019). *A Digital Asset Transfer Platform designed for financially marginalized.* Obtido de Sikka: https://www.sikka.me/docs/SikkaConceptPaper.pdf

Anexos

Modelo - “aid.cto”

```
/**
 * Defines a data model for aid exchange
 */
namespace org.jan.aid.exchange

/**
 * The types of goods that could be exchanged
 */
enum GoodType {
  o FOOD
  o CLOTH
  o DRUG
  o CONSTRUCTION_MATERIAL
  o VOUCHER
}

enum ListingState {
  o IN_NEED
  o IN_TRANSIT
  o IN_WAREHOUSE
  o CLOSED
}

/**
 * The movement status for an animal
 */
enum MovementStatus {
  o IN_WAREHOUSE
  o IN_TRANSIT
  o ASSIGNED
}

/**
 * Basic class to define a participant
 */
abstract participant User identified by vatNumber {
  o String vatNumber
  o String firstName
  o String lastName
  o String email
}

/**
 * A Member participant
```

```
 */
participant Member extends User {
    o String address1
    o String address2
    o String county
    o String postcode
    --> Location location optional
}

/**
 * A Regulator participant
 */
participant Regulator extends User {
}
/**
 * A Location asset which is owned by a Member, is related
 * to a list of warehouses and has a list of incoming goods.
 */
asset Location identified by lid {
     o String lid
     o String address1
     o String address2
     o String county
     o String postcode
     --> Member owner
     --> Good[] incomingGoods optional
}

/**
 * An Good asset, which can be stored in a Wharehouse
 */
asset Good identified by id {
  o String id
  o String description
  o GoodType type
  o Integer amount
  o MovementStatus movementStatus
  --> Location location optional
  --> Member owner
}
/**
 * Listing of goods that are in need
 */
asset GoodListing identified by listingId {
  o String listingId
  o String description
  o Integer quantity
  o ListingState state
```

```
  o Donation[] donation optional
  --> Member owner
}

/**
 * An abstract transaction type for Good movements
 */
abstract transaction GoodMovement {
  o String[] logs optional
    --> Good good
    --> Location from
    --> Location to
}

/**
 * A transaction type for an Good leaving a location
 */
transaction GoodShipping extends GoodMovement {
  --> Location fromLocation
}

/**
 * A transaction type for an Good arriving at a location
 */
transaction GoodArrival extends GoodMovement {
  --> Location arrivalLocation
}

/**
 * Member makes a Good donation to a specific Listing
 */
transaction Donation {
  --> Member member
  --> Good good
  --> GoodListing listing
}

/**
 * Listing owner closed the listing
 */
transaction CloseDonation {
  --> GoodListing listing
}

/**
 * Member assign a Listing to beneficiary
 */
transaction AssignListing {
```

```
    --> GoodListing listing
    --> Member beneficiary
}
/**
 * Test Transcation to setup the network
 */
transaction SetupDemo  {
}
```

Transações - “Lojic.js”

```
'use strict';

/* global getAssetRegistry getParticipantRegistry getFactory */

/**
 *
 * @param {org.jan.aid.exchange.GoodShipping} movementDeparture - model instance
 * @transaction
 */
async function onGoodMovementDeparture(movementDeparture) {  // eslint-disable-line no-unused-vars
    console.log('onGoodMovementDeparture');
    if (movementDeparture.good.movementStatus !== 'IN_WAREHOUSE') {
        throw new Error('Good is already IN_TRANSIT');
    }

     // set the movement status of the Good
    movementDeparture.good.movementStatus = 'IN_TRANSIT';

     // save the Good
    const ar = await getAssetRegistry('org.jan.aid.exchange.Good');
    await ar.update(movementDeparture.good);

    // add the Good to the incoming Good of the
    // destination Location
    if (movementDeparture.to.incomingGoods) {
        movementDeparture.to.incomingGoods.push(movementDeparture.good);
    } else {
        movementDeparture.to.incomingGoods = [movementDeparture.good];
    }

    // save the Location
    const br = await getAssetRegistry('org.jan.aid.exchange.Location');
    await br.update(movementDeparture.to);
}

/**
 *
 * @param {org.jan.aid.exchange.GoodArrival} movementArrival - model instance
 * @transaction
 */
async function onGoodMovementArrival(movementArrival) {  // eslint-disable-line no-unused-vars
    console.log('onGoodMovementArrival');

    if (movementArrival.good.movementStatus !== 'IN_TRANSIT') {
        throw new Error('Good is not IN_TRANSIT');
    }
```

```
     // set the movement status of the Good
    movementArrival.good.movementStatus = 'IN_WAREHOUSE';

     // set the new owner of the Good
     // to the owner of the 'to' Location
    movementArrival.good.owner = movementArrival.to.owner;

     // set the new location of the Good
    movementArrival.good.location = movementArrival.arrivalLocation;

     // save the good
    const ar = await getAssetRegistry('org.jan.aid.exchange.Good');
    await ar.update(movementArrival.good);

    // remove the Good from the incoming Goods
    // of the 'to' Location
    if (!movementArrival.to.incomingGoods) {
        throw new Error('Incoming Location should have incomingGoods on GoodArrival.');
    }

    movementArrival.to.incomingGoods = movementArrival.to.incomingGoods
      .filter(function(good) {
          return good.id !== movementArrival.good.id;
      });

    // save the Location
    const br = await getAssetRegistry('org.jan.aid.exchange.Location');
    await br.update(movementArrival.to);
}

/**
 * Make an donation for a GoodListing
 * @param {org.jan.aid.exchange.Donation} donation - the donation
 * @transaction
 */
async function makeDonation(donation) {  // eslint-disable-line no-unused-vars
    let listing = donation.listing;
    if (listing.state !== 'IN_NEED') {
        throw new Error('Listing is not IN_NEED');
    }
 /*   if (listing.member !== donation.good.owner) {
        throw new Error('Not the owner!!!');
    }
 */   if (listing.quantity >= donation.good.amount) {
        listing.quantity -=  donation.good.amount;

        donation.good.owner = listing.owner;
```

```
        donation.good.location= listing.owner.location;

    } else {
        throw new Error('Good amount hihger than needed');
    }
    if (!listing.donation) {
        listing.donation = [];
    }
    listing.donation.push(donation);

    // save the Good listing
    const goodListingRegistry = await getAssetRegistry('org.jan.aid.exchange.GoodListing');
    await goodListingRegistry.update(listing);

    // save the Goods
    const goodRegistry = await getAssetRegistry('org.jan.aid.exchange.Good');
    await goodRegistry.update(donation.good);
}

/**
 * Make an assing for a GoodListing
 * @param {org.jan.aid.exchange.AssignListing} assignment - the assignment
 * @transaction
 */
async function makeAssignListing(assignment) {  // eslint-disable-line no-unused-vars
    const listing = assignment.listing;
    const listofGoods = [];
    if (listing.state !== 'CLOSED') {
        throw new Error('Listing is not CLOSED');
    }
    listing.donation.forEach(function(donation) {
        donation.good.owner = assignment.beneficiary;
        donation.good.location = assignment.beneficiary.location;
        donation.good.movementStatus = 'ASSIGNED';
        listofGoods.push(donation.good);
    });
    // save the Goods
    const goodRegistry = await getAssetRegistry('org.jan.aid.exchange.Good');
    await goodRegistry.updateAll(listofGoods);
}

/**
 *
 * @param {org.jan.aid.exchange.SetupDemo} setupDemo - SetupDemo instance
 * @transaction
 */
async function setupDemo(setupDemo) {  // eslint-disable-line no-unused-vars
    const factory = getFactory();
```

```
const NS = 'org.jan.aid.exchange';

const members = [
    factory.newResource(NS, 'Member', 'DONOR_1'),
    factory.newResource(NS, 'Member', 'DONOR_2'),
    factory.newResource(NS, 'Member', 'DONOR_3'),
    factory.newResource(NS, 'Member', 'DONOR_4'),
    factory.newResource(NS, 'Member', 'LNGO_5'),
    factory.newResource(NS, 'Member', 'INGO_6'),
    factory.newResource(NS, 'Member', 'BENEFICIARY_7'),
    factory.newResource(NS, 'Member', 'BENEFICIARY_8')
];

const locations = [
    factory.newResource(NS, 'Location','LOCATION_1'),
    factory.newResource(NS, 'Location','LOCATION_2'),
    factory.newResource(NS, 'Location','LOCATION_3'),
    factory.newResource(NS, 'Location','LOCATION_4'),
    factory.newResource(NS, 'Location','LOCATION_5'),
    factory.newResource(NS, 'Location','LOCATION_6'),
    factory.newResource(NS, 'Location','LOCATION_7'),
    factory.newResource(NS, 'Location','LOCATION_8')
];

const goods = [
    factory.newResource(NS, 'Good', 'GOOD_1'),
    factory.newResource(NS, 'Good', 'GOOD_2'),
    factory.newResource(NS, 'Good', 'GOOD_3'),
    factory.newResource(NS, 'Good', 'GOOD_4'),
    factory.newResource(NS, 'Good', 'GOOD_5'),
    factory.newResource(NS, 'Good', 'GOOD_6'),
    factory.newResource(NS, 'Good', 'GOOD_7'),
    factory.newResource(NS, 'Good', 'GOOD_8')
];

const regulator = factory.newResource(NS, 'Regulator', 'REGULATOR');
regulator.email = 'REGULATOR';
regulator.firstName = 'Ronnie';
regulator.lastName = 'Regulator';
const regulatorRegistry = await getParticipantRegistry(NS + '.Regulator');
await regulatorRegistry.addAll([regulator]);

members.forEach(function(member) {
    const lid = 'LOCATION_' + member.getIdentifier().split('_')[1];
    member.firstName = member.getIdentifier();
    member.lastName = '';
    member.email = "email@acme.com"
    member.address1 = 'Address1';
```

```
        member.address2 = 'Address2';
        member.county = 'County';
        member.postcode = 'PO57C0D3';
        member.location = factory.newResource(NS, 'Location', lid);
    });
    const memberRegistry = await getParticipantRegistry(NS + '.Member');
    await memberRegistry.addAll(members);

    locations.forEach(function(location, index) {
        const member = 'MEMBER_' + (index + 1);
        location.address1 = 'Address1';
        location.address2 = 'Address2';
        location.county = 'County';
        location.postcode = 'PO57C0D3';
        location.owner = factory.newRelationship(NS, 'Member', member);
    });
    const locationRegistry = await getAssetRegistry(NS + '.Location');
    await locationRegistry.addAll(locations);

    goods.forEach(function(good, index) {
        const location = 'LOCATION_' + ((index % 2) + 1);
        const member = 'MEMBER_' + ((index % 2) + 1);
        good.type = 'FOOD';
        good.description = '';
        good.amount = 1;
        good.movementStatus = 'IN_WAREHOUSE';
        good.location = factory.newRelationship(NS, 'Location', location);
        good.owner = factory.newRelationship(NS, 'Member', member);
    });
    const goodRegistry = await getAssetRegistry(NS + '.Good');
    await goodRegistry.addAll(goods);
}
```

Controlo de Acessos – "Permissions.acl"

```
/**
 * Access Control List for the Aid Exchange network.
 */
rule Regulator {
    description: "Allow the Regulator full access"
    participant: "org.jan.aid.exchange.Regulator"
    operation: ALL
    resource: "org.jan.aid.exchange.*"
    action: ALLOW
}
rule Member {
    description: "Allow the member Full access only when is the owner"
    participant: "org.jan.aid.exchange.Member"
    operation: READ
    resource: "org.jan.aid.exchange.*"
    action: ALLOW
 }
rule GoodOwner {
    description: "Allow the owner of a Good total access"
    participant(m): "org.jan.aid.exchange.Member"
    operation: ALL
    resource(g): "org.jan.aid.exchange.Good"
    condition: (g.owner.getIdentifier() == m.getIdentifier())
    action: ALLOW
}
rule GoodListingOwner {
    description: "Allow the owner of a good total access to their goods listing"
    participant(m): "org.jan.aid.exchange.Member"
    operation: ALL
    resource(gl): "org.jan.aid.exchange.GoodListing"
    condition: (gl.good.owner.getIdentifier() == m.getIdentifier())
    action: ALLOW
}
rule SystemACL {
    description:  "System ACL to permit all access"
    participant: "org.hyperledger.composer.system.Participant"
    operation: ALL
    resource: "org.hyperledger.composer.system.**"
    action: ALLOW
}
rule NetworkAdminUser {
    description: "Grant business network administrators full access to user resources"
    participant: "org.hyperledger.composer.system.NetworkAdmin"
    operation: ALL
    resource: "**"
    action: ALLOW
```

```
}
rule NetworkAdminSystem {
    description: "Grant business network administrators full access to system resources"
    participant: "org.hyperledger.composer.system.NetworkAdmin"
    operation: ALL
    resource: "org.hyperledger.composer.system.**"
    action: ALLOW
}
```