



POLITECNICO DI BARI

DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELL'INFORMAZIONE
CORSO DI LAUREA TRIENNALE IN INGEGNERIA INFORMATICA E
DELL'AUTOMAZIONE

Chain4Good

*Progettazione e sviluppo di una piattaforma di
crowdfunding con tecnologia blockchain*

Candidati:

Angelica DE FEUDIS
Johnatan CAPUTO
Luca GENTILE

Docente:

Prof.ssa Marina
MONGIELLO

Academic Year: 2025/2026

Indice

Acronimi	iv
1 Introduzione	1
2 Background	2
2.1 Blockchain	2
2.1.1 Transazioni	2
2.1.2 Blocchi	2
2.1.3 Protocolli di consenso	3
2.1.4 Vantaggi della blockchain	3
2.2 Ethereum Blockchain	4
2.2.1 Smart Contract	4
2.2.2 DApps	5
3 Stato dell'arte	6
3.1 Crowdfunding	6
3.1.1 Limiti delle piattaforme tradizionali di CF	7
3.2 Crowdfunding e blockchain	8
4 Metodologia di progetto	9
4.1 Modello di processo	9
4.1.1 Organizzazione del team	9
4.2 Pianificazione delle attività	10
4.3 Analisi dei rischi	10
4.4 Stima dei costi	10
5 Progettazione e implementazione	11
5.1 L'obiettivo di Chain4Good	11
5.2 Analisi dei requisiti	11
5.2.1 Requisiti funzionali	11
5.2.2 Requisiti non funzionali	11
5.3 Architettura del Sistema	11
5.3.1 Architettura del Software	11
5.3.2 Strumenti di sviluppo e deployment	11
6 Prototipo	12
6.1 Login e autenticazione	12
6.1.1 Ente	12
6.1.2 Donatore	12
6.2 Dashboard donatore	12

6.3	Creazione progetto	12
6.4	Inserimento e valutazione spesa	12
6.4.1	Meccanismo di validazione	12
7	Validazione e discussione	17
7.1	Valutazione dell'applicazione	17
7.2	Realizzazione dei requisiti	17
8	Conclusioni e sviluppi futuri	18
	Riferimenti bibliografici	19

Elenco delle figure

1	Dashboard del donatore	14
2	Inserimento di un nuovo progetto	15
3	Valutazione di una richiesta di spesa	16

Acronimi

CF Crowdfunding.

DCF Donation-based Crowdfunding.

ECF Equity-based Crowdfunding.

LCF Lending-based Crowdfunding.

ONP Organizzazioni no-profit.

RBC Reward-based Crowdfunding.

1 Introduzione

2 Background

2.1 Blockchain

Una blockchain è un registro distribuito, condiviso e immutabile gestito da una rete di nodi interconnessi secondo un'architettura [Peer-to-Peer \(P2P\)](#). Tale paradigma architettonico è stato introdotto per la prima volta nel 2008 con la pubblicazione del *white paper* “Bitcoin: A Peer-to-Peer Electronic Cash System” da parte di Satoshi Nakamoto, con l’obiettivo di creare un sistema di pagamento elettronico decentralizzato che permettesse a due controparti di effettuare transazioni in modo sicuro, senza l’ausilio di autorità centrali [1].

Sul piano architettonico, la blockchain è strutturata come una catena sequenziale di blocchi, ciascuno contenente un insieme di transazioni validate dalla rete.

2.1.1 Transazioni

Una transazione costituisce l’unità fondamentale di informazione all’interno della blockchain. Essa descrive un’operazione richiesta da un utente, come il trasferimento di un asset digitale (*token*) [2]. Ogni transazione è creata da un nodo e viene firmata digitalmente utilizzando una coppia di chiavi crittografiche (crittografia asimmetrica).

Prima di essere registrata nella blockchain, ogni transazione viene validata dalla rete secondo regole condivise e, una volta confermata, viene inserita in un blocco.

2.1.2 Blocchi

Un blocco è una struttura dati che aggredisce un insieme di transazioni valide in un determinato intervallo di tempo. Oltre alle transazioni, ogni blocco contiene un *header* con informazioni di gestione fondamentali, tra cui:

- l’*hash* del blocco precedente, che collega crittograficamente i blocchi tra loro formando la catena;
- un *timestamp*, che indica il momento di creazione del blocco;
- il *Merkle Root*, ovvero la radice di un *Merkle Tree*, una struttura ad albero binario che consente di riassumere tutte le transazioni del blocco in un unico *hash*;

- dati aggiuntivi richiesti dal protocollo di consenso adottato.

L'inclusione dell'*hash* del blocco precedente rende la blockchain una struttura immutabile per costruzione: anche una minima modifica a una singola transazione altererebbe la *Merkle Root* e di conseguenza l'*hash* del blocco, invalidando l'intera catena successiva. Questo meccanismo costituisce uno dei principali fattori di sicurezza della blockchain [3].

2.1.3 Protocolli di consenso

Come precedentemente affermato, la blockchain è un sistema intrinsecamente decentralizzato, per cui la validazione delle transazioni non dipende più da un'autorità centrale ma è subordinata all'approvazione collettiva da parte tutti i nodi della rete. Il processo di validazione di un blocco impone dunque l'implementazione di algoritmi di consenso, come:

- **Proof of Work (PoW)**: rappresenta il protocollo di consenso più noto e storicamente rilevante (introdotto da Bitcoin). In questo meccanismo, i nodi della rete, detti *miner*, competono per risolvere un problema crittografico computazionalmente complesso ma facilmente verificabile. Il *miner* che per primo individua una soluzione valida al problema, acquisisce il diritto di creare un nuovo blocco e propagarlo alla rete. Tuttavia, l'aggiunta effettiva del blocco avviene solo se questo viene correttamente validato e accettato dalla maggioranza dei nodi. Se queste operazioni vanno a buon fine, il *miner* riceve una ricompensa, generalmente erogata sotto forma di criptovaluta [4].
- **Proof of Stake (PoS)**: in questo protocollo, il processo di validazione è affidato ad un insieme di nodi scelti in funzione della quantità di criptovaluta vincolata come garanzia, in un processo noto come *staking*. I nodi selezionati come *validatori* sono responsabili della creazione dei blocchi e della verifica delle transazioni. Una volta raggiunto il consenso, il blocco viene aggiunto immutabilmente alla blockchain, aggiornando lo stato del *ledger* distribuito.

2.1.4 Vantaggi della blockchain

L'adozione della tecnologia blockchain offre vantaggi significativi rispetto ai sistemi centralizzati tradizionali, come:

- **Decentralizzazione**: la replicazione del registro su più nodi consente al sistema di operare correttamente anche in presenza di guasti, eliminando in questo modo i *single point of failure* [5];

- **Immutabilità:** ogni transazione è irreversibile, per cui una volta registrata nella blockchain, non può più essere cancellata o modificata, in quanto farlo altererebbe l'intera catena dei blocchi [3];
- **Trasparenza:** tutte le transazioni sono verificabili pubblicamente (nelle *blockchain permissionless*) o dai membri autorizzati (nelle *permissioned*) [2];
- **Sicurezza:** l'uso combinato di crittografia asimmetrica (per l'autenticazione) e protocolli di consenso distribuito (per l'integrità del registro) rende il sistema resistente ad attacchi e frodi;

2.2 Ethereum Blockchain

Ethereum è una piattaforma decentralizzata e *open-source* basata su tecnologia blockchain, proposta da Vitalik Buterin nel 2013 e resa operativa con il primo rilascio stabile nel 2015 [6].

A differenza delle blockchain di prima generazione, concepite prevalentemente come registri distribuiti per il trasferimento di valore (come Bitcoin), Ethereum nasce come una piattaforma *general-purpose* progettata per l'esecuzione di programmi direttamente sulla blockchain.

Questa estensione del modello tradizionale ha permesso di superare alcune limitazioni delle piattaforme precedenti, come la mancanza di Turing-completezza [7], e ha reso possibile la realizzazione di applicazioni decentralizzate, le **Decentralized Applications (DApps)**, la cui logica applicativa è implementata tramite *Smart Contract*.

2.2.1 Smart Contract

Gli *Smart Contract* sono programmi immutabili memorizzati sulla blockchain di Ethereum. Come illustrato nel *whitepaper* che ha introdotto questa tecnologia, gli *Smart Contract* possono essere concepiti come entità crittografiche capaci di detenere e trasferire valore, il cui comportamento è rigidamente definito dal codice e la cui esecuzione è consentita esclusivamente al verificarsi delle condizioni prestabilite [7].

Per supportare la loro esecuzione, Ethereum adotta un modello basato su *account*, distinguendo tra:

- ***Externally Owned Account (EOA)*:** controllati direttamente dagli utenti tramite una coppia di chiavi crittografiche. La chiave privata è utilizzata per firmare digitalmente le transazioni, mentre la chiave pubblica funge da indirizzo dell'account sulla rete Ethereum. Una

transazione è valida solo se firmata dal mittente, che ne garantisce l'autenticità e l'integrità;

- *Contract Accounts*: account associati a uno *Smart Contract*, privi di chiave privata. Essi non possono avviare transazioni, ma vengono attivati esclusivamente in risposta a chiamate provenienti da un **EOA** o da un altro *Smart Contract*.

Ogni interazione con uno *Smart Contract* avviene tramite una transazione e comporta l'esecuzione di istruzioni all'interno dell'**Ethereum Virtual Machine (EVM)**. Il costo computazionale associato a tale esecuzione è quantificato in *gas* ed è sostenuto dal mittente della transazione sotto forma di Ether, criptovaluta nativa di Ethereum.

2.2.2 DApps

Le **DApps** sono applicazioni decentralizzate che operano su una rete P2P. A differenza delle applicazioni tradizionali, la logica di *backend* non risiede su un server centralizzato, ma è implementata mediante *Smart Contract* [8]. L'esecuzione di una DApp, infatti, avviene all'interno di una **Virtual Machine (VM)** distribuita: quando una funzione di uno *Smart Contract* viene invocata tramite una transazione, ogni nodo della rete esegue la **VM** ed elabora il codice del contratto in modo autonomo e deterministico.

Questa architettura conferisce alle **DApps** vantaggi significativi rispetto ai sistemi centralizzati in termini di affidabilità, trasparenza e immutabilità [9].

3 Stato dell'arte

3.1 Crowdfunding

Il *Crowdfunding (CF)* è un modello di finanziamento collettivo in cui una pluralità di individui decide di destinare il proprio denaro, prevalentemente tramite piattaforme digitali, a supporto di progetti e iniziative di varia natura [10].

In ragione della sua etimologia, dall'inglese *crowd* "folla" e *funding*, finanziamento, il **CF** è stato definito come una pratica di microfinanziamento "dal basso" [11], la cui peculiarità risiede nella capacità di aggregare numerosi contributi finanziari di modesta entità a partire da un'ampia platea di sostenitori.

La letteratura attribuisce al Web 2.0 il principale catalizzatore del successo del **CF** [12]. Lo sviluppo di Internet e la capillare diffusione di canali digitali di comunicazione, come i *social-media*, infatti, ha permesso non solo di ampliare la platea di donatori, ma anche di abbattere i limiti geografici, trasformando la "folla" in una comunità attiva e globale.

Inoltre, la nascita di infrastrutture digitali dedicate, come *Kickstarter* e *GoFundme*, è stato determinante per garantire la scalabilità del fenomeno.

In questo scenario, il modello contemporaneo di **CF** si articola in un'architettura tripartita, che vede l'interazione sinergica di tre attori chiave: il promotore dell'iniziativa, i sostenitori e la piattaforma digitale [13]. Quest'ultima non funge da mera vetrina, ma rappresenta l'infrastruttura tecnologica che media le interazioni tra le parti, facilitando il processo di raccolta fondi, la diffusione delle informazioni e il coordinamento delle attività connesse alla realizzazione del progetto. Sebbene la struttura relazionale del **CF** rimanga invariata, la natura del contributo richiesto e le aspettative di ritorno dei sostenitori, rappresentano gli elementi chiave che ne definiscono la tassonomia. E' sulla base di questi criteri, infatti, che gli studi convergono nel classificare le seguenti tipologie di **CF**:

- **Donation-based Crowdfunding (DCF):** i contributi economici sono erogati senza alcuna aspettativa di ritorno materiale o finanziario. La donazione è motivata esclusivamente dal desiderio di sostenere una causa di interesse collettivo o di pubblica utilità; per questa ragione, la **DCF** è stata definita come la forma più "pura" di *crowdfunding* [14];
- **Reward-based Crowdfunding (RBC):** i sostenitori finanziano un progetto in cambio di una ricompensa, generalmente di natura non finanziaria (come riconoscimenti simbolici oppure ricompense tangi-

bili, configurandosi spesso come un vero e proprio "pre-ordine" del prodotto) [15];

- **Equity-based Crowdfunding (ECF)**: il finanziatore, sia esso un individuo o un ente, riceve quote societarie o titoli partecipativi dell'azienda, in cambio del capitale investito [16]
- **Lending-based Crowdfunding (LCF)**: noto anche come *debt-based crowdfunding*, prevede che il capitale versato dai sostenitori venga rimborsato dal promotore entro una scadenza prestabilita, comprensivo di un tasso di interesse pattuito [17];

Nonostante la natura prettamente finanziaria degli ultimi due modelli, l'elemento che li riconduce univocamente al paradigma del *crowdfunding* è la modalità di raccolta: il capitale non è più appannaggio di un singolo grande istituto di credito, ma deriva dalla somma di innumerevoli micro-investimenti operati da una moltitudine di individui. E' proprio questo a sancire la natura "dal basso" di tali strumenti, trasformando ogni cittadino in un potenziale nodo di una rete di finanziamento globale e democratico.

3.1.1 Limiti delle piattaforme tradizionali di CF

La letteratura converge nel considerare le piattaforme di **CF** caratterizzate da una serie di criticità strutturali, riconducibili principalmente al loro modello architettonurale. Si distinguono:

- **mancanza di trasparenza nell'utilizzo dei fondi**: i donatori non dispongono di strumenti efficaci per verificare l'intero ciclo di vita delle donazioni [13]. La tracciabilità delle quote donate è, infatti, demandata al fruitore della donazione, il quale ha il compito di fornire aggiornamenti sullo stato di avanzamento dell'iniziativa finanziata;
- **scarsa fiducia e frodi**: l'assenza di meccanismi di verifica e controllo automatizzati rende le piattaforme di **CF** tradizionali vulnerabili a comportamenti fraudolenti, quali la creazione di campagne ingannevoli o la mancata realizzazione dei progetti finanziati. Tale criticità accentua l'asimmetria informativa tra promotori e donatori, alimentando una diffusa sfiducia nei confronti delle piattaforme [18].
- **centralizzazione**: l'architettura centralizzata utilizzata dalla maggior parte delle piattaforme (come GoFundMe) introduce **Single Point of Failure (SPoF)** e attribuisce la gestione dei fondi raccolti interamente alla piattaforma [19];

- **mancanza di sicurezza:** la gestione centralizzata dei dati espone le piattaforme ad attacchi malevoli, con conseguenze rilevanti in termini di perdita di fondi e fiducia degli utenti [19].

Nel complesso, queste criticità evidenziano come le piattaforme tradizionali di CF si fondino su un modello fortemente fiduciario, nel quale il corretto funzionamento del sistema dipende dal comportamento onesto degli intermediari e dei promotori delle iniziative. Sebbene questo paradigma abbia dunque democratizzato l'accesso al capitale, l'architettura adottata introduce inefficienze strutturali che limitano il potenziale del modello di *crowdfunding*.

3.2 Crowdfunding e blockchain

4 Metodologia di progetto

4.1 Modello di processo

Per lo sviluppo di questo sistema è stato adottato un modello di processo *Agile* di tipo *Incrementale*. Questa scelta è motivata dalla necessità di coniugare la flessibilità dei metodi agili, con la capacità del modello incrementale di gestire le fasi di sviluppo in maniera concorrente e sovrapposta.

Il coordinamento del *team*, invece, ha seguito la tecnica *Scrum*. In particolare, le riunioni periodiche hanno permesso una gestione dinamica del *product backlog* (elenco delle attività da svolgere) e un monitoraggio costante dello stato di avanzamento del progetto, garantendo un'integrazione continua dei risultati discussi.

L'orientamento Agile si è manifestato sin dalle fasi iniziali. Le sessioni di *brainstorming* effettuate hanno permesso di proporre e analizzare diverse alternative progettuali. La decisione di abbandonare la proposta iniziale in favore di una più rispondente alle indicazioni dei referenti riflette i principi cardine del Manifesto Agile, quali: collaborazione con gli *stakeholder* e risposta al cambiamento.

L'adozione del modello incrementale, invece, ha permesso di ottimizzare i tempi di sviluppo. Il progetto, infatti, non è stato condotto secondo una sequenza rigida di fasi, ma ha previsto lo svolgimento in parallelo di più attività.

4.1.1 Organizzazione del team

Lo sviluppo concorrente ha richiesto la suddivisione delle responsabilità di progetto in macro-aree (*front-end*, *back-end* e documentazione tecnica), favorendo l'avanzamento simultaneo dei diversi incrementi del sistema. Tale ripartizione, tuttavia, non ha comportato una compartimentazione stagna dei compiti. Al contrario, ogni membro del gruppo ha mantenuto una visione olistica del progetto, partecipando attivamente alla risoluzione delle criticità anche al di fuori della propria area di competenza primaria. Tale impostazione ha favorito una dinamica di supporto reciproco e interdisciplinare. Il *team* ha, inoltre, operato seguendo il principio della *Collective Ownership*, estendendo a ciascun membro la responsabilità della qualità globale del prodotto.

Complessivamente, l'approccio adottato ha permesso sia di valorizzare i punti di forza di ogni singolo membro che di trasformare le riunioni in opportunità di apprendimento trasversale e di crescita collettiva. Il successo

della metodologia adottata è risultato fortemente legato ai fattori umani, quali competenza tecnica, condivisione degli obiettivi e cooperazione proattiva all'interno del *team*.

4.2 Pianificazione delle attività

La pianificazione delle attività è stata articolata in due momenti distinti: inizialmente è stata definita una **Work Breakdown Structure (WBS)** per scomporre gerarchicamente il progetto in macro-attività. Successivamente, per gestire la sequenzialità e il parallelismo dei *task* individuati, è stato redatto Diagramma di Gantt.

Questo approccio ha permesso di confrontare costantemente i tempi effettivi di esecuzione con la durata stimata in fase di pianificazione, garantendo il rispetto delle scadenze.

4.3 Analisi dei rischi

4.4 Stima dei costi

La stima dei costi del progetto è stata effettuata in termini di tempo/persona, tenendo conto della dimensione del sistema, delle tecnologie adottate e dell'esperienza di ciascuno. Considerata la natura prototipale del progetto, è stata adottata una stima qualitativa ispirata ai modelli algoritmici dei costi, come il CoCoMo.

5 Progettazione e implementazione

5.1 L'obiettivo di Chain4Good

Chain4Good nasce per superare i limiti strutturali delle piattaforme di [CF](#) esistenti.

5.2 Analisi dei requisiti

5.2.1 Requisiti funzionali

5.2.2 Requisiti non funzionali

5.3 Architettura del Sistema

Prima di poter procedere alla progettazione dell'architettura del sistema da realizzare si è resa necessaria l'individuazione delle tecnologie da utilizzare in fase di sviluppo per poter comprendere come queste potessero interagire tra loro e soddisfare tutti i requisiti funzionali e non funzionali emersi dalla precedente fase di analisi.

5.3.1 Architettura del Software

Backend e frontend

5.3.2 Strumenti di sviluppo e deployment

6 Prototipo

6.1 Login e autenticazione

6.1.1 Ente

6.1.2 Donatore

6.2 Dashboard donatore

La *dashboard* del donatore è l'interfaccia attraverso la quale gli utenti possono interagire direttamente con la piattaforma a seguito della procedura di autenticazione.

Considerata la natura dei proponenti (Enti del terzo settore e [Organizzazioni no-profit \(ONP\)](#)), il sistema implementa il meccanismo di raccolta del *Keep-it-all*, secondo cui l'ente beneficiario trattiene i fondi raccolti anche qualora l'obiettivo non venga raggiunto entro i termini prefissati.

6.3 Creazione progetto

6.4 Inserimento e valutazione spesa

A differenza dei sistemi centralizzati in cui l'Ente ha piena e immediata disponibilità del *budget* donato, l'architettura proposta prevede che i fondi raccolti rimangano vincolati all'interno di uno *Smart Contract*.

Per accedere a tali risorse, il beneficiario deve formalizzare una "Richiesta di Spesa" (3) attraverso l'inserimento dei seguenti parametri: identificativo della spesa, importo richiesto, finalità dell'esborso e preventivo.

Una volta sottomessa la richiesta, la piattaforma permette ai donatori di analizzare la documentazione ed esercitare il proprio diritto di voto.

6.4.1 Meccanismo di validazione

Per garantire l'operatività del sistema ed evitare lo stallo decisionale, lo *Smart Contract* è stato programmato per agire secondo le seguenti regole:

- La richiesta è approvata se la maggioranza dei votanti (rappresentata dalla comunità di individui che finanzia un'iniziativa) esprime un parere favorevole. In presenza di una partecipazione parziale, la soglia di maggioranza viene ricalcolata in funzione dei soli voti espressi.
- Qualora non venga registrata alcuna attività di voto, il sistema approva automaticamente la richiesta.

Al soddisfacimento dei requisiti di approvazione, lo *Smart Contract* esegue in modo autonomo e irreversibile il trasferimento della somma raccolta verso il *Wallet* del beneficiario.

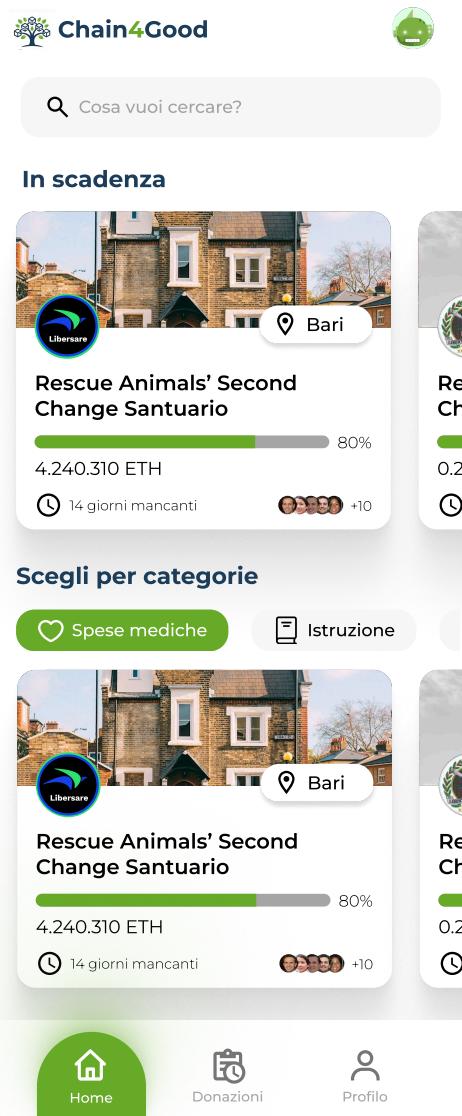


Figura 1: Dashboard del donatore

The image displays two side-by-side screenshots of a mobile application interface for creating a new project on the 'Chain4Good' platform.

Left Screenshot (Step 1 di 2):

- Header:** 'Chain4Good' logo and user profile picture.
- Title:** 'Crea un Nuovo Progetto'
- Progress Bar:** 'Step 1 di 2' with two steps numbered 1 and 2.
- Fields:** 'Nome del progetto' (Project Name) input field.
- Category Selection:** A grid of categories with rounded corners:
 - Spese mediche** (highlighted in green)
 - Istruzione
 - Emergenze
 - Ambiente
 - Sport
- Budget Target:** 'Budget target' input field.
- Deadline:** 'Scadenza' input field.
- Buttons:** 'Proseguì' (Next) in dark blue, 'Annulla' (Cancel) in red, and 'Carica progetto' (Upload project) in green.

Right Screenshot (Step 2 di 2):

- Header:** 'Chain4Good' logo and user profile picture.
- Title:** 'Crea un Nuovo Progetto'
- Progress Bar:** 'Step 2 di 2' with two steps numbered 1 and 2.
- Fields:** 'Descrizione' (Description) input field.
- Text:** 'Come useremo i fondi' (How we will use the funds) input field.
- Image:** 'Immagine di copertina' (Cover image) input field.
- Buttons:** '←' (Back), 'Carica progetto' (Upload project) in green, and a large light gray button at the bottom.

Figura 2: Inserimento di un nuovo progetto

Nuova spesa

Nome spesa

EUR **100,00**

La tua spesa verrà valutata
Prima di sbloccare i fondi,
dovrà essere approvata dai donatori

Descrizione spesa

Allega preventivo

Invia richiesta

RICHIESTA DI SPESA

Acquisto furgoncino fantastico (usato) **1570 USDC**

Grazie al furgone potremmo andare a ricercare i
trotatelli in giro per la città di Bari.

[Preventivo-1.pdf](#)

Valuta se è una spesa appropriata

Pubblicata il 12/12/2025
 Ha ricevuto finora 23 voti positivi
 Ha ricevuto finora 10 voti negativi

Approva **Nega**

Figura 3: Valutazione di una richiesta di spesa

7 Validazione e discussione

7.1 Valutazione dell'applicazione

7.2 Realizzazione dei requisiti

8 Conclusioni e sviluppi futuri

Riferimenti bibliografici

- [1] S. Nakamoto, B. Bit et al., «Bitcoin: A peer-to-peer electronic cash system,» 2008, 2007.
- [2] M. Crosby, P. Pattanayak, S. Verma, V. Kalyanaraman et al., «Blockchain technology: Beyond bitcoin,» *Applied innovation*, vol. 2, n. 6-10, p. 71, 2016.
- [3] A. A. Monrat, O. Schelén e K. Andersson, «A survey of blockchain from the perspectives of applications, challenges, and opportunities,» *Ieee Access*, vol. 7, pp. 117134–117151, 2019.
- [4] F. Tschorisch e B. Scheuermann, «Bitcoin and beyond: A technical survey on decentralized digital currencies,» *IEEE Communications Surveys & Tutorials*, vol. 18, n. 3, pp. 2084–2123, 2016.
- [5] R. Rodrigues e P. Druschel, «Peer-to-peer systems,» *Communications of the ACM*, vol. 53, n. 10, pp. 72–82, 2010.
- [6] V. Buterin et al., «A next-generation smart contract and decentralized application platform,» *white paper*, vol. 3, n. 37, pp. 2–1, 2014.
- [7] V. Buterin et al., «Ethereum white paper,» *Github repository*, vol. 1, n. 22-23, pp. 5–7, 2013.
- [8] S. A. Renu e B. G. Banik, «Implementation of a secure ridesharing DApp using smart contracts on Ethereum blockchain,» *International Journal of Safety and Security Engineering*, vol. 11, n. 2, pp. 167–173, 2021.
- [9] W. Metcalfe et al., «Ethereum, smart contracts, DApps,» *Blockchain and Crypt Currency*, vol. 77, pp. 77–93, 2020.
- [10] A. Brunello, «Crowdfunding,» *Podręcznik. Warszawa: Wydawnictwo CeDeWu*, 2016.
- [11] Y. Thalassinos, «Crowdfunding: A Different Approach to Investment,» *European Research Studies Journal*, vol. 26, n. 2, pp. 318–333, 2023.
- [12] F. Brunetti, «Web 2.0 as Platform for the Development of Crowdfunding,» in *Crowdfunding for SMEs: A European Perspective*, Springer, 2016, pp. 45–60.
- [13] O. K. Alia, D. M. Suleiman e H. A. Noman, «IHSAN: A Secure and Transparent Crowdfunding Platform Leveraging Comprehensive Decentralized Technologies,» *IEEE Access*, 2024.
- [14] N. Salido-Andres, M. Rey-Garcia, L. I. Alvarez-Gonzalez e R. Vazquez-Casielles, «Mapping the field of donation-based crowdfunding for charitable causes: systematic review and conceptual framework,» *VOLUNTAS: International Journal of Voluntary and Nonprofit Organizations*, vol. 32, n. 2, pp. 288–302, 2021.

- [15] S. Hohen, C. Hüning e L. Schweizer, «Reward-based crowdfunding—a systematic literature,» 2025.
- [16] M. Kuti, Z. Bedő e D. Geiszl, «Equity-based crowdfunding,» *Financial and Economic Review*, vol. 16, n. 4, pp. 187–200, 2017.
- [17] M. Hossain e G. O. Oparaocha, «Crowdfunding: Motives, definitions, typology and ethical challenges,» *Entrepreneurship Research Journal*, vol. 7, n. 2, p. 20150045, 2017.
- [18] A. Rejeb, K. Rejeb, A. Appolloni, S. Zailani e M. Iranmanesh, «Mapping the research landscape of blockchain and crowdfunding,» *Financial Innovation*, vol. 11, n. 1, p. 22, 2025.
- [19] K. Mukherjee, A. Rana e S. Rani, «Crowdfunding Platform using Blockchain,» in *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*, IEEE, 2024, pp. 1–6.