# **Roadmap del Progetto "Sistema Evolutivo MIU"**

**Ultimo Aggiornamento:** 19 Giugno 2025, ore 10:50

Questo documento delinea le fasi di sviluppo del "Sistema Evolutivo MIU", un progetto mirato a creare un'intelligenza artificiale capace di derivare stringhe secondo le regole MIU, apprendendo e ottimizzando le proprie strategie di ricerca nel tempo.

## **Fase 1: Infrastruttura e Persistenza Dati**

* **Obiettivo:** Creare un'infrastruttura solida per la gestione del sistema MIU, inclusa la persistenza dei dati relativi a regole, stati, ricerche e applicazioni delle regole.
* **Stato:** **COMPLETATA** (20/06/2025).
* **Risultati Chiave:**
  + Integrazione di un sistema di logging robusto (MasterLog).
  + Configurazione del database SQLite (miu\_data.db) e creazione delle tabelle (RegoleMIU, MIU\_States, RuleApplications, Search).
  + Implementazione del MIURepository per interfacciarsi con il database (caricamento regole, salvataggio stati, applicazioni regole e ricerche).
  + Gestione degli eventi (OnRuleApplied, OnSolutionFound) per la persistenza automatica dei dati di ricerca e applicazione delle regole.

## **Fase 2: Apprendimento Continuo e Ottimizzazione della Ricerca**

* **Obiettivo:** Dotare il sistema della capacità di apprendere dalle proprie esperienze di ricerca per ottimizzare le future derivazioni delle stringhe, rendendo gli algoritmi più intelligenti e adattivi.
* **Stato:** **IN CORSO**.

### **Fase 2.1: Apprendimento Basato sulle Regole Generali (RuleStatistics)**

* **Obiettivo:** Implementare la raccolta e l'utilizzo delle statistiche aggregate per ogni singola regola MIU.
* **Stato:** **COMPLETATA**.
* **Risultati Chiave:**
  + Introduzione della tabella RuleStatistics (ID, ApplicationCount, SuccessfulCount, EffectivenessScore).
  + Logica per aggiornare ApplicationCount su OnRuleApplied.
  + Logica per aggiornare SuccessfulCount e ricalcolare EffectivenessScore su OnSolutionFound.
  + Persistenza delle RuleStatistics alla chiusura dell'applicazione.

### **Fase 2.2: Integrazione e Utilizzo dei Parametri di Ricerca e Ordinamento Regole (Apprendimento Generale)**

* **Obiettivo:** Consentire la configurazione dinamica dei parametri di ricerca e utilizzare le statistiche delle regole per ottimizzare l'ordine di applicazione delle regole.
* **Stato:** **COMPLETATA**.
* **Risultati Chiave:**
  + Caricamento di ProfonditaDiRicerca e MassimoPassiRicerca dal database.
  + Gestione interna di questi parametri in RegoleMIUManager tramite proprietà statiche.
  + Ordinamento dinamico delle regole in RegoleMIUManager (nei metodi ApplicaRegole, TrovaDerivazioneDFS, TrovaDerivazioneBFS) basato su RuleStatistics.EffectivenessScore e ApplicationCount.

### **Fase 2.3: Apprendimento Granulare (TransitionStatistics) e Raccolta Caratteristiche Stringhe**

* **Obiettivo:** Estendere l'apprendimento al livello delle singole transizioni (stato di partenza → stato di arrivo tramite una regola specifica) e **preparare il terreno per la scelta dinamica dell'algoritmo basata sulle caratteristiche specifiche delle stringhe**.
* **Stato:** **IN CORSO (Fase di Preparazione Dati e Struttura)**.

#### **Sotto-fase 2.3.1: Implementazione TransitionStatistics**

* **Obiettivo:** Tracciare le statistiche di successo e fallimento per specifiche transizioni di stato generate dall'applicazione di una regola.
* **Stato:** **COMPLETATA (già implementato dal gestore del progetto!)**.
* **Risultati Chiave:**
  + Introduzione della tabella TransitionStatistics (RuleID, OriginalStateHash, NewStateHash, ApplicationCount, SuccessfulCount, EffectivenessScore).
  + Logica in RegoleMIUManager\_OnRuleApplied per aggiornare ApplicationCount per la transizione specifica.
  + Logica in RegoleMIUManager\_OnSolutionFound per aggiornare SuccessfulCount e ricalcolare EffectivenessScore per le transizioni nel percorso soluzione.
  + Persistenza delle TransitionStatistics.

#### **Sotto-fase 2.3.2: Estensione della Tabella Search con Caratteristiche delle Stringhe**

* **Obiettivo:** Modificare lo schema della tabella Search per includere metriche chiave delle stringhe di partenza e di arrivo. Questo è fondamentale per abilitare l'apprendimento contestuale e la successiva scelta dinamica dell'algoritmo.
* **Stato:** **PROSSIMO PASSO CONCRETO**.
* **Colonne da Aggiungere alla tabella Search:**
  + InitialStringLength INTEGER
  + TargetStringLength INTEGER
  + InitialIcount INTEGER
  + InitialUcount INTEGER
  + TargetIcount INTEGER
  + TargetUcount INTEGER
* **Modifiche al Codice:**
  + MIURepository.cs: Aggiornamento del metodo InsertSearch per accettare e salvare i nuovi parametri delle caratteristiche.
  + Program.cs: Calcolo di queste caratteristiche (lunghezza, conteggio 'I' e 'U') per le stringhe iniziali e target prima di chiamare \_repository.InsertSearch, e passaggio dei nuovi parametri.

### **Fase 2.4: Motore di Decisione Dinamica sull'Algoritmo**

* **Obiettivo:** Implementare una logica di alto livello che, basandosi sulle caratteristiche delle stringhe di input (raccolte in 2.3.2) e sulle statistiche storiche aggregate per problemi simili, scelga dinamicamente l'algoritmo di ricerca più promettente (BFS, DFS, o futuri IDDFS/A\*). Questo permetterà al sistema di adattarsi alla natura specifica di ogni problema di derivazione.
* **Stato:** **DA INIZIARE**.
* **Sotto-fasi e Risultati Attesi:**
  + **2.4.1: Analisi Storica Dati:** Sviluppo di query e logica per analizzare l'efficacia (tasso di successo, tempo medio, nodi esplorati) di BFS e DFS su tipi di problemi categorizzati dalle caratteristiche delle stringhe (es. lunghezza, conteggio I/U).
  + **2.4.2: Logica di Selezione Algoritmo:** Sviluppo di un "Decision Maker" (potenzialmente un nuovo metodo in Program.cs o una nuova classe) che valuti le caratteristiche di InitialString e TargetString di una nuova ricerca e, basandosi sulle analisi storiche, decida quale algoritmo (BFS o DFS) chiamare.
  + **2.4.3: Potenziale IDDFS/A\*:** Considerazione dell'implementazione di algoritmi ibridi o informati (come Iterative Deepening DFS o A\*) come opzioni di scelta dinamica, se i dati indicano un vantaggio per specifici scenari.

## **Fase 3: Ottimizzazione del Riconoscimento di Pattern e Funzionalità Avanzate**

* **Obiettivo:** Affinare la capacità del sistema di riconoscere e applicare pattern in modo più intelligente, estendendo le sue capacità oltre la semplice derivazione di stringhe.
* **Stato:** **DA INIZIARE**.

### **Impatto della Scelta Dinamica dell'Algoritmo (Fase 2.4) sulla Fase 3:**

L'implementazione della Fase 2.4 **potenzia significativamente la Fase 3**, rendendo i suoi obiettivi più realizzabili e performanti:

* **Maggiore Efficienza per la Fase 3.1 (Ottimizzazione dei Pattern):** Un motore di ricerca che sceglie l'algoritmo più efficiente per il problema specifico (grazie alla Fase 2.4) renderà il processo di "scoperta di meta-regole" e "sequenze efficaci" molto più rapido ed efficiente. Meno tempo speso in ricerche inefficienti significa un apprendimento di livello superiore più veloce e risultati di pattern più significativi.
* **Base più Solida per la Generazione di Stringhe (Fase 3.2):** La capacità di adattare la strategia di ricerca al compito di generazione (che spesso implica la ricerca in ampi spazi di stati o la ricerca a ritroso) sarà un vantaggio enorme, rendendo la generazione di stringhe più pratica ed efficiente.
* **Miglioramento delle Prestazioni Complessive:** L'efficienza sottostante del motore di ricerca (garantita dalla Fase 2.4) è un prerequisito fondamentale per raggiungere elevate prestazioni nelle funzionalità avanzate della Fase 3.

### **Sotto-fasi e Risultati Attesi:**

#### **Fase 3.1: Ottimizzazione e Astrazione dei Pattern (Base per Riconoscimento Avanzato)**

* **Obiettivo:** Rivedere e astrarre i pattern delle regole MIU per consentire un riconoscimento più flessibile e l'identificazione di "meta-regole" o sequenze di regole che producono risultati efficaci.
* **Attività:** Analisi delle sequenze di regole applicate nei percorsi di soluzione più efficienti; sviluppo di un sistema per identificare e memorizzare pattern di applicazione di regole complesse; potenziale introduzione di un meccanismo di "compressione" o "astrazione" di queste sequenze in nuove "regole composite".
* **Impatto Atteso:** Miglioramento dell'efficienza nel trovare soluzioni complesse, riducendo il numero di passi espliciti e migliorando la "comprensione" del sistema.

#### **Fase 3.2: Integrazione con la Generazione di Stringhe (Potenziale)**

* **Obiettivo:** Esplorare la possibilità di utilizzare le regole MIU per generare nuove stringhe con proprietà desiderate, non solo derivare stringhe esistenti.
* **Attività:** Ricerca e prototipazione di algoritmi per la generazione inversa o basata su obiettivi.
* **Impatto Atteso:** Estensione delle capacità del sistema, trasformandolo in uno strumento più creativo e predittivo.

#### **Fase 3.3: Interfaccia Utente e Visualizzazione (Potenziale)**

* **Obiettivo:** Se il contesto del progetto lo richiederà, sviluppare un'interfaccia utente grafica per visualizzare il processo di derivazione, l'albero degli stati e le statistiche di apprendimento in tempo reale.
* **Attività:** Design UI/UX, implementazione frontend.
* **Impatto Atteso:** Miglioramento dell'usabilità, della comprensione e dell'interazione con il sistema.

### **Prossimi Passi Concreti**

Il prossimo passo immediato è l'implementazione della **Fase 2.3.2: Estensione della Tabella Search con Caratteristiche delle Stringhe**. Questo getterà le fondamenta per la **Fase 2.4: Motore di Decisione Dinamica sull'Algoritmo**, che a sua volta rafforzerà le capacità future della **Fase 3**.