E-Cycle: Analisi

[Introduzione 1](#_Toc198254761)

[Obiettivo dell’Applicazione 1](#_Toc198254762)

[Glossario 2](#_Toc198254763)

[Funzionamento dell’Applicazione 3](#_Toc198254764)

[Registrazione e Accesso 3](#_Toc198254765)

[Pubblicazione di un’Offerta 3](#_Toc198254766)

[Creazione di una Richiesta 3](#_Toc198254767)

[Matching Automatizzato 3](#_Toc198254768)

[Ciclo vita di Richieste e Offerte 4](#_Toc198254769)

[Il Modello Concettuale di E-Cycle 5](#_Toc198254770)

[Le Entità 5](#_Toc198254771)

[Le Relazioni 5](#_Toc198254772)

[Il Modello Concettuale 6](#_Toc198254773)

[I Vincoli 6](#_Toc198254774)

[Il Modello Logico di E-Cycle 8](#_Toc198254775)

# Introduzione

Oggi chi vuole disfarsi di un bene ha due strade: l’accumulo, sia esso privato o in discariche, e il *riutilizzo*.

Con *riutilizzo* si intendono pratiche tre pratiche principali:

* up-cycling, con cui il bene assume una funzione diversa da quella designata;
* riciclaggio, con cui il bene viene scomposto in parti rientranti nel ciclo produttivo;
* e rivendita, con cui il bene viene ceduto intatto a un altro soggetto, che poi né farà ciò che desidera.

L’attività di riutilizzo più semplice ed efficace è senza dubbio la rivendita: il venditore non compie alcuna lavorazione sul bene e si limita a guadagnare, il compratore ottiene l’oggetto desiderato pronto all’uso che più desidera, e la collettività trae beneficio dallo scambio con la riduzione dei rifiuti.

Ancora oggi però vendere un bene inutilizzato risulta complicato. I marketplace sono dispersivi e privi di alcuna logica strutturale: la domanda e l’offerta faticano ad incontrarsi. Manca uno spazio dove si possa cercare non un prodotto specifico, ma un bene con determinate caratteristiche.

# Obiettivo dell’Applicazione

E-Cycle nasce per colmare questo vuoto: lo scopo dell’app è facilitare la rivendita dei beni inutilizzati dando loro una nuova vita.

Chi possiede beni vacanti li può offrire, chi ne ha bisogno li può cercare. Entrambi agiscono in base alle caratteristiche reali del bene, non solo in base al suo nome commerciale.

Con pochi click una richiesta può trovare una risposta, e un oggetto dimenticato un nuovo contesto d’uso.

E-Cycle è più di un semplice marketplace: è un’intelligenza collettiva applicata ai beni materiali.

# Glossario

Per comprendere il funzionamento dell’applicazione, è necessario essere messi a conoscenza di alcuni termini specifici del progetto.

* *Utente*: qualsiasi soggetto registrato sull’app, sia esso privato o azienda.
* *Caratteristiche* (o Insieme Caratteristico): l’insieme delle proprietà che definiscono un bene.
* *Marca* e *Modello*: rispettivamente, il produttore e la denominazione specifica dell’asset.
* *Categoria*: macro-classe di beni a cui un asset appartiene (es. “dispositivo elettronico”, “veicolo industriale”, “mobile d’ufficio”).
* *Natura*: il nome comune con cui si identifica l’oggetto (es. “laptop”, “forno industriale”, “trapano”).
* *Richiesta*: la formulazione, da parte di un utente, di un bisogno relativo a uno o più beni dotati di caratteristiche specifiche. Una richiesta può riguardare beni omogenei (identici tra loro) o eterogenei (diversi ma compatibili con usi distinti), e può quindi essere costituita da più richieste singole.
* *Richiesta singola*: la richiesta, da parte di un utente, di un singolo bene con determinate caratteristiche.
* *Offerta*: la disponibilità, da parte di un utente, a cedere uno o più beni dotati di caratteristiche specifiche. Un’offerta può riguardare beni omogenei o eterogenei, e può quindi essere costituita da più offerte singole.
* *Offerta singola*: l’offerta, da parte di un utente, di un singolo bene con determinate caratteristiche.
* *Matching*: il processo attraverso cui l’applicazione confronta una richiesta singola con le offerte singole disponibili, identificando quelle compatibili per caratteristiche e prezzo.
* *Negoziazione*: la fase in cui un utente visualizza un’offerta singola in relazione ad una sua richiesta singola e può decidere di accettare l’acquisto o rifiutarlo.

# Funzionamento dell’Applicazione

## Registrazione e Accesso

Ogni utente, privato o azienda, accede alla piattaforma tramite un account. Dopo la registrazione, può interagire con l’applicazione in qualità di offerente (chi cede beni) o richiedente (chi li cerca).

## Pubblicazione di un’Offerta

Un utente che intende vendere beni inutilizzati procede come segue:

* accede all’area di creazione di un’offerta;
* dà un nome all’offerta;
* specifica le caratteristiche dei beni che desidera cedere: marca, modello, categoria, natura, stato qualitativo, colore principale e funzione.;
* indica la quantità disponibile di tale bene;
* assegna un prezzo unitario a ciascuno di essi;
* e salva l’offerta, che si tradurrà in più offerte singole, entranti nel sistema e rimanenti disponibile al matching fino alla loro scadenza.

## Creazione di una Richiesta

Un utente con un bisogno definito procede come segue:

* accede all’area di creazione di una richiesta;
* descrive l’insieme caratteristico dei beni desiderato;
* indica la quantità necessaria dei beni;
* opzionalmente, inserisce un prezzo massimo disposto a pagare per ciascun asset;
* e salva la richiesta, che si tradurrà in più richieste singole, entranti nel sistema e rimanenti disponibili al matching fino alla loro scadenza.

## Matching Automatizzato

Il sistema esegue periodicamente un confronto tra richieste singole e offerte singole. Un’offerta singola viene considerata compatibile se:

* il suo insieme caratteristico coincide o è contenuto in quello richiesto;
* il prezzo richiesto è inferiore o uguale a quello massimo indicato nella richiesta (se presente);
* o l’offerta non è scaduta.

Quando vengono trovate corrispondenze:

* al richiedente viene mostrata la richiesta singola con l’offerta singola compatibile;
* si apre così una negoziazione: il richiedente ha 24 ore per accettare o rifiutare l’acquisto;
* e se accetta, l’offerta singola viene considerata assegnata. Se rifiuta o non risponde entro il termine, la trattazione decade.

## Ciclo vita di Richieste e Offerte

* Un’offerta singola ha una durata temporale limitata: dopo la data di scadenza non viene più considerata nei matching.
* Una richiesta singola resta attiva finché non viene soddisfatta o cancellata dall’utente richiedente.
* Le offerte singole e non ancora accettate possono partecipare ad una singola trattazione al momento.

# Il Modello Concettuale di E-Cycle

## Le Entità

Le entità individuate nella progettazione di E-Cycle sono:

* *user*, caratterizzata da *username, name, surname, email, password, address* (a sua volta caratterizzato da *state, region, province, city, street* e *civic*)e *ID*;
* *request*, caratterizzata da *ts\_creation, title* e *ID*;
* *sing\_request*, carratterizzata da *max\_price*, *ts\_deletion* e *ID*;
* *offer*,caratterizzata da *ts\_creation, title* e *ID*;
* *sing\_offer*, caratterizzata da *ts\_creation*, *title* e *ID*;
* *characteristics* (degli asset), caratterizzata da *main\_color, function, quality, prod\_year* e *ID*;
* *model*, *brand*, *category* e *nature*, ciascuna delle quali caratterizzate da un *ID*;
* e *negotiation*, caratterizzata da *ts\_creation*, *ts\_closure*, *closure* (*Accepted*/*Refused*) e *ID*.

## Le Relazioni

Le relazioni individuate sono:

* tra *user* e *request*, di tipo n:1, con vincoli di cardinalità rispettivamente di “(0, n)” e “(1, 1)”;
* tra *user* e *offer*, di tipo n:1, con vincoli di cardinalità rispettivamente di “(0, n)” e “(1, 1)”;
* tra *request* e *sing\_request*, di tipo n:1, con vincoli di cardinalità rispettivamente di “(1, n)” e “(1, 1)”;
* tra *sing\_request* e *characteristics*, di tipo 1:n, con vincoli di cardinalità rispettivamente di “(1, 1)” e “(0, n)”;
* tra *sing\_request* e *negotiation*, di tipo n:1, con vincoli di cardinalità rispettivamente di “(0, n)” e “(1, 1)”;
* tra *offer* e *sing\_offer*, di tipo n:1, con vincoli di cardinalità rispettivamente di “(1, n)” e “(1, 1)”;
* tra *sing\_offer* e *characteristics*, di tipo 1:n, con vincoli di cardinalità rispettivamente di “(1, 1)” e “(0, n)”;
* tra *sing\_offer* e *negotiation*, di tipo n:1, con vincoli di cardinalità rispettivamente di “(0, n)” e “(0, 1)”;
* tra *model* e *characteristics*, di tipo n:1, con vincoli di cardinalità rispettivamente di “(1, n)” e “(1, 1)”;
* tra *brand* e *model*, di tipo n:1, con vincoli di cardinalità rispettivamente di “(1, n)” e “(1, 1)”;
* tra *category* e *characteristics*, di tipo n:1, con vincoli di cardinalità rispettivamente di “(1, n)” e “(1, 1)”;
* e tra *nature* e *characteristics*, di tipo n:1, con vincoli di cardinalità rispettivamente di “(1, n)” e “(1, 1)”.

## Il Modello Concettuale

Di seguito una rappresentazione cartacea del modello concettuale.

A diagram of a diagram

AI-generated content may be incorrect.

*Nota: dove è scritto “ts” si intende dire “timestamp”*, mentre dove è scritto *“sing”* si intende dire *“singular”.*

## I Vincoli

Oltre ai vincoli di cardinalità, già espressi nella rappresentazione cartacea del modello concettuale e mantenuti intrinsecamente dalla struttura del database, in E-Cycle ne esistono ulteriori che vengono imposti dal modus operandi dell’applicazione.

Questi vincoli aggiuntivi sono:

* nella tabella *users* solamente *name* e *surname* sono attributi di inserimento facoltativo;
* nella tabella *users* non possono esistere due o più tuple con attributo *username* uguale;
* nella tabella *users* non possono esistere due o più tuple con attributo *email* uguale;
* nella tabella *sing\_requests* solamente *max\_price* e *ts\_deletion* possono essere NULL;
* nella tabella *sing\_offers* solamente *ts\_deletion* può essre NULL;
* quando si inseriscono tuple in *sing\_offers* queste non possono avere *expiration* inferiore rispetto alla data corrente;
* nella tabella *negotiations* solamente gli attributi *closure* e *ts\_closure* possono essere NULL;
* al momento di creazione di una nuova tupla nella tabella *negotiations*, la richiesta singola che verrà associata ad essa non può avere *ts\_deletion* diverso da NULL e non può essere associata ad un’altra tupla di *negotiations* in cui l’attributo *ts\_closure* equivale a NULL;
* al momento di creazione di una nuova tupla nella tabella *negotiations*, l’offerta singola che verrà associata ad essa non può avere *expiration* pari alla data del giorno corrente, non può avere *ts\_deletion* diverso da NULL, e non può essere già associata ad un’altra tupla di *negotiations* in cui l’attributo *ts\_closure* equivale a NULL;
* tutte le tuple di *negotiations* di cui il campo *ts\_creation* sottratto al timestamp corrente produce una differenza pari o superiore a 24 ore non possono avere *ts\_closure* diverso da NULL;
* tutte le tuple di *negotiations* di cui il campo *closure* è diverso da NULL non possono avere il proprio campo *ts\_closure* pari a NULL;
* nelle tabelle *offers*, *requests*, *natures*, *models*, *characteristics*, *categories* e *brands* tutti gli attributi non possono essere NULL;
* tutte le tuple di *characteristics* non possono avere il campo *prod\_year* inferiore a 0 o maggiore rispetto all’anno corrente.

# Il Modello Logico di E-Cycle

A paper with writing on it

AI-generated content may be incorrect.