

# RELAZIONE PROGETTO DI BASI DI DATI

FRANCESCO GIUSEPPE IELO - 0001070883  
MATTEO PIO TROTTA - 0001093699  
VALERIA SENSINI - 0001069533

# 02

# RACCOLTA/ANALISI DEI REQUISITI

## PREMESSA

Si vuole realizzare la piattaforma BOSTARTER per supportare la creazione di campagne di crowdfunding finalizzate alla realizzazione di progetti hardware/software. La piattaforma è liberamente ispirata alla piattaforma Kickstarter (<https://www.kickstarter.com>). La piattaforma consente agli utenti di creare un progetto hardware o software che si vuole finanziare, indicando il budget da raggiungere e la data limite entro cui ottenere tale importo. Altri utenti della piattaforma possono finanziare il progetto, ricevendo in cambio una qualche reward (non economica) dai creatori. È prevista la possibilità di candidarsi allo sviluppo di un progetto software, se le skill possedute dall'utente fanno matching con quelle dei profili richiesti dal progetto. Infine, la piattaforma consente l'inserimento di commenti.

## SPECIFICA DELLA PIATTAFORMA

La piattaforma BOSTARTER gestisce i dati degli utenti registrati. Ogni utente dispone di indirizzo email (univoco), nickname, password, nome, cognome, anno di nascita, luogo di nascita. Inoltre, ogni utente può indicare le proprie skill di curriculum. Le skill di curriculum consistono in una sequenza di , dove la competenza è una stringa ed il livello è un numero tra 0 e 5 (es. ). La lista delle competenze è comune a tutti gli utenti della piattaforma. Alcuni utenti -ma non tutti- possono appartenere a due sotto-categorie: utenti amministratori, o utenti creatori. Gli utenti amministratori dispongono anche di un codice di sicurezza.

# 03

## SPECIFICA DELLA PIATTAFORMA

Solo gli utenti amministratori possono popolare la lista delle competenze. Un utente creatore dispone anche dei campi: #nr\_progetti (ridondanza concettuale, vedi sotto) ed affidabilità. Un utente creatore – e solo lui – può inserire uno o più progetti. Ogni progetto dispone di un nome (univoco), un campo descrizione, una data di inserimento, una o più foto, un budget da raggiungere per avviare il progetto, una data limite entro cui raggiungere il budget, uno stato. Lo stato è un campo di tipo enum (aperto/chiuso). Ogni progetto è associato ad un solo utente creatore. Inoltre, ogni progetto prevede una lista di reward: una reward dispone di un codice univoco, una breve descrizione, una foto. I progetti appartengono esclusivamente a due categorie: progetti hardware o progetti software. Nel caso dei progetti hardware, è presente anche la lista delle componenti necessarie: ogni componente ha un nome univoco, una descrizione, un prezzo, una quantità ( $>0$ ). Nel caso dei progetti software, viene elencata la lista dei profili necessari per lo sviluppo. Ogni profilo dispone di un nome (es. “Esperto AI”) e di skill richieste: come nel caso delle skill di curriculum, esse consistono in una sequenza, dove la competenza è una stringa - tra quelle presenti in piattaforma- ed il livello è un numero tra 0 e 5. Ogni utente della piattaforma può finanziare un progetto: ogni finanziamento dispone di un importo ed una data. Un utente potrebbe inserire più finanziamenti per lo stesso progetto, ma in date diverse. Nel momento in cui la somma totale degli importi dei finanziamenti supera il budget del progetto, oppure il progetto resta in stato aperto oltre la data limite, lo stato di tale progetto diventa pari a chiuso: un progetto chiuso non accetta ulteriori finanziamenti. Ad ogni finanziamento è associata una sola reward, tra quelle previste per il progetto finanziato. Un utente può inserire commenti relativi ad un progetto.

# 04

## SPECIFICA DELLA PIATTAFORMA

Ogni commento dispone di un id (univoco), una data ed un campo testo. L'utente creatore può eventualmente inserire una risposta per ogni singolo commento (un commento ha al massimo 1 risposta). Infine, è prevista la possibilità per gli utenti di candidarsi come partecipanti allo sviluppo di un progetto software. Un utente può candidarsi ad un numero qualsiasi di profili. Un progetto software può ricevere un numero qualsiasi di candidature per un certo profilo. La piattaforma consente ad un utente di inserire una candidatura su un profilo SOLO se, per ogni skill richiesta da un profilo, l'utente dispone di un livello superiore o uguale al valore richiesto. L'utente creatore può accettare o meno la candidatura.

Infine, si vuole tenere traccia di tutti gli eventi che occorrono nella piattaforma, relativamente all'inserimento di nuovi dati (es. nuovi utenti, nuovi progetti, etc). Tali eventi vanno inseriti, sotto forma di messaggi di testo, all'interno di un log, implementato in un'apposita collezione MongoDB.

05

## LISTA DELLE OPERAZIONI

### OPERAZIONI CHE RIGUARDANO TUTTI GLI UTENTI

1. Autenticazione/registrazione sulla piattaforma
2. Inserimento delle proprie skill di curriculum
3. Visualizzazione dei progetti disponibili
4. Finanziamento di un progetto (aperto). Un utente può finanziare anche il progetto di cui è creatore.
5. Scelta della reward a valle del finanziamento di un progetto
6. Inserimento di un commento relativo ad un progetto
7. Inserimento di una candidatura per un profilo richiesto per la realizzazione di un progetto software

### OPERAZIONI CHE RIGUARDANO SOLO GLI AMMINISTRATORI

1. Inserimento di una nuova stringa nella lista delle competenze
2. In fase di autenticazione, oltre a username e password, viene richiesto anche il codice di sicurezza

## 06

# LISTA DELLE OPERAZIONI

## OPERAZIONI CHE RIGUARDANO SOLO GLI UTENTI CREATORI

1. *Inserimento di un nuovo progetto*
2. *Inserimento delle reward per un progetto*
3. *Inserimento di una risposta ad un commento*
4. *Inserimento di un profilo -solo per la realizzazione di un progetto software*
5. *Accettazione o meno di una candidatura*

## STATISTICHE (VISIBILI DA TUTTI GLI UTENTI)

1. *Visualizzare la classifica degli utenti creatori, in base al loro valore di affidabilità. Mostrare solo il nickname dei primi 3 utenti.*
2. *Visualizzare i progetti APERTI che sono più vicini al proprio completamento (= minore differenza tra budget richiesto e somma totale dei finanziamenti ricevuti). Mostrare solo i primi 3 progetti.*
3. *Visualizzare la classifica degli utenti, ordinati in base al TOTALE di finanziamenti erogati. Mostrare solo i nickname dei primi 3 utenti.*

07

## TAVOLA MEDIA DEI VOLUMI

Concetto	Volume
Progetti	10
Finanziamenti per Progetto	3
Utenti	5
Progetti per Utente	2

### COEFFICIENTI PER L'ANALISI:

$$wI = 1$$

$$wB = 0.5$$

$$\alpha = 2$$

08

## GLOSSARIO DEI DATI

Termine	Descrizione
Utente	Chi usufruisce di un bene o servizio.
Skill	Abilità che un utente possiede.
Amministratore	Incaricato dell'amministrazione del sito.
Creatore	Professionista che crea progetti.
Progetto	Elaborati tecnici relativi all'ideazione di un prodotto hardware o software.
Software	Applicativo sviluppato da professionisti.
Hardware	Prodotto sviluppato da professionisti.
Componente	Parte che potrebbe essere necessaria in un Progetto Hardware.
Profilo	Insieme di caratteristiche che definiscono una figura professionale.
Candidatura	Domanda presentata per ricoprire una posizione lavorativa descritta da un Profilo.
Finanziamento	Contributo per l'avanzamento di un Progetto.
Reward	Ricompensa per Utenti che hanno effettuato un Finanziamento.
Commento	Giudizio espresso da un utente riguardo ad un progetto.

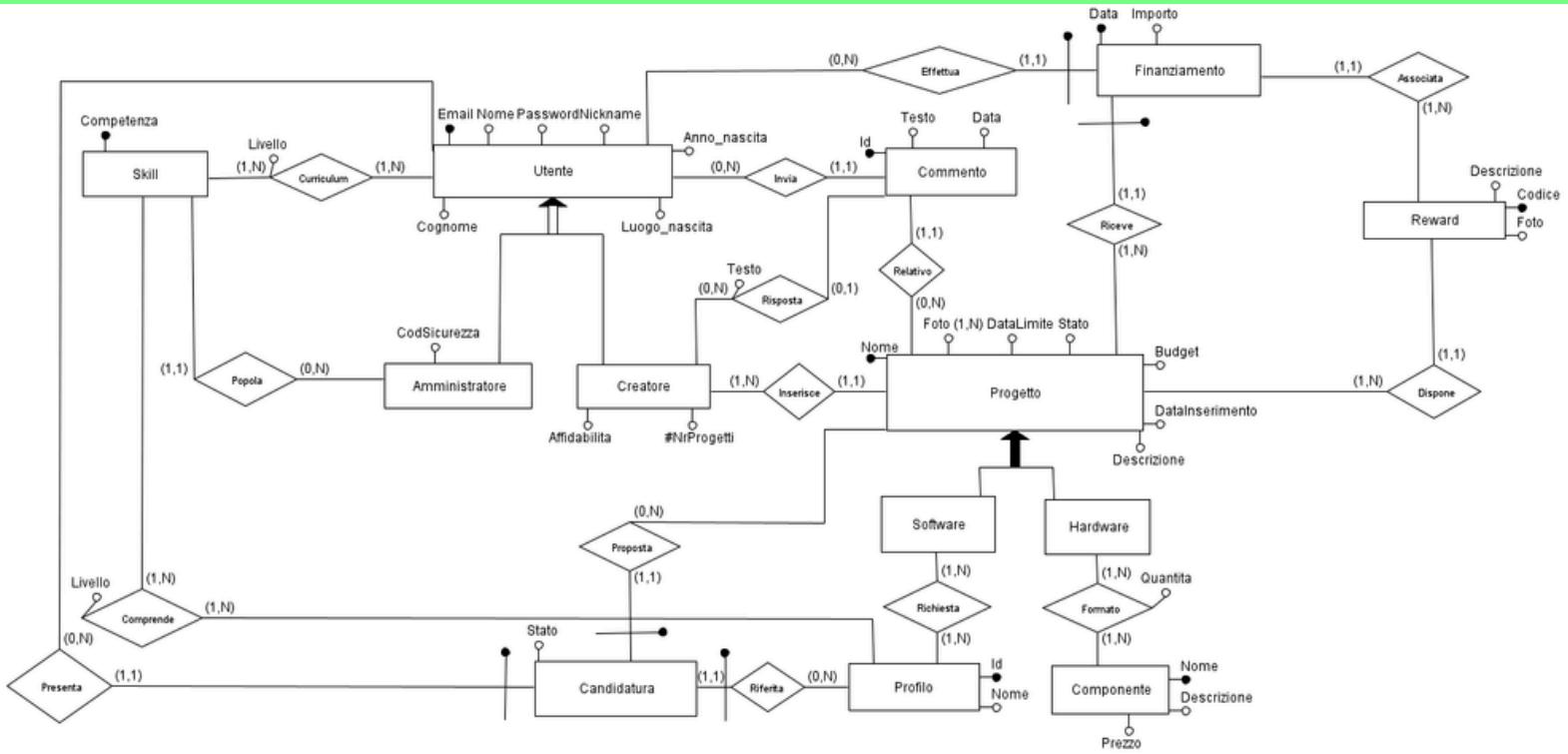
09

# PROGETTAZIONE CONCETTUALE

Ogni Profilo può essere presente in più progetti, si pensi ad un profilo di "Sviluppatore SW" richiesto in diversi progetti, per questo abbiamo aggiunto l'attributo ID.

Abbiamo deciso di non inserire una relazione diretta tra Candidatura e Creatore, perché essendo la Candidatura riferita a un Progetto Software, il nome del Creatore è ricavabile da esso.

Abbiamo inoltre inserito all'interno delle Business Rules, che solo il Creatore associato al progetto per cui si è effettuata la Candidatura possa aggiornarne lo stato.



10

# DIZIONARIO DELLE ENTITÀ

Entità	Descrizione	Attributi	Identificatore
Utente	Utente che si autentica sulla piattaforma	Email, Nome, Cognome, Password, Nickname, Anno_Nascita, Luogo_Nascita	Email
Skill	Competenze che un utente può possedere	Competenza	Competenza
Amministratore	Utente amministratore	CodSicurezza	Email
Creatore	Utente creatore	Affidabilità, #nr_progetti	Email
Progetto	Progetto che può essere inserito nella piattaforma da un Creatore	Nome, Foto(1,N), DataLimite, Stato, DataInserimento, Budget, Descrizione	Nome
Software	Tipologia di Progetto di sviluppo Software		Nome
Hardware	Tipologia di Progetto di sviluppo Hardware		Nome
Componente	Parte che potrebbe essere necessaria in un Progetto Hardware	Nome, Descrizione, Prezzo	Nome

11

# DIZIONARIO DELLE ENTITÀ

Entità	Descrizione	Attributi	Identificatore
Profilo	Insieme di caratteristiche richieste per ricoprire un ruolo in Progetto software	Id, Nome	Id
Candidatura	Richiesta per ricoprire un ruolo descritto da un Profilo	Stato	Id (Profilo), Email (Utente)
Finanziamento	Contributo di un Utente per lo sviluppo di un Progetto	Data, Importo	Data, Email (Utente), Nome (Progetto)
Reward	Ricompensa per Utenti che hanno effettuato un Finanziamento	Codice, Foto, Descrizione	Codice
Commento	Commento lasciato da un Utente relativo ad un Progetto	Id, Testo, Data	Id

12

## DIZIONARIO DELLE RELAZIONI

Relazione	Descrizione	Componenti	Attributi
Curriculum	Elenco di coppie competenza - livello che appartengono ad un Utente	Utente, Skill	Livello
Popola	Amministratore che Popola la lista delle Skills	Amministratore, Skill	
Inserisce	Utente Creatore inserisce un nuovo Progetto	Creatore, Progetto	
Formato	Lista Componenti necessarie per la realizzazione di un Progetto Hardware	Hardware, Componente	Quantità
Richiesta	Necessità di determinati Profili per la realizzazione di un Progetto Software	Software, Profilo	
Riferita	Afferenza di una Candidatura per un determinato profilo	Candidatura, Profilo	
Presenta	Utente presenta una Candidatura per un Profilo	Utente, Candidatura	

13

## DIZIONARIO DELLE RELAZIONI

Relazione	Descrizione	Componenti	Attributi
Valutazione	Accettazione o rifiuto di una Candidatura da parte di un Creatore	Candidatura, Creatore	
Comprende	Lista delle Skills in un Profilo	Profilo, Skill	Livello
Dispone	Reward di cui un Progetto dispone	Progetto, Reward	
Associata	Associazione tra Reward e Finanziamento	Reward, Finanziamento	
Effettua	Utente effettua un Finanziamento per un Progetto	Utente, Finanziamento	
Riceve	Progetto riceve un Finanziamento	Progetto, Finanziamento	
Invia	Utente invia un Commento su un Progetto	Utente, Commento	
Relativo	Commento associato a un Progetto	Commento, Progetto	
Risposta	Creatore risponde ad un Commento	Creatore, Commento	Testo

# 14

## BUSINESS RULES

<b>Regole di Vincolo</b>
1. Solo gli utenti amministratori possono popolare la lista delle competenze.
2. Un utente creatore – e solo lui- può inserire uno o più progetti.
3. La quantità di una Componente presente in un Progetto Hardware deve essere >0.
4. Nel momento in cui la somma totale degli importi dei finanziamenti supera il budget del progetto, oppure il progetto resta in stato aperto oltre la data limite, lo stato di tale progetto diventa pari a chiuso: un progetto chiuso non accetta ulteriori finanziamenti.
5. La piattaforma consente ad un utente di inserire una candidatura su un profilo SOLO se, per ogni skill richiesta da un profilo, l'utente dispone di un livello superiore o uguale al valore richiesto.
6. L'utente creatore del progetto a cui è riferita la candidatura può accettarla o meno.

15

# PROGETTAZIONE LOGICA

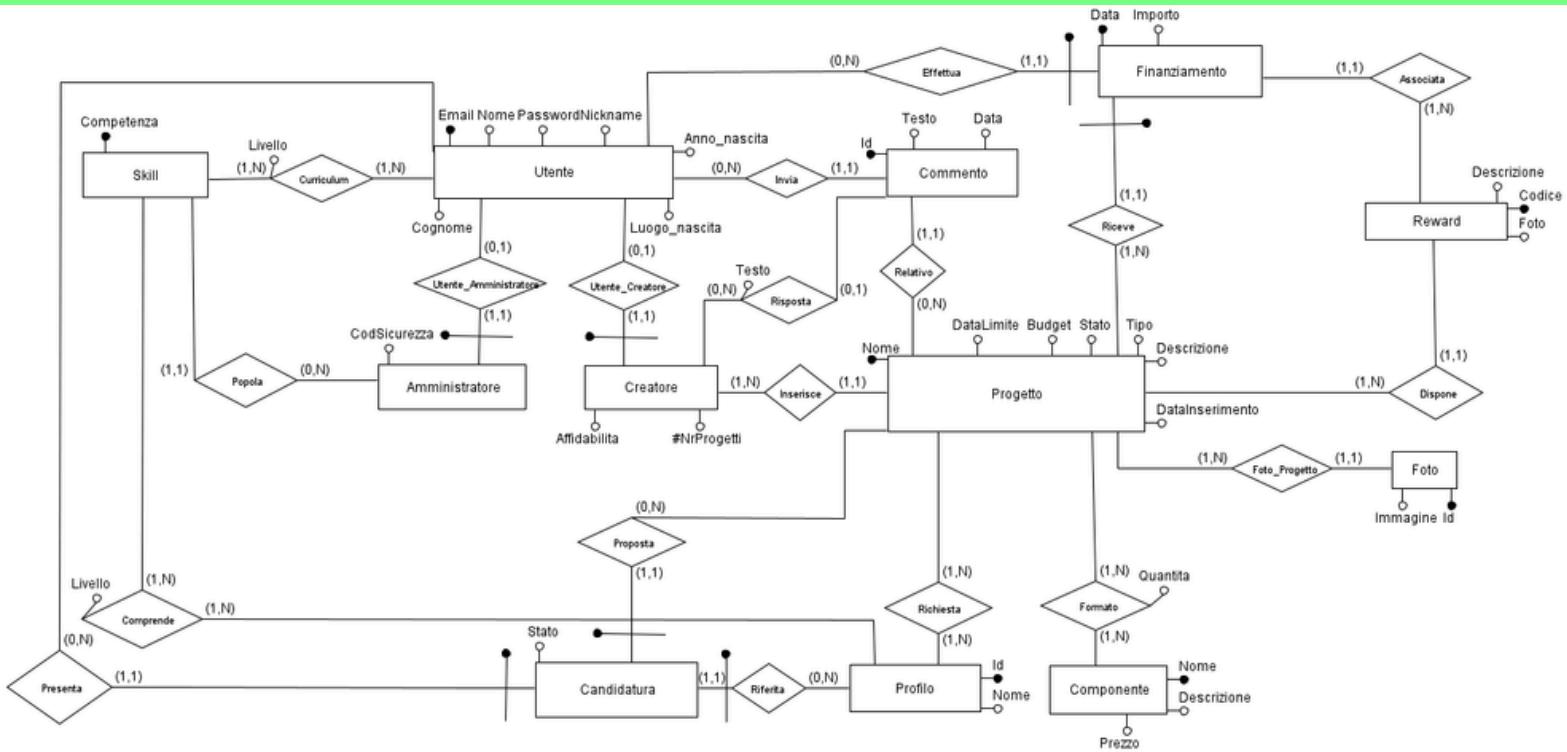
## Ristrutturazione dello schema concettuale

Tutte le relazioni con molteplicità 1 a molti sono state accorpate all'interno dell'entità con cardinalità massima uguale a 1.

Le due generalizzazioni presenti nel diagramma concettuale sono state tradotte in maniera differenti:

- Amministratore e Creatore sono diventate entità collegate con delle relazioni all'entità padre.
  - Hardware e Software sono state accorpate all'interno di Progetto aggiungendo l'attributo tipo che può assumere valore Hardware o Software.

Inoltre, è stata aggiunta l'entità Foto per tradurre l'attributo Foto di Progetto che aveva molteplicità 1,N.



16

# ANALISI DELLE RIDONDANZE

## OPERAZIONE 1 - SENZA RIDONDANZA

$$C(Op1) = 1 * 1 * (2*2 + 0) = 4$$

*TAVOLA DEGLI ACCESSI*

Concetto	Costrutto	Accessi	Tipo
Inserisce	R	1	W
Progetto	E	1	W

## OPERAZIONE 1 - CON RIDONDANZA

$$C(Op1) = 1 * 1 * (2*3 + 0) = 6$$

*TAVOLA DEGLI ACCESSI*

Concetto	Costrutto	Accessi	Tipo
Inserisce	R	1	W
Progetto	E	1	W

17

# ANALISI DELLE RIDONDANZE

## OPERAZIONE 2 - CON E SENZA RIDONDANZA

Questa operazione ha lo stesso costo sia in caso di presenza della ridondanza che in caso di assenza di essa.

$$C(Op2) = 1 * 0.5 * (2*0 + 40) = 20$$

*TAVOLA DEGLI ACCESSI*

Concetto	Costrutto	Accessi	Tipo
Finanziamento	E	30	R
Progetto	E	10	R

Abbiamo deciso di non includere tra gli accessi i 30 passaggi dalla relazione <Riceve> perché andavano visualizzati tutti i Progetti e tutti i Finanziamenti indistintamente da come essi fossero collegati.

## 18

# ANALISI DELLE RIDONDANZE

## OPERAZIONE 3 - SENZA RIDONDANZA

$$C(Op3) = 3 * 0.5 * (2*0 + 2) = 3$$

### TAVOLA DEGLI ACCESSI

Concetto	Costrutto	Accessi	Tipo
Inserisce	R	2	R

## OPERAZIONE 3 - CON RIDONDANZA

$$C(Op3) = 3 * 0.5 * (2*0 + 1) = 1.5$$

### TAVOLA DEGLI ACCESSI

Concetto	Costrutto	Accessi	Tipo
Creatore	E	1	R

## 19

# ANALISI DELLE RIDONDANZE

## ANALISI DEI COSTI

L'operazione 2 resta invariata a prescindere dalla presenza o meno di una ridondanza, perciò analizzo i costi delle operazioni 1 e 3.

→ **Caso S senza uso di ridondanze**

CostoOp1 = 4

CostoOp3 = 3

→ **Caso Srid con ridondanza #nr\_progetti**

CostoOp1 = 6

CostoOp3 = 1.5

Riassumendo:

$c(S) = 7$

$c(Srid) = 7.5$

→ **Speedup =  $7/7.5 = 0,93$ .**

**Risultato dell'analisi delle ridondanze:** è sconveniente mantenere (o introdurre) l'attributo ridondante perché, oltre a introdurre un overhead di memoria, non migliora lo speedup delle operazioni, peggiorando le prestazioni.

20

# LISTA DELLE TABELLE CON VINCOLI DI CHIAVE

**Utente**(email, nickname, password, nome, cognome, anno\_nascita, luogo\_nascita)

**Skill**(competenza, email\_Amministratore)

**Curriculum**(email\_Utente, nome\_competenza, livello)

**Amministratore**(email\_Utente, codice\_sicurezza)

**Creatore**(email\_Utente, nr\_progetti, affidabilità)

**Progetto**(nome, descrizione, data\_inserimento, budget, data\_limite, stato, tipo, email\_creatore)

**Foto**(id, nome\_Progetto, immagine)

**Reward**(codice, descrizione, foto, nome\_Progetto)

**Componente**(nome, descrizione, prezzo)

**Lista\_Componenti**(nome\_Componente, nome\_Progetto, quantita)

**Profilo**(id, nome)

**Profilo\_Software**(id\_profilo, nome\_Software)

**ProfiloSkill**(id\_profilo, nome\_competenza, livello)

**Finanziamento**(email\_Utente, nome\_Progetto, importo, data, codice\_Reward)

**Commento**(id, email\_Utente, nome\_Progetto, data, testo)

**Risposta**(id\_Commento, testo, email\_Creatore)

**Candidatura**(email\_Utente, id\_Profilo, nome\_progetto, stato)

21

A black and white abstract network graph with numerous small white dots connected by thin white lines, forming a complex web-like structure.

# LISTA DEI VINCOLI INTER-RELAZIONALI

Skill.email\_Amministratore → Amministratore.email\_Utente  
Curriculum.email\_Utente → Utente.email  
Curriculum.nome\_competenza → Skill.competenza  
Amministratore.email\_Utente → Utente.email  
Creatore.email\_Utente → Utente.email  
Progetto.email\_creatore → Creatore.email\_Utente  
Foto.nome\_Progetto → Progetto.nome  
Reward.nome\_Progetto → Progetto.nome  
Lista\_Componenti.nome\_Progetto → Progetto.nome  
Lista\_Componenti.nome\_Componente → Componente.nome  
Profili\_Software.id\_Profilo → Profilo.id  
Profili\_Software.nome\_Software → Progetto.nome  
ProfiloSkill.id\_profilo → Profilo.id  
ProfiloSkill.nome\_competenza → Skill.competenza  
Finanziamento.email\_Utente → Utente.email  
Finanziamento.nome\_Progetto → Progetto.nome  
Finanziamento.codice\_Reward → Reward.codice  
Commento.email\_Utente → Utente.email  
Commento.nome\_Progetto → Progetto.nome  
Risposta.id\_Commento → Commento.id  
Risposta.email\_Creatore → Creatore.email\_Utente  
Candidatura.email\_Utente → Utente.email  
Candidatura.id\_Profilo → Profilo.id  
Candidatura.nome\_Progetto → Progetto.nome

## 22

# NORMALIZZAZIONE

Non è stata necessaria la Normalizzazione poiché in tutte le tabelle viene rispettata la Terza Forma Normale, questo perché ogni attributo non chiave dipende interamente dalla chiave primaria e non ci sono dipendenze transitive.

La ridondanza #nr\_progetti, potendo essere calcolata derivandola dalla tabella Progetto contando le occorrenze dell'email\_Creatore, viola la TFN. Considerando quindi l'esito del calcolo dell'analisi della ridondanza e la violazione della TFN avremmo deciso di rimuovere tale ridondanza.

Però, essendo richiesto dalle specifiche di implementare un trigger per incrementare di un'unità il campo #nr\_progetti ogni qualvolta un utente creatore inserisce un progetto, abbiamo ritenuto opportuno conservare il campo in questione per rispettare le richieste.

## 2 3

# FUNZIONALITÀ DELL'APPLICAZIONE

L'applicazione è pensata per essere utilizzata solo da utenti registrati, perciò come prima schermata si fa decidere all'utilizzatore che tipo di utente sarà: creatore, amministratore o utente base. Dopodiché si visualizza la pagina di login, con eventuale link ad una pagina di registrazione nel caso in cui non si sia ancora creato un account.

Una volta effettuato l'accesso, tutti gli utenti vedranno le statistiche in homepage e avranno a disposizione un menù a scomparsa contenente le funzionalità. Gli utenti base vedranno solamente le funzionalità dedicate a qualsiasi utente mentre creatori e amministratori, in base al tipo di account, vedranno delle funzionalità personalizzate.

### Funzionalità visibili a tutti gli utenti

- Inserimento delle proprie skill di curriculum
- Visualizzazione dei progetti disponibili, dove sarà possibile per ogni progetto inserire un commento, per ogni progetto aperto effettuare un finanziamento e successivamente scegliere la reward, e nel caso di progetti software inserire una candidatura per un profilo richiesto.
- Logout

### Funzionalità visibili solo agli utenti amministratori

- Inserimento di una nuova competenza nella lista delle skills

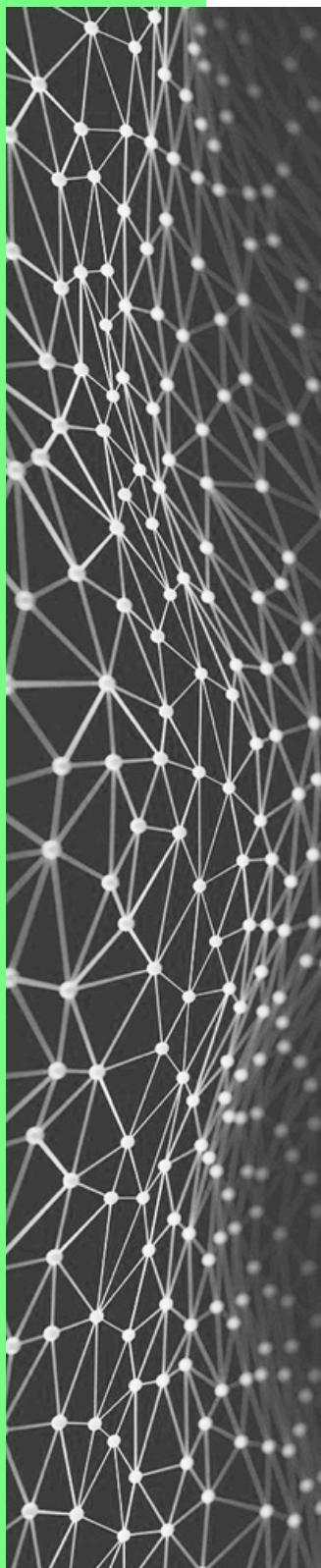
### Funzionalità visibili solo agli utenti creatori

- Inserimento di un nuovo progetto
- Inserimento delle reward per un progetto
- Dalla sezione “I Tuoi progetti” il creatore visualizzerà solo i progetti da lui creati e per ognuno potrà: inserire una risposta ad un commento, nel caso di progetti software inserire un profilo richiesto per la sua realizzazione e gestire le candidature, accettazione o rifiuto delle candidature inserite degli utenti.

## 24

## APPENDICE:

“Codice SQL dello schema della base di dati”



```
-- Tabella Utente
CREATE TABLE Utente (
    email VARCHAR(50) PRIMARY KEY,
    nickname VARCHAR(50) UNIQUE NOT NULL,
    password VARCHAR(50) NOT NULL,
    nome VARCHAR(50) NOT NULL,
    cognome VARCHAR(50) NOT NULL,
    anno_Nascita INT NOT NULL,
    luogo_Nascita VARCHAR(50) NOT NULL
)
engine= "InnoDB";

-- Tabella Skill
CREATE TABLE Skill (
    competenza VARCHAR(50) PRIMARY KEY,
    email_Amministratore VARCHAR(50) NOT NULL,
    FOREIGN KEY (email_Amministratore) REFERENCES Utente(email)
)
engine= "InnoDB";

-- Tabella Curriculum
CREATE TABLE Curriculum (
    email_Utente VARCHAR(50) NOT NULL,
    nome_Competenza VARCHAR(50) NOT NULL,
    livello INT NOT NULL, -- Il valore compreso tra 0 e 5 è stato gestito nelle procedure
    PRIMARY KEY (email_Utente, nome_Competenza),
    FOREIGN KEY (email_Utente) REFERENCES Utente(email),
    FOREIGN KEY (nome_Competenza) REFERENCES Skill(competenza)
)
engine= "InnoDB";
```

## 25

“*Codice SQL dello schema della base di dati*”

```
-- Tabella Amministratore
CREATE TABLE Amministratore (
    email_Utente VARCHAR(50) PRIMARY KEY,
    codice_Sicurezza VARCHAR(50) NOT NULL,
    FOREIGN KEY (email_Utente) REFERENCES Utente(email)
);

-- Tabella Creatore
CREATE TABLE Creatore (
    email_Utente VARCHAR(50) PRIMARY KEY,
    nr_progetti INT DEFAULT 0,
    affidabilita FLOAT, -- Da Rivedere
    FOREIGN KEY (email_Utente) REFERENCES Utente(email)
)
engine= "InnoDB";

-- Tabella Progetto
CREATE TABLE Progetto (
    nome VARCHAR(50) PRIMARY KEY,
    descrizione TEXT NOT NULL,
    data_Inserimento DATE NOT NULL,
    budget FLOAT NOT NULL,
    data_Limite DATE NOT NULL,
    stato ENUM('aperto', 'chiuso') DEFAULT 'aperto',
    tipo ENUM('hardware', 'software') NOT NULL,
    email_Creatore VARCHAR(50) NOT NULL,
    FOREIGN KEY (email_Creatore)
    REFERENCES Creatore(email_Utente)
)
engine= "InnoDB";

-- Tabella Foto
CREATE TABLE Foto(
    id INT AUTO_INCREMENT PRIMARY KEY,
    immagine BLOB,
    nome_Progetto VARCHAR(50) NOT NULL,
    FOREIGN KEY (nome_Progetto)
    REFERENCES Progetto(nome)
)
engine= "InnoDB";

-- Tabella Reward
CREATE TABLE Reward (
    codice INT AUTO_INCREMENT PRIMARY KEY,
    descrizione TEXT NOT NULL,
    foto BLOB,
    nome_Progetto VARCHAR(50) NOT NULL,
    FOREIGN KEY (nome_Progetto)
    REFERENCES Progetto(nome)
)
engine= "InnoDB";

-- Tabella Componente
CREATE TABLE Componente (
    nome VARCHAR(50) PRIMARY KEY,
    descrizione TEXT NOT NULL,
    prezzo FLOAT NOT NULL
)
engine= "InnoDB";
```

## 26

“*Codice SQL dello schema della base di dati*”

```
-- Tabella Lista_Componenti
CREATE TABLE Lista_Componenti (
    nome_Componente VARCHAR(50) NOT NULL,
    nome_Progetto VARCHAR(50) NOT NULL,
    quantita INT NOT NULL,
    PRIMARY KEY (nome_Componente,
    nome_Progetto),
    FOREIGN KEY (nome_Componente)
    REFERENCES Componente(nome),
    FOREIGN KEY (nome_Progetto)
    REFERENCES Progetto(nome)
)
engine= "InnoDB";
```

```
-- Tabella Profilo
CREATE TABLE Profilo (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(50) NOT NULL
)
engine= "InnoDB";
```

```
CREATE TABLE Profilo_Software(
    nome_Software VARCHAR(50),
    id_Profilo INT AUTO_INCREMENT,
    PRIMARY KEY (nome_Software, id_Profilo),
    FOREIGN KEY (nome_Software)
    REFERENCES Progetto(nome),
    FOREIGN KEY (id_Profilo) REFERENCES
    Profilo(id)
)
engine= "InnoDB";
```

```
-- Tabella ProfiloSkill
CREATE TABLE ProfiloSkill (
    id_Profilo INT NOT NULL,
    nome_Compетенция VARCHAR(50) NOT NULL,
    level INT,
    PRIMARY KEY (id_Profilo,
    nome_Compетенция),
    FOREIGN KEY (id_Profilo) REFERENCES
    Profilo(id),
    FOREIGN KEY (nome_Compетенция)
    REFERENCES Skill(competence)
)
engine= "InnoDB";
```

```
-- Tabella Finanziamento
CREATE TABLE Finanziamento (
    email_Utente VARCHAR(50) NOT NULL,
    nome_Progetto VARCHAR(50) NOT NULL,
    importo FLOAT NOT NULL,
    data DATE NOT NULL,
    codice_Reward INT,
    PRIMARY KEY (email_Utente,
    nome_Progetto, data),
    FOREIGN KEY (email_Utente) REFERENCES
    Utente(email),
    FOREIGN KEY (nome_Progetto)
    REFERENCES Progetto(nome),
    FOREIGN KEY (codice_Reward)
    REFERENCES Reward(codice)
)
engine= "InnoDB";
```

## 27

“*Codice SQL dello schema della base di dati*”

```
-- Tabella Commento
CREATE TABLE Commento (
    id INT AUTO_INCREMENT PRIMARY KEY,
    email_Utente VARCHAR(50) NOT NULL,
    nome_Progetto VARCHAR(50) NOT NULL,
    data DATE NOT NULL,
    testo TEXT NOT NULL,
    FOREIGN KEY (email_Utente) REFERENCES Utente(email),
    FOREIGN KEY (nome_Progetto)
    REFERENCES Progetto(nome)
)
engine= "InnoDB";
```

```
-- Tabella Risposta
CREATE TABLE Risposta (
    id_Commento INT PRIMARY KEY,
    testo TEXT NOT NULL,
    email_Creatore VARCHAR(50) NOT NULL,
    FOREIGN KEY (id_Commento) REFERENCES Commento(id),
    FOREIGN KEY (email_Creatore)
    REFERENCES Creatore(email_Utente)
)
engine= "InnoDB";
```

```
-- Tabella Candidatura
CREATE TABLE Candidatura (
    email_Utente VARCHAR(50) NOT NULL,
    id_Profilo INT NOT NULL,
    stato ENUM('accettata', 'rifiutata', 'in attesa') DEFAULT 'in attesa',
    nome_Progetto VARCHAR(50) NOT NULL,
    PRIMARY KEY (email_Utente, id_Profilo,
    nome_Progetto),
    FOREIGN KEY (email_Utente) REFERENCES Utente(email),
    FOREIGN KEY (id_Profilo) REFERENCES Profilo(id),
    FOREIGN KEY (nome_Progetto)
    REFERENCES Progetto(nome)
)
engine= "InnoDB";
```