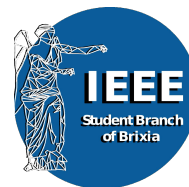




# PROGRAMMA ARNALDO

*TamaGolem*

*A.A. 2024/25*



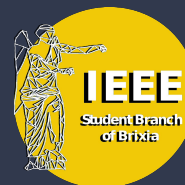


Il delicato Equilibrio del Mondo si basa da sempre sull'interazione fra le diverse forze naturali, dalle più miti alle più distruttive. Ogni elemento in natura ha i suoi punti forti e le sue debolezze, caratteristiche che mantengono il nostro Universo stabile e sicuro.

Da migliaia di anni, L'Accademia studia le tecniche per governare tali elementi: utilizzando alcune pietre particolari e dandole in pasto a strane creature denominate TamaGolem, infatti, è possibile conservare il potere degli elementi per liberarlo al bisogno.

Gli allievi dell'Accademia, per questo motivo, sono soliti sfidarsi in combattimenti clandestini fra TamaGolem. L'abilità dei combattenti, in questo caso, sta nella scelta delle giuste Pietre degli Elementi in modo che lo scontro abbia il risultato sperato. Tale scelta non è scontata, poiché gli Equilibri del Mondo sono mutevoli, e possono modificarsi radicalmente da una battaglia all'altra.

Solamente il TamaGolem che resiste fino alla fine decreta la vittoria del proprio combattente.



# Specifiche del programma

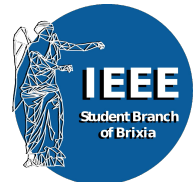
---



Il programma che dovete implementare permette a due giocatori di sfidarsi in un combattimento fra TamaGolem. Per “giocare”, l’utente deve comunicare al programma l’inizio di una nuova partita. Ogni partita si divide in tre fasi:

- La prima è di *setup*, non richiede l’interazione dell’utente e il risultato non viene rivelato. Serve a stabilizzare gli equilibri fra le forze del Mondo, ossia a stabilire il funzionamento delle interazioni fra gli elementi per quella specifica partita.
- La seconda comprende lo *Scontro* fra i due giocatori, secondo meccaniche che verranno spiegate nelle diapositive successive.
- La fase finale prevede la dichiarazione del vincitore, e l’eventuale rivelazione dell’*Equilibrio del Mondo* (vedi slide successive).

Il programma, infine, chiede all’utente se vuole iniziare una nuova partita o se intende terminare l’esecuzione dello stesso.



# L'Equilibrio del Mondo

---



All'inizio di ogni nuova partita l'Universo stabilisce casualmente le regole che governano le interazioni fra le forze naturali. L'insieme di queste interazioni è detto ***Equilibrio del Mondo***. Definire questo *Equilibrio* è ciò che deve fare il programma durante la **Fase 1**.

Come già detto, l'*Equilibrio* non è noto ai giocatori, e la sua creazione avviene "in background" all'inizio di una partita. Nonostante ciò, l'*Equilibrio* è fondamentale per definire l'esito degli scontri durante la fase di combattimento.

L'*Equilibrio* coinvolge tutti gli elementi presenti nella partita. Il numero di elementi  $N$  è arbitrario, può essere deciso dai programmatori e influenza direttamente la difficoltà del gioco. Tuttavia, affinché esista l'equilibrio,  $N$  deve essere almeno pari a 3; al contrario, valori di  $N$  superiori a 10 potrebbero rendere la partita impossibile per un umano non allenato.



# Creazione dell'Equilibrio

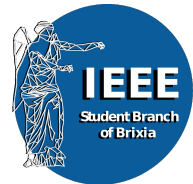
---



Una volta definiti gli elementi in gioco, l'*Equilibrio* viene generato in maniera pseudo-casuale, facendo attenzione che vengano rispettate le seguenti regole:

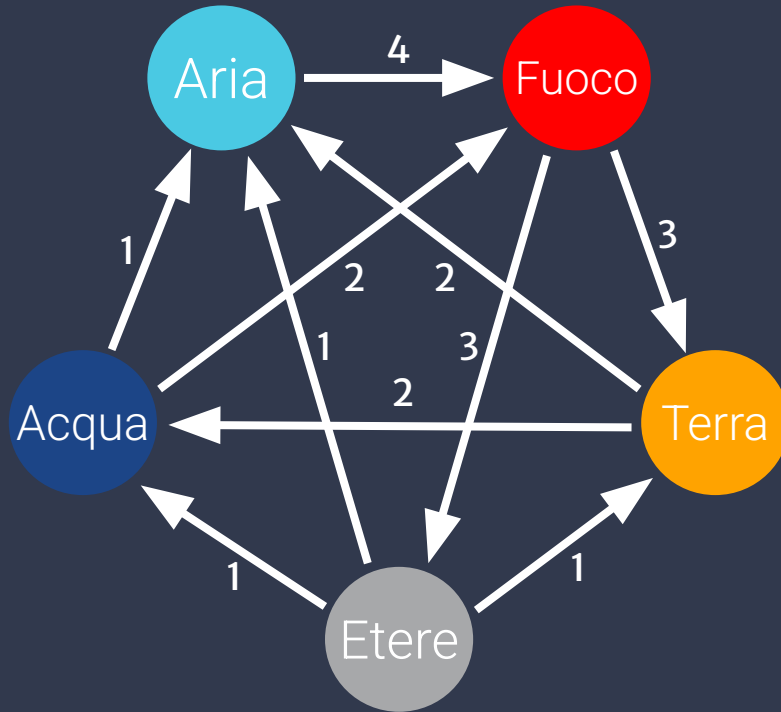
- Dati due elementi **differenti**, la loro interazione non è mai nulla.
- In particolare, per ciascuna coppia di elementi, uno solo dei due svolgerà il ruolo di elemento *forte*, mentre l'altro sarà l'elemento *debole*. Questi ruoli non si modificano nel corso della partita.
- Dati due elementi **uguali**, la loro interazione è sempre nulla.
- L'interazione di due elementi è caratterizzata da uno specifico valore, detto *potenza dell'interazione* (o anche solo *potenza*). Anch'esso non si modifica durante la partita.
- Per ciascun elemento, la somma delle *potenze* delle interazioni in cui tale elemento è forte è uguale alla somma delle *potenze* delle interazioni in cui è debole (**proprietà fondamentale dell'Equilibrio**).

Trovare l'algoritmo migliore per generare l'*Equilibrio* è la sfida che voi programmatori dovrete affrontare per implementare la Fase 1.

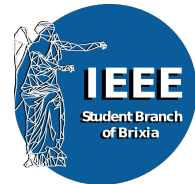
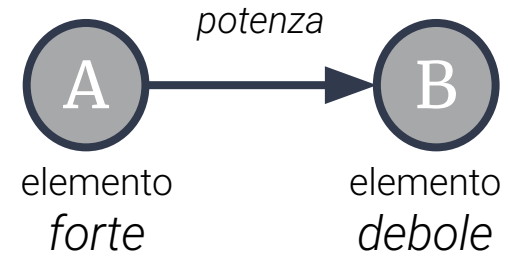


# Esempio

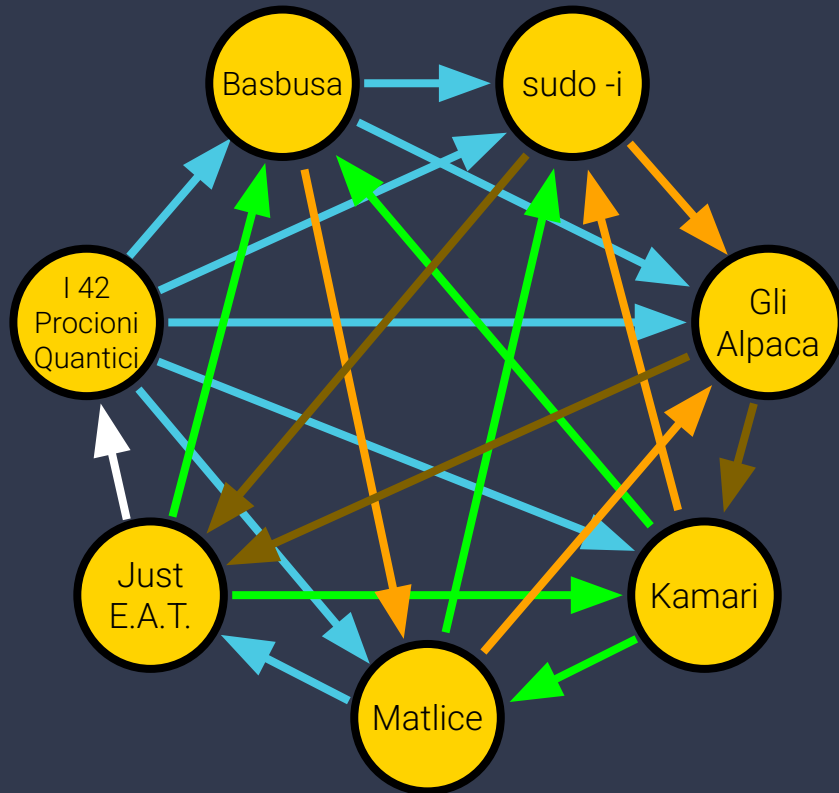
Equilibrio con 5 elementi:



Legenda:



## Esempio 4.2.2

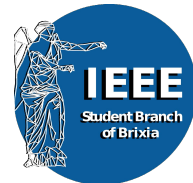


### Equilibrio con 7 elementi

I sette tondi in cerchio rappresentano sette elementi, a prescindere dai nomi, che interagiscono secondo le frecce.

#### Legenda dei valori delle frecce:

- 1
- 2
- 3
- 4
- 5



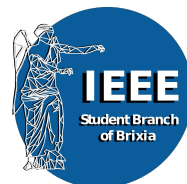
# Panoramica dello Scontro

---



Una volta che è stato costruito l'*Equilibrio* comincia la fase dello *Scontro*. Per la fase dello scontro, in generale succede che:

- Ogni giocatore ha a disposizione una quantità  $G$  fissata di TamaGolem, che può evocare nel corso dello *Scontro*.
- Ogni TamaGolem può ingurgitare fino a  $P$  **pietre degli elementi**, ossia pietre speciali caratterizzate dal loro legame con uno (e **un solo**) degli  $N$  elementi.
- Può scendere in campo solo un TamaGolem per giocatore alla volta. I due TamaGolem in campo si sfideranno scagliando uno contro l'altro pietre degli elementi.
- Due pietre scagliate l'una contro l'altra scatenano il potere dei relativi elementi, che **interagiscono** secondo l'*Equilibrio* generato nella *Fase 1*. L'elemento debole causa dei **danni** al TamaGolem che l'ha scagliato.
- Quando un TamaGolem subisce un certo **numero massimo** di danni, denominato *Vita* del TamaGolem, viene eliminato dallo scontro. Il rispettivo giocatore deve quindi evocarne un altro.
- Chi perde  $G$  TamaGolem è considerato sconfitto. L'altro giocatore è il **vincitore**.





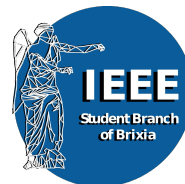
# Struttura dello Scontro

---



Lo scontro ha la seguente struttura:

1. Viene decretato il primo giocatore, d'ora in poi denotato con A.
  2. Il giocatore A schiera il primo TamaGolem, seguendo le regole per l'*evocazione* di un *TamaGolem* (spiegate più avanti).
  3. Dopodichè, il giocatore B fa lo stesso.
  4. I due TamaGolem cominciano un turno:
    - Essi scagliando la pietra del turno l'uno contro l'altro: i due elementi interagiscono.
    - Il TamaGolem che ha scagliato l'elemento debole fra i due subisce un numero di danni pari alla *potenza dell'interazione* fra i due elementi, e la sua *Vita* diminuisce.
  5. Se un TamaGolem ha raggiunto un valore di *Vita* **nullo o negativo**, allora:
    - Quel TamaGolem è **eliminato** dal gioco.
    - Se possibile, il relativo giocatore segue la fase di *evocazione* del nuovo TamaGolem.
- Altrimenti, se il TamaGolem ferito ha ancora *Vita* positiva:
- Viene segnalato ai giocatori il numero di danni inflitti al TamaGolem in quel turno.
- Se nessun giocatore è stato decretato vincitore, si torna al passo 4.



# Evocazione di un TamaGolem

---



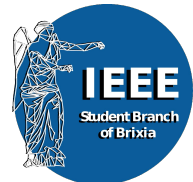
La procedura di *evocazione* di un TamaGolem è una sottofase della **fase 2** che ricorre più volte durante lo scontro. In generale l'*evocazione* interrompe lo *Scontro* ogni qualvolta sia necessario schierare in campo un nuovo TamaGolem da parte di un giocatore, evento che si verifica quando il TamaGolem precedente del giocatore è eliminato dallo *Scontro*.

L'*evocazione* si verifica solo se il giocatore ha ancora TamaGolem disponibili, ossia il numero di TamaGolem eliminati è strettamente inferiore a G. In caso **affermativo**:

1. Un nuovo TamaGolem viene selezionato per la fase dell'*evocazione*; la sua *Vita* viene inizializzata al valore massimo V.
2. Il giocatore seleziona esattamente P pietre degli elementi da far mangiare al TamaGolem, fra quelle a sua disposizione. Le pietre selezionate verranno scagliate ciclicamente finché il TamaGolem resterà in vita.
3. Il TamaGolem viene schierato in campo, l'*evocazione* termina e lo *Scontro* può procedere.

In caso **negativo**, invece:

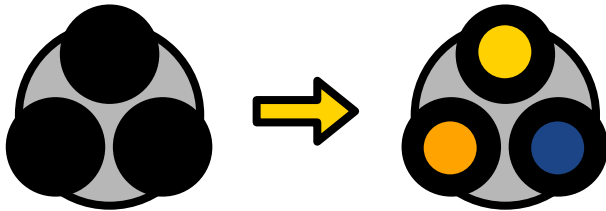
1. Il giocatore con G TamaGolem eliminati è considerato sconfitto.
2. L'altro giocatore è decretato vincitore.



# Caricamento delle Pietre



Sottofase di *evocazione*



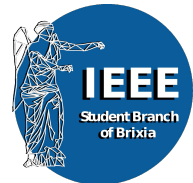
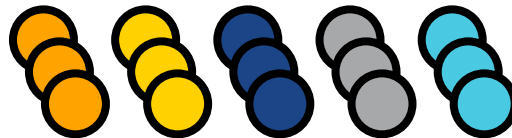
Esempio di caricamento di un TamaGolem  
con  $P = 3$  pietre degli elementi.

Durante la sottofase di *evocazione*, il giocatore attivo seleziona un set di  $P$  pietre fra quelle disponibili.

Questo set verrà quindi assegnato al TamaGolem e **non potrà più essere modificato** (né durante lo scontro né durante le successive *evocazioni*) e resterà in gioco finché il TamaGolem non verrà eliminato.

Le pietre selezionate per il set verranno sottratte da una scorta comune ad entrambi i giocatori; tale scorta, all'inizio della partita, prevede un totale di  $S$  pietre (calcolato secondo una formula che verrà mostrata più avanti).

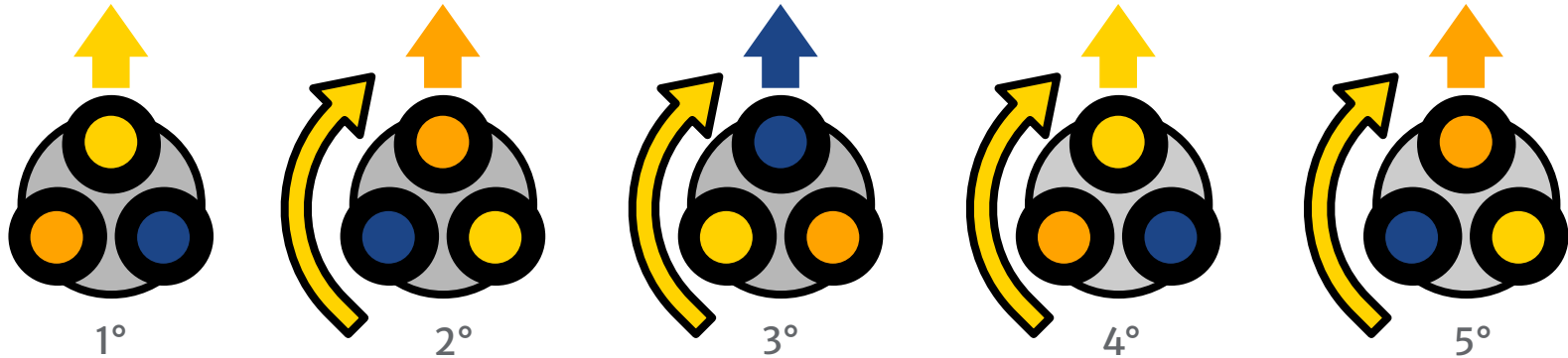
Esempio:  $S = 15$   
(con  $N=5$ ,  $P=3$  e  $G=2$ )



# Liberazione delle Pietre



## Fase 2 - Scontro



Come già anticipato, la particolare conformazione dell'apparato digerente dei TamaGolem permette a queste creature di "sputare" le pietre degli elementi in maniera **ciclica**, finché essi restano in vita. Può succedere, perciò, che un TamaGolem ripeta più di una volta la sequenza di pietre a lui assegnata, anche contro avversari differenti.



# Finale

---



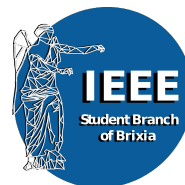
All'eliminazione dell'ultimo TamaGolem del giocatore sconfitto, si entra nella **fase finale**.

Questa fase prevede la **dichiarazione** del vincitore della partita, secondo le regole mostrate nelle slide precedenti, e la **rivelazione** dell'intero *Equilibrio*.

La rivelazione può essere fatta (ad esempio) visualizzando la lista di tutte le interazioni possibili fra i vari elementi, oppure attraverso una matrice. La scelta è libera, ma deve prevedere unicamente output da linea di comando.

## Note:

- Deve essere possibile incominciare una nuova partita senza dover azzerare l'esecuzione del programma.
- Non è richiesta alcuna memorizzazione delle partite passate, dei giocatori o dei vecchi *Equilibri* generati.
- Non è richiesto il calcolo di alcun punteggio per i giocatori, ma solamente chi è il giocatore vincente e chi quello sconfitto.



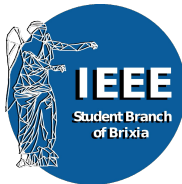
# Note sui parametri

---



Alcune note sui parametri citati nelle slides precedenti:

- $N, G, P, S, V$  sono quantità intere e positive.
- La scelta del numero di elementi è arbitraria; si consideri che:
  - $3 \leq N \leq 5$  per una partita di livello facile (anche se  $N=3$  ha poco senso).
  - $6 \leq N \leq 8$  per una partita di livello intermedio.
  - $9 \leq N \leq 10$  per una partita di livello difficile.
  - Altri valori di  $N$  potrebbero compromettere le meccaniche di gioco.
- Il numero di TamaGolem, in genere, vale  $G = \lceil (N - 1)(N - 2) / (2 * P) \rceil$ . È possibile diminuire (aumentare)  $G$  per incrementare (decrementare) la difficoltà di gioco.
- Il numero di pietre vale  $P = \lceil (N + 1) / 3 \rceil + 1$ .
- La quantità di pietre nella scorta comune vale  $S = \lceil (2 * G * P) / N \rceil * N$ . In particolare, per ciascun elemento sono presenti  $S / N = \lceil (2 * G * P) / N \rceil$  pietre.
- $V$  è arbitrario. Ad esempio,  $V = 10$ .
- Sia  $W$  l'insieme di tutti i valori di *potenza dell'interazione* presenti nell'*Equilibrio*. Allora vale:
  - $\sup(W) \leq V$ , oppure -se riuscite-  $\sup(W) = V$ .
  - $\inf(W) \geq 1$  (escluse le interazioni fra un elemento e se stesso).



# Esempio di gioco

Xing è stato scelto come primo giocatore. Durante la prima *evocazione*, nutre il suo TamaGolem con una pietra di Fuoco e due pietre d'Acqua.

Tocca a Yang, che nella sottofase *evocazione* fornisce al suo TamaGolem pietre di Aria, Fuoco e Terra.

Inizia lo **scontro**: entrambi i TamaGolem scagliano la prima pietra. L'Aria ha il sopravvento sul Fuoco, infliggendo 4 danni al TamaGolem di Xing.

Nel **secondo turno** l'Acqua vince sul Fuoco, infliggendo 2 danni al TamaGolem di Yang. Al **terzo turno**, il programma avvisa Xing che il suo TamaGolem ha perso tutti i suoi punti *Vita*, senza però specificare la *potenza dell'interazione* tra Acqua e Terra.

Xing è chiamato a svolgere una nuova *evocazione*: seleziona quindi Terra, Aria e Fuoco. In questo modo, vince i due turni successivi.

Commette però un errore, dando a Yang la possibilità di *evocare* un TamaGolem prima del turno in cui Xing lancerà la pietra di Fuoco. Yang, che ha appreso le leggi dell'*Equilibrio*, seleziona Aria, Acqua ed Etere e vince la partita.



Siano stati stabiliti i seguenti valori:

$$N = 5$$

$$G = 2$$

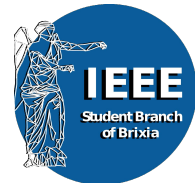
$$P = 3$$

$$S = 15$$

$$V = 5$$

e considero l'*Equilibrio* a cinque elementi mostrato nell'Esempio 4.2.1.

Si noti che questi valori non soddisfano a pieno tutte le richieste sui parametri (perché?).



# Modalità di Consegna

---

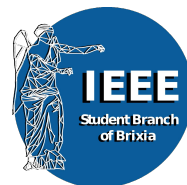


Una volta terminato il programma, esso dovrà essere caricato sul server **Forgejo** del programma, come repository privata all'interno dell'organizzazione denominata con il vostro nome gruppo.

La **repository** dovrà chiamarsi "TamaGolem", e dovrà contenere l'intero progetto del vostro gruppo con le eventuali dipendenze, in modo che poi si possa clonare e **sia già funzionante**. Il link deve essere inviato per email a [pgmarnaldo@googlegroups.com](mailto:pgmarnaldo@googlegroups.com) dal capogruppo, mettendo come oggetto il nome del gruppo e dell'esercizio/repository.

L'intero progetto è da consegnare entro la data sotto riportata. Eventuali modifiche successive all'orario di consegna non verranno accettate.

**Scadenza di consegna:  
ore 23:59, 15/05/2025**





# Note sulla valutazione

---



Oltre a valutare - come di consueto - il programma nel suo complesso, verranno valutate nello specifico le seguenti caratteristiche:

- Presenza di documentazione.
- **Progettazione** del software, struttura complessiva del codice e suddivisione in classi.
- Ideazione e implementazione dell'**algoritmo** per la creazione dell'*Equilibrio*, in particolar modo dal punto di vista dell'efficienza.
- Utilizzo delle **strutture dati** adatte per gestire alcune parti del programma.
- Comodità di **interazione** con l'utente, chiarezza nella visualizzazione dei dati e facilità d'inserimento di input durante la fase 2.
- **Rappresentazione** dell'*Equilibrio* durante la fase finale.
- **Solidità** del programma di fronte a scelte "peculiari" dell'utente (ossia: quanto è a prova di idiota il vostro codice? Avete previsto *tutti* i casi particolari?).
- Eventuali scelte di **implementazione** originali e interessanti che migliorino l'efficienza e l'esperienza dell'utente.
- Utilizzo corretto di Git e Forgejo.





- Che cosa sono quelle parentesi quadre “incomplete” nelle formule dei parametri?

[Floor and ceiling functions - Wikipedia](#)

- Guida pratica per Git e Gitea/GitHub:

[git - la guida tascabile - niente di complicato!](#)

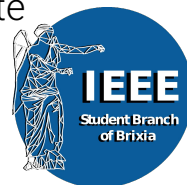
- [Bonus] Immagine di esempio di un TamaGolem:

[http://pre00.deviantart.net/daea/th/pre/i/2014/080/a/9/stone\\_golem\\_by\\_george\\_eracleous-d7b2y7y.jpg](http://pre00.deviantart.net/daea/th/pre/i/2014/080/a/9/stone_golem_by_george_eracleous-d7b2y7y.jpg)

- *Suggerimento nascosto*: non fatevi ingannare dalle “note sui parametri”: si tratta solo di fare molta attenzione a quali parametri fissate per primi!

- Se non vi vengono idee per i nomi degli elementi:

[Tipo - Pokémon Central Wiki](#) (non c'entra nulla con il programma, guardate solo i termini).

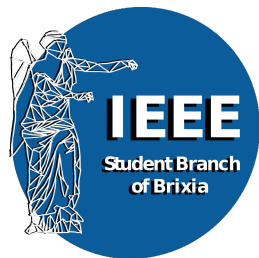


Presentazione realizzata per lo  
Student Branch IEEE  
dell'Università degli Studi di  
Brescia, in occasione del  
Programma Arnaldo 2025

*Si prega di non modificare o distribuire  
il contenuto di tale documento senza  
essere in possesso dei relativi  
permessi*

corazzinamarco33@ieee.org  
matteo.mottinelli@ieee.org  
matteo.boniotti@ieee.org

[ieeesb.unibs.it](http://ieeesb.unibs.it)



## Grazie per l'attenzione

**BUON LAVORO!**