Tópicos Especiais em Sistemas de Informação

Minicursos do XVII Simpósio Brasileiro em Sistemas de Informação (SBSI 2021)

ORGANIZADORES:

Davi Viana (UFMA)
Thiago P. Ribeiro (UFU)
Rafael D. Araújo (UFU)
Fabiano A. Dorça (UFU)

Realização

UFU

CESI
COMISSÃO ESPECIAL DE SISTEMAS DE INFORMAÇÃO

Promoção

SBC
Sociedade Brasileira de Computação

Organizadores
Davi Viana
Thiago P. Ribeiro
Rafael D. Araújo
Fabiano A. Dorça

TÓPICOS EM SISTEMAS DE INFORMAÇÃO
Minicursos SBSI 2021

Sociedade Brasileira da Computação
Porto Alegre
2021

TÓPICOS EM SISTEMAS DE INFORMAÇÃO
Minicursos SBSI 2021

Sociedade Brasileira de Computação –SBC
CNPJ: 29.532.264/0001-78

Coordenação Geral
Rafael D. Araújo
Fabiano A. Dorça

Coordenação do Comitê de Programa - Minicursos
Davi Viana
Thiago P. Ribeiro

Edição dos Anais
Awdren Fontão

Realização

Organização

Em cooperação

Apoiadores Institucionais

Patrocinadores

Editores

Davi Viana
Thiago P. Ribeiro
Awdren Fontão
Renata Araujo
Sean Siqueira
Rafael D. Araújo
Fabiano A. Dorça

Comitê técnico

Coordenação Geral
Rafael D. Araújo (UFU)
Fabiano A. Dorça (UFU)

Coordenação do Comitê de Programa
Renata Araujo (UPM)
Sean Siqueira (UNIRIO)

Coordenação dos Anais
Awdren Fontão (UFMS)

Coordenação do Comitê - Minicursos
Davi Viana (UFMA)
Thiago P. Ribeiro (UFU)

Comissão Especial de Sistemas de Informação da SBC (CESI)

Coordenador Geral
Rodrigo Pereira dos Santos (UNIRIO)

Vice-coordenador
Davi Viana (UFMA)

Comitê Gestor
André Pimenta Freire (UFLA)
Andrea M. Magdaleno (UFF)
Fabio Gomes Rocha (UNIT)
Flávio E. Aoki Horita (UFABC)
Leonardo Guerreiro Azevedo (IBM)
Luis J. E. Rivero Cabrejos (UFMA)
Marcelo Fornazin (UFF)
Rafael Dias Araujo (UFU)
Renata Araujo (UPM)
Rita S. Pitangueira Maciel (UFBA)
Sean W. Matsui Siqueira (UNIRIO)
Scheila de Ávila e Silva (UCS)

Comitê de Programa – Minicursos

Alex Borges - UFJF
Alexandre Cidral - UNIVILLE
André de Oliveira - UFJF
Andre Martinotto - UCS
Awdren Fontão - UFMS
Carla Merkle Westphall - UFSC
Carlos Eduardo Santos Pires - UFCG
Carlos Eduardo de Barros Paes - PUC-SP
Daniel Notari - UCS
Edmundo Spoto - UFG
Eliomar Lima - UFG
Emanuel Coutinho - UFC
Emilio Francesquini - UFABC
Fernanda Baião - PUC-Rio
Fernanda Santos - UFU
Flávio Soares Corrêa da Silva - USP
Heitor Costa - UFLA
Jorge Barbosa - Unisinos
Jorge Júnior - UFPB
José Maria David - UFJF
Juliano Lopes de Oliveira - UFG
Leonardo Azevedo - IBM Research - Brazil
Leticia Peres - UFPR
Marcos Chaim - USP
Maria Istela Cagnin - UFMS
Morganna Diniz - UNIRIO
Paulo Sérgio Santos - UNIRIO
Renata Araujo - UPM
Ricardo Choren - IME / RJ
Rita Suzana Pitangueira Maciel - UFBA
Rodrigo Miani - UFU
Rodrigo Santos - UNIRIO
Scheila de Avila e Silva - UCS
Valdemar Vicente Graciano Neto - UFG

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

S612b Simpósio Brasileiro em Sistemas de Informação (SBSI 2021) (17. : 2021 : Porto Alegre, RS)
Anais de Tópicos em Sistemas de Informação: minicursos SBSI 2021 [recurso eletrônico] / organizadores: Davi Viana, Thiago P. Ribeiro, Rafael D. Araújo, Fabiano A. Dorça -- Porto Alegre : Sociedade Brasileira da Computação, 2021.
62 p. : il.

ISBN: 978-65-87003-53-5
Modo de acesso: Internet.
Disponível em: https://sol.sbc.org.br/livros
Inclui bibliografia.

1. Computação 2. Blockchain 3. Ética da pesquisa 4. Sistemas inteligentes de controle I. Santos, Davi Viana dos, 1988-, (Org.). II. Ribeiro, Thiago Pirola, 1978-, (Org.). III. Araújo, Rafael Dias, 1986-, (Org.). VI. Dorça, Fabiano Azevedo, 1979-, (Org.). V. Universidade Federal de Uberlândia. VI. Sociedade Brasileira de Computação. VII. Título.

CDU: 681.51

Rejâne Maria da Silva - CRB6/1925
Bibliotecária

Prefácio Minicursos SBSI 2021

Este livro reúne trabalhos apresentados nos minicursos ministrados no XVII Simpósio Brasileiro de Sistemas de Informação (SBSI 2021), realizado online e organizado pela Universidade Federal de Uberlândia, no período de 07 a 10 de junho de 2021. Participam do SBSI, fórum nacional de debates da área de Sistemas de Informação (SI), estudantes e pesquisadores com apresentação de trabalhos científicos e discussão de temas contemporâneos relacionados à área.

Neste ano, foram submetidas 7 (sete) propostas de minicursos válidas e 2 duas foram selecionadas, tal qual decidido na última reunião da Comissão Especial de Sistemas de Informação. Tais propostas foram avaliadas por, no mínimo, três pesquisadores que fazem parte de um comitê composto por 34 professores pesquisadores.

Os dois minicursos contidos neste livro abordam tópicos de interesse da comunidade de Sistemas de Informação e correlatos ao tema da 17a. Edição do evento, intitulado “Sistemas de Informação Inteligentes e Onipresentes: novos desafios e oportunidades”. O primeiro capítulo, “Blockchain, Contratos Inteligentes, Sistemas Web: Teoria e Prática”, aborda os conceitos fundamentais, mecanismos e as plataformas utilizadas e que estão envolvidas na implementação de aplicações Web confiáveis e distribuídas. O segundo capítulo, “Ética da Pesquisa em Sistemas de Informação: Por que e como submeter meu projeto ao Comitê de Ética”, discute os aspectos técnicos de pesquisas envolvendo seres humanos no contexto de Sistemas de Informação. Inclusive, são apresentados os trâmites necessários para a submissão e aprovação de um projeto de pesquisa envolvendo seres humanos.

Esperamos que este livro auxilie estudantes, pesquisadores e profissionais da área de Sistemas de Informação na construção do conhecimento em temas específicos relacionados ao que foi aqui apresentado.

Davi Viana (UFMA) e Thiago Pirola Ribeiro (UFU)
Coordenadores da Trilha de Minicursos do SBSI 2021
Uberlândia/MG, Junho de 2021.

Coordenadores dos Minicursos do SBSI 2021

Davi Viana é Graduado em Ciência da Computação pela Universidade Federal do Amazonas (UFAM), Mestre e Doutor em Informática pelo Programa de Pós-Graduação em Informática da UFAM. Durante o doutorado, esteve em período sanduíche na *Blekinge Institute of Technology* (BTH, Suécia). Possui curso técnico em informática pela Fundação Nokia de Ensino. Atualmente é Professor Adjunto da Universidade Federal do Maranhão (UFMA). Além disso, é membro permanente do Programa de Pós-Graduação em Ciência da Computação (PPGCC) da UFMA e do Programa de Doutorado em Ciência da Computação Associação UFMA/UFPI (DCCMAPI). Também atua como Analista de TI na Diretoria de Tecnologias na Educação (DTED/UFMA) e Universidade Aberta do Sistema Único de Saúde (UNA-SUS UFMA). É vice coordenador da Comissão Especial de Sistemas de Informação (CESI) da SBC e Secretário Regional da SBC no Maranhão. Tem interesse nas áreas de sistemas de informação, qualidade de software e sistemas, melhoria processo de software (MPS), implementação de programas de MPS com ênfase na adoção de modelos de maturidade, educação em sistemas de informação e engenharia de software, metodologias experimentais em sistemas de informação e engenharia de software. Por fim, também atua na aplicação de metodologias de desenvolvimento de sistemas de informação e engenharia de software no contexto de: internet das coisas; startup de software; e computação aplicada à saúde.

Thiago P. Ribeiro é Graduado em Análise de Sistema pela Universidade Metodista de Piracicaba - UNIMEP, Mestrado em Ciência da Computação pela Universidade Federal de São Carlos - UFSCar e Doutorado em Ciência da Computação pela Universidade Federal de Uberlândia - UFU (2019). Atualmente é professor adjunto na Faculdade de Computação da Universidade Federal de Uberlândia - Campus de Monte Carmelo. Pertence ao Banco de Avaliadores do Sistema Nacional de Avaliação da Educação Superior (BASis) do INEP/MEC. Tem interesse nas áreas de sistemas de informação, processamento digital de imagens, software livre, educação em sistemas de informação, computação aplicada à saúde, dispositivos móveis e internet das coisas.

Organizadores Gerais do SBSI 2021

Rafael D. Araújo é Bacharel, Mestre e Doutor em Ciência da Computação pela Universidade Federal de Uberlândia. Realizou um estágio na área de desenvolvimento de software na France Telecom durante um período de intercâmbio acadêmico entre o *Institut National des Sciences Appliquées* de Lyon (INSA, França) e a UFU. Realizou um período de estágio de doutorado sanduíche durante um ano no laboratório de Sistemas Web Adaptativos e Personalizados (PAWS) na *School of Information Sciences* (iSchool) da Universidade de Pittsburgh (USA). Também trabalhou em empresas de desenvolvimento de software e possui experiência com arquitetura de soluções Web e distribuídas. Atualmente é professor adjunto da Faculdade de Computação (FACOM/UFU) - Campus Monte Carmelo e professor do Programa Pós-Graduação em Ciência da Computação (PPGCO/UFU). É editor associado da Revista Brasileira de Informática na Educação (RBIE), foi membro do Comitê Gestor da Comissão Especial da Informática na Educação (CEIE) e, atualmente, é membro do Comitê Gestor da Comissão Especial de Sistemas de Informação (CESI) da Sociedade Brasileira de Computação (SBC) e do Comitê Diretivo do Simpósio Brasileiro de Sistemas de Informação. É pesquisador nas áreas de Recomendação e Personalização de Conteúdo, Computação Ubíqua, Interação Humano-Computador, Sistemas Web e Multimídia Interativos, Informática na Educação e Sistemas de Informação.

Fabiano A. Dorça Possui graduação em Ciência da Computação pela Universidade Federal de Uberlândia - UFU (2000), mestrado (2004) e doutorado (2012) pela mesma instituição. Professor da Faculdade de Computação (FACOM/UFU), ministra disciplinas nos cursos de graduação em Ciência da Computação, Sistemas de Informação e Gestão da Informação, e atua como membro permanente do Programa de Pós-graduação em Ciência da Computação da FACOM/UFU. Pesquisador nas áreas de Inteligência Artificial e Engenharia de Software, com ênfase em sistemas tutores inteligentes, sistemas adaptativos para educação, aprendizagem de máquina, ontologias, web semântica, design smells e simulação. Participou da coordenação geral do “V Congresso Brasileiro de Informática na Educação (2016)” e foi membro da Comissão Especial de Informática na Educação (CEIE) da Sociedade Brasileira de Computação (SBC). Participou do comitê de programa de diversas conferências (inclusive como coordenador geral) e é revisor de vários periódicos nacionais e internacionais, como Expert Systems with Applications, IEEE Transactions on Education, IEEE Transactions on Emerging Topics in Computing, IEEE Transactions on Learning Technologies, Smart Learning Environments, International Journal of Learning Technology, Revista Brasileira de Informática na Educação, dentre outros.

Sumário

Capítulo 1

Blockchain, Contratos Inteligentes, Sistemas Web: Teoria e Prática

Jauberth Abijaude, Henrique Serra, Levy Santiago,
Péricles Sobreira, Fabíola Greve

Abstract

Blockchain is a disruptive technology that offers a digital trust network for carrying out transactions between peers. Smart contracts are codes hosted on the blockchain and establish contractual clauses to be followed. In this mini-course, we present the fundamental concepts, mechanisms, and platforms used by these technologies, to develop in their participants the needs for implementing required and distributed Web applications (DApps). For this, we will demonstrate, through practical exercises, a model for implementing smart contracts interacting with Web systems, in addition to discussing the advances, opportunities, and challenges in research related to this area of knowledge.

Resumo

A blockchain é uma tecnologia disruptiva que oferece uma rede de confiança digital para a realização de transações entre pares. Os contratos inteligentes são códigos hospedados na blockchain e estabelecem cláusulas contratuais a serem seguidas. Neste minicurso, apresentamos os conceitos fundamentais, os mecanismos e as plataformas utilizadas por estas tecnologias, de forma a desenvolver em seus participantes as competências necessárias à implementação de aplicações Web confiáveis e distribuídas (DApps). Para isto, demonstraremos, através de exercícios práticos, um modelo de implementação de contratos inteligentes interagindo com sistemas Web, além de discutirmos os avanços, as oportunidades e os desafios em pesquisas relacionadas a esta área do conhecimento.

1.1. Introdução

Este capítulo tem como objetivo desenvolver competências na criação de sistemas web integrados com blockchain e contratos inteligentes, com foco na plataforma *Ethereum*, para formar estudantes da área de Sistemas de Informação, Análise e Desenvolvimento

de Sistemas, Engenharia de Software, Ciência da Computação e afins, principalmente pela existência de uma crescente demanda de profissionais, aliado ao fato destes assuntos não serem abordados nos currículos universitários atuais.

Será apresentada a Blockchain (BC), os Contratos Inteligentes (CI) e sua integração com sistemas Web. A abordagem é dividida em seis seções: a primeira e segunda, teóricas, abordam histórico, conceitos fundamentais e apresentação da plataforma *Ethereum*. A terceira, apresenta a linguagem de programação *Solidity* e as características fundamentais para o desenvolvimento de CIs na plataforma *Ethereum*, introduzindo a primeira prática do curso. A quarta seção, teórica e prática, explora a integração com sistemas web e o desenvolvimento de uma aplicação distribuída (DApp). A quinta, discute como um professor pode abordar este assunto em uma sala de aula, relatando a experiência dos autores. Por fim, a última seção apresenta os desafios e perspectivas na área.

1.1.1. Histórico e Conceitos Fundamentais

Nick Szabo, em 1997, apresenta os CIs como uma combinação de protocolos com interfaces de usuário para formalizar e proteger relacionamentos em redes de computadores [Szabo 1997]. Esta foi a primeira vez que este termo foi usado, porém a ideia permaneceu adormecida por não encontrar um sistema computacional que oferecesse os requisitos mínimos necessários para a sua implementação.

Uma década depois, em agosto de 2008, Satoshi Nakamoto descreve uma habilidosa combinação de técnicas para dar suporte à criptomoeda *bitcoin* [Nakamoto 2008]. Este protocolo descreve a rede (BC) e suas propriedades; em janeiro de 2009, ela entrou em operação. Esta BC original incorpora uma máquina de estados simplificada, com transações voltadas para a moeda *bitcoin*. Ela é composta por uma combinação de técnicas robustas provenientes da computação distribuída confiável (tolerância a falhas bizantinas, sistemas P2P), criptografia (chave assimétrica, funções *hash*, desafios criptográficos) e teoria dos jogos (mecanismos de incentivos) [Greve e outros 2018].

Vitalik Buterin, após observar e acompanhar as discussões da BC *Bitcoin*, em 2013, propôs uma nova plataforma - o *Ethereum*, que entrou em operação em julho de 2015 [Buterin e outros. 2014]. Esta nova proposta de BC estabeleceu uma nova criptomoeda, o *ether*, e retomou a ideia de Nick Szabo sobre os contratos inteligentes, permitindo transacionar a criptomoeda e códigos de programação mais avançados. Tais códigos são os CIs que passam a ser implementados na rede BC e permitem a execução de uma máquina de *Turing* completa, possibilitando as chamadas Aplicações Descentralizadas ou DApps [Antonopoulos e Wood 2018].

Atualmente, existem diversas plataformas de blockchain com forte investimento da indústria para desenvolvimento de aplicações robustas e descentralizadas em vários segmentos. Novas plataformas de BC como *Hyperledger Fabric* [Androulaki e outros 2018], *Nano* [LeMahieu 2018], *Iota* [Popov 2018], e diversas outras surgiram, cada uma com suas características particulares.

1.1.2. Propriedades da Blockchain

A BC possui propriedades que cooperam no desenvolvimento das DApps e permitem que estas as herdem, por conseguinte. Dentre elas pode-se destacar [Greve e outros 2018]:

- Descentralização: Sistemas e aplicações que usam a BC não precisam de uma entidade central para coordenar as ações, as tarefas são executas de forma distribuída;
- Disponibilidade e integridade: Os dados e as transações são replicados para todos os participantes da BC, mantendo o sistema seguro e consistente;
- Transparência e auditabilidade: A cadeia de blocos que registra as transações é pública e pode ser auditada e verificada;
- Imutabilidade e Irrefutabilidade: os registros são imutáveis e a correção só pode ser feita a partir de novos registros. O uso de recursos criptográficos garante que os lançamentos não podem ser refutados;
- Privacidade e Anonimidade: As transações são anônimas, com base nos endereços dos usuários. Os servidores armazenam apenas fragmentos criptografados dos dados do usuário;
- Desintermediação: A BC consegue eliminar terceiros em suas transações, atuando como um conector de sistemas de forma confiável e segura.
- Cooperação e incentivos: Uso do modelo de teoria dos jogos como forma de incentivo.

1.1.3. Componentes de uma Rede Blockchain

As redes BC podem ser públicas ou privadas [Greve e outros 2018]. A primeira é também conhecida como não permissionada ou de acesso aberto. A segunda é comumente chamada de permissionada ou federada. Nas BC públicas, o acesso é anônimo, não há nenhum controle sobre a entrada e saída de nós na rede e eles não possuem confiança mútua. Como exemplos destas redes, tem-se a *Bitcoin* e a *Ethereum*. Já nas BC privadas ou federadas, os nós são conhecidos e precisam ser autenticados. São redes voltadas normalmente para ambientes corporativos, onde cada participante tem um papel definido. A BC *Hyperledger Fabric* é um exemplo de BC privada.

Os componentes essenciais de uma BC pública ou privada são [Greve e outros 2018] [Antonopoulos e Wood 2018]:

- Uma rede ponto a ponto (P2P) conectando participantes e propagando transações e blocos de transações verificadas, com base em um protocolo padronizado;
- Mensagens, na forma de transações, representando transições de estado;
- Um conjunto de regras de consenso, definindo o que constitui uma transação e o que contribui para uma transição de estado válida;
- Uma máquina de estado que processa transações de acordo com as regras de consenso;
- Uma cadeia de blocos protegidos criptograficamente que atua como um diário de todas as transições de estado verificadas e aceitas;

- Um algoritmo de consenso que descentraliza o controle sobre a blockchain, forçando os participantes a cooperarem na aplicação das regras de consenso;

Em adição a esses componentes, as BC públicas apresentam, a partir de elementos de teoria dos jogos:

- Um esquema de incentivos teoricamente sólido para proteger economicamente a máquina de estados em um ambiente aberto, como por exemplo, os custos do algoritmo de consenso *Proof of Work*[Nakamoto 2008], em conjunto com custos de recompensas em bloco.

A grande totalidade destes componentes geralmente é combinada em um único cliente de software. Por exemplo, no Bitcoin, a implementação de referência é desenvolvida pelo projeto de código aberto *Bitcoin Core* e implementada como o cliente bitcoind. No Ethereum, em vez de uma implementação de referência, há uma especificação de referência, uma descrição matemática do sistema no *yellow paper* [Buterin e outros. 2014]. Existem vários clientes, que são construídos de acordo com a especificação de referência, dentre eles, o mais popular é o *Geth* [Go-ethereum 2013].

1.2. Plataforma Ethereum e Contratos Inteligentes

Nesta seção, discute-se a teoria da BC *Ethereum* e dos CIs. Os principais componentes da BC *Ethereum*, as carteiras, a estrutura das transações, redes disponíveis e outros detalhes serão discutidos.

1.2.1. Plataforma *Ethereum*

O acesso a plataforma *Ethereum* pode ser classificado sob dois aspectos: (a) Acesso para os desenvolvedores e (b) Acesso para os usuários. Em cada um deles há ferramentas e métodos diferentes.

Em (a), os desenvolvedores, comumente, usam a `web3`. Ela é uma coleção de bibliotecas que permitem a interação com um nó *Ethereum*, local ou remoto, usando HTTP, IPC ou *WebSocket* [Web3js 2016], com APIs(*Application Programming Interface*) disponíveis para Java, Android e Javascript.

Em (b), os usuários conectam-se à rede *Ethereum* usando um cliente remoto (um aplicativo de software que implementa a especificação *Ethereum* e se comunica pela rede ponto a ponto com outros clientes *Ethereum*). Estes clientes remotos oferecem um subconjunto da funcionalidade de um cliente completo. Eles não armazenam a blockchain *Ethereum* completa, são mais rápidos de configurar e requerem menos armazenamento de dados.

Geralmente, os clientes remotos permitem: (1) Gerenciar chaves privadas e endereços *Ethereum* em uma carteira; (2) Criar, assinar e transmitir transações; (3) Interagir com contratos inteligentes, usando a carga útil de dados; (4) Navegar e interagir com DApps; (5) Oferecer links para serviços externos, como exploradores de blocos; (6) Converter unidades de *ether* e recuperar taxas de câmbio de fontes externas; (7) Injetar uma instância `web3` no navegador web como um objeto JavaScript; (8) Usar uma instância

`web3` fornecida/injetada no navegador por outro cliente; e/ou (9) Acessar os serviços RPC em um nó *Ethereum* local ou remoto.

As carteiras móveis (*wallets*) são clientes remotos, já que os smartphones não têm recursos adequados para executar um cliente *Ethereum* completo. Os mais populares são *Jaxx* [Jaxx 2018], *Status* [Status 2019] , *Trust Wallet* [Trust 2019] e *Coinbase* [Coinbase 2018].

Os usuários podem também usar navegadores, onde as carteiras estão disponíveis como plugins ou extensões do Chrome ou Firefox, por exemplo. Estes clientes remotos são executados no navegador. Os mais populares são *Metamask* [Metamask 2018], *Jaxx*, *MyEtherWallet* [Myetherwallet 2019], *Nifty* e MyCrypto [MyCrypto 2019].

Usando uma destas carteiras é possível acessar a plataforma *Ethereum*, constituída da rede principal e da rede de testes. Na rede principal são transacionados *ethers* reais que precisam ser comprados e possuem valor financeiro. Na rede de testes, transacionam-se *ethers* fictícios, sem valor financeiro, adquiridos gratuitamente através de geradores de *ethers* na internet.

O *ether* é subdividido em unidades menores. A menor unidade possível é denominada *wei*. Um *ether* equivale a 1 quintilhão de *weis* ($1*10^{18}$ ou 1.000.000.000.000.000.000). Uma tabela de conversão e diversas informações sobre *gas* e *ethers* podem ser encontradas em [GasStation 2017].

1.2.1.1. Rede Principal e de Testes

A rede principal, endereçável na porta TCP 30303 trabalha com *ethers* que precisam ser comprados com dólares e as transações sofrem consequências reais.

As redes de teste trabalham com *ethers* que não possuem valor real e que podem ser adquiridos em geradores de *ethers* na Internet sem custos financeiros. A rede *Ropsten* é uma rede de teste pública de blockchain. A rede de teste *Kovan* é uma rede de teste pública que usa o protocolo de consenso Aura com prova de autoridade (Esta é uma rede permissionada). A rede de teste *Rinkeby* utiliza o protocolo de consenso *"Clique"* com prova de autoridade (Esta é uma rede permissionada). A opção *localhost* 8545 conecta-se a um nó em execução no mesmo computador que o navegador. A opção *Custom RPC* permite que conexão a qualquer nó com uma interface de Chamada de Procedimento Remoto (RPC) compatível com *Geth*. O nó pode fazer parte de qualquer blockchain pública ou privada.

Este minicurso usa a rede de testes *Rinkeby* e o plugin *Metamask*. Para abastecer a carteira do metamask com *ethers* sem valor comercial na rede *Rinkeby*, usa-se, por exemplo, o site `https://faucet.rinkeby.io/`.

O *metamask* é um gateway para aplicações da plataforma *Ethereum*. Ele fornece acesso a todas as redes da plataforma com uma única conta, permitindo que se gerencie as carteiras de todas as redes *Ethereum*. A Figura 1.1 ilustra isto. O *metamask* pode ser instalado nos navegadores Chrome ou Firefox sob forma de extensão. Ao instalar, você receberá 12 palavras mnemônicas, e deve guardá-las sob o maior sigilo, pois são a

única forma de recuperar a sua conta ou fazer transações usando a `web3`. Estas palavras devem ser informadas na ordem em que são apresentadas, na criação da conta, quando for necessário. O procedimento para instalação do *Metamask* pode ser acessado na página do curso [Abijaude e outros 2020].

O *Metamask* pode criar outras contas de acesso às redes *Ethereum*. Isto quer dizer que com as mesmas palavras mnemônicas e com a mesma instância instalada no navegador, o usuário pode ter vários endereços de contas. Isto é muito importante, principalmente no momento em que formos usar o CI e a DApp, descritas no decorrer do texto.

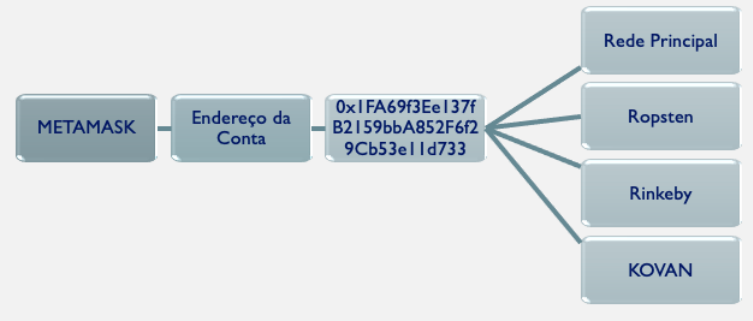


Figura 1.1. Conta na carteira metamask para acesso as redes *Ethereum*.

1.2.1.2. Transações

As transações *Ethereum* são mensagens de rede que incluem os campos ***nonce*** (número de sequência, emitido pela origem, usado para evitar a reprodução da mensagem), ***recipient*** (endereço para onde será enviada uma quantidade de *ether*), **value** (quantidade de *ether* a ser enviada), ***gasPrice*** (valor a ser pago por unidade de *gas* para a transação ser processada), ***startGas/gasLimit*** (quantidade de *gas* que esta transação pode consumir), ***data*** (A carga útil de dados binários de comprimento variável) e "**r**", "**v**"e "**s**"(variáveis criptográficas).

A função do *gasPrice* e *gasLimit* em uma transação precisa ficar bem clara. O *gas* é como se fosse o combustível do *Ethereum*. O *gas* não é *ether*, é uma moeda virtual separada com sua própria taxa de câmbio em relação ao *ether*. O *Ethereum* utiliza o *gas* separado do *ether* como forma de isolar a cotação da moeda *ether* no mundo real do valor das transações na rede, pelos quais o *gas* paga (computação, memória e armazenamento).

O campo *gasPrice* em uma transação permite que o emissor da transação defina o preço que está disposto a pagar para adquirir o *gas*. O preço é medido em *wei* por unidades de *gas*. O campo *limitGas* indica qual o limite de *gas* a ser consumido pela transação, ou seja, o número máximo de unidades de *gas* que o emissor da transação está disposto a comprar para concluir a transação.

Uma transação simples, de transferência de *ether* de uma conta para outra, custa 21.000 unidades de *gas*. O valor a ser pago em *ether* é encontrado multiplicando-se 21.000 x *gasPrice*. As carteiras geralmente possuem um valor médio cobrado na rede

para que uma transação seja confirmada dentro de um tempo aceitável (no momento da escrita deste texto estava em torno de 12,3 *gwei*). Neste caso, uma transferência de *ether* custaria 21.000 x 12,3 = 266.700 *gwei*,o que equivale a 0.0002667 *ether*.

As carteiras podem ajustar o *gasPrice* nas transações originadas para obter uma confirmação mais rápida das transações. Quanto maior o *gasPrice*, mais rápido a transação provavelmente será confirmada. Por outro lado, as transações de baixa prioridade podem ter um preço reduzido, resultando em uma confirmação mais lenta. O valor mínimo que *gasPrice* pode ser definido é zero, o que significa uma transação sem taxas. Durante os períodos de demanda por espaço em um bloco, essas transações podem ser preteridas.

1.2.1.3. Consenso

O *Ethereum* usa o modelo de consenso do *Bitcoin* (*PoW - Proof of Work*). No entanto, há planos em um futuro próximo de mudar para um sistema de votação ponderada (*PoS - Proof of Stake*), cujo codinome é *Casper*. As regras de consenso da *Ethereum* são definidas em [Wood e outros. 2014]. O algoritmo PoW usado pelo *Ethereum* é o *Ethash* [Wiki 2017].

As transições de estado do *Ethereum* são processadas pela Máquina Virtual *Ethereum* EVM (do inglês *Ethereum Virtual Machine*), baseada em uma pilha que executa bytecodes gerados pela compilação dos CIs.

O estado de *Ethereum* é armazenado localmente em cada nó como um banco de dados (geralmente o *LevelDB* do Google), que contém as transações e o estado do sistema em uma estrutura de dados em *hash* serializada chamada de *Merkle Patricia Tree*.

Estes nós clientes são um aplicativo de software que implementa a especificação *Ethereum*, comunicando-se pela rede ponto a ponto com outros clientes *Ethereum*. Há diferentes implementações, mas interoperáveis, e dentre as mais comuns, temos: *Geth* (escrito em *Go*), *Parity* (escrito em *Rust*), *Cpp-ethereum* (escrito em C++), *Pyethereum* (escrito em Python), *Mantis* (escrito em Scala) e *Harmony* (escrito em Java).

1.2.2. Contratos Inteligentes

Os CIs, segundo Nick Szabo [Szabo 1997], representam "um conjunto de promessas, especificado em formato digital, incluindo protocolos nos quais as partes cumprem estas promessas". Este conceito evoluiu, especialmente após a introdução de plataformas blockchain descentralizadas.

Recentemente, Antonopoulos reformou este conceito para se referir a programas de computador imutáveis, que são executados de forma determinística, no contexto de uma EVM, como parte do protocolo de rede *Ethereum* - ou seja, no computador mundial *Ethereum* descentralizado [Antonopoulos e Wood 2018].

Portanto, os CIs são simplesmente programas de computador. A palavra **contrato** não tem significado legal neste contexto. Eles são imutáveis, por que uma vez implementado em uma rede *Ethereum*, o código não pode ser alterado nem substituído. A única forma de se modificar o seu conteúdo é implementando um novo contrato, o qual terá um

novo endereço.

Assim como os softwares, os contratos são determinísticos, pois o resultado de sua execução é sempre o mesmo para todos os que o executam, conservando-se o contexto no momento da execução. Os CI estão em constante evolução e operam com um contexto muito limitado, por enquanto. No caso dos CIs para a rede *Ethereum*, existem diversas versões do compilador (*Solc*), com mudanças significativas entre elas. De modo geral, os CIs acessam seu próprio estado, o contexto da transação que os chamou e algumas informações sobre os blocos mais recentes.

As linguagens de programação de alto nível atualmente suportadas para construção dos contratos inteligentes são *LLL*, *Serpent*, *Vyper*, *Bambu* e *Solidity*. Esta última é a mais popular, suportada pela plataforma *Ethereum* e utilizada neste minicurso. Após escritos, os contratos precisam ser compilados para depois serem implementados em uma rede *Ethereum*. O resultado do processo de compilação são os *bytecodes* e a *Application Binary Interface* (ABI), conforme ilustrado na Figura 1.2.

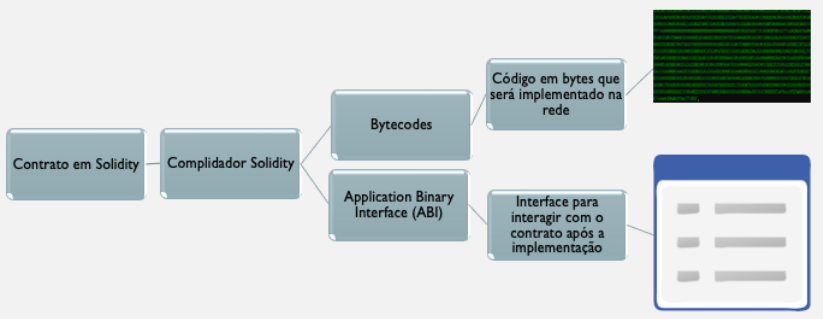


Figura 1.2. O contrato, após compilado, gera duas saídas - Os *bytecodes* e a ABI.

Os *bytecodes*, de baixo nível, são implementados na plataforma *Ethereum* usando uma transação de criação de contrato enviada para um endereço especial de criação de contratos. Cada contrato, portanto, possui um endereço *Ethereum*, que é derivado da transação de criação do contrato em função da conta e do nonce de origem. Ele é proprietário de seu próprio endereço, o qual pode ser usado, em uma transação, para receber *ethers*, por exemplo, de uma outra conta contrato ou de uma conta *Ethereum* cliente.

Para tanto é necessário acessar uma função do contrato, por intermédio da ABI. Somente através desta interface é que se pode ter acesso às funções do contrato e executar as rotinas previamente programadas.

É importante acrescentar que os contratos somente executam funções se forem chamados por uma transação. Os contratos nunca podem chamar a si próprios ou atuarem em *background*, mas podem chamar outros contratos em cadeia. As transações são atômicas, e caso a execução ocorra sem erros até o final, toda a transação é registrada.

As transações envolvendo os CIs possuem algumas particularidades em relação às transações envolvendo apenas as carteiras. Uma destas particularidades é que a transação envolvendo CIs contém dois campos: valor e dados. Estes valores podem alternadamente serem preenchidos ou não. Quando o endereço de destino de uma transação for relativa a um contrato, a EVM executará o contrato e tentará chamar a função nomeada na carga

útil de dados de sua transação. Se não houver dados, o EVM irá chamar uma função de *fallback* e, se esta função envolver pagamentos, irá executá-la para determinar o que fazer a seguir.

Outra particularidade é que as contas que representam CI possuem diferenças em relação às contas que representam apenas uma carteira eletrônica. Enquanto nestas, uma única conta pode acessar as diversas redes Ethereum, nas contas de CI isto não é possível. Um conta que representa um CI só pode acessar a rede na qual ela foi implementada. Para que este contrato possa ser implementado em outra rede *Ethereum* é necessário implementar uma nova instância do contrato nesta nova rede, com outro pagamento das taxas da transação.

Os CIs podem ser gerados basicamente de duas formas. Usando editores online como *Studio Ethereum* [StudioEthereum 2019], *Ethfidlle* [Ethfiddle 2017] e o *Remix* [Remix 2015], ou através de qualquer editor de texto, após configurar adequadamente um ambiente local para desenvolvimento, o qual será discutido na seção 1.3.

O *Remix* é um ambiente online configurado para programar, compilar e implementar CIs. Além de um editor de texto integrado, ele possui diversas versões de compiladores prontos para usar. Nele, existem 3 modos de se implementar os contratos: (a) Através de uma máquina virtual JavaScript, implementada no navegador; (b) usando a `web3` injetada pelo *Metamask*; ou (c) fornecendo um endereço para conexão de um provedor `web3`.

A figura 1.3 ilustra o ambiente de desenvolvimento do *Remix*. O contrato em tela é um exemplo didático e pode ser encontrado na página do minicurso. Este contrato é compilado na versão 7.4 do `solc` (compilador do *solidity*). Na linha 1 é informado o tipo de licença para o contrato. Em seguida, na linha 2 informa-se qual a versão do `solc` será usada para compilar o contrato.

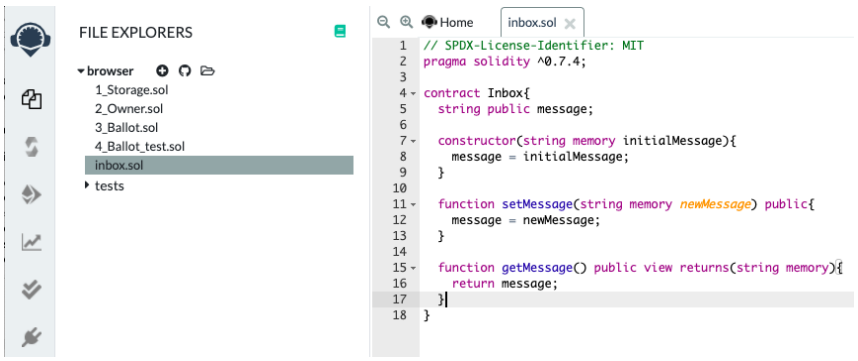


Figura 1.3. Exemplo didático de CI utilizando o editor on-line Remix.

Entre as linhas 4 e 18 está o CI propriamente dito. A linha 4 define o nome do contrato e a linha 5 declara a variável que será utilizada. As linhas 7 e 8 declaram o construtor do contrato, que será executado uma única vez, recebendo como parâmetro uma mensagem inicial.

Por fim, temos duas funções definidas no contrato. A função `setMessage`, na linha 11, quando invocada, recebe como parâmetro uma mensagem nova e atualiza a variável do contrato. A função `getMessage`, na linha 15, retorna o valor armazenado

na variável `message`. Observe que, enquanto a primeira mensagem modifica o valor de uma variável do contrato, a segunda apenas lê o seu conteúdo.

Isto implica que a função `setMessage`, quando for invocada, vai gerar custos para executar a transação e será necessário desembolsar *ethers* da carteira para que a transação seja completada. Já a função `getMessage` não tem custo nenhum para ser executada.

A função `getMessage` está presente neste contrato apenas como exemplo didático. Todas as variáveis declaradas no contrato, automaticamente, terão uma função `get` associada no momento da compilação.

1.3. Desenvolvimento de Contratos Inteligentes

O desenvolvimento de CI em *Solidity* pode ser feito no *Remix*, como mostrado na seção 1.2.2 ou usando um ambiente local, onde se tenha mais liberdade para escolher ferramentas que possam tornar o trabalho mais produtivo. Esta seção apresenta a linguagem *Solidity*, explica como configurar um ambiente de desenvolvimento e logo em seguida, apresenta um CI que será usado com uma DApp.

Ainda não há um Ambiente de Desenvolvimento Integrado (IDE, do inglês *Integrated Development Environment*) ou um ambiente amigável para o desenvolvimento dos contratos e de DApps. A solução encontrada pelos autores e já testada em cursos na Universidade Estadual de Santa Cruz (UESC), Universidade Estadual do Sudoeste da Bahia (UESB) e Universidade Federal da Bahia (UFBA) será descrita abaixo. Toda a documentação, *scripts* e códigos estão disponíveis na página do curso [Abijaude e outros 2020].

1.3.1. A Linguagem Solidity

O primeiro exemplo de um CI foi muito simples e sem explorar recursos da linguagem *Solidity*. Evidentemente que este não é um minicurso de *Solidity*, pois esta é uma linguagem poderosa e em constante evolução. No entanto, serão apresentados pontos da linguagem, como tipos de variáveis, métodos e funções com o objetivo de fornecer uma base suficiente para que os alunos possam explorar sozinhos novos conhecimentos. Os tipos básicos de dados do *Solidity* são listados na tabela 1.1.

Quando criamos uma transação e a enviamos para a rede, algumas informações podem ser encapsuladas na mensagem. Estas opções são pré-definidas pelo *Solidity* e agrupadas em 3 categorias: `msg, block e tx`.

`msg` - O objeto `msg` é uma chamada de transação originada de um cliente *Ethereum* ou de um contrato. Ela contém uma série de atributos úteis:

- `msg.sender`: Representa o endereço que iniciou a chamada de contrato
- `msg.value`: O valor de *ether* enviado com esta chamada (em *wei*).
- `msg.gas`: A quantidade de *gas* restante no suprimento de *gas* desse ambiente de execução. Isso foi descontinuado no *Solidity* v0.4.21 e substituído pela função `gasleft()`.
- `msg.data`: A carga útil de dados desta chamada no contrato.

Tabela 1.1. Tabela com os tipos de dados disponíveis no *Solidity*

Tipo	Descrição
`int`	Inteiros positivos ou negativos (`int`) declarados em incrementos de 8 bits (`int8`, `int16`, ... `int256`).
`uint`	Inteiros positivos declarados em incrementos de 8 bits (`uint8`, `uint16`... `uint256`).
`bool`	Valor lógico, verdadeiro ou falso, com operadores lógicos ! (não), && (e), || (ou), == (igual) e! = (diferente).
`fixed/ ufixed`	Números de ponto fixo, declarados com `(u)fixedMxN` em que M é o tamanho em bits (incrementos de 8 até 256) e N é o número de decimais após o ponto (até 18); por exemplo, `ufixed32x2`.
`address`	Usado para armazenar endereços *Ethereum* de 20 bytes. O objeto de endereço tem muitas funções membro úteis, como `balance` (retorna o saldo da conta) e `transfer` (transfere *ether* para uma conta).
`byte array (fixed)`	Matrizes de bytes de tamanho fixo, declaradas com bytes.
`byte array (dynamic)`	Matrizes de bytes de tamanho variável, declaradas com bytes ou `string`.
`enum`	Tipo definido pelo usuário para enumerar valores discretos `enum name {rotulo1, rotulo2...}`.
`array`	Um `array` de qualquer tipo, fixo ou dinâmico.
`struc`	`Conteiners` de dados definidos pelo usuário para agrupar variáveis `struct Car { String year; int color; }`.
`Mapping`	Tabelas de pesquisa de *hash* para pares chave => `mapping (key_type=>value_type)`.

- `msg.sig`: Os primeiros quatro bytes da carga de dados, que é o seletor de função.

`block` - O objeto de bloco contém informações sobre o bloco atual:

- `block.blockhash(blockNumber)`: O *hash* de um bloco específico. Em desuso e substituído pela função `blockhash()` no *Solidity* v0.4.22.
- `block.coinbase`: O endereço do destinatário das taxas do bloco atual e da recompensa do bloco.
- `block.difficulty`: A dificuldade (prova de trabalho) do bloco atual.
- `block.gaslimit`: A quantidade máxima de *gas* que pode ser gasta em todas as transações incluídas no bloco atual.
- `block.number`: O número do bloco atual.
- `block.timestamp`: O carimbo de data/hora colocado no bloco atual pelo minerador.

`tx` - O objeto *tx* fornece um meio de acessar informações relacionadas à transação:

- `tx.gasprice`: o preço do *gas* na transação de chamada.
- `tx.origin`: O endereço da conta *Ethereum* de origem para esta transação. Esta é uma operação considerada insegura!

O *Solidity* oferece uma facilidade de programação para manipular os dados relativos aos endereços passados como entrada. Eles facilitam a escrita dos contratos e são apresentados na forma de atributos e métodos.Os principais estão listados abaixo:

- `address.balance`: O saldo do endereço, em *wei*. Por exemplo, o saldo do contrato atual é `address(this).balance`.
- `address.transfer(quantidade)`: Transfere o valor (em *wei*) para este endereço, lançando uma exceção para qualquer erro.
- `address.send(quantidade)`: Semelhante ao `transfer`. Ao invés de lançar uma exceção, ele retorna falso em caso de erro.
- `address.call(payload)`: pode construir uma chamada de mensagem arbitrária com uma carga de dados. Retorna falso em caso de erro. Mas o destinatário pode (acidentalmente ou maliciosamente) esgotar todo o seu *gas*, fazendo com que seu contrato seja interrompido com uma exceção.

Dentro de um contrato, é necessário definir funções que podem ser chamadas por uma transação originada em uma carteira *Ethereum* ou em outro contrato. A sintaxe usada para declarar uma função no *Solidity* é a seguinte:

`function FunctionName ([parâmetros]) public|private|internal|external [pure|constant|view|payable] [modifier] [return (tipos de retorno)]`, onde:

`FunctionName` é o nome usado para chamar a função em uma transação de uma carteira *Ethereum*, de outro contrato ou de dentro do mesmo contrato. Uma função pode ser definida sem um nome. Neste caso, é a função de *fallback*, que é chamada quando nenhuma outra função é nomeada. A função de *fallback* não pode ter argumentos ou retornos.

Os parâmetros vêm após o nome, especificado os argumentos que devem ser passados para a função, com seus nomes e tipos.

O próximo atributo especifica a visibilidade da função. O padrão são funções públicas que podem ser chamadas por outros contratos, transações de carteiras *Ethereum*, ou de dentro do contrato. As funções com atributo `external` são como funções públicas, exceto que não podem ser chamadas de dentro do contrato, a menos que explicitamente prefixadas com a palavra-chave this.

As funções com atributo `internal` são acessíveis apenas de dentro do contrato ou por contratos derivados de outro contrato. Elas não podem ser chamadas por outro

contrato ou transações de carteiras *Ethereum*. As funções com o atributo `private` são como funções `internal`, mas não podem ser chamadas por contratos derivados.

Lembre-se de que os termos `internal` e `private` são um tanto enganosos. Qualquer função ou dado dentro de um contrato está sempre visível na blockchain pública, o que significa que qualquer pessoa pode ver o código ou os dados. As palavras-chave descritas aqui afetam apenas como e quando uma função pode ser chamada.

O segundo conjunto de palavras-chave `(pure, constant, view, payable)` afetam o comportamento da função:

Uma função `constant` ou `view` promete não modificar nenhum estado. Os termos possuem o mesmo objetivo e o primeiro será descontinuado em uma versão futura.

Uma função `pure` é aquela que não lê nem grava nenhuma variável no armazenamento. Ele só pode operar em argumentos e retornar dados, sem referência a nenhum dado armazenado.

Uma função `payable` é aquela que pode aceitar pagamentos recebidos. Funções não declaradas como `payable` rejeitarão pagamentos recebidos.

Existem 3 tipos especiais de funções que deve-se ficar atento: construtoras, auto-destruição e modificadoras.

As funções construtoras são executadas apenas uma vez, durante a criação do contrato e possuem a palavra-chave `constructor()`. As funções de auto-destruição possuem a palavra-chave `destroy()` e são utilizadas, como o próprio nome diz, para destruir o contrato implementado. As funções modificadoras são aplicadas adicionando-se o nome do `modifier` na declaração da função. São usados para criar condições que se aplicam a muitas situações em um contrato, e para isto, basta acrescentar o seu nome na declaração de uma função.

1.3.2. Construindo um Contrato Inteligente

Será apresentado agora um novo e mais sofisticado CI, explorando mais as funcionalidades da linguagem Solidity. Este CI simula uma loteria, de forma que os participantes apostam e depois, realiza-se um sorteio ente eles. O ganhador receberá o saldo do contrato em sua conta *Ethereum*.

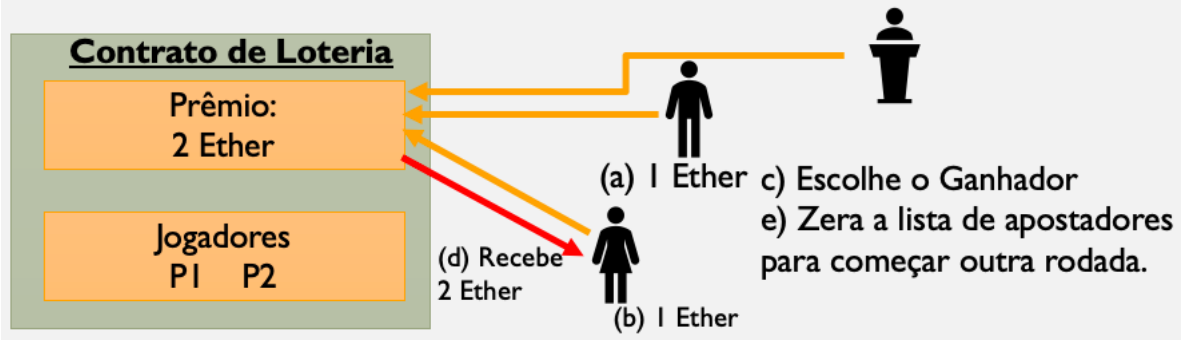


Figura 1.4. Lógica do CI que simula um sorteio entre apostadores.

A Figura 1.4 mostra como funciona a lógica a ser implementada no contrato. Em (a) um jogador aposta 1 ether, enviando-o para o contrato. Em (b), outro jogador aposta mais 1 ether. O gerente, representado pelo endereço da conta que implementou o contrato na rede *Ethereum*, decide fazer o sorteio em (c). Isto executa uma função que faz uma escolha aleatória e conclui que o segundo jogador ganhou, por exemplo, transferindo para ele o saldo de 2 *ethers*, em (d). Terminado o processo, o gerente zera a lista de apostadores e inicia outra rodada em (e).

Este contrato necessita de 5 funções: (`jogar`, `random`, `sorteio`, `verificaGerente`, `getJogadores`) e as variáveis `gerente` e `jogadores`.

O código do contrato é exibido na Figura 1.5. A linha 1 indica sob qual licença de uso está o contrato. A linha 2 informa a versão do compilador. É obrigatório que um contrato inicie com estas informações. A linha 4 define o nome do CI (`Loteria`). Este bloco do contrato começa na linha 4 e vai até a linha 36.

```
1  // SPDX-License-Identifier: MIT
2  pragma solidity ^0.7.4;
3  contract Loteria{
4      address public gerente;
5      // cria a variável para receber o endereço do gerente
6      address payable[] public jogadores;
7      // cria o array para reeber o endereço dos jogadores com
8      //capacidade de pagamento
9      constructor(){
10         gerente = msg.sender;
11         //atribui o endereço do gerente à variável
12     }
13     function jogar() public payable{
14         require(msg.value > 0.1 ether);
15         jogadores.push(msg.sender);
16         //adiciona no array o endereço do jogador
17     }
18     function random() private view returns (uint){
19         return uint(keccak256(abi.encodePacked(block.difficulty,
20          block.timestamp, jogadores)));
21         // keccak256 gera um hash
22     }
23     function sorteio() public verificaGerente{
24         uint indice = random() % jogadores.length;
25         //usamos o operador modulo (%) para sortear um indice
26         jogadores[indice].transfer(address(this).balance);
27         jogadores = new address payable[](0);
28     }
29     modifier verificaGerente(){
30         require(msg.sender == gerente);
31         _;
32     }
33     function getJogadores() public view returns(address payable[] memory){
34         return jogadores;
35     }
36 }
```

Figura 1.5. Código-fonte do CI `Loteria.sol`, que simula a aposta e sorteio entre jogadores.

As linhas 4 e 6 definem as variáveis a serem utilizadas no contrato. A variável `gerente` é do tipo `address` e de escopo público. Ela armazena o endereço da conta

que implementa o contrato. A variável `jogadores` é um `array` de escopo público, do tipo `payable`, para receber os endereços das contas *Ethereum* que irão realizar apostas, inclusive a conta do próprio administrador. Como estes endereços vão transacionar valores, eles precisam ser do tipo `payable`.

Entre as linhas 9 e 12, define-se um construtor, o qual é executado apenas uma vez durante o ciclo de vida do contrato. Este construtor captura o endereço da conta *Ethereum* que implementou o contrato e armazena-o na variável `gerente`.

A função `jogar()`, entre as linhas 13 e 17, realiza duas tarefas. A primeira é restringir as apostas, permitindo apenas aquelas com valor superior a 0.1 *ether*. A instrução `require` executa esta tarefa. Se o resultado for falso, a execução da função é abandonada e uma exceção é gerada. Se for verdadeiro, como é uma função `payable`, ela transfere o valor para a conta contrato e continua a execução na linha seguinte. Começa então a segunda tarefa: adicionar na matriz de jogadores o endereço da carteira *Ethereum* que fez a aposta.

A função `random()`, entre as linhas 18 e 22, possui escopo privado e não modifica dados no contrato, retornando um inteiro, que representa um *hash* dos parâmetros repassados. Este valor é obtido através da função `keccak256`, que recebe como parâmetros a dificuldade do bloco atual (`block.difficulty`), o tempo em que este bloco foi gerado (`block.timestamp`) e a matriz `jogadores` e retorna um inteiro.

A função `sorteio()`, linhas 23 a 28, é uma função pública que usa a função modificadora `verificaGerente()`. Isto implica que, antes de esta função executar o seu próprio conteúdo, o fluxo do programa é desviado para a função modificadora `verificaGerente()`. Na linha 29 temos a declaração desta função. Na linha 30, compara-se o remetente da mensagem com o endereço da variável `gerente`. Se for verdadeiro, a execução segue para a próxima linha. O símbolo `"_"` quer dizer que o fluxo do programa retornará para a primeira linha da função que invocou a função modificadora. Caso o resultado da expressão `require`, na função modificadora, fosse falso, seria gerada uma exceção e o fluxo não retornaria para a função `sorteio()`.

Considerando que os requisitos da função modificadora foram atendidos, finalmente a função `sorteio()` será executada. A linha 24 declara a variável `indice`, do tipo inteiro, que calcula o módulo do valor gerado pela função `random()` pelo tamanho da matriz de jogadores. Isto garante que o resultado será sempre um número compreendido entre zero e o tamanho desta matriz. A linha 26 localiza o índice na matriz de jogadores e transfere o saldo da conta contrato para este endereço. Na linha 27, a matriz de jogadores é zerada para que seja feita uma nova rodada de apostas.

A última função, `getJogadores()`, é do tipo pública, não modifica dados do contrato, e retorna os endereços da matriz de jogadores. Observe que esta função não é utilizada na lógica do jogo. Ela será usada na DApp desenvolvida na seção 1.4.

1.3.3. Configuração do Ambiente

Esta seção vai detalhar como configurar um ambiente de desenvolvimento de CIs no computador. Existe um guia completo na página do minicurso com informações detalhadas sobre este processo. Lá estão três roteiros práticos para desenvolvimento de CIs que serão

executados na apresentação do minicurso. O projeto que será usado como referência para configuração do ambiente é o Loteria.

O primeiro passo é ter o Node.js e um editor de código de sua preferência instalados e configurados no computador. Após isto, pode-se instalar os pacotes, executar os *scripts* para compilar e implementar os contratos e preparar uma rotina de testes.

Após instalar o Node.js, deve-se criar uma pasta no computador para armazenar os arquivos do projeto. Em seguida, acesse a página do minicurso, escolha o projeto Loteria e siga as instruções. Ao final, a estrutura de diretório do projeto estará pronta e todos os arquivos disponíveis para uso.

Os seguintes pacotes adicionais para o Node.js precisam ser instalados: a) `solc`: compilador *Solidity*; b) `mocha`: *framework* para testar os contratos antes de implementá-los em uma rede BC; c) `web3`: coleção de bibliotecas que permite interagir com um nó *Ethereum* local ou remoto usando HTTP; d) `ganache-cli`: é uma BC pessoal para desenvolvimento rápido de aplicativos distribuídos *Ethereum* e *Corda* em um ambiente seguro e determinístico; e) `truffle-hdwallet-provider`: para realizar as assinaturas usando as palavras mnemônicas.

A instalação de todos estes pacotes pode ser feita facilmente executando, dentro da pasta do projeto, o comando `npm install --save`. Isto pode demorar um pouco, mas, quando o procedimento de instalação for encerrado, seu ambiente estará configurado com todos os arquivos necessários para escrever, compilar e implementar CIs.

Os autores fornecem dois *scripts*, escritos na linguagem JavaScript, que permitem compilar e implementar os CIs. Procure na pasta raiz do projeto por `compile.js` e `deploy.js`. Ambos precisam, para cada contrato, de pequenos ajustes.

O *script* `compile.js` é mostrado na figura 1.6. As linhas de 1 a 6 declaram variáveis com alguns requisitos necessários, como por exemplo o compilador `solc`. Observe que na linha 5 é informado o nome do contrato que se deseja compilar. Neste exemplo, o *script* está preparado para compilar o contrato `Loteria.sol`. O mesmo acontece nas linhas 12, 27 e 29. Estas são as modificações que precisam ser feitas quando este *script* for utilizado em outros contratos. A chamada para o compilador `solc` ocorre na linha 25, armazenando o resultado na variável `contratoCompilado`. É neste momento que o contrato é de fato compilado. A linha 27 exibe no console o resultado da compilação. Ela está aí apenas para fins didáticos, com o objetivo de exibir na tela a saída do compilador. Quando houver a necessidade de compilar um contrato para posterior implementação, a linha 27 deve ser comentada e retirado o `//` da linha 29.

O *script* `deploy.js`, exibido na figura 1.7, trabalha em conjunto com o *script* `compile.js`. Entre as linhas 1 e 4 são definidas variáveis que importam bibliotecas. A linha 5 envia como parâmetro informações que devem ser colocadas no arquivo `.env`, a ser criado no diretório raiz do projeto. Ele deve conter 2 linhas: `mnemonic = 'Suas palavras mnemônicas'` e `provider = 'endereço do site infura'`.

A linha 9 instancia o web3 e passa como referência o endereço da rede Infura para implementação do contrato. A linha 10 declara uma função assíncrona, cujo objetivo é enviar para a blockchain o pedido de implementação do contrato e aguardar

```
1  const path = require("path"); // linhas para indicar o caminho onde o arquivo será lido
2  const fs = require("fs"); // e garantir a compatibilidade de sistemas operacionais
3  const solc = require("solc");
4  // Pega o arquivo Inbox.sol e atribui a variável
5  const LoteriaPath = path.resolve(__dirname, "contracts", "Loteria.sol");
6  const source = fs.readFileSync(LoteriaPath, "utf8");
7  // * Mais informações sobre o input e output
8  // * https://docs.soliditylang.org/en/v0.7.4/using-the-compiler.html#output-description
9  var input = {
10   language: "Solidity",
11   sources: {
12     "Loteria.sol": {
13       content: source,
14     },
15     // Pode-se adicionar otros contratos, caso exista
16   },
17   settings: {
18     outputSelection: {
19       "*": {
20         "*": ["*"],
21       },
22     },
23   },
24 };
25 let contratoCompilado = JSON.parse(solc.compile(JSON.stringify(input)));
26 // Gera o log para investigação
27  console.log(contratoCompilado.contracts["Loteria.sol"].Loteria);
28 // Pedimos apenas o nosso contrato para exportação
29 //module.exports = contratoCompilado.contracts["Loteria.sol"].Loteria;
```

Figura 1.6. *Script* `compile.js` para compilação de contratos inteligentes no ambiente Node.js.

que o processo de mineração seja concluído. A linha 11 e 12 definem a conta onde será debitada as taxas de implementação do contrato. Das linhas 13 a 16 preparamos a chave privada para assinar a transação. Entre as linhas 18 e 20 declaramos a variável `contract` com informações sobre os bytecodes. As linhas de 21 a 24 definem a variável `transactionObject` que define o conteúdo da transação. Nesta parte, pode-se observar que define-se 4.000.000 de unidades de gas. Entre as linhas 26 e 28, assina-se a transação com a chave privada, e finalmente na linha 30 e 31 envia-se o a transação para a rede blockchain.

A execução do *script* fica paralisada. Enquanto isto, esta transação vai para uma piscina de transações da rede *Ethereum* e aguarda a sua vez para ser inserida em um bloco e publicada como válida. Somente depois deste processo é que o *script* prossegue a execução, imprimindo no console o endereço atribuído ao contrato (linha 33) e fechando a conexão com o provedor na linha 37. Anote o endereço do contrato, pois esta informação será necessária quando a DApp for construída na seção 1.4.

Existem dois sites muito importantes que hospedam aplicativos e que complementam as atividades a serem desenvolvidas com os CI. O primeiro (`www.infura.io`) permite o acesso à rede BC, criando um endereço nos servidores *Ethereum* para que se possa implementar nossos SCs. O segundo (`https://faucet.rinkeby.io`) é usado para

```
require("dotenv").config();
const HDWalletProvider = require("@truffle/hdwallet-provider");
const Web3 = require("web3");
const { abi, evm } = require("./compile");
const provider = new HDWalletProvider({
  mnemonic: { phrase: process.env.mnemonic },
  providerOrUrl:process.env.provider,
});
const web3 = new Web3(provider);
const deploy = async () => {
  const accounts = await web3.eth.getAccounts();
  const deploymentAccount = accounts[0];
  const privateKey = provider.wallets[
    accounts[0].toLowerCase()
  ].privateKey.toString("hex");
  console.log("Conta usada para o deploy ", accounts[0]);
  try {
    let contract = await new web3.eth.Contract(abi)
      .deploy({ data: evm.bytecode.object, arguments: [] })
      .encodeABI();
    let transactionObject = {
      gas: 4000000,
      data: contract,
      from: deploymentAccount,
    };
    let signedTransactionObject = await web3.eth.accounts.signTransaction(
      transactionObject,
      "0x" + privateKey
    );
    let result = await web3.eth.sendSignedTransaction(
      signedTransactionObject.rawTransaction
    );
    console.log("Contract deployed to", result.contractAddress);
  } catch (error) {
    console.log(error);
  }
  provider.engine.stop();
};
deploy();
```

Figura 1.7. *Script* `deploy.js` para implementação de contratos na rede *Rinkeby*.

a geração de *ethers* sem valor comercial, empregados nas redes de teste da plataforma *Ethereum*, conforme mencionado anteriormente.

O site *Infura* permite o acesso instantâneo às redes *Ethereum* através de *websockets* ou HTTPS. O serviço permite até 3 projetos gratuitos. O nó *Infura* cria a solicitação de acesso a rede desejada (principal ou teste) e retorna um link, o qual aponta para o endereço válido. Este link deve ser adicionado na segunda linha do arquivo `.env`. Neste momento, os alunos são convidados para acessarem a página do curso e executarem o tutorial da segunda prática.

1.3.4. Rotina de Testes para CI

Os CIs são programas especiais que gerenciam ativos digitais na BC. É difícil recuperar a perda se os usuários fizerem transações por meio de CIs com bugs, que não podem ser corrigidos diretamente. Portanto, é importante garantir a exatidão dos contratos inteligentes antes de implementá-los [Wu e outros 2019].

A referência [Wu e outros 2019] propõe uma estrutura sistemática para teste de mutação para contratos inteligentes no *Ethereum*. Quinze novos operadores de mutação foram projetados para os CIs, em termos de palavra-chave, variável/função global, unidade de variável e tratamento de erros.

A referência [Andesta e outros 2019] propõe um mecanismo de testes para CIs na linguagem *Solidity*, baseado em testes de mutação. Analisando uma lista abrangente de *bugs* conhecidos nos contratos inteligentes do *Solidity* foi possível catalogar 10 classes de operadores de mutação inspirados nas falhas reais.

Uma técnica automatizada, *SolAnalyser* [Akca e outros 2019], para detecção de vulnerabilidade em CIs desenvolvidos em *Solidity* usa análise estática e dinâmica. Ela oferece suporte à detecção automatizada de 8 tipos diferentes de vulnerabilidades que atualmente carecem de amplo suporte nas ferramentas existentes e podem ser facilmente estendidas para oferecer suporte a outros tipos.

Estas referências, no entanto, não apresentaram detalhes suficientes para que, em tempo hábil, fossem analisadas e incluídas nas práticas deste minicurso. Existem ainda muitos artigos versando sobre este tema nas bases de pesquisa.

Optou-se em usar o *framework* `mocha` [OpenJS 2017]. As razões que contribuíram para isto são a farta documentação disponível, a experiência dos autores com este *framework* e o amplo uso deste em projetos envolvendo Javascript e CIs na rede *Ethereum*.

O ciclo de testes do `mocha` é bastante simples e segue 4 passos: Início -> Implementação de um contrato -> Manipulação do Contrato -> Comparações. Uma vez realizado o primeiro e segundo passos, o terceiro e quarto se repetem tanto quanto forem os testes a serem realizados.

A figura 1.8 ilustra um fragmento de código do arquivo de testes. A primeira parte consiste em preparar o teste, importando os requisitos necessários e definindo as variáveis que serão usadas (veja as linhas de 1 a 7). Esta parte representa o primeiro passo (Início).

As linhas de 8 a 13 representam o segundo passo e implementam o contrato na rede `ganache`. Esta rede simula localmente uma rede *Ethereum*. A partir da linha 14 começam efetivamente a manipulação dos contratos e as comparações para testes, agrupados em blocos definidos pelo comando `describe`. Estes são o terceiro e quarto passos, respectivamente. Cada um destes blocos manipula e depois testa o contrato. No exemplo que compreende as linhas de 14 até 18 é feito um teste para verificar se o contrato foi implementado na rede *ganache*. Isto é verificado na linha 17, quando solicita-se que seja retornado o endereço do contrato implementado anteriormente. Caso haja um endereço associado ao contrato, então ele foi implementado com sucesso.

O arquivo `test/Loteria.test.js` contém todo o arquivo de teste. Ao abrí-

```
1   const assert = require("assert");
2   const ganache = require("ganache-cli");
3   const Web3 = require("web3");
4   const web3 = new Web3(ganache.provider());
5   const { abi, evm } = require("../compile");
6   let loteria;
7   let contas;
8   beforeEach(async () => {
9     contas = await web3.eth.getAccounts();
10    loteria = await new web3.eth.Contract(abi)
11      .deploy({ data: evm.bytecode.object })
12      .send({ from: contas[0], gas: "1000000" });
13  });
14  describe("Contrato Loteria", () => {
15    it("Deploy a contract", () => {
16      // console.log(inbox);
17      assert.ok(loteria.options.address);
18    });
```

Figura 1.8. Fragmento do arquivo de testes usado no contrato `Loteria.sol` com o *framework* `mocha`.

lo, observe que foram feitos testes para recuperar o endereço do contrato Loteria; permitir que uma conta seja adicionada ao sorteio; permitir que várias contas sejam adicionadas ao sorteio; verificar a quantidade mínima de ether a ser apostado; verificar se o gerente solicita que seja feito o sorteio; e o teste geral, envolvendo todos os passos do contrato `Loteria.sol`. Para executar os testes basta digitar `npm run test` na pasta do projeto e aguardar o resultado. Há um tutorial na página do curso para guiá-lo.

1.4. Desenvolvendo DApps

Uma DApp é um aplicativo que é parcialmente ou totalmente descentralizado e pode oferecer características únicas, exigindo conhecimentos em novas linguagens e em conceitos até então pouco explorados.

Aspectos como por exemplo o *back-end*, o *front-end*, o armazenamento de dados, a comunicação e até mesmo a questão de resolução de nomes podem ser centralizados ou descentralizados. Por exemplo, um *front-end* pode ser desenvolvido como um aplicativo da web executado em um servidor centralizado ou como um aplicativo móvel executado em seu celular ou *tablet*. O *back-end* e o armazenamento podem estar em servidores privados e bancos de dados proprietários, ou você pode usar um CI e o armazenamento P2P [Antonopoulos e Wood 2018].

Além disto, as DApps oferecem peculiaridades que os sistemas centralizados não conseguem, como por exemplo a resiliência, a transparência e a resistência a censura.

A interface do lado do cliente de uma DApp pode usar tecnologias da web padrão (HTML, CSS, JavaScript, etc.). Isso permite que um desenvolvedor da web tradicional use ferramentas, bibliotecas e estruturas familiares. As interações com a rede *Ethereum*, como assinatura de mensagens, envio de transações e gerenciamento de chaves, geralmente são conduzidas por meio do navegador web utilizando uma extensão como *Meta-*

mask.

O *front-end* é geralmente vinculado à rede *Ethereum* por meio da `web3.js`, uma biblioteca, empacotada com os recursos do próprio *front-end* e servida a um navegador por um servidor web. O *Metamask* injeta uma biblioteca `web3` no navegador, mas para que as DApps funcionem adequadamente, é necessário rejeitar esta biblioteca injetada pelo *Metamask* e injetar a biblioteca `web3.js`. A figura 1.9 ilustra isto. Ambas as bibliotecas conseguem acessar a rede *Rinkeby*, mas a `web3.js`, proveniente da aplicação, deve assumir o controle, conforme discutiremos na seção 1.4.1.

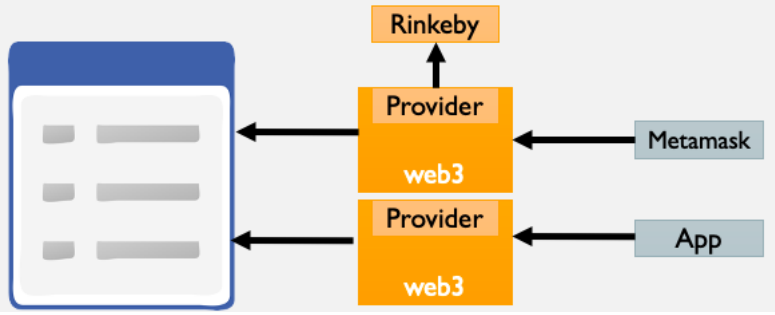


Figura 1.9. Biblioteca `web3` injetada pelo *metamask* x injetada pela aplicação.

1.4.1. Criando a DApp para interagir com o contrato `Loteria.sol`

Esta seção descreve os passos para criar uma DApp que interage com o contrato criado na seção 1.3.2. A página do curso possui o projeto `loteria_react` com todos os arquivos e configurações necessárias, inclusive o tutorial para a prática realizada ao final desta seção. Crie uma nova pasta e salve o projeto `loteria_react` da página do curso. Após isto siga o roteiro descrito para instalar as dependências do projeto.

Para que a DApp funcione, o primeiro passo é implementar o contrato na rede *Rinkeby* e guardar o endereço gerado pelo site *Infura*. Este passo já foi executado no laboratório anterior.

Na estrutura de diretório criada, existe a pasta `src` com 4 arquivos: `web3.js`, `index.js`, `loteria.js` e `App.js`. Os três primeiros preparam o ambiente para a execução da DApp. Todos possuem comentários explicativos e são de fácil entendimento. O arquivo `web3.js` realiza as configurações necessárias para injetar a `web3` na DApp que iremos criar. O arquivo `index.js` importa o `react`, o `react-dom` e o arquivo `app.js`, imprescindíveis para a construção da página web. O arquivo `loteria.js` possui a ABI gerada pela compilação do contrato `loteria.sol`.

O arquivo `app.js` é onde as coisas acontecem. Este arquivo é responsável por exibir uma tela no navegador, conforme a figura 1.10. Observe que há componentes dinâmicos, destacados no retângulo laranja. Estes componentes são capturados diretamente do CI e atualizados todas as vezes que se clica nos botões Jogar ou Sortear.

Ao clicar no botão Jogar, o usuário já deve ter preenchido a quantidade de *ether* a ser enviada com um valor superior a 0.1 *ether*. Caso não faça isto uma exceção será gerada e a aposta não será concluída. A extensão do *Metamask* precisa estar ativa no

Figura 1.10. Tela apresentada pela DApp com os comentários sobre os campos.

navegador para que uma conta seja registrada no momento da aposta.

Ao clicar no botão Sortear, a conta selecionada no *Metamask* será enviada ao contrato para verificar se o endereço coincide com a conta que implementou o contrato. Se a verificação for verdadeira, o sorteio irá acontecer e a conta sorteada receberá os *ethers* da conta contrato, aumentando o seu saldo. Para verificar isto, basta navegar pelas contas no *Metamask* e observar os saldos das contas.

O código do arquivo `App.js` é divido em 5 seções. A primeira seção importa o `react`, com o conjunto de bibliotecas para construção do site; o arquivo `web3.js`, cuja função é trabalhar com as requisições da rede com a BC; e o `loteria.js` com a ABI, necessária para acessar o CI.

A segunda seção é composta pela função assíncrona `carregarDados()`. As linhas de 18 e 20, exibidas na figura 1.11 acionam o contrato para executar as funções `gerente()` e `getJogadores()`, ambas no CI implementado na BC. A linha 22 solicita ao contrato que envie o saldo em *ethers* da conta contrato.

```
15  // Função assincrona que carrega os dados do contrato
16  const carregarDados = async () => {
17    // Pega a carteira do gerente do contrato
18    const _gerente = await loteria.methods.gerente().call();
19    // Pega a carteira dos jogadores
20    const _jogadores = await loteria.methods.getJogadores().call();
21    // Pega o valor total vinculado ao contrato
22    const _saldo = await web3.eth.getBalance(loteria.options.address);
```

Figura 1.11. Função assíncrona para acessar o CI na BC e carregar informações para uso na DApp.

A terceira seção do código possui a função assíncrona `apostar()` (veja figura 1.12). Esta função faz um tratamento de exceções devido à condição do valor mínimo para a aposta. A linha 42 configura uma mensagem para alertar ao usuário, no momento adequado, que ele precisa aguardar a validação da transação de aposta submetida ao CI, que por sua vez depende da publicação desta transação na cadeia de blocos da rede *Ethe-*

reum. A linha 44 usa o `web3` para capturar a conta selecionada no *Metamask*. A linha 48 invoca a função `jogar()` no CI, enviando a conta e o valor que farão parte da aposta. A linha 53 recarrega os dados da página, atualizando a mensagem, assim que a rede *Ethereum* validar a transação feita na linha 48. As linhas de 56 a 62 fazem o tratamento das exceções que podem ocorrer.

```
37  const apostar = async (event) => {
38    try {
39      // Evita que a página seja recarregada
40      event.preventDefault();
41      // Altera valor da mensagem exibida
42      setMensagem("Aguardando a  validação da transação...");
43      // Pega contas do metamask
44      const contas = await web3.eth.getAccounts();
45      // console.log(contas);
46
47      // Joga passando valor da conta principal e o valor de ether em wei
48      await loteria.methods.jogar().send({
49        from: contas[0],
50        value: web3.utils.toWei(value, "ether"),
51      });
52      // Recarrega dados da página
53      await carregarDados();
54      // Altera mensagem
55      setMensagem("Transação concluida!");
56    } catch (error) {
57      // Caso o usuário cancele a solicitação no metamask
58      if (error.code === 4001) {
59        setMensagem("Transação cancelada!");
60      } else {
61        // Caso algo esteja fora das políticas do contrato
62        setMensagem("Transação vai contra regras do contrato");
63      }
64    }
65  };
```

Figura 1.12. Função assíncrona para realizar as apostas. Esta função roda em um bloco *Try/Cath* para tratar as exceções geradas.

A quarta seção descreve a função assíncrona `sortear()` (veja figura 1.13). A linha 72, através da `web3`, captura a conta selecionada no *Metamask* para a variável `contas`. Isto permite a verificação da conta que implementou o contrato, condição exigida para a realização do sorteio. Esta etapa, então é realizada na linha 74, com o envio de uma transação para a rede *Ethereum*, invocando a função do CI `sorteio()`. Antes de executar esta função, o fluxo de execução do CI será redirecionado para a função modificadora `verificaGerente()`. Se a conta ativa no momento da execução for a mesma que implementou o contrato, a função do CI `sorteio()` segue a execução do código e sorteia um ganhador para a rodada. Após o processo de validação desta transação, a mensagem é exibida na tela da DApp para o usuário (linhas 78 e 80). O bloco de linhas entre 81 e 87 trata as possíveis exceções que podem acontecer.

A quinta e última seção, compreendida entre as linhas 92 e 118 exibe a tela da DApp. O código está ilustrado na figura 1.14. A linha 97 consulta o contrato, solicitando o valor da variável `saldo`. Este valor é obtido do CI `Loteria.sol` na linha 22. Esta informação é dinâmica e atualizada cada vez que a página é lida pelo navegador. A linha

```
67  const sortear = async () => {
68    try {
69      // Altera mensagem
70      setMensagem("Aguardando processamento...");
71      // Pega contas do metamask
72      const contas = await web3.eth.getAccounts();
73      // Solicita sorte e manda conta que está realizando o sorteio
74      await loteria.methods.sorteio().send({
75        from: contas[0],
76      });
77      // Recarrega dados da página
78      await carregarDados();
79      // Altera mensagem
80      setMensagem("Um vencedor foi escolhido!");
81    } catch (error) {
82      // Caso o usuário cancele a solicitação no metamask
83      if (error.code === 4001) {
84        setMensagem("Transação cancelada!");
85      } else {
86        // Caso algo esteja fora das políticas do contrato
87        setMensagem("Transação vai contra regras do contrato");
88      }
89    }
90  };
```

Figura 1.13. Função assíncrona para realizar o sorteio entre as contas que apostaram.

100 invoca a função local `apostar()` quando o botão Jogar do formulário (definido na linha 110) for acionado. E por fim, quando o botão Sorteio, definido na linha 114 for acionado, a função local `sortear()` será invocada.

```
92      <div>
93        <h2>Contrato de Loteria</h2>
94        <p>Este contrato é gerenciado por {gerente}</p>
95        <p>
96          Existem agora {jogadores.length} pessoas apostando para ganhar{" "}
97          {web3.utils.fromWei(saldo, "ether")} ether
98        </p>
99        <br />
100       <form onSubmit={apostar}>
101         <h4>Quanto deseja apostar?</h4>
102         <div>
103           <label>Quantidade de ether para ser enviado: </label>
104           <input
105             value={value}
106             // Altera o valor que está sendo apostado
107             onChange={(event) => setValue(event.target.value)}
108           />
109         </div>
110         <button>Jogar</button>
111       </form>
112       <hr />
113       <h4>Realizar sorteio? </h4>
114       <button onClick={sortear}> Sortear</button>
115       <hr />
116       {/* Mostra mensagem ao usuário */}
117       <h1>{mensagem}</h1>
118     </div>
119   );
120 };
```

Figura 1.14. Código para exibição da tela da DApp.

1.5. Como montar um curso de Desenvolvimento Web com Blockchain e Contratos Inteligentes

A programação de CIs não é uma tarefa trivial. Envolve conceitos, propriedades e conhecimentos que vão além da linguagem de programação *Solidity*. Existe uma escassez de material teórico e prático para o ensino de tecnologias emergentes como Blockchain, CIs e desenvolvimento de DApps.

Uma alternativa para isto são as plataformas MOOC (*Massive Open Online Course*, como Udemy, Coursera, edX. Algumas Universidades promovem cursos de extensão ou treinamentos para seus alunos [Rao e Dave 2019, Delmolino e outros 2016, Dettling 2018, Araujo e outros 2019].

Os autores deste minicurso elaboraram e ministraram um curso em três universidades na Bahia: A Universidade Estadual de Santa Cruz (UESC), a Universidade Estadual do Sudoeste da Bahia (UESB) e a Universidade Federal da Bahia UFBA).

Os cursos são compostos de três módulos. O primeiro, básico, serviu de base para a criação deste minicurso e aborda conceitos de BC, CIs e DApps, criando um contrato simples e uma DApp. O segundo, explora mais profundamente as funções da linguagem *Solidity*, construindo um contrato bem mais complexo, utilizando técnicas de engenharia de software, e criando uma DApp multipágina. O terceiro módulo, ainda em elaboração, prevê a integração com a Internet das Coisas, coletando dados dos sensores, armazenando-os em CIs e disparando ações quando determinadas situações forem alcançadas.

O hardware empregado para o desenvolvimento dos laboratórios é bastante simples, uma vez que não há plataformas que necessitem de muitos recursos computacionais. Nos 3 treinamentos ministrados havia computadores equipados com processadores que vão desde o Core 2 Duo com 4 Gb de RAM, até o Core i7 com 32 Mb de RAM. O espaço em disco também não é um fator limitante, uma vez que a maioria das máquinas possui espaço de armazenamento suficiente os experimentos realizados.

O conjunto de softwares necessários à realização de todas as atividades práticas do treinamento é composto por navegadores, extensões para navegadores, ferramentas, pacotes e aplicativos hospedados em sites. Os softwares são compatíveis com praticamente todos os sistemas operacionais, como por exemplo Windows, Linux, Unix e MacOs.

O curso básico foi ministrado com carga horária de 16h. Em 2 dias de aula, com 4 horas no período da manhã e 4 horas no período da tarde. No primeiro dia, durante o período da manhã foi abordada toda a parte teórica e conceitual da Blockchain com ênfase na plataforma Ethereum, além da apresentação do editor de contratos on-line *Remix*. Durante o período da tarde, configuramos as máquinas locais e realizamos rotinas de testes.

No segundo dia, durante o período da manhã apresentamos mais um pouco de teoria com foco na interação com as redes *Ethereum*. Aprendemos mais um pouco sobre a linguagem de programação *Solidity* e escrevemos, compilamos, testamos e implementamos um contrato na rede *Rinkeby*. No período da tarde desenvolvemos uma DApp que interagia com o contrato implementado.

É possível encontrar mais recursos na internet, como por exemplo:

a) *CryptoZombies*: uma plataforma online onde o intuito é ensinar sobre contratos inteligentes de forma interativa. O usuário desenvolve um jogo com foco em zumbis onde a logística é administrada por um contrato e com interação visual através de html, css e javascript. Disponível em `https://cryptozombies.io/pt/`.

b) *Ethernaut*: uma plataforma online que apresenta diversos tutoriais voltados para jogos. Ela foca puramente na criação de contratos, sem implementação visual. Alguns exemplos possibilitam a interação através da ferramenta do desenvolvedor do navegador. Disponível em `https://ethernaut.openzeppelin.com/`.

c) *Vyper Tutorials*: semelhante a ideia do CryptoZombies, esta plataforma propõe a criação de um jogo de *pokémon*. Atualmente está em fase de desenvolvimento, mas já é possível aprender como funciona a criação de contratos. Futuramente serão adicionados interações através de interface assim como *CryptoZombies*. Disponível em `https://vyper.fun/#/`.

d) *Ethereum Studio*: uma ferramenta para desenvolvedores que desejam aprender sobre como construir aplicações na rede *Ethereum*. Os modelos ensinam como escrever um contrato inteligente, implementá-lo e interagir com os CIs por meio de um aplicativo baseado na web. Disponível em `https://studio.ethereum.org/`.

1.6. Desafios e Perspectivas

Embora existam ferramentas que auxiliam no desenvolvimento e nos testes de CIs e da BC, ainda há desafios em aberto. Nesta seção, serão apresentados alguns destes desafios e possíveis propostas que estão em andamento para solucionar problemas encontrados e aperfeiçoar os métodos de desenvolvimento e análise.

Considerando o que foi explicado anteriormente, percebe-se que ainda existe um problema no que se refere à garantia da segurança no desenvolvimento de CIs. Como explica [Atzei e outros 2016], mesmo existindo ferramentas que auxiliam na verificação destas falhas, o uso de uma linguagem *Turing*-completa (como o *Solidity*) pode limitar o processo de verificação. A sugestão é a criação de linguagens não *Turing*-completas para este fim, sendo uma direção a ser seguida pela literatura. Partindo deste princípio, [Jansen e outros 2019] faz um estudo para avaliar se realmente os contratos da rede *Ethereum* utilizam todo o aparato de controle de fluxo oferecido pelo *Solidity*. O resultado deste estudo mostrou que uma pequena parte dos contratos analisados (35,3% de 53757 contratos) utilizam os mecanismos complexos de controle de fluxo oferecidos pela linguagem, e afirmam portanto, que o uso de linguagens não *Turing*-completas no contexto de sistemas baseados em blockchain faria todo o sentido.

Há uma grande quantidade de artigos buscando aprimorar a forma como os CIs são desenvolvidos, de forma a aumentar o nível de abstração para que, além de abstrair a estrutura de implementação, diminua a responsabilidade do desenvolvedor de ter que analisar o CI a fim de encontrar possíveis falhas. A referência [Frantz e Nowostawski 2016] utiliza uma estrutura de gramática institucional para representar CIs, de modo a permitir que tanto desenvolvedores quanto outras pessoas possam ser capazes de criar uma estrutura e, a partir desta, gerar automaticamente o código do CI. Uma ferramenta semelhante é proposta por [Mavridou e Laszka 2018], que utiliza estrutura de máquina de estados finita

para modelar e gerar os códigos dos CIs já aplicando métricas para evitar falhas de segurança. Como uma forma de abstrair ainda mais, [Qin e outros 2019] propõe transformar a linguagem natural em código de CI, para isso, utiliza uma gramática como um dicionário para ajudar na representação do CI em linguagem melhor entendível pelo humano.

A verificação formal é outro desafio que contribui para o estágio de análise de CIs. Com o objetivo de investigar se o CI está se comportando de acordo com a especificação correta, aplica-se uma prova matemática e constrói-se um modelo formal do CI para certificar que este comportamento é válido [Wang e outros 2019]. Alguns métodos de verificação formal são abordados em [Murray e Anisi 2019], mas observa-se que é uma área relativamente nova e ainda não existem padrões definidos. A referência [Bhargavan e outros 2016] apresenta uma forma de traduzir o código *Solidity* e os *bytecodes* da EVM em F*, uma linguagem funcional para verificação de CIs. É uma ferramenta que não suporta toda a sintaxe do *Solidity*, mas é capaz de encontrar algumas vulnerabilidades e erros de sintaxe. A referência [Amani e outros 2018], propõe descompilar os *bytecodes* para dividir o código em blocos e utilizar um provador lógico *Isabelle/HOL* para finalizar a verificação do CI.

Junto com as diversas plataformas blockchain existentes, variadas linguagens foram criadas para implementação de CIs. Em cada uma delas, aumenta-se a complexidade em desenvolver contratos para diferentes plataformas [Coblenz e outros 2019]. Pensando em uma solução para este e outros desafios apresentados, uma nova linguagem de programação de CIs que envolve propriedades que são comuns às linguagens de várias plataformas BC, além de oferecer verificação formal e análises de CIs de forma automática é proposta por [Coblenz e outros 2020].

Diante deste cenário que os autores apresentaram, espera-se que este material auxilie e encoraje os pesquisadores e alunos a compreenderem as relações entre a Blockchain, os Contrato Inteligentes e os Sistemas Web a fim de que possam ampliar suas pesquisas e trabalhos correlatos.

Referências

[Abijaude e outros 2020] Abijaude, J., Serra, H., Santiago, L., Sobreira, P., e Greve, F. (2020). Blockchain, contratos inteligentes sistemas web:teoria e prática. `https://github.com/lifuesc/minicurso-blockchain`. Acessado em 03/03/2021.

[Akca e outros 2019] Akca, S., Rajan, A., e Peng, C. (2019). Solanalyser: A framework for analysing and testing smart contracts. In *2019 26th Asia-Pacific Software Engineering Conference (APSEC)*, pages 482–489. IEEE.

[Amani e outros 2018] Amani, S., Bégel, M., Bortin, M., e Staples, M. (2018). Towards verifying ethereum smart contract bytecode in isabelle/hol. In *Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pages 66–77.

[Andesta e outros 2019] Andesta, E., Faghih, F., e Fooladgar, M. (2019). Testing smart contracts gets smarter. *arXiv preprint arXiv:1912.04780*.

[Androulaki e outros 2018] Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., e outros. (2018). Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, page 30. ACM.

[Antonopoulos e Wood 2018] Antonopoulos, A. M. e Wood, G. (2018). *Mastering ethereum: building smart contracts and dapps*. O'reilly Media.

[Araujo e outros 2019] Araujo, P., Viana, W., Veras, N., Farias, E. J., e de Castro Filho, J. A. (2019). Exploring students perceptions and performance in flipped classroom designed with adaptive learning techniques: A study in distributed systems courses. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 30, page 219.

[Atzei e outros 2016] Atzei, N., Bartoletti, M., e Cimoli, T. (2016). A survey of attacks on ethereum smart contracts. *IACR Cryptol. ePrint Arch.*, 2016:1007.

[Bhargavan e outros 2016] Bhargavan, K., Delignat-Lavaud, A., Fournet, C., Gollamudi, A., Gonthier, G., Kobeissi, N., Kulatova, N., Rastogi, A., Sibut-Pinote, T., Swamy, N., e outros. (2016). Formal verification of smart contracts: Short paper. In *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security*, pages 91–96.

[Buterin e outros. 2014] Buterin, V. e outros. (2014). A next-generation smart contract and decentralized application platform. *white paper*, 3(37).

[Coblenz e outros 2020] Coblenz, M., Oei, R., Etzel, T., Koronkevich, P., Baker, M., Bloem, Y., Myers, B. A., Sunshine, J., e Aldrich, J. (2020). Obsidian: Typestate and assets for safer blockchain programming. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 42(3):1–82.

[Coblenz e outros 2019] Coblenz, M., Sunshine, J., Aldrich, J., e Myers, B. (2019). Smarter smart contract development tools. In *2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*, pages 48–51. IEEE.

[Coinbase 2018] Coinbase (2018). Coinbase wallet. `https://wallet.coinbase.com/`. Acessado em 03/03/2021.

[Delmolino e outros 2016] Delmolino, K., Arnett, M., Kosba, A., Miller, A., e Shi, E. (2016). Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab. In *International conference on financial cryptography and data security*, pages 79–94. Springer.

[Dettling 2018] Dettling, W. (2018). How to teach blockchain in a business school. In *Business Information Systems and Technology 4.0*, pages 213–225. Springer.

[Ethfiddle 2017] Ethfiddle (2017). Ethfiddle editor rinkeby. `https://ethfiddle.com/`. Acessado em 03/03/2021.

[Frantz e Nowostawski 2016] Frantz, C. K. e Nowostawski, M. (2016). From institutions to code: Towards automated generation of smart contracts. In *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS* W)*, pages 210–215. IEEE.

[GasStation 2017] GasStation (2017). Gas station. `https://ethgasstation.info/index.php`. Acessado em 03/03/2021.

[Go-ethereum 2013] Go-ethereum, T. (2013). Go ethereum - official go implementation of the ethereum protocol. `https://geth.ethereum.org/`. Acessado em 03/03/2021.

[Greve e outros 2018] Greve, F. G., Sampaio, L. S., Abijaude, J. A., Coutinho, A. C., Valcy, Í. V., e Queiroz, S. Q. (2018). Blockchain e a revolução do consenso sob demanda. *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)-Minicursos*.

[Jansen e outros 2019] Jansen, M., Hdhili, F., Gouiaa, R., e Qasem, Z. (2019). Do smart contract languages need to be turing complete? In *International Congress on Blockchain and Applications*, pages 19–26. Springer.

[Jaxx 2018] Jaxx (2018). Jaxx safely manager ethereum. `https://jaxx.io/`. Acessado em 03/03/2021.

[LeMahieu 2018] LeMahieu, C. (2018). Nano: A feeless distributed cryptocurrency network. *Nano [Online resource]. URL: https://nano. org/en/whitepaper (date of access: 24.03. 2018)*.

[Mavridou e Laszka 2018] Mavridou, A. e Laszka, A. (2018). Tool demonstration: Fsolidm for designing secure ethereum smart contracts. In *International Conference on Principles of Security and Trust*, pages 270–277. Springer.

[Metamask 2018] Metamask (2018). Metamask crypto wallet and gateway. `https://metamask.io/`. Acessado em 03/03/2021.

[Murray e Anisi 2019] Murray, Y. e Anisi, D. A. (2019). Survey of formal verification methods for smart contracts on blockchain. In *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–6. IEEE.

[MyCrypto 2019] MyCrypto (2019). Mycrypto. `https://mycrypto.com/`. Acessado em 03/03/2021.

[Myetherwallet 2019] Myetherwallet (2019). Myetherwallet original wallet. `https://www.myetherwallet.com/`. Acessado em 03/03/2021.

[Nakamoto 2008] Nakamoto, S. (2008). A peer-to-peer electronic cash system. *Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf*, 4. Acessado em 03/03/2021.

[OpenJS 2017] OpenJS (2017). Mocha test framework. `https://mochajs.org/`. Acessado em 03/03/2021.

[Popov 2018] Popov, S. (2018). The tangle, iota whitepaper. https://iota.org/IOTA_Whitepaper.pdf. Acessado em 03/03/2021.

[Qin e outros 2019] Qin, P., Guo, J., Shen, B., e Hu, Q. (2019). Towards self-automatable and unambiguous smart contracts: Machine natural language. In *International Conference on e-Business Engineering*, pages 479–491. Springer.

[Rao e Dave 2019] Rao, A. R. e Dave, R. (2019). Developing hands-on laboratory exercises for teaching stem students the internet-of-things, cloud computing and blockchain applications. In *2019 IEEE Integrated STEM Education Conference (ISEC)*, pages 191–198. IEEE.

[Remix 2015] Remix (2015). Remix ide. https://remix.ethereum.org/. Acessado em 03/03/2021.

[Status 2019] Status (2019). Status private, secure communication. https://status.im/. Acessado em 03/03/2021.

[StudioEthereum 2019] StudioEthereum (2019). Studio ethereum. https://studio.ethereum.org//. Acessado em 03/03/2021.

[Szabo 1997] Szabo, N. (1997). Formalizing and securing relationships on public networks. *First Monday*.

[Trust 2019] Trust (2019). Trust wallet - secure crypto wallet. https://trustwallet.com/. Acessado em 03/03/2021.

[Wang e outros 2019] Wang, S., Ouyang, L., Yuan, Y., Ni, X., Han, X., e Wang, F.-Y. (2019). Blockchain-enabled smart contracts: architecture, applications, and future trends. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(11):2266–2277.

[Web3js 2016] Web3js (2016). Ethereum javascript api. https://web3js.readthedocs.io/en/v1.3.0/index.html. Acessado em 03/03/2021.

[Wiki 2017] Wiki, E. (2017). Ethash. *GitHub Ethereum Wiki. https://github.com/ethereum/wiki/wiki/Ethash*. Acessado em 03/03/2021.

[Wood e outros. 2014] Wood, G. e outros. (2014). Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32.

[Wu e outros 2019] Wu, H., Wang, X., Xu, J., Zou, W., Zhang, L., e Chen, Z. (2019). Mutation testing for ethereum smart contract. *arXiv preprint arXiv:1908.03707*.

Capítulo

2

Ética da Pesquisa em Sistemas de Informação: Por que e como submeter meu projeto ao Comitê de Ética?

Valéria Farinazzo Martins[1,2], Michelle Asato Junqueira[3] e Renata Mendes de Araujo[1,4]

[1]Faculdade de Computação e Informática, Universidade Presbiteriana Mackenzie; [2]Programa de Pós-Graduação em Distúrbios do Desenvolvimento da Universidade Presbiteriana Mackenzie; [3]Faculdade de Direito, Universidade Presbiteriana Mackenzie; [4]Programa de Pós-Graduação em Sistemas de Informação da EACH-USP

Abstract

Research ethics has been increasingly discussed today, reflecting the need for higher impact research results to society, innovation stimulus and research quality improvement concerning transparency about risks and benefits for those who participate in it. Research institutions, as well as scientific journals and conferences, have stimulated this discussion by requiring research involving human beings to be submitted to the Research Ethics Committees (CEP), linked to the National Council for Ethics in Research (CONEP). Researchers are often unaware of the regulations related to the topic and, in general, believe that their research material does not involve ethical issues. This chapter aims to discuss the ethical aspects of research involving human beings and their importance in the scope of Information Systems research. It also details the necessary procedures for the submission and approval of a research project involving human beings using the Plataforma Brasil (Brazilian digital platform for research ethics approval).

Resumo

A ética na pesquisa tem sido um tema cada vez mais discutido nos dias atuais, reflexo da necessidade dos resultados de pesquisa alcançarem e impactarem a sociedade, do estímulo à inovação e geração de produtos e da preocupação com o aumento da qualidade das pesquisas em relação ao esclarecimento quanto aos seus riscos e benefícios para os que participam da mesma. As instituições de pesquisa, bem como periódicos e conferências, têm estimulado esta discussão mediante a exigência da submissão de pesquisas que envolvam seres humanos aos Comitês de Ética em Pesquisa (CEP). Os pesquisadores, por muitas vezes, desconhecem as normativas relacionadas ao tema e, em geral, acreditam que seu material de pesquisa não envolva questões éticas. Este capítulo tem como objetivo discutir os aspectos éticos de pesquisas envolvendo seres humanos no âmbito de Sistemas de Informação e sua importância. Também apresenta os

trâmites necessários para a submissão e aprovação de um projeto de pesquisa envolvendo seres humanos junto aos CEPs, vinculados à Comissão Nacional de Ética em Pesquisa (CONEP), usando a Plataforma Brasil.

1.1. Introdução

De acordo com a Resolução 466/2012 do Conselho Nacional de Saúde (CNS) [Brasil, 2012], no Brasil, as pesquisas, em qualquer área do conhecimento, envolvendo seres humanos, precisam seguir procedimentos éticos. Questões éticas devem ser analisadas em relação a diretrizes a respeito dos critérios de participação dos seres humanos, do consentimento livre e esclarecido, da análise de riscos e benefícios, do direito dos participantes, da responsabilidade e capacitação do pesquisador e do acompanhamento da pesquisa.

Assim, os Comitês de Ética em Pesquisa (CEP), respondendo à Comissão Nacional de Ética em Pesquisa (CONEP), se baseiam em três aspectos: a) delimitação conceitual do que são consideradas pesquisas envolvendo seres humanos: todas aquelas que os envolvem direta ou indiretamente, de forma individual ou coletiva; b) nível de formação dos pesquisadores: devem ser apresentadas pesquisas delineadas por estudantes de graduação, pós-graduação e por profissionais; e, c) espectro das investigações: devem ser avaliadas pesquisas em todas as áreas de conhecimento.

Embora a Resolução 466/2012 seja clara em estabelecer o que deve ser entendido como participante da pesquisa, para todas as áreas do conhecimento, na área de Computação, por muitas vezes, corre-se o risco de negligenciar tal resolução, seja por desconhecimento, por falta de delimitação do alcance de tal normativa, por acreditar que o material de pesquisa não envolva questões éticas, ou por considerar a norma excessiva.

A área de pesquisa em Sistemas de Informação tem como objetivo principal o estudo da concepção, aplicação, uso e impacto de novas tecnologias computacionais em diferentes domínios organizacionais e sociais. Como uma área de pesquisa teórico-aplicada, boa parte da pesquisa desenvolvida envolve a concepção de artefatos visando a ampliação de capacidades humanas em suas atividades de interação profissionais e/ou sociais. É esperado que uma parcela significativa das pesquisas na área envolva a validação de soluções junto a seres humanos. Ademais, a participação da tecnologia nos novos ecossistemas de informação organizacionais e sociais, a complexidade da interação entre humanos e tecnologias e a necessidade fundamental de se compreender o aspecto sociotécnico (sócio e técnico) nestes novos ecossistemas de informação, foram estabelecidos pela comunidade científica na área de Sistemas de Informação como um de seus principais desafios para os próximos anos [Boscarioli et al., 2017].

A ética na área de computação tem sido discutida desde a década de 40, envolvendo questões ligadas à confidencialidade dos dados dos usuários, privacidade, propriedade intelectual, qualidade no trabalho, justiça e não discriminação, entre outros [Amorim et al., 2019][Kizza, 2013]. Recentemente, dada a imbricada e cada vez maior relação entre a computação e a sociedade, o tema da ética tem sido amplamente debatido, sobretudo no que se refere à atuação profissional na área, incluindo no Brasil [Santoro e Costa, 2020].

Embora a discussão desse tema no contexto da área de Computação seja amplo, o intuito deste capítulo restringe-se a discutir questões éticas relacionadas a projetos de pesquisa, em particular na área de Sistemas de Informação. Se, na prática de pesquisa científica, todas as discussões a respeito da ética profissional podem ser também discutidas, há aspectos específicos relacionados à ética neste contexto que precisam ser observados, como, especificamente, a necessidade de submissão e aprovação de projetos de pesquisa por uma comissão especializada, no caso, os comitês de ética.

De modo geral, instituições, revistas científicas e órgãos de fomento têm exigido a aprovação dos projetos de pesquisa pelos comitês de ética. Mais do que isso, é importante pensar que um projeto que tenha sido estruturado levando em consideração os preceitos éticos em sua elaboração, no que concerne à participação de usuários, pode tornar-se um projeto com mais qualidade e mais capaz de produzir impacto com seus resultados. No entanto, considerar questões éticas nos projetos de pesquisa envolve uma mudança de mentalidade por parte dos pesquisadores responsáveis pelos projetos, bem como um entendimento da responsabilidade dos comitês de ética em pesquisa e o processo de submissão de projetos a estes comitês e o diálogo que se estabelece a respeito. Em nossa experiência, percebemos que o processo de planejar as atividades que envolvem questões éticas em projetos de pesquisa na área, o diálogo entre pesquisadores e comitês de ética em pesquisa e as regras de aprovação ética dos projetos permanecem ainda confusos e, no limite, polêmicos.

Vale ainda ressaltar que a motivação na escrita deste capítulo é também oriunda de dois aspectos de implicação pessoal das autoras no tema: no primeiro caso, uma das autoras, como pesquisadora na área de SI, percebeu a necessidade em conhecer os trâmites do sistema CEP/CONEP, de acordo com as exigências da instituição onde atua, e, mais do que isso, conhecer as razões para a submissão de um projeto à avaliação ética; de outro lado, as outras duas autoras, por já participarem como membros do CEP há três anos e serem, atualmente, coordenadora geral e vice coordenadora do CEP da instituição a que são ligadas, visam compartilhar a experiência no desempenho das suas funções, fortalecendo, inclusive, o caráter educativo do CEP.

Nosso objetivo, com este texto, é trazer à luz a importância de considerar os aspectos éticos de pesquisa envolvendo seres humanos, apontando questões que dizem respeito à adequação dos projetos de pesquisa na área de Sistemas de Informação (SI) em relação às intervenções usuais realizadas em pesquisas na área, como, por exemplo: as avaliações de artefatos computacionais com potenciais usuários, o acesso a bases de dados não públicas, o acesso a dados pessoais, os estudos de caso e pesquisa-ação em organizações ou grupos sociais. O conteúdo deste texto tem caráter introdutório, destinado aos pesquisadores com interesse em conhecer os principais aspectos e os trâmites para a aprovação ética de seus projetos, seja ele no âmbito da graduação, pós-graduação, pesquisa ou inovação.

O capítulo é organizado da seguinte forma: na Seção 1.2, discutimos o conceito de ética em pesquisa e um histórico das principais normativas que regem o conceito no Brasil e no mundo. Na Seção 1.3, apresentamos as iniciativas das principais associações científicas nacionais e internacionais em Computação e Sistemas de Informação, resultantes em seus códigos de ética profissional e de publicações científicas. A Seção 1.4. apresenta as legislações nacionais que regem o tema e os atores principais de sua regulação - os comitês de ética em pesquisa ou CEPs. A Seção 1.5 discute como a área

de pesquisa em SI tem abordado a questão da ética em suas pesquisas e qual a relevância de submeter projetos da área à aprovação ética, ilustrando casos reais. A Seção 1.6. detalha o processo de submissão e tramitação de projetos para aprovação ética aos CEPs institucionais, com uso da Plataforma Brasil. A Seção 1.7 conclui o texto com considerações finais.

1.2. Ética em Pesquisa

Os comitês de ética em pesquisa compreendem mecanismos de controle social com base em princípios éticos para a garantia da integridade e dignidade dos participantes de pesquisa. O controle social é um dos pilares do Estado Democrático de Direito, na medida em que possibilita a participação da sociedade em questões de relevância, permitindo que contribua para a construção de instituições baseadas na cidadania, impedindo abusos e arbítrios desvinculados da vontade popular por parte dos detentores do poder.

Do ponto de vista da configuração jurídica no Brasil, vale apontar que a Constituição Brasileira de 1988 [Brasil, 1988] não menciona a palavra "ética", mas sim faz menção à integridade na concepção de unidade nacional ou, ainda, no sentido de manutenção do meio ambiente ou integridade física e moral, reportando-se sempre, portanto, para a identificação do todo, do uno, do inteiro. Por sua vez, a moralidade aparece por quatro vezes, sempre associada à gestão da Administração Pública.

Postas tais premissas, é importante diferenciar ética e moral. A ética tem origem na palavra grega ethos, que significa "caráter", ou seja, é a qualidade do ser. Ética é ciência, que integra a filosofia e oferta critérios para escolha da melhor conduta para a comunidade humana. Ética é o agir, é a intenção. Por sua vez, a moral deriva de *mos (moralis)*, palavra de origem latina, que significa "costume". Assim, a moral é um conjunto de hábitos, usos e regras que não incorpora a dimensão pessoal, mas o sentido comunitário, determina, portanto, uma prescrição de conduta, ou seja, é um fenômeno social de caráter normativo, orientando a conduta [Camillo, 2019]. Parte-se, então, da premissa, que a ética é que dá a essência da moralidade, que se traduz na normativa necessária para o respeito à dignidade humana, fundamento do Estado Brasileiro e postulado interpretativo dos direitos humanos e fundamentais.

A ética na pesquisa compreende "a ciência da conduta humana, é o princípio sistemático da conduta moralmente correta" [Bongertz, 1999]. Se a pesquisa é a busca incessante por respostas, a ética na pesquisa pode ser traduzida como a procura por respostas mediante condutas moralmente corretas. Nesta linha argumentativa, acrescente-se que as questões éticas em pesquisa ganham especial relevância após as diversas atrocidades cometidas durante a Segunda Guerra Mundial.

O regime nazista propunha três categorias de experiências, especialmente nos campos de concentração: (i) as que tinham por finalidade facilitar a sobrevivência dos militares do Eixo, como testes de reações às altas altitudes para determinar a altitude máxima que as equipes de aeronaves poderiam saltar de pára-quedas, bem como experiências de congelamento para se descobrir um método eficaz para a hipotermia e testes visando a trasnformação de água marinha em potável; (ii) testagem de

medicamentos e métodos de tratamento para ferimentos e enfermidades como malária, tifo, tuberculose, febre tifóide, febre amarela e hepatite infecciosa, inoculando os prisioneiros com essas doenças, pode-se ainda citar as experiências com enxertos ósseos e (iii) exepriências que buscavam aprofundar princípios raciais e ideológicos na visão nazista, com a utilização de pessoas para determinar como diversas "raças" resistiam a doenças infecciosas, visando comprovar a superioridade ariana, além das crueis experiências de esterilização [Enciclopedia do Holocausto, 2020].

Após o julgamento de Nuremberg, foi necessária a criação do Código de Nuremberg em 1947 [CREMESP, 2020], estabelecendo a necessidade do consentimento humano voluntário e estabelecendo princípios éticos mínimos para a pesquisa. Em 1948, com a Declaração de Direitos Humanos da ONU [ONU, 2020] de se estabelecer paradigmas para o reconhecimento e proteção dos direitos humanos, o paradigma da dignidade humana é reforçado. Assim, em 1964 foi formulada a Declaração de Helsinque da Associação Médica Mundial [CREMESP, 2020a], válida, após algumas emendas, até os dias atuais.

A Declaração de Helsinque traça, pela primeira vez, o dever dos médicos nas pesquisas médicas: proteger a vida, a saúde, dignidade, integridade, direito à autodeterminação, privacidade e confidencialidade das informações pessoais dos sujeitos de pesquisa (artigo 6º), prescrevendo: "a responsabilidade pela proteção aos sujeitos de pesquisa deve sempre recair no médico ou outros profissionais da saúde e nunca ao sujeito da pesquisa, mesmo que eles tenham dado consentimento". Observe-se que, embora destinada às pesquisas médicas, seus postulados estão presentes em toda e qualquer pesquisa envolvendo seres humanos.

Desde então, os países e a comunidade científica vêm tentando adaptar estes princípios éticos e de respeito à dignidade humana ao seu ordenamento interno, visando dar efetividade às declarações universais, cabendo destaque, ainda, à Declaração Universal sobre Bioética e Direitos Humanos, firmada por diversos países, na Conferência Geral da Unesco, em 2005 [CREMESP, 2020b].

1.3. Códigos de Ética em Pesquisa em Computação

Segundo Amorim et al. (2019), a área de pesquisa em Computação discute os aspectos éticos de sua aplicação desde a década de 40, principalmente em consequência de seus impactos durante a Segunda Guerra Mundial. De lá para cá, diversas associações científicas em Computação internacionais, como a *Association for Computing Machinery* (ACM) [ACM, 2018] e o *Institute of Electrical and Electronic Engineers* (IEEE) [IEEE, 2020], e nacionais, como a Sociedade Brasileira de Computação (SBC) [SBC, 2013][SBC, 2020], elaboraram seus respectivos códigos de ética. Na área de Sistemas de Informação, a *Association for Information Systems* (AIS), principal associação científica internacional no tema, também divulga seu código de conduta em pesquisa [AIS, 2014]. Em 2019, a AIS inicia a discussão sobre o seu código de ética profissional, com base no documento da ACM[1]. Agências de fomento nacionais, como a FAPESP, também

[1] https://aisnet.org/news/474415/Invitation-to-Comment-on-Proposed-AIS-Member-Code-of-Ethics-and-Professional-Conduct.htm

apresentam suas visões éticas em orientação aos pesquisadores [FAPESP, 2014 apud Amorim et al, 2019].

A preocupação de todos estes códigos está em apoiar a reflexão de profissionais e pesquisadores da área quanto aos impactos éticos de suas atividades profissionais ou práticas, bem como regular e induzir a execução de procedimentos de avaliação ética em suas pesquisas e na divulgação científica. Portanto, notem que o foco de cada documento varia entre **códigos de conduta profissional**, **códigos de conduta em pesquisa** e **códigos de conduta para publicações científicas**. Nos interessa discutir, no presente caso, os códigos de conduta em pesquisa - recomendações éticas para o desenvolvimento de projetos de pesquisa científica que, de forma natural, compreende também as recomendações éticas para publicações, uma das etapas do processo de pesquisa.

Iniciamos esta discussão apresentando um resumo dos conteúdos dos códigos de conduta em pesquisa das sociedades científicas que mais se aproximam da área de Sistemas de Informação - a SBC, a AIS e a ACM - e exemplos de orientações de instituições de fomento nacionais, como a FAPESP. O Código de Ética do Profissional de Informática [SBC, 2013] orienta a respeito da prática profissional em empresas e na sociedade, o que, em certa medida, pode se aplicar também às atividades profissionais como pesquisadores inseridos em instituições de pesquisa. O recém divulgado Código de Conduta para Publicações da SBC [SBC, 2020], se preocupa com desvios éticos relacionados à plágio e submissão de publicações nos principais veículos gerenciados pela sociedade (Tabela 1). O código de conduta profissional da ACM segue a linha de orientações para a prática profissional na área de Computação que, conforme dito anteriormente, podem se aplicar às práticas profissionais de pesquisa (Tabela 2). Finalmente, a AIS define seu código de conduta para pesquisa que, embora tenha este nome, traz em seu conteúdo, recomendações para publicações de resultados de pesquisa e sua disseminação (Tabela 3). O Código de Boas práticas Científicas da FAPESP [FAPESP, 2014], por sua vez, pretende abordar recomendações para todo o processo de pesquisa científica, desde sua concepção até sua publicação, incluindo os procedimentos usuais de discussão científica por meio de revisões por pares (Tabela 4).

Tabela 1 - Códigos de Conduta da SBC.

Sociedade Brasileira de Computação (SBC)	
Código de Ética do Profissional de Informática [SBC, 2013]	Código de Conduta para Publicações da SBC [SBC, 2020] (resumido)
São deveres dos profissionais de Informática: Art. 1o : Contribuir para o bem-estar social, promovendo, sempre que possível, a inclusão de todos setores da sociedade. Art. 2o : Exercer o trabalho profissional com responsabilidade, dedicação, honestidade e justiça, buscando sempre a melhor solução.	Parte I - Condutas Não Aceitáveis É de responsabilidade dos autores evitar a ocorrência de: Art. 1º Plágio Art. 2º Autoplágio Art. 3º Submissão múltipla Art. 4º Exclusão de artigo Parte II - Ações Recomendáveis

Art. 3o : Esforçar-se para adquirir continuamente competência técnica e profissional, mantendo-se sempre atualizado com os avanços da profissão. Art. 4o : Atuar dentro dos limites de sua competência profissional e orientar-se por elevado espírito público. Art. 5o : Guardar sigilo profissional das informações a que tiver acesso em decorrência das atividades exercidas. Art. 6o : Conduzir as atividades profissionais sem discriminação, seja de raça, sexo, religião, nacionalidade, cor da pele, idade, estado civil ou qualquer outra condição humana. Art. 7o : Respeitar a legislação vigente, o interesse social e os direitos de terceiros. Art. 8o : Honrar compromissos, contratos, termos de responsabilidade, direitos de propriedade, copyrights e patentes. Art. 9o : Pautar sua relação com os colegas de profissão nos princípios de consideração, respeito, apreço, solidariedade e da harmonia da classe. Art. 10: Não praticar atos que possam comprometer a honra, a dignidade, privacidade de qualquer pessoa. Art. 11: Nunca apropriar-se de trabalho intelectual, iniciativas ou soluções encontradas por outras pessoas. Art. 12: Zelar pelo cumprimento deste código.	É recomendado aos autores observarem as seguintes ações: Art. 5º Reprodutibilidade de resultados de pesquisa: nos casos pertinentes, recomenda-se que artigos indiquem a disponibilidade pública de material utilizado na pesquisa, de modo a facilitar a reprodução dos respectivos resultados por outros pesquisadores, como códigos utilizados e base de dados. Art. 6º Participação em autoria: espera-se que todos os autores de um trabalho publicado ou submetido para publicação, tenham tido efetiva participação no respectivo trabalho. Parte III - Disposições Gerais Art. 7º Violações ao código: a ocorrência de violações ao código de conduta pode ser apresentada ao editor dos anais do evento ou do periódico que, por sua vez, deverão dar a solução adequada ou encaminhar ao Comitê de Ética da SBC, que decidirá pela aplicação ou não de alguma penalidade. Art. 8º Abrangência: este código deve ser seguido por todos os eventos e publicações realizados ou apoiados pela SBC. Art. 9º Situações não previstas neste código serão analisadas pelo Comitê de Ética da SBC.

Tabela 2 - Código de Conduta da ACM.

***Association for Computing Machinery* (ACM)**
ACM Code of Ethics and Professional Conduct [ACM, 2018] (resumido)
1. GENERAL ETHICAL PRINCIPLES. *A computing professional should…* *1.1 Contribute to society and to human well-being, acknowledging that all people are stakeholders in computing.* *1.2 Avoid harm.* *1.3 Be honest and trustworthy.* *1.4 Be fair and take action not to discriminate.* *1.5 Respect the work required to produce new ideas, inventions, creative works, and computing artifacts.*

1.6 Respect privacy.

1.7 Honor confidentiality.

2. PROFESSIONAL RESPONSIBILITIES.

A computing professional should…

2.1 Strive to achieve high quality in both the processes and products of professional work.

2.2 Maintain high standards of professional competence, conduct, and ethical practice.

2.3 Know and respect existing rules pertaining to professional work.

2.4 Accept and provide appropriate professional review.

2.5 Give comprehensive and thorough evaluations of computer systems and their impacts, including analysis of possible risks.

2.6 Perform work only in areas of competence.

2.7 Foster public awareness and understanding of computing, related technologies, and their consequences.

2.8 Access computing and communication resources only when authorized or when compelled by the public good.

2.9 Design and implement systems that are robustly and usably secure.

3. PROFESSIONAL LEADERSHIP PRINCIPLES.

A computing professional, especially one acting as a leader, should…

3.1 Ensure that the public good is the central concern during all professional computing work.

3.2 Articulate, encourage acceptance of, and evaluate fulfillment of social responsibilities by members of the organization or group.

3.3 Manage personnel and resources to enhance the quality of working life.

3.4 Articulate, apply, and support policies and processes that reflect the principles of the Code.

3.5 Create opportunities for members of the organization or group to grow as professionals.

3.6 Use care when modifying or retiring systems.

3.7 Recognize and take special care of systems that become integrated into the infrastructure of society.

4. COMPLIANCE WITH THE CODE.

A computing professional should…

4.1 Uphold, promote, and respect the principles of the Code.

4.2 Treat violations of the Code as inconsistent with membership in the ACM.

Tabela 3 - Código de Conduta em Pesquisa da AIS.

Association for Information Systems **(AIS)**
AIS Code of Research Conduct [AIS, 2014] (resumido)
CATEGORY ONE: Codes that must ALWAYS be adhered to. *1. Do not plagiarize.* *2. Do not fabricate or falsify data, research procedures, or data analysis.*

3. Do not use other people's unpublished writings, information, ideas, concepts or data that you may see as a result of processes such as peer review without permission of the author.
4. Do not make misrepresentations to editors and conference program chairs about the originality of papers you submit to them.

CATEGORY TWO: Codes in this category are "recommended ethical behavior".

5. Give priority to the public interest, particularly when designing or implementing new information systems or other designed artefacts.
6. Respect the rights of research subjects, particularly their rights to information privacy, and to being informed about the nature of the research and the types of activities in which they will be asked to engage.
7. Do not abuse the authority and responsibility you have been given as an editor, reviewer or supervisor, and ensure that personal relationships do not interfere with your judgment.
8. Do not take or use published data of others without acknowledgement; do not take or use unpublished data without both permission and acknowledgement.
9. Declare any material conflict of interest that might interfere with your ability to be objective and impartial when reviewing submissions, grant applications, software, or undertaking work from outside sources.
10. Acknowledge the substantive contributions of all research participants, whether colleagues or students, according to their intellectual contribution.
11. Use archival material only in accordance with the rules of the archival source.

ADVICE: The following suggestions are provided on how to protect yourself from authorship disputes, mis-steps, mistakes, and even legal action.

Keep the documentation and data necessary to validate your original authorship for each scholarly work with which you are connected.

Do not republish old ideas of your own as if they were a new intellectual contribution.

12. Settle data set ownership issues before data compilation.
13. Consult appropriate colleagues if in doubt.

Tabela 4 - Boas Práticas Científicas da FAPESP.

Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP)
Código de Boas Práticas Científicas [FAPESP, 2014] (resumido)
1. Diretrizes para as atividades científicas a. Sobre a concepção, a proposição e a realização da pesquisa b. Sobre a comunicação dos resultados da pesquisa e a autoria c. Sobre o registro, conservação e acessibilidade de dados e informações d. Sobre o conflito potencial de interesses e. Sobre a avaliação pelos pares f. Sobre a tutoria 2. Sobre as más condutas científicas 3. Sobre a responsabilidade das instituições de pesquisa 4. Sobre a alegação, a investigação e a declaração de más condutas científicas

Os esforços das associações científicas e agências de fomento em publicar recomendações éticas são significativos e importantes para promover a reflexão das

comunidades de pesquisadores e, principalmente, definir políticas de verificação e decisões a respeito de ocorrências indesejáveis em seu processo de operação. Reparem que, em todos os códigos apresentados, veremos menções a questões importantes relacionadas à interação com sujeitos e ao uso de suas informações ao longo da pesquisa: privacidade, respeito às diferenças, respeito à propriedade intelectual, uso de dados, o direito à informação por parte dos participantes das pesquisas, entre outras.

No Brasil, a preocupação com a participação de seres vivos, seres humanos e o uso de suas informações em processos de pesquisa é assunto de legislação específica, que precisa ser conhecida pelos pesquisadores em geral, principalmente os de Sistemas de Informação, que lidam com artefatos tecnológicos em uso em organizações e na sociedade.

1.4. Regulamentação ética da pesquisa envolvendo humanos

No Brasil, em 1996 foi editada a **Resolução 196 do Conselho Nacional de Saúde** [CNS, 1996], criando a CONEP - Comissão Nacional de Ética em Pesquisa e um sistema que se divide em comitês, presentes em instituições que promovem a gestão e o desenvolvimento de pesquisas.

A CONEP tem um papel coordenador da rede de Comitês de Ética em Pesquisa – CEPs, criados nas instituições, com os quais forma o Sistema CEP-CONEP. Constitui-se também, em órgão consultor junto ao Ministério da Saúde e órgãos do SUS. Tem, ainda, a atribuição de apreciar projetos de pesquisa a serem desenvolvidos em áreas temáticas especiais.

A seguir trataremos das diretrizes e normas regulamentadoras de pesquisa envolvendo seres humanos, em especial a **Resolução 466, de 12 de dezembro de 2012 do Conselho Nacional de Saúde** [Brasil, 2012], bem assim a **Resolução 510, de 07 de abril de 2016, também do Conselho Nacional de Saúde** [Brasil, 2016], específica para as pesquisas que se relacionam às ciências humanas e sociais.

A Resolução CNS 466/2012 parte das premissas já exploradas neste capítulo, de respeito à dignidade humana e do imanente progresso científico e tecnológico, já nas disposições preliminares esclarece que incorpora, sob a ótica do indivíduo e das coletividades referenciais da bioética: autonomia, não maleficência, beneficência, justiça e equidade, dentre outros, determinando, em seguida, que os projetos de pesquisa envolvendo seres humanos devem atender o disposto na resolução.

Iniciando-se com os termos e definições utilizados na resolução, a Resolução CNS 466/2012 menciona os aspectos éticos da pesquisa envolvendo os seres humanos, mencionando que a eticidade importa (item III.1): a) respeito ao participante da pesquisa em sua dignidade e autonomia, reconhecendo sua vulnerabilidade, assegurando sua vontade de contribuir e permanecer, ou não, na pesquisa, por intermédio de manifestação expressa, livre e esclarecida; b) ponderação entre riscos e benefícios, tanto conhecidos como potenciais, individuais ou coletivos, comprometendo-se com o máximo de benefícios e o mínimo de danos e riscos; c) garantia de que danos previsíveis serão evitados; e d) relevância social da pesquisa, o que garante a igual consideração dos interesses envolvidos, não perdendo o sentido de sua destinação sócio-humanitária.

Sendo assim e partindo destes pressupostos, fica clara a exigência de que a pesquisa deve ser pautada no respeito ao ser humano, sua vontade, sua correta ciência do consentimento e que não o coloque diante de riscos desnecessários. Posto isso, um projeto detalhado da pesquisa será necessário para responder às exigências previstas no item III. 2 que se relacionam aos fundamentos teóricos da pesquisa, eventuais experimentações prévias, métodos, benefícios e riscos que envolvem a pesquisa, a composição dos grupos que farão parte da pesquisa, o consentimento livre e esclarecido, os recursos humanos e materiais necessários à realização da pesquisa, os procedimentos que serão realizados, bem como expõe preocupações com a pesquisa internacional (para assegurar a soberania nacional) e com mulheres grávidas e seus direitos reprodutivos. As metodologias experimentais, especialmente na área biomédica, foram especificamente tratadas (item III.3).

Importante ressaltar que ainda há menções específicas ao respeito aos valores culturais, sociais, morais, religiosos e éticos, como também hábitos e costumes, quando envolverem comunidades (item III.2, k); bem assim que pesquisas em comunidades devem continuar com seus efeitos quando representar benefícios após a sua conclusão (item III.2, l), ressaltando a responsabilidade social do pesquisador, bem como os valores de igualdade.

A pesquisa científica não deve ser um fim em si mesma, mas deve ter relação com os propósitos relacionados ao progresso da ciência, sem descurar dos valores humanos, muitas vezes aqui já mencionados. Assim, a relevância social é item imprescindível na análise ética.

Com a finalidade de garantir ao participante da pesquisa a total ciência da pesquisa com que colaborará, a Resolução ainda disciplina o processo de consentimento livre e esclarecido, impondo os itens obrigatórios que devem ser inseridos no Termo de Consentimento Livre e Esclarecido (TCLE), pautados na liberdade do consentimento. Vale citar que a dispensa do TCLE deve ser justificada em razão da sua inviabilidade ou de riscos substanciais à privacidade e confidencialidade dos dados dos participantes ou aos vínculos de confiança entre pesquisador e pesquisado (Item IV. 8).

Quanto aos riscos e benefícios, a Resolução 466/2012 é clara ao dispor: "Toda pesquisa com seres humanos envolve risco em tipos e gradações variados. Quanto maiores e mais evidentes os riscos, maiores devem ser os cuidados para minimizá-los e a proteção oferecida pelo Sistema CEP/CONEP aos participantes. Devem ser analisadas possibilidades de danos imediatos ou posteriores, no plano individual ou coletivo. A análise de risco é componente imprescindível à análise ética, dela decorrendo o plano de monitoramento que deve ser oferecido pelo Sistema CEP/CONEP em cada caso específico" (item V). Desta forma, os riscos podem ser escalonados em riscos mínimo, médio ou máximo e apenas são admissíveis quando se justifique pelo benefício esperado, garantindo-se ao participante da pesquisa a indenização em caso de danos.

A Resolução CNS 466/2012 ainda traz as atribuições do Sistema CEP/CONEP, as competências e o procedimento de análise ética, bem assim a responsabilidade do pesquisador, consolidando um sistema de múltiplas responsabilidades. No item XIII.3, a citada Resolução esclarece que "As especificidades éticas das pesquisas nas ciências sociais e humanas e de outras que se utilizam de metodologias próprias dessas áreas serão contempladas em resolução complementar, dadas suas particularidades".

Em cumprimento, foi editada a Resolução CNS 510/2016 para tratar das especificidades das pesquisas realizadas no âmbito das Ciências Humanas e Sociais, justificando que "nelas prevalece uma acepção pluralista de ciência na qual decorre a adoção de múltiplas perspectivas teórico-metodológicas, bem como lidam com atribuições de significados, práticas e representações, sem intervenção direta no corpo humano, com natureza e grau de risco específico".

Importante mencionar que a Resolução CNS 466/2012 continua aplicável no que couber (ou seja, no que não for contraditório), sendo utilizada, inclusive, para suprir eventuais lacunas da Resolução CNS 510/2016.

A exemplo da Resolução CNS 466/2012, a Resolução CNS 510/2016 também destina capítulo específico aos termos e definições, bem como aos princípios éticos aplicáveis às ciências humanas e sociais, com vistas às garantias de confidencialidade e privacidade e às garantias quanto ao uso de imagem e voz. Há especial preocupação quanto aos procedimentos de consentimento e assentimento livre e esclarecido e da análise dos riscos, que deve ser graduada observando-se os procedimentos metodológicos. Não se omite em relação aos procedimentos de análise ética e da responsabilidade do pesquisador.

Se, por um lado a CNS 466/2012 aborda questões relacionadas à necessidade de submeter projetos de pesquisa que envolvam seres humanos para análise ética, vale frisar que a principal novidade instituída pela Resolução CNS 510/2016 está nas hipóteses de dispensa autorizadas pelo parágrafo único do artigo 1º que esclarece que não serão registradas nem avaliadas pelo sistema CEP/CONEP: I - pesquisa de opinião pública com participantes não identificados; II- pesquisa que utilize informações de acesso público, nos termos da **Lei no 12.527, de 18 de novembro de 2011 (Lei do Acesso à Informação)** [Brasil, 2011]; III - pesquisa que utilize informações de domínio público; IV - pesquisa censitária; V - pesquisa com bancos de dados, cujas informações são agregadas, sem possibilidade de identificação individual; e VI - pesquisa realizada exclusivamente com textos científicos para revisão da literatura científica; VII - pesquisa que objetiva o aprofundamento teórico de situações que emergem espontânea e contingencialmente na prática profissional, desde que não revelem dados que possam identificar o sujeito; e IX - atividade realizada com o intuito exclusivamente de educação, ensino ou treinamento sem finalidade de pesquisa científica, de alunos de graduação, de curso técnico, ou de profissionais em especialização. As situações não são exemplificativas, mas taxativas, não devendo ser interpretadas de maneira ampliativa.

Os parágrafos 1º e 2º ainda salientam que as exclusões das atividades de ensino, educação e treinamento não excluem os trabalhos de conclusão de curso e equivalentes, bem como que se, no decorrer da atividades, surgir a intenção de utilização dos dados para pesquisa, deve ser procedido o protocolo no sistema CEP/CONEP. Desta forma, se reafirma que a preocupação se dá com as atividades de pesquisa e utilização de dados para este fim, deixando ao professor a responsabilidade pelas atividades típicas de sala de aula. Cabe refletir que a normativa exclui da apreciação ética as pesquisas de opinião pública, que, inclusive, não deve incluir a análise de dados sensíveis.

O artigo 5º, II, da **Lei nº 13.709, de 14 de agosto de 2018, conhecida como a Lei Geral de Proteção de Dados Pessoais – LGPD** [Brasil, 2018], prescreve que constitui dado pessoal sensível: "dado pessoal sobre origem racial ou étnica, convicção religiosa, opinião política, filiação a sindicato ou a organização de caráter religioso, filosófico ou

político, dado referente à saúde ou à vida sexual, dado genético ou biométrico, quando vinculado a uma pessoa natural".

A LGPD visa à proteção e disciplina do tratamento de dados pessoais, inclusive por meios digitais e tem algumas correspondências com a análise ética desempenhada pelos CEPs no que se refere ao respeito à privacidade, à autodeterminação informativa e à inviolabilidade da intimidade, honra e imagem, tida por fundamentos da proteção de dados pessoais. Confirma alguns pressupostos já assegurados nas normativas do CNS, disciplinando a possibilidade de dados para fins de pesquisa, assegurando que, sempre que possível, os dados pessoais devem ser anonimizados (art. 7º, IV; art. 11, II, 'c' e art. 16, II). Salientando sempre que o consentimento é fator indispensável para a pesquisa e utilização dos dados pessoais. A matéria não será aqui aprofundada, na medida em que comportaria um segundo minicurso, tal sua importância reflexiva.

Para encerrar o presente tópico, vale mencionar que há, em tramitação no Congresso Nacional, o **Projeto de Lei n. 7.082/2017** [Câmara de Deputados, 2017] que visa dar nova regulamentação à pesquisa clínica com seres humanos (o que refletirá em outras áreas de pesquisa, ao certo), instituindo, inclusive, o Sistema Nacional de Ética em Pesquisa Clínica com Seres Humanos.

1.4.1. Os Comitês de Ética em Pesquisa

Os Comitês de Ética cumprem a missão de zelar pela proteção aos sujeitos da pesquisa em nome da sociedade e de forma independente (*múnus público*) ao qualificar eticamente os projetos. Tornam-se, assim, corresponsáveis pela parte ética, juntamente com: o pesquisador (cuja responsabilidade é indelegável e intransferível), a instituição e o patrocinador, para assegurar o respeito aos direitos dos sujeitos de pesquisa.

Atualmente, há 848 (oitocentos e quarenta e oito) comitês de ética em pesquisa em funcionamento no Brasil [CNS, 2020]. O sistema "busca universalizar a fiscalização das pesquisas e a manutenção dos direitos humanos, como prerrogativa de todos os membros da sociedade, o que é considerada também uma ação educativa e de controle social no campo científico". A obrigação para o respeito à ética deve ser estimulada por toda a sociedade, responsável pela eficácia horizontal dos direitos fundamentais, independentemente da origem dos recursos.

Além disso, a Universidade e as instituições de pesquisa devem ter especial cuidado com as pesquisas geridas em seu bojo, haja vista que independentemente do fomento público das pesquisas em si, há a obrigação social advinda da sua função enquanto instituição secular. A análise ética prévia de projetos de pesquisa visa assegurar a integridade do participante de pesquisa, mas também contribui para o desenvolvimento da ciência em um ambiente de respeito ao bom uso dos recursos públicos.

É possível sintetizar as características do CEP: (i) diferem dos comitês de ética hospitalar em sua composição, suas funções e suas normas; (ii) não são compostos somente de cientistas naturais, incluindo representantes das disciplinas sociais e da comunidade; (iii) a participação de outros profissionais ou membros da comunidade não se rege por um princípio de representatividade, mas sim de idoneidade; (iv) seguindo o

modelo dos comitês institucionais de revisão ética, prefere-se o comitê de ética local, que conhece sua própria instituição e seus pesquisadores, podendo convocá-los com mais facilidade para levar adiante a pesquisa; (v) os comitês de ética em pesquisa são duplamente obrigatórios: toda pesquisa deve ser revisada por eles, e todo pesquisador deve acatar as correções éticas que o comitê exigir; (vi) a deliberação do comitê de ética em pesquisa não apenas garante a conformidade com normas gerais como também analisa individualmente cada protocolo; (vii) os comitês de ética em pesquisa asseguram o consentimento livre e esclarecido, a proporcionalidade dos riscos, os detalhes do método científico que possam incidir em riscos, os aspectos econômicos que velam pela probidade e a utilização pertinente dos resultados; (viii) os comitês de ética em pesquisa devem funcionar de forma regulamentada e documentada, tanto para fundamentar suas deliberações quanto para criar jurisprudência [Kottow, 2008].

Um dos principais papéis desempenhados pelos CEPs consiste no recebimento e análise de projetos de pesquisa que envolvam seres humanos, sob o aspecto ético. Assim, o olhar do comitê sempre se respalda na proteção ao participante da pesquisa desde o seu recrutamento, acolhimento e cuidados na condução da pesquisa, esclarecimentos sobre os procedimentos e acesso a informações. Sua multidisciplinaridade reflete o próprio sistema de análise ética e a representação de caráter social, suprindo, inclusive, a recomendação contida na Resolução CNS 510/2016.

1.5. Mas, por que submeter meu projeto ao comitê de ética?

A pesquisa na área de Sistemas de Informação tem uma característica aplicada e intervencionista, no sentido de que se preocupa em estudar a concepção, construção, uso e impactos de artefatos tecnológicos em contexto organizacionais e sociais. Pessoas, processos, organizações e tecnologias, encaradas sob um olhar sistêmico são o foco das pesquisas nesta área.

Araujo, Fornazin e Pimentel (2017) analisaram as pesquisas publicadas na Revista Brasileira de Sistemas de Informação (iSys)[2] no período de 2008 a 2017 (10 anos) e constataram a grande ênfase das pesquisas desta comunidade na produção de artefatos tecnológicos. Embora a pesquisa nesta área possa ter forte componente de construção de produtos, frequentemente, será parte fundamental da pesquisa a validação destes artefatos em uso por pessoas representativas de seus usuários. Araujo, Fornazin e Pimentel (2017) ainda enfatizam a importância e oportunidade da comunidade de pesquisa em desenvolver investigações de caráter sociotécnico como forma de estudar a complexidade e as relações entre tecnologia e contextos organizacionais e sociais. Desenvolver estudos de caso, pesquisa-ação e observações de contextos sociais e organizacionais será cada vez mais necessário na área, mediante a cada vez mais forte relação entre tecnologia e pessoas na atualidade, derivadas do intenso processo de digitalização de atividades.

No documento que retrata os grandes desafios para a pesquisa em SI no Brasil até 2026 [Boscarioli et. al. 2017], o termo "ethical" surge 19 vezes, principalmente associado aos desafios "Sistemas de Informação e os Desafios do Mundo Aberto" e "Visão Sociotécnica de Sistemas de Informação". Entretanto, uma busca por palavras-chave "ética/ético" ou "ethics/ethical" nas bases de artigos dos principais canais de

[2] https://sol.sbc.org.br/index.php/sbsi

disseminação científica nacionais na área de SI - o Simpósio Brasileiro de Sistemas de Informação e a Revista Brasileira de Sistemas de Informação[3] não retorna nenhum resultado.

Não temos um levantamento das razões pelas quais os pesquisadores nacionais na área de SI não estão submetendo seus projetos à avaliação dos CEPs, similar, por exemplo, ao realizado por pesquisadores na área de IHC [Amorim et al, 2019], porém, é possível identificar algumas questões principais. A primeira delas, o desconhecimento da necessidade da submissão, por não identificar em sua pesquisa riscos éticos, embora com a participação de humanos. Conforme veremos em alguns casos a seguir, riscos éticos existem de forma direta e indireta, em situações que os CEPs terão, em geral, condições de identificá-los junto ao pesquisador. Outra questão é o entendimento e interpretação superficial das resoluções, principalmente da Resolução 510, supondo que o projeto possa ser dispensado de aprovação ética. Há também o receio da burocracia e da necessidade de planejamento antecipado e detalhado das atividades do projeto para submissão ao CEP, quando, em geral, os projetos possuem dinâmicas menos planejadas ou mais *ad-hoc*. Por fim, há o receio do atraso ou mesmo impedimento do desenvolvimento da pesquisa devido a pareceres negativos quanto às questões éticas.

No entanto, os prejuízos advindos de um projeto mal conduzido em relação aos seus aspectos éticos, podem ser consideráveis, envolvendo prejuízos ao direito e integridade alheios, passíveis de sanções legais. A submissão de projetos de pesquisa para aprovação dos CEPs é uma forma do pesquisador se resguardar a si e sua instituição de problemas e questionamentos legais, de proteger os participantes de sua pesquisa de prejuízos em sua participação desconhecidos pelo pesquisador, bem como garantir tranquilidade na comunicação de sua pesquisa para a sociedade. Vejamos alguns casos a seguir.

1.5.1 Casos envolvendo Ética

Nesta seção são apresentados alguns casos de projetos de pesquisa fictícios (mas que poderiam ser reais), a fim de se fazer compreender questões éticas envolvidas.

Caso 1: Era uma Pesquisa de Opinião Pública?

O pesquisador A, juntamente com seu grupo de pesquisa ligado à área de Ciências Sociais, resolveu iniciar uma pesquisa que caracterizou como de opinião pública, com início na Pandemia de Covid-19. De acordo com a Resolução 510, de 2016, "pesquisa de opinião pública: consulta verbal ou escrita de caráter pontual, realizada por meio de metodologia específica, através da qual o participante, é convidado a expressar sua preferência, avaliação ou o sentido que atribui a temas, atuação de pessoas e organizações, ou a produtos e serviços; sem possibilidade de identificação do participante".

Como o pesquisador tinha algum conhecimento sobre as Resoluções do CNS, foi examinar a Resolução 510, de 2016, e encontrou no artigo 1o, Parágrafo Único, Inciso I, que as pesquisas de opinião pública com participantes não identificados estariam isentas de passar pelo sistema CEP/CONEP. Entendendo que este era o caso, realizou a preparação do instrumento de coleta de dados juntamente com a equipe. Então, iniciou a

[3] https://sol.sbc.org.br/journals/index.php/isys

coleta com professores de sua universidade. Em dado momento da coleta, foi alertado por um membro do CEP de sua instituição que tomou ciência do instrumento, que possivelmente a sua pesquisa deveria ter passado por apreciação ética.

Analisando o que aconteceu: o projeto trata-se de uma pesquisa sobre saúde mental de professores em época de Pandemia do Covid-19. Ao analisar o instrumento de coleta de dados, percebeu-se que havia questões sobre dados considerados "sensíveis", ou seja, relacionados a questões psicológicas. Assim, o recrutamento dos participantes e a aplicação do instrumento de coleta de dados não poderia ter sido iniciado antes de sua aprovação ética. O pesquisador não submeteu o projeto ao CEP por má fé e sim por desconhecimento. Então ele decidiu desistir do projeto.

Como conclusão deste caso, pode-se perceber que há a necessidade de se verificar o tipo de questões que serão tratadas em um instrumento de coleta de dados. Uma pesquisa com um instrumento que capte, por exemplo, opinião sobre determinado software como Facebook, sem identificação dos participantes, poderia estar isenta de necessidade de aprovação ética. Já uma pesquisa, mesmo que sem identificação dos participantes, que envolva questões "sensíveis", como o caso de questões psicológicas, necessitam passar por aprovação ética.

Caso 2: Usava Dados Públicos?

O pesquisador B iniciou uma pesquisa sobre episódios de *Cyberbullying* a partir de informações não identificadas em redes sociais abertas. Então, o pesquisador ficou em dúvida se deveria ou não enviar seu projeto de pesquisa para apreciação ética. Lembrava-se que havia uma resolução que previa alguns casos em que projetos estavam dispensados de tramitar pelo sistema CEP/CONEP, então realizou uma busca. De acordo com a Resolução 510, de 2016, artigo 1o, Parágrafo Único, Inciso II, pesquisa que utilize informações de acesso público não precisam passar pela aprovação pelo sistema CEP/CONEP. Então, o pesquisador se tranquilizou e deu andamento em sua pesquisa.

Enviou o artigo com os resultados da pesquisa para uma revista científica e teve uma surpresa. A revista negou seu artigo argumentando que o projeto usava informações sensíveis disponíveis em redes sociais e, que, como usava trechos transcritos das informações coletadas, era possível rastrear tais informações e identificar o participante. Informações públicas não se confundem com informações publicizadas. É importante a segurança e privacidade dos dados pessoais dos participantes da pesquisa, se eles podem de alguma forma ser identificados, devem consentir e ser protegidos. Assim, conclui-se o pesquisador, mesmo para casos de dados públicos (ou publicados) ter a certeza de que não serão identificados os participantes da pesquisa. Para resolver tal problema, ele poderia indicar o teor das informações coletadas sem, contudo, apresentar trechos transcritos e assim garantir o anonimato e a privacidade.

Caso 3: Era uma Pesquisa de Opinião Pública ou um Teste de Usabilidade?

O pesquisador C queria fazer a validação de um software através de testes de usabilidade, envolvendo potenciais usuários. Ouviu dizer que pesquisa de opinião pública não necessitava passar pela aprovação ética do CEP, então iniciou os seus testes. Além disso, o software utiliza capacetes de visualização, em que o usuário fica imerso no ambiente

virtual e perde a noção do mundo real, momentaneamente. Dos 20 participantes da pesquisa, dois se sentiram bastante desconfortáveis com o dispositivo de visualização e um chegou a ter fortes náuseas. O pesquisador não sabia que medidas deveria tomar, porque não previu tal risco.

Assim, como o disposto no Caso 1, pesquisa de opinião pública a respeito de softwares não precisam tramitar no sistema CEP/CONEP. No entanto, parece haver um engano no que seja pesquisa de opinião pública e avaliação com participação de usuário quando em contato com a tecnologia. Assim, o avaliador deve garantir que os direitos dos participantes sejam garantidos, além do respeito a seus limites. Entre tais direitos estão a confidencialidade dos dados do participante e que o uso dos dados será restrito aos propósitos declarados [Amorim et al, 2019] [Sharp et al., 2011].

Também é bastante comum que seja utilizada tecnologia para realizar avaliações ou intervenções na área de Saúde. Assim, mais do que nunca, é necessária a conscientização da necessidade de se tramitar tais projetos de pesquisa para aprovação do CEP. Como conclusão sobre este caso, pode-se pensar que dar opinião sobre um software como Facebook (preferências de uso, design, etc.) é bastante diferente de se solicitar que um participante teste um software específico. Dar a opinião sobre um software estaria isento de aprovação ética, pois se enquadraria como pesquisa de opinião; por outro lado, um projeto de pesquisa que solicite ao participante que teste um determinado software deve tramitar pelo CEP.

Caso 4: Era uma pesquisa, que assegurava privacidade, envolvendo empresa?

O pesquisador D queria conduzir sua pesquisa através de entrevistas com gestores, especificamente, com gerentes de projetos de uma empresa na área de Tecnologia da Informação. Queria conhecer, dentro do ambiente de trabalho, as questões relacionadas à qualidade de vida das equipes de desenvolvimento de software, ou seja, questões sobre a quantidade de horas trabalhadas das equipes, pagamento de hora extra/ banco de horas das equipes, horas de descanso, relacionamento hierárquico, entre outros. Não achou que era importante ou necessário que o projeto fosse aprovado pelo CEP. E, assim, conduziu a pesquisa com os quatro gerentes de projeto, sem identificação, e publicou os resultados numa revista científica brasileira. No entanto, nos dados da empresa, embora não constasse o nome, eram exibidas a cidade da empresa e descrições que permitiam identificá-la, por ser a única empresa do ramo na pequena cidade com aquelas características. Assim, foi possível mapear quem eram os gerentes de projetos e, pelos dados coletados sobre o perfil, a identificação de cada um. Dois deles foram demitidos. A privacidade das informações foi falha e não foi possível que o CEP pudesse ter tido um olhar crítico sobre o projeto a fim de alertar o pesquisador. A conclusão sobre este caso é que o pesquisador deve atentar-se ao tipo de dados que ele possa estar coletando que possam colocar em risco o participante da sua pesquisa. Assim, ter tramitado este projeto pelo CEP, poderia ter trazido um feedback a respeito do seu problema.

Caso 5: Ajudinha para a pesquisa no ambiente acadêmico, que mal tem?

Na época da pandemia pelo Covid-19, o pesquisador E quis verificar como andava a saúde mental dos estudantes de sua universidade. Passou o projeto pelo sistema CEP/CONEP e como tinha competência para atender a qualquer problema decorrente da pesquisa e os

documentos estivessem corretos, com a metodologia bastante clara, recebeu aprovação para esta pesquisa. Primeiramente, começou a pressionar os alunos de seus cursos e também os professores da sua equipe. Como, ainda assim, o número de participantes estivesse aquém de suas expectativas, passou a oferecer horas de atividade complementar a todos para responderem ao questionário. Em tal caso, vê-se dois problemas éticos: o primeiro é que todo participante é livre para decidir se quer ou não participar da pesquisa e, ainda mais, pode desistir a qualquer momento [Brasil, 2012]. O segundo problema surge à medida que o participante não pode ter custos, mas, tampouco privilégios advindos de sua participação na pesquisa. É autorizada a compensação material, exclusivamente de despesas do participante e seus acompanhantes, quando necessário, tais como transporte e alimentação [Brasil, 2021]. Como conclusão deste caso, percebe-se que o pesquisador deve lançar mão de outras ferramentas (tais como ampla divulgação) para conseguir o número almejado de participantes de pesquisa, sendo vetado compensação material a não ser o que está previsto em lei.

Caso 6: A revista pediu e agora?

O pesquisador F estava disposto a fazer uma pesquisa e se tratava claramente de pesquisa de opinião pública sobre redes sociais. Tratava-se de obter informações sobre facilidade de uso, preferências e sentimento de segurança do ambiente. Então, o pesquisador, apoiado na Resolução 510, de 2016, e encontrou no artigo 1o, Parágrafo Único, Inciso I, que as pesquisas de opinião pública com participantes não identificados estariam isentas de passar pelo sistema CEP/CONEP. Assim, realizou sua pesquisa e escreveu um artigo com os resultados. Quando foi realizar a submissão do artigo para a revista científica, esta pedia o número de aprovação ética. Então o pesquisador ficou impedido de submeter o seu artigo para tal revista, ficando bastante frustrado. A conclusão para este caso é que, caso haja dúvida quanto à necessidade de apresentação do número de aprovação ética solicitado pela revista (ou por desconhecer para qual revista científica o trabalho será submetido a priori), é recomendável que o projeto passe pelo Sistema CEP/CONEP.

1.6. Como submeter meu projeto ao Comitê de Ética em Pesquisa?

Nesta seção, apresentaremos os passos e as informações necessárias para a submissão de projetos de pesquisa aos CEPs institucionais e a plataforma principal de tramitação de processos - a Plataforma Brasil.

1.6.1. Plataforma Brasil

Para solicitar aprovação ética do seu projeto de pesquisa através do sistema CEP/CONEP, o pesquisador deve tramitar seus documentos pela Plataforma Brasil (http://plataformabrasil.saude.gov.br/). Esta plataforma é onde se concentram todos os projetos de pesquisa que envolvam seres humanos que tenham sido aprovados ou estejam em fase de tramitação para sua aprovação ética. Permite a automatização dos processos que são necessários para a apreciação dos protocolos de pesquisa, além de ser possível, assim, estabelecer uma uniformidade das informações que devem ser entregues.

A utilização da Plataforma Brasil visa dar maior transparência aos processos de análise ética, bem assim vincula a todos ao cumprimento e acompanhamento de prazos e gestão da informação dos projetos tanto pelo pesquisador, quanto do próprio sistema

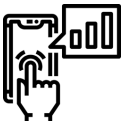
CEP/CONEP. A plataforma permite que as pesquisas sejam acompanhadas em diferentes estágios, desde sua submissão até a sua aprovação final pelo CEP e pela CONEP [Ministério da Saúde, 2020].

1.6.2. Documentos necessários

O pesquisador, quando deseja que seu projeto tenha a aprovação do CEP, deve submeter uma série de documentos (Tabela 5). Estes documentos devem ter como objetivo fundamental trazer informações que dêem os esclarecimentos necessários para que o parecerista compreenda as fases do projeto e que permita a análise dos documentos à luz do correto tratamento do participante da pesquisa. É possível acessar mais informações na página da CONEP [CNS, 2020][CNS, 2018] que facilitam o entendimento destes documentos. Basicamente, são exigidos:

Tabela 5. Documentos exigidos para submissão de projetos.

Informações/Documentos	Descrição/Comentários
 Cadastro do pesquisador	Em algumas instituições, o projeto deve estar vinculado ao pesquisador com vínculo empregatício. Contém informações básicas do pesquisador, assim como documentos, foto e link para o currículo Lattes.
 Cadastro da instituição	Cadastro da instituição a que o pesquisador se relaciona.
 Dados do Projeto	Preenchimento, na Plataforma Brasil, de informações básicas do projeto, tais como: metodologia do estudo, desenho, resumo, quantidade de participantes, cronograma, financiamento, entre outros. Deve estar em consonância com as informações fornecidas no projeto detalhado e demais documentos.
 Folha de Rosto	Folha de rosto assinada pelo representante legal da instituição. Esta folha é gerada na última página de preenchimento do projeto na Plataforma Brasil.

 Modelo dos Termos de Consentimento Livre e Esclarecido (TCLE)	Modelo dos Termos de Consentimento Livre e Esclarecido (TCLE) a ser utilizado para solicitar consentimento dos participantes da pesquisa. De acordo com a Norma Operacional No 1, de 2013 [Brasil, 2013], o TCLE "é um documento público específico para cada pesquisa, incluindo informações sobre as circunstâncias sob as quais o consentimento será obtido, sobre o responsável por obtê-lo e a natureza da informação a ser fornecida aos participantes da pesquisa, ou a dispensa do TCLE deve ser justificadamente solicitada pelo pesquisador responsável ao Sistema CEP/CONEP, para apreciação". Os TCLEs devem conter informações que esclareçam ao participante detalhes da pesquisa em que ele participará: passos envolvidos, tempo necessário, local, riscos (e sua forma de mitigá-lo) e benefícios da pesquisa, garantia do sigilo das informações, que as informações serão guardadas por 5 anos, acesso aos resultados da pesquisa, acesso a uma cópia do termo, possibilidade de desistência da pesquisa em qualquer momento, indicação de que os participantes não terão custos. Havendo custos (por exemplo, de deslocamento), deve estar previsto ressarcimento. Neste caso, deve haver orçamento previsto no projeto.
 Modelos dos Termos de Assentimento Livre e Esclarecido (TALE)	Modelos dos Termos de Assentimento Livre e Esclarecido (TALE), caso o participante da pesquisa seja menor de 18 anos ou incapazes. De acordo com a Resolução 510, de 2016 [Brasil, 2016], TALE é "anuência do participante da pesquisa –criança, adolescente ou indivíduos impedidos de forma temporária ou não de consentir, na medida de sua compreensão e respeitadas suas singularidades, após esclarecimento sobre a natureza da pesquisa, justificativa, objetivos, métodos, potenciais benefícios e riscos. A obtenção do assentimento não elimina a necessidade do consentimento do responsável".
 Projeto Detalhado	Projeto que contém as informações de toda a pesquisa, desde introdução, referencial teórico, objetivos, método, procedimentos para a análise de dados, cronograma e referências. O projeto deve explicitar a capacidade da equipe para executar o projeto. Assim, também, os textos e os métodos devem ser claros o suficiente para permitir uma compreensão completa pelo parecerista.
 Instrumentos de Coleta de Dados	Descrição dos instrumentos de coleta de dados, com as questões que serão perguntadas aos participantes da pesquisa.
 Anuência da Instituição	Carta em que a Instituição onde acontecerão as coletas de dados dá permissão para que a pesquisa aconteça.

Outros documentos	Outros documentos que possam compor o completo entendimento do projeto ou que são solicitados pelo CEP em que o projeto tramitará.

1.6.3. Tramitação dos projetos

Depois de inseridos os documentos na Plataforma Brasil e enviados, estes documentos serão recebidos pelo CEP de sua instituição (caso a instituição não esteja cadastrada, o projeto será primeiramente destinado à CONEP). Estes documentos passarão, então, por validação documental (até 10 dias). Se aprovados, serão encaminhados para um parecerista que emitirá um parecer que será discutido na próxima reunião do CEP. Na reunião, o CEP decidirá se o projeto está aprovado, reprovado ou aprovado com pendências.

Será, então, emitido um parecer consubstanciado (até 30 dias a contar da reunião) que o pesquisador terá acesso. Se houver pendências, o pesquisador deve realizar as modificações pedidas e responder ao CEP (em até 30 dias). O projeto, juntamente com a carta de encaminhamento de respostas, é analisado novamente e novo parecer consubstanciado é emitido. Se todas as pendências forem atendidas, o projeto está autorizado a começar [Brasil, 2013]. Após aprovação do protocolo de pesquisa, torna-se de responsabilidade do pesquisador o envio de relatórios parciais (semestralmente), e do relatório final, quando o estudo for finalizado. O pesquisador deve se atentar sobre os prazos necessários para a aprovação do seu projeto de pesquisa a fim de não comprometer seu correto andamento. A seguir é apresentado um fluxograma (Figura 1) que apresenta os passos até a aprovação.

Alguns tipos de projetos, por sua própria natureza, têm uma tramitação diferente do que foi apresentado. Um primeiro exemplo é um projeto que tenha coparticipantes; neste caso, é necessário que ele tramite por todos os CEPs das instituições participantes da pesquisa.

Por outro lado, pesquisas internacionais, quando do preenchimento da Plataforma Brasil, deve ser feita a indicação do estudo internacional. Desta maneira, o projeto será enquadrado na área "Pesquisa com Cooperação Estrangeira" da CONEP (ver Resolução CNS 292/99) [Brasil, 1999], cabendo à CONEP a aprovação do projeto depois da aprovação do CEP local.

Finalmente, projetos com temas de áreas temáticas especiais, de acordo com a Resolução 466/2012-CNS, item IX.4, devem ter apreciação ética do CEP e, também, da CONEP, entre eles: genética humana, reprodução humana, equipamentos e dispositivos terapêuticos, novos ou não registrados no país, novos procedimentos terapêuticos invasivos, estudos com populações indígenas, projetos que envolvam organismos geneticamente modificados, células-tronco embrionárias e organismos que representem alto risco coletivo, protocolos de constituição e funcionamento de biobancos para fins de pesquisa, pesquisas com coordenação e/ou patrocínio originados fora do Brasil, excetuadas aquelas com copatrocínio do Governo Brasileiro. É importante destacar que

estes projetos supracitados devem ser identificados como de "Área Temática Especial" durante o preenchimento dos dados na Plataforma Brasil.

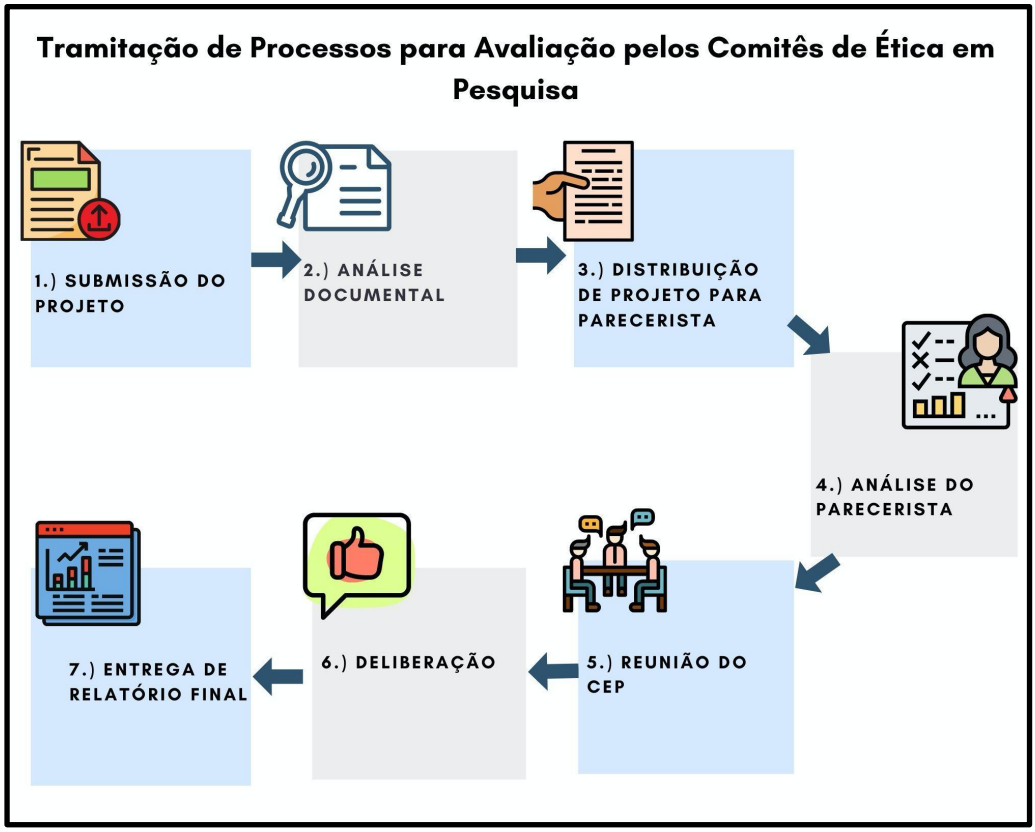


Figura 1. Fluxo de tramitação de processos. Fonte: as autoras.

1.6.4. Pendências e Reprovação

Com base na experiência das autoras, reunimos nesta seção alguns pontos que podem comprometer o projeto de pesquisa e gerar pendências ao projeto submetido à aprovação ética, resumidos na Figura 2. Tanto nas reprovações, como na indicação de pendências, o objetivo da avaliação é garantir que o projeto, o pesquisador e os participantes da pesquisa estejam protegidos, de acordo com a legislação.

Embora não seja comum, a reprovação de um projeto ocorre quando os riscos para os participantes se mostram insuperáveis - os danos às dimensões física, psíquica, moral, intelectual, social, cultural, espiritual são graves e impossíveis de serem minimizados. O projeto está também passível de reprovação por sua incapacidade de apresentar seus objetivos de forma compreensível, em razão de falta de dados ou imensa quantidade de informações conflitantes.

Em relação a pendências, boa parte dos problemas que as geram diz respeito à ausência de informações. Pendências documentais (vide seção 1.6.2) são apontadas após a submissão do projeto e pendências mais específicas, por meio da análise do parecerista e membros do CEP. As pendências são comumente geradas em virtude da pouca experiência ou maturidade dos projetos em relação aos procedimentos éticos,

principalmente no que envolve o planejamento da proteção aos riscos, base de toda a orientação ética.

Itens muito comuns a gerar pendências compreendem o baixo detalhamento do planejamento do projeto a respeito do desenvolvimento das atividades da pesquisa em relação às preocupações éticas: falta de informações sobre a forma de recrutamento dos participantes; falta de local onde será realizada a pesquisa (falta de informações sobre se haverá deslocamento do participante e se o local protege sua confidencialidade); falta de informações sobre como será assegurada a não identificação dos participantes e se as informações coletadas são confidenciais; falta de descrição dos riscos, benefícios e ações para minimizar os riscos, caso ocorram; e, em caso de grupo controle, a falta de garantia de que os mesmos procedimentos serão oferecidos aos participantes, caso os resultados da pesquisa sejam satisfatórios. Percebe-se também muitas lacunas no esclarecimento ao participante quanto a: o método a ser aplicado e o tempo necessário, a indicação de ausência de custos aos participantes ou de ressarcimento; a indicação de suporte aos participantes e a indenização em caso de constatação de danos, nos termos da lei.

Outro ponto muito comum de geração de pendências é em relação aos TCLE e TALE, por exemplo: o uso de linguagem não adaptada ao contexto do participante; falta de numeração de páginas e de espaço para rubrica; falta da informação de que os dados serão guardados por, no mínimo, 5 anos; falta de informação de como será dada a devolutiva da pesquisa ao participante; falta de informação de que serão fornecidas duas vias (uma para o participante e outra para o pesquisador responsável); falta de informações sobre o CEP (endereço, função e horário de atendimento); entre outras.

Por fim, pendências podem ser geradas quando são identificadas informações conflitantes entre o número de participantes e grupos, cronograma e método nas Informações Básicas, Projeto Detalhado e Termos apresentados.

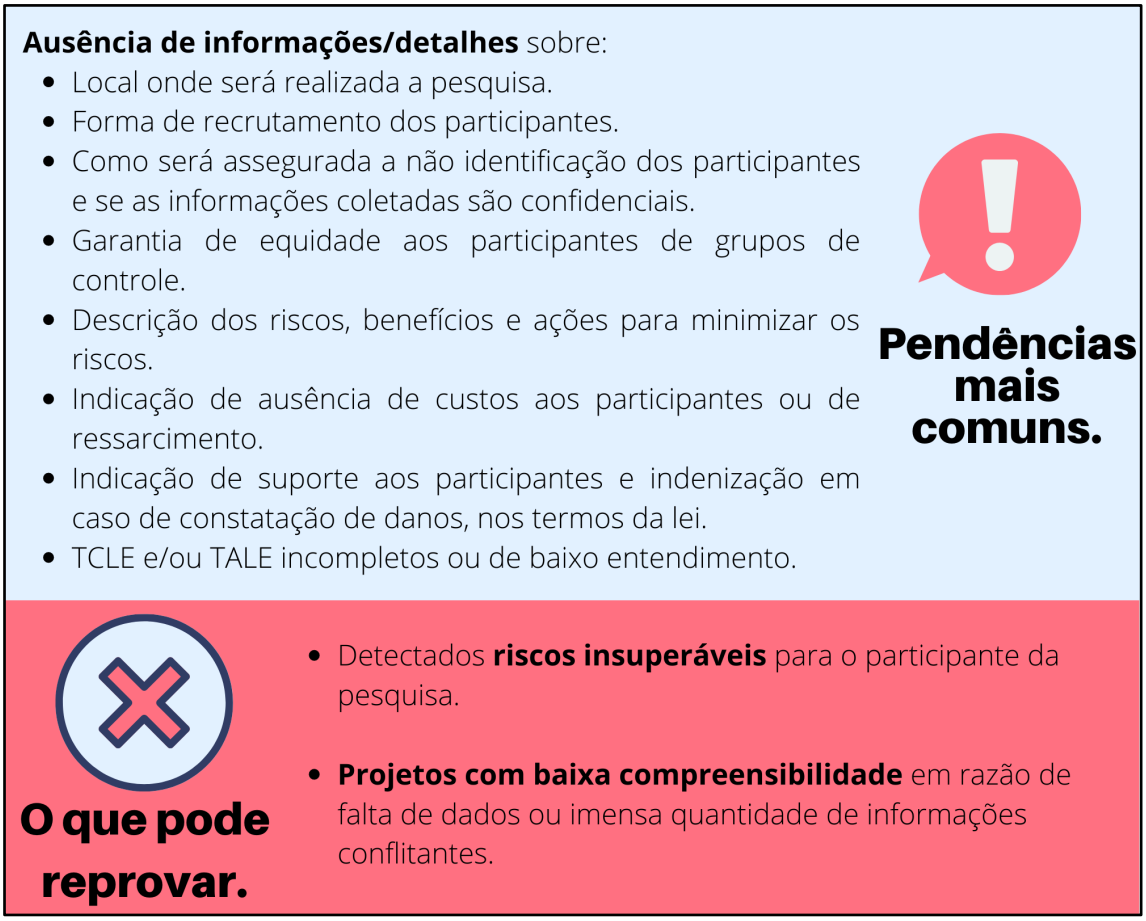


Figura 2. Principais aspectos que comprometem a submissão de um projeto à aprovação ética. Fonte: as autoras.

1.7 Conclusões

O tema da ética na Computação é bastante amplo e envolve questões curriculares, profissionais, jurídicas e sociais. Neste capítulo, colocamos nosso olhar nos desafios de pensar a ética nas pesquisas na área de SI que envolvem seres humanos. O capítulo foi palco para a discussão dos aspectos éticos de pesquisas envolvendo seres humanos no âmbito de Sistemas de Informação, assim como apresentar a sua importância. Apresentou uma visão geral das regulamentações que regem o CEP, assim como os trâmites para a submissão de projetos de pesquisa a este comitê para a sua aprovação ética. Abordou alguns casos fictícios, porém factíveis de acontecerem, a fim de sensibilizar o leitor.

Quando da análise ética de um projeto de pesquisa que envolva seres humanos, a principal questão que envolve um parecer estará sempre centrada na proteção ao participante da pesquisa, em um processo de empatia. Este olhar de proteção deve perpassar todas as etapas que envolvem o participante, desde o seu recrutamento, acolhimento e cuidados na condução da pesquisa, até esclarecimentos sobre os procedimentos e acesso a informações.

A preparação e submissão de um projeto de pesquisa que envolve seres humanos ao CEP traz à luz várias questões metodológicas em relação aos participantes da pesquisa

que podem passar despercebidas pelo pesquisador que não executa tal exercício. Também, através da discussão de possíveis riscos aos participantes que podem não ter sido considerados pelo pesquisador, além de alertá-lo sobre a responsabilidade em riscos ocorridos.

A área de pesquisa em Computação tem sido cada vez mais desafiada a construir soluções com impacto econômico, social e individual, em empresas, instituições e sociedade. Artigo recentemente publicado na Communications of ACM [Connolly, 2020], um dos periódicos mais representativos da área, discute até a visão de que a Computação seja, cada vez mais, uma área das Ciências Sociais. Também muito recentemente, a SBC publica uma série de livros denominados "Computação e Sociedade", com foco em temáticas multidisciplinares. A área de Sistemas de Informação, por sua vez, sempre se preocupou com os aspectos aplicados e os impactos da tecnologia nos contextos organizacionais e sociais, estimulando as abordagens sociotécnicas de pesquisa [Boscarioli et al., 2017].

Isto pode significar que esta área de pesquisa precisará, e muito, realizar projetos com resultados que impliquem na participação destes que são autores fundamentais de sua existência hoje - humanos. Para que estas pesquisas sejam bem sucedidas e tragam resultados reais e impactantes para realmente resolver os problemas que nos assolam como sociedade, as questões éticas precisarão ser compreendidas, aplicadas e aprofundadas por seus pesquisadores. Dizemos, aprofundadas, porque ainda não são perfeitas e exigem esforço de debate e evolução pela própria comunidade de pesquisa. Por outro lado, nós, pesquisadores, precisamos enxergar o processo de discussão ética para além da mera burocracia. A dimensão ética em nossos projetos merece uma mudança de mentalidade de pesquisa que passa por incluir uma reflexão crítica sobre as implicações de nossas pesquisas, o conhecimento da legislação pertinente ao tema, os procedimentos de diálogo com o CEP e, principalmente, um visão de planejamento que inclua essas atividades na programação de nossas ações de pesquisa.

Referências

ACM. Association for Computing Machinery. (2018). ACM Code of Ethics and Professional Conduct. https://ethics.acm.org/

AIS. Association for Information Systems. (2014). AIS Code of Research Conduct. Code of Research Conduct - Association for Information Systems (AIS) (aisnet.org)

Amorim, P. F., Sacramento, C., Capra, E. P., Tavares, P. Z., Ferreira, S. B. L. (2019). "Submit or Not My HCI Research Project to the Ethics Committee, That is the Question". 18th Brazilian Symposium on Human Factors in Computing Systems (IHC'19). Outubro 21-25, Vitória - ES, Brasil. ACM, New York, NY. https://doi.org/10.1145/3357155.3358473.

Araujo, R. M., Fornazin, M.; Pimentel, M. (2017). "An Analysis of the Production of Scientific Knowledge in Research Published in the First 10 years of iSys (2008-2017)". iSys - Brazilian Journal of Information Systems, Porto Alegre, v. 10, n. 4, p. 45-65, dez. 2017. ISSN 1984-2902. DOI: https://doi.org/10.5753/isys.2017.351

Bongertz, Vera. (1999). O dia-a-dia na pesquisa científica: considerações éticas. In: Carneiro, F. (Org.). A Moralidade dos Atos Cientificos – questões emergentes dos Comitês de Ética em Pesquisa, Rio de Janeiro, FIOCRUZ, 1999. http://www.dbbm.fiocruz.br/ghente/publicacoes/moralidade/dia_a_dia.pdf

Boscarioli, C., Araujo, R. M., Maciel, R. S. P. (2017). I GranDSI-BR – Grand Research Challenges in Information Systems in Brazil 2016-2026. Special Committee on Information Systems (CE-SI). Brazilian Computer Society (SBC). ISBN: [978-85-7669-384-0]. 184p. https://sol.sbc.org.br/livros/index.php/sbc/catalog/book/28

Brasil. (1988) Constituição da República Federativa do Brasil de 1988. http://www.planalto.gov.br/ccivil_03/constituicao/constituicaocompilado.htm

Brasil. (1999) Conselho Nacional de Saúde. Resolução nº 292, de 08 de julho de 1999. Aprova a norma no que diz respeito à área temática especial "pesquisas coordenadas do exterior ou com participação estrangeira e pesquisas que envolvam remessa de material biológico para o exterior". https://bvsms.saude.gov.br/bvs/saudelegis/cns/1999/res0292_08_07_1999.html

Brasil. (2011) Lei Nº 12.527, de 18 de novembro de 2011.http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2011/lei/l12527.htm Acesso em 08 de dezembro de 2020.

Brasil. (2012) Conselho Nacional de Saúde. Resolução nº 466, de 12 de dezembro de 2012. Aprova as diretrizes e normas regulamentadoras de pesquisas envolvendo seres humanos. Diário Oficial da União. Brasília, n. 12, p. 59, 13 jun 2013, Seção 1. https://conselho.saude.gov.br/resolucoes/2012/Reso466.pdf

Brasil. (2013) Norma Operacional nº 001/2013 Dispõe sobre a organização e funcionamento do Sistema CEP/CONEP, e sobre os procedimentos para submissão, avaliação e acompanhamento da pesquisa e de desenvolvimento envolvendo seres humanos no Brasil, nos termos do item 5, do Capítulo XIII, da Resolução CNS n° 466 de 12 de dezembro de 2012. CONEP: Resoluções. 2014. http://conselho.saude.gov.br/web_comissoes/conep/aquivos/CNS%20%20Norma%20Operacional%20001%20-%20conep%20finalizada%2030-09.pdf

Brasil. (2016) Conselho Nacional de Saúde. Resolução nº 510, de 07 de abril de 2016. Normas aplicáveis a pesquisas em Ciências Humanas e Sociais. Diário Oficial da União. Brasília, n. 98, p. 44-46, 24 mai 2016, Seção 1. http://conselho.saude.gov.br/images/comissoes/conep/documentos/NORMAS-RESOLUCOES/Resoluo_n_510_-_2016_-_Cincias_Humanas_e_Sociais.pdf

Brasil. (2018) Lei Geral de Proteção de Dados Pessoais (LGPD). http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm Acesso em 08 de dezembro de 2020.

Câmara dos Deputados. (2017). PL 7082/2017. Dispõe sobre a pesquisa clínica com seres humanos e institui o Sistema Nacional de Ética em Pesquisa Clínica com Seres Humanos. https://www.camara.leg.br/proposicoesWeb/fichadetramitacao?idProposicao=2125189 Acesso em 08 de dezembro de 2020.

Camillo, C. (2019). Manual da Teoria Geral do Direito. São Paulo: Almeida.

CNS. Conselho Nacional de Saúde. Comitês de Ética em Pesquisa. http://conselho.saude.gov.br/comites-de-etica-em-pesquisa-conep?view=default Acesso em 08 de dezembro de 2020.

CNS. Conselho Nacional de Saúde. (2018). Manual de Usuário - Pesquisador. https://conselho.saude.gov.br/images/comissoes/conep/documentos/PB/MANUAL_PESQUISADOR.pdf Acesso em 08 de dezembro de 2020.

CNS. Conselho Nacional de Saúde. (1996). Resolução do Conselho Nacional de Saúde No 196/1996. http://www.aids.gov.br/pt-br/legislacao/resolucao-do-conselho-nacional-de-saude-no-1961996#:~:text=Aprova%20diretrizes%20e%20normas%20reguladoras%20de%20pesquisas%20envolvendo%20seres%20humanos. Acesso em 08 de dezembro de 2020.

Connolly, B. (2020). "Why Computing Belongs Within the Social Sciences". Communications of the ACM, August 2020, Vol. 63 No. 8, Pages 54-59. https://doi.org/10.1145/3383444

CREMESP. Centro de Bioética. Código de Nuremberg. http://www.bioetica.org.br/?siteAcao=DiretrizesDeclaracoesIntegra&id=2 Acesso em 08 de dezembro de 2020.

CREMESPa. Centro de Bioética. Declaração de Helsinque. http://www.bioetica.org.br/?siteAcao=DiretrizesDeclaracoesIntegra&id=4 Acesso em 08 de dezembro de 2020.

CREMESPb. Centro de Bioética. Declaração Universal sobre Bioética e Direitos Humanos. http://www.bioetica.org.br/?siteAcao=DiretrizesDeclaracoesIntegra&id=17 Aceesso em 08 de dezembro de 2020.

Enciclopédia do Holocausto. (2020) "As Experiências Médicas Nazistas". https://encyclopedia.ushmm.org/content/pt-br/article/nazi-medical-experiments Acesso em 08 de dezembro de 2020.

FAPESP. (2014). Fundação de Amparo à Pesquisa do Estado de São Paulo. (2014). Código de Boas Práticas Científicas. FAPESP-Codigo_de_Boas_Praticas_Cientificas_2014.pdf

IEEE. (2020) The Institute of Electrical and Electronic Engineers. IEEE Code of Ethics. https://www.ieee.org/about/corporate/governance/p7-8.html

Kizza, J. M., (2013). Ethical and Social Issues in the Information Age. Springer Verlag. Londres. UK.

Kottow, M. (2008). "História da Ética em pesquisa em seres humanos". RECIIS – Revista Eletrônica de Comunicação, Informação e Inovação em Saúde. Rio de Janeiro, v.2, Sup.1, p.Sup.7-Sup.18, Dez., 2008. DOI: 10.3395/reciis.v2.Sup1.203pt. https://www.arca.fiocruz.br/bitstream/icict/17570/2/2.pdf

Ministério da Saúde (Brasil). Plataforma Brasil. Acessado em 28 de novembro de 2020, de: https://plataformabrasil.saude.gov.br/login.jsf

ONU. Organização das Nações Unidas. Universal Declaration of Human Rights. https://www.un.org/en/universal-declaration-human-rights/ Acesso em 08 de dezembro de 2020.

Santoro, F. M., Costa, R. M. E. M. (2020). Ética Profissional em Computação. Em: Maciel, C., Viterbo, J. (eds). Computação e Sociedade. Vol. 1. EdUFMT Digital.

Sharp, H., Rogers, Y., Preece, J. (2011). Design de Interação: além da interação homem-computador. Bookman, Porto Alegre, Brasil.

SBC. Sociedade Brasileira de Computação. (2013). Código de Ética do Profissional de Informática. https://www.sbc.org.br/jdownloads/02.codigo_de_etica_da_sbc.pdf

SBC. Sociedade Brasileira de Computação. (2020). Código de Conduta para Publicações da SBC. https://www.sbc.org.br/documentos-da-sbc/send/144-institucional/1298-codigo-de-conduta-para-publicacoes-da-sbc