

ALMA MATER STUDIORUM  
UNIVERSITÀ DEGLI STUDI DI BOLOGNA

## Learning Path Recommendation

*Studio e progettazione di modelli di raccomandazione di learning paths  
e di risorse didattiche.*

Laurea Magistrale in  
Ingegneria e Scienze Informatiche

---

Relatore: Prof.ssa Antonella Carbonaro

Presentata da: Sokol Guri

ANNO ACCADEMICO 2020-2021  
Cesena

# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
<b>2</b>	<b>Background</b>	<b>4</b>
2.1	Ambito . . . . .	4
2.2	Sistemi e tecniche di raccomandazione . . . . .	5
2.2.1	Path Generation . . . . .	5
2.2.2	Path Sequence . . . . .	7
2.2.3	IRT (Item Response Theory) . . . . .	9
2.2.3.1	Assunzioni del IRT . . . . .	9
2.2.3.2	Parametri degli oggetti . . . . .	10
2.2.4	Three parameter logistic model . . . . .	10
2.3	Accessibilità . . . . .	11
<b>3</b>	<b>Web Semantico</b>	<b>12</b>
3.1	Ontologia . . . . .	12
3.1.1	lom . . . . .	12
3.1.2	accessible . . . . .	12
3.2	Knowledge base . . . . .	12
3.3	Linked Data . . . . .	12
3.4	Rule base model . . . . .	12
<b>4</b>	<b>Analisi</b>	<b>13</b>
4.1	Scopo del progetto . . . . .	13
4.1.1	Modellazione ontologica . . . . .	13
4.1.2	Rule and query base model . . . . .	14
4.1.3	Pathadora engine . . . . .	14
4.2	Analisi dei requisiti . . . . .	14
4.2.1	Requisiti funzionali . . . . .	14
4.2.2	Requisiti non funzionali . . . . .	15

<b>5</b>	<b>Progettazione</b>	<b>16</b>
5.1	Architettura . . . . .	17
5.2	Progettazione del knowledge graph . . . . .	18
5.2.1	L'università . . . . .	18
5.2.2	Lo studente . . . . .	19
5.3	Gestione delle iterazioni e richieste . . . . .	20
5.3.1	Inserimento . . . . .	20
5.3.2	Generazione di facoltà . . . . .	20
5.3.3	Generazione di corsi . . . . .	20
5.3.4	Generazione di risorse . . . . .	21
<b>6</b>	<b>Implementazione</b>	<b>22</b>
6.1	Fase di sviluppo . . . . .	22
6.2	Tecnologie . . . . .	22
6.2.1	RDF . . . . .	22
6.2.2	OWL . . . . .	22
6.2.3	SWRL . . . . .	22
6.2.4	Web Scrapping . . . . .	22
6.3	Tools . . . . .	22
6.3.1	SWRL & OWL API . . . . .	22
6.3.2	Protege . . . . .	22
6.3.3	Stardog . . . . .	22
6.3.3.1	Docker . . . . .	22
<b>7</b>	<b>Risultati</b>	<b>23</b>
<b>8</b>	<b>Conclusioni</b>	<b>24</b>
	<b>References</b>	<b>24</b>

# Capitolo 1

## Introduzione

Il progresso nell'era digitale degli ultimi anni ha influenzato su tutti gli aspetti della vita quotidiana. Il progresso fatto dalle innovazioni tecnologiche sta ridefinendo e ristrutturando le metodologie di apprendimento. Con tale progresso e miglioramento, oggi gli utenti scelgono l'autoapprendimento attraverso l'internet. Al centro di questo cambiamento radicale sta l'E-learning.

Nei giorni d'oggi l'utilizzo di sistemi e-learning che forniscono risorse educative digitalizzate agli utenti, è diventata una normalità. L'e-learning porta tantissimi vantaggi rispetto ai metodi di apprendimento tradizionali con un insegnante che svolge il ruolo principale. Il vantaggio principale è l'aumento dell'accessibilità e della disponibilità delle risorse, riducendo i costi e rispettando la flessibilità degli utenti. L'e-learning è un tipo di apprendimento a distanza che viene svolto tramite internet dove l'utente può accedere alle risorse in qualsiasi momento da qualsiasi posto.

Le metodologie tradizionali di apprendimento incentrate sull'insegnante sono state utilizzate per tantissimi anni come la soluzione più efficace e fattibile da implementare per ottenere i risultati migliori. Con il progresso fatto a livello tecnologico sono state rilevate tantissime problematiche che i metodi tradizionali portavano. Nella maggior parte degli utenti tali metodi portavano una disorganizzazione e disorientamento all'utente, fornendo una varietà di risorse disorganizzate e non strutturate. Questa problematica risultava critica se l'utente aveva un'esperienza di apprendimento limitata.

Da tantissime ricerche fatte sull'ambito dell'apprendimento è stato affermato che oltre alla digitalizzazione delle risorse, anche l'ordine delle risorse ha un grande impatto sulla qualità dell'apprendimento. Gradualmente alla metodologia di e-learning, sono stati associati modelli per l'organizzazione

delle risorse didattiche strutturate in una sequenza di materiali che rispettano l'ordine. Tale sequenza rappresenta un percorso di apprendimento personalizzato per un utente che lo guiderà al raggiungimento degli obiettivi prefissati. L'utilizzo di E-learning con i modelli di raccomandazione potrebbe ridurre significativamente il tempo necessario per raccogliere e organizzare le risorse e in questo modo migliorare l'esperienza dell'apprendimento.

I modelli di raccomandazione forniscono una sequenza di materiali didattici come *learning-path*, però possono essere applicati su diversi domini. Uno di questi domini, sempre collegato con l'insegnamento e l'apprendimento, è *academic-program-path*. I modelli applicati a tale dominio prevedono un percorso educativo adatto per un utente in base alle informazioni che si forniscono al modello. Rispettando l'organizzazione strutturale delle istituzioni educative come le università, questi modelli prevedono le scuole, i dipartimenti e le facoltà da raccomandare all'utente in base alle sue caratteristiche (passioni, obiettivi, stile di apprendimento, etc).

Il progetto di tesi propone un modello di raccomandazione del percorso accademico e del percorso di apprendimento per un utente basato sulla modellazione semantica del dominio. *Pathadora*, l'ontologia progettata per rappresentare e modellare i componenti di questo dominio, incorporando ontologie già esistenti sulla accessibilità e l'organizzazione strutturale delle istituzioni educative. Il modello di raccomandazione si basa su regole che estendono e inferiscono nuove relazioni semantiche tra i componenti della ontologia. Per ricevere le richieste di interrogazione e manipolazione della *knowledge* dell'ontologia è stato implementato *pathadora-recommender*, che svolge il ruolo di una engine sempre in esecuzione in attesa per computare la risposta alle richieste ricevute.

In questo documento di relazione verranno introdotti una varietà di modelli di raccomandazione che utilizzano diverse metodologie di progettazione, oltre alle regole semantiche. In seguito verrà spiegato in profondità la soluzione ontologica scelta per il modello di raccomandazione e le problematiche incontrate, focalizzandosi sugli vantaggi e svantaggi di tale scelta. Una sezione della relazione sarà dedicata alla progettazione del *pathadora-client* da parte di Andrea, che interagisce con l'engine e fornisce le richieste aspettando la risposta.

La relazione è organizzata come segue:

- Sezione 1- Introduction: introduce senza approfondire il problema e il

tema del progetto, paragonando le soluzioni tradizionali con quelle più innovative.

- Sezione 2- Background: *to be done*
- Sezione 3- Progettazione: *to be done*
- Sezione 4- Implementazione: *to be done*
- Sezione 5- Conclusioni: *to be done*

# Capitolo 2

## Background

### 2.1 Ambito

E-learning è uno strumento importantissimo per il miglioramento della qualità dell'istruzione e formazione, basandosi sulle tecnologie d'informazione. La mancanza di adattamento dei contenuti educativi per gli studenti è un problema che e-learning ha portato sin dall'inizio. Mettendo a disposizione degli studenti le stesse risorse allo stesso modo, non è la soluzione migliore e più efficiente possibile.

Provando di risolvere il problema della personalizzazione della raccomandazione di risorse didattiche è nato il concetto di *learning path*. Un learning path (percorso di apprendimento) consiste nella progettazione di una sequenza di attività di apprendimento che aiutino lo studente a raggiungere gli obiettivi prefissati. Durante gli anni la personalizzazione dei learning paths è diventata un problema importante da risolvere, perché tale soluzione si potrebbe applicare in vari domini, oltre all'aspetto didattico. Oltre alla raccomandazione delle risorse didattiche, si potrebbe raccomandare una sequenza di corsi da seguire, di video tutorial da vedere, di libri da leggere, di medicine da prendere, etc. La dinamicità di applicazione lo rende un problema molto complesso e complicato da trovare una soluzione unica e condivisa per tutti.

Dalla fine degli anni '60, i ricercatori hanno tentato di affrontare la personalizzazione, utilizzando diversi parametri, approcci e algoritmi. In seguito si spiegano i sistemi e le tecniche di raccomandazione studiate e implementate seguendo diverse soluzioni e modelli di progettazione.

## 2.2 Sistemi e tecniche di raccomandazione

Lo sviluppo di modelli di raccomandazione inizia dagli anni 60' utilizzando una sequenza direzionale di risorse didattiche basandosi sui meccanismi di sequenza del curriculum presentato. Questi meccanismi generavano i learning paths, offrendo un'unica soluzione per tutti, siccome fornivano le stesse risorse didattiche. Come già accennato prima, la soluzione "one-fit-all" causava tanti problemi, ignorando la diversità degli studenti.

Per risolvere il problema di una soluzione unica, sono stati introdotti "personalized learning paths", focalizzandosi sulla personalizzazione e l'adattamento della soluzione allo specifico studente. In base ad alcuni parametri iniziali che rappresentano le caratteristiche dello studente, i modelli di raccomandazione generavano diverse soluzioni.

Con l'introduzione di nuove tecniche di raccomandazione comparivano nuove problematiche. Con le personalized learning paths, veniva ignorato il progresso che lo studente stava facendo durante tale percorso. Questo problema influenzava negativamente l'efficienza del percorso di apprendimento. Non tenendo in considerazione una tale problematica si rischiava che lo studente non utilizzasse più questo sistema mostrando una mancanza di correlazione tra il processo e il metodo di apprendimento.

Secondo Nabizadeh, Jorge, and Leal (2015) and Nabizadeh et al. (2017), i metodi di personalizzazione dei percorsi possono essere classificati in due macro categorie:

- *Path Generation*: questi metodi si focalizzano sulla generazione dell'intero percorso in un'unica raccomandazione e la valutazione dell'esperienza viene fatta alla fine del percorso.
- *Path Sequence*: questi metodi si focalizzano nella generazione di una sequenza di passi, che presi insieme formano il learning path. La raccomandazione del percorso intero viene fatta man mano che l'utente valuta l'andamento fino a quel punto.

### 2.2.1 Path Generation

Il processo di generazione del percorso di apprendimento intero consiste in diverse fasi. Durante la prima fase si stabiliscono le caratteristiche e i requisiti che l'utente dovrebbe specificare come parametri iniziali. Durante



la seconda fase, in base a tali parametri allo studente viene generato e consigliato un unico percorso.

Al posto di generare un percorso per ogni studente, alcune soluzioni tendono di raggruppare gli studenti tramite alcuni parametri in comune e poi raccomandare lo stesso percorso per tutti. Kardan et al. ha chiamato tale metodo come ACO-Map e consiste in una soluzione con due fasi principali. All'inizio si applica l'algoritmo K-means per dividere e raggruppare gli studenti e successivamente si ottimizza la soluzione tramite Ant-Colony-optimization in modo da generare un percorso per ogni gruppo.

GLPD (Groupized learning path discovering) è un'altra tecnica di raccomandazione di gruppo, dove inizialmente viene generato un grafo di topics e poi si raccolgono le conoscenze e preferenze dello studente. Il modello dovrà stimare i limiti temporali per tutti i gruppi di studenti sul completamento di un percorso. In base a tali limiti una certa strategia viene scelta per generare il percorso.

Basandosi sulla teoria dei grafi, Belacel propone una tecnica di raccomandazione dove i vertici rappresentano le risorse e gli archi le relazioni e le dipendenze tra le risorse. La fase iniziale consiste nella potatura del grafo, togliendo i vertici irrilevanti e raggruppando quelli simili. L'algoritmo Branch-and-Bound viene utilizzato per individuare il percorso più breve.

Seguendo la logica di Belacel, CourseNavigator è un altro metodo della categoria di Path Generation che genera un percorso di apprendimento tramite i grafi. CourseNavigator si basa sulla logica dei grafi con un algoritmo di ricerca sul un insieme di percorsi generata dalle caratteristiche dichiarate dell'utente. Le caratteristiche servono come vincoli nell'elenco dei percorsi generati.

Seguendo la stessa logica Xu ha progettato un metodo di raccomandazione che ha come obiettivo finale il completamento del percorso nel minor tempo possibile massimizzando il voto finale.

ECM (*Educational Concept Map*) viene considerato come uno dei metodi più efficienti della categoria Path Generation per la raccomandazione dei percorsi di apprendimento. Presentata da Adorni e Koceva nel 2015, si focalizza sulla conoscenza iniziale che uno studente ha, prima di iniziare il percorso. Lo studente definisce la sua conoscenza selezionando un insieme di argomenti nella mappa dei concetti. ECM utilizza l'algoritmo ENCODE

di Koceva, per linearizzare la mappa in base agli argomenti iniziali e target scelti dallo studente.

RUTICA è un metodo introdotto da Nabizadeh che ha come obbiettivo finale la generazione di un percorso che massimizza il voto finale allo studente con un limite temporale. Utilizzando l'algoritmo DFS (*Depth First Search*) si identificano tutti i possibili percorsi tenendo in considerazione il tempo. Per ogni percorso generato viene stimato e calcolato il voto finale.

Utilizzano i Markov chain, Xia ha progettato un sistema di raccomandazione basandosi su una sequenza di domande che lo studente dovrà rispondere. Le domande sono personalizzate in base allo studente e si focalizzano a estrarre lo storico dell'esercitazione dello studente e altri concetti comuni che lo studente potrebbe avere con altri studenti.

Christudas, nel 2018, ha proposto una tecnica di raccomandazione basandosi sul CGA (*Compatible Genetic algorithm*). Tale tecnica si basa sullo stile di apprendimento, la conoscenza e l'interattività dello studente.

In uno degli studi più recenti in questo campo, Liu utilizza complex-network-theory definendo i percorsi in tre scenari diversi basandosi sullo storico di apprendimento dello studente.

I metodi di Path Generation sono utilizzati ampiamente anche se portano alcuni svantaggi. Tali metodi ignorano e non prendono in considerazione step intermediari durante il percorso generato. Nel caso dei percorsi di apprendimento, i modelli non considerano l'andamento dello studente e un probabile aggiornamento dei parametri dichiarati. Il percorso è statico ed è impossibile aggiornarlo step-by-step. Un percorso non efficiente porterebbe uno spreco di risorse e un risultato non efficiente per l'utente finale.

### 2.2.2 Path Sequence

Diversamente dai metodi Path Generation, i metodi Path Sequence raccomandano gli step del percorso man mano che lo studente avanza tenendo traccia dell'andamento. Per risolvere tale problema sono stati utilizzati diversi approcci come: ALN (Association Link Network), EAs (Evolutionary Algorithms), IRT (Item Response Theory), Bayes, etc.

Govindarajan, nel 2016, applico un algoritmo EA, in questo caso PSO (*Parallel Particle Swarm Optimizatio*) per restituire un percorso personaliz-

zato e dinamico per gli studenti. Govindarajan raggruppo gli studenti in base alla loro competenza ovvero mappando una misura numerica alla possibilità di raggiungimento di un obiettivo target e il cambiamento di tale unità con l'andamento dello studente. La dinamicità del percorso consiste nella raccomandazione dello step successivo basandosi sul cambiamento di tale unità di misura.

Seguendo la logica presentata da Govindarajan, Li sviluppo un metodo per la generazione dei percorsi dinamici basandosi su due algoritmi, MLE (*Maximum Likelihood Estimation*) e GA (*Genetic Algorithm*). La fase iniziale del metodo consisteva nella creazione di una sequenza di risorse definendo anche la loro difficoltà in base ai feedback dello studente. Successivamente viene applicato MLE per analizzare le abilità e gli obiettivi di ogni studente. Alla fine un algoritmo GA e PSO vengono utilizzati per generare il percorso tramite i risultati ottenuti negli step precedenti. Per ogni step compiuto, i feedback dello studente servono ad aggiustare il livello di difficoltà delle risorse ed aggiornare le abilità dello studente.

Nel 2013, Yarandi pubblicava un studio sulla modellazione dei learning path utilizzando meccanismi del web semantico. Yarandi proponeva un sistema adattivo di e-learning utilizzando il modello ontologico del dominio focalizzandosi sulle abilità dello studente. Tale sistema riceveva come parametri d'input le abilità dello studente, la conoscenza, lo stile di apprendimento e le preferenze e alla fine generava un percorso basandosi su questi parametri. L'analisi delle risposte veniva fatta tramite IRT e successivamente aggiornava le abilità dell'utente.

Nello stesso anno, anche Salahli aveva proposto un sistema di raccomandazione utilizzando IRT. Il sistema inizialmente identificava i temi, le loro relazioni e la loro difficoltà. L'algoritmo IRT viene applicato per calcolare il grado di comprensione dei temi e il livello di conoscenza necessario per ogni tema. Quando un utente utilizza il sistema, il suo livello di conoscenza e la difficoltà del tema selezionato sono i parametri che servono per stimare il grado di comprensione.

La generazione di percorsi dinamici offre risultati efficienti della risoluzione del problema dei learning path, però porta alcuni svantaggi per quanto riguarda l'istante di tempo quando si devono aggiornare i profili degli studenti in base ai risultati ottenuti. L'utilizzo di un tempo statico e prefissato potrebbe risultare non efficiente in tutti i casi e porterebbe un risultato non corretto. L'aggiornamento continuo dei profili degli studenti nella maggior

parte dei casi porterebbe uno spreco di risorse computazionali.

### 2.2.3 IRT (Item Response Theory)

La teoria IR (Item Response) viene nota come la teoria della risposta latente, si riferisce a un insieme di modelli matematici con l'obiettivo di spiegare le relazioni, i risultati, le risposte e le prestazioni tra tratti latenti. Questa teoria stabilisce un collegamento tra le proprietà degli elementi, gli individui che rispondono e il tratto misurato. IRT si è basata sulla relazione tra le prestazioni degli individui su un elemento di prova e i livelli di prestazione dei partecipanti al test su una misura complessiva dell'abilità che l'elemento è stato progettato per misurare. Ogni risposta a un determinato elemento fornisce un grado di inclinazione sul livello o sull'abilità dell'individuo del tratto latente. L'abilità di un individuo è la probabilità di approvare la risposta come corretta e maggiore è l'abilità, maggiore è la probabilità di una risposta corretta.

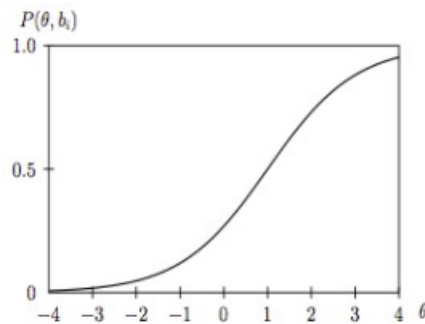


Figura 2.1: Curva caratterista dell'elemento.

*La forma del grafico consiste in una S (Sigmoide/Ogiva). La probabilità di ottenere una risposta corretta dipende dalla abilità del intervistato la quale nelle applicazioni rispetta il range  $-3$  and  $+3$ .*

#### 2.2.3.1 Assunzioni del IRT

- Monotonicità: l'assunzione specifica che la probabilità di ottenere una risposta corretta aumenterebbe aumentando il livello del tratto;
- Unidimensionalità: si assume che sia un tratto latente dominante da misurare che guida le risposte osservate per ogni elemento della misura;
- Indipendenza locale: si assume che le risposte date dagli elementi in un test sono reciprocamente indipendenti dato un livello di abilità;

- Invarianza: si possono stimare i parametri di un elemento da qualsiasi gruppo di soggetto che ha già risposto all'elemento.

Basandosi sulle ipotesi iniziali e considerandole valide, la variazione del tratto latente da parte degli intervistati definisce le differenze nell'osservazione delle risposte corrette. Il modello prevede le risposte degli intervistati su uno strumento basato su elementi tramite la loro posizione nel grafico continuo del tratto latente e alle caratteristiche di tali elementi. Queste caratteristiche vengono considerate parametri d'input per il modello.

### 2.2.3.2 Parametri degli oggetti

Le abilità e le capacità delle persone variano e sono molto dinamiche e per questo motivo la loro posizione nel grafico cambia. Il campione degli intervistati e i parametri degli oggetti definiscono la posizione precisa nel grafico.

- Item Difficulty (bi): parametro che definisce il comportamento dell'oggetto lungo la scala delle abilità. Viene calcolato dalla capacità con cui il 50% degli intervistati approvano una risposta come corretta. Gli oggetti che è difficile approvarli sono spostati più a destra, mentre quelli più facili più a sinistra;
- Item Discrimination (ai): parametro che definisce la velocità con cui la probabilità di approvare un oggetto corretto cambia in base ai livelli di abilità. Per ottenere una misura precisa, vengono inclusi elementi con discriminazione alta;
- Guessing (ci): parametro che tiene in considerazione le ipotesi su un determinato elemento limitando la probabilità di approvare la risposta se l'abilità va negativamente all'infinito.

### 2.2.4 Three parameter logistic model

Nel modello logistico a tre parametri (3PL), la probabilità di una risposta corretta a un oggetto dicotomico  $i$  per una domanda a scelta multipla, è:

$$p_i(\theta) = c_i + \frac{1 - c_i}{1 + e^{-a_i(\theta - b_i)}}$$

dove:

- $\theta$  indica che le abilità degli intervistati sono definite in base ad una distribuzione normale per stimare i parametri dell'oggetto. In seguito sono stimate le abilità di ogni intervistato;

- $a_i, b_i, c_i$  sono i parametri dell'oggetto;

Il cambiamento dei parametri viene rappresentato come il cambiamento della forma di una funzione logistica:

$$P(t) = \frac{1}{1 + e^{-t}}$$

dove:

- $a$  - discriminazione, scala e pendenza massimale:  $p'(b) = a \cdot (1 - c)/4$ ; Tale parametro rappresenta il grado della discriminazione dell'oggetto tra regioni diverse del grafico. Per esempio le persone con capacità basse hanno una probabilità molto minore di rispondere correttamente alla domanda.
- $b$  - difficoltà e posizione dell'oggetto:  $p(b) = a(1 + c)/2$ .  $p(b)$  in questo caso è il punto medio tra  $c_i$  (min) e 1 (max). In questo punto la pendenza è massimale.
- $c$  - ipotesi: minimo asintotico  $p(-\infty) = c$ . Il parametro tende a spiegare il caso quando un intervistato con basse capacità sceglie la risposta corretta.

Quando  $c = 0$ ,  $p(b) = 1/2$  e  $p'(b) = a/4$ ,  $b$  è uguale 50% probabilità di avere successo (difficoltà) e  $a$  è la pendenza massimale (discriminazione).

## 2.3 Accessibilità

ToDo

## Capitolo 3

# Web Semantico

Per tanti anni si parlava di web di documenti, considerando i documenti come l'elemento più importante della navigazione nei sistemi web based. Nei giorni d'oggi si menziona sempre di più, web dei dati, come nei database. L'obiettivo del web dei dati è di abilitare i compilatori ad effettuare un lavoro più efficiente e di sviluppare sistemi che possano supportare iterazioni sicure nella rete. Il termine web semantico rappresenta una grande quantità di dati interconnessi tramite relazioni e facilmente estendibili e accessibili dagli utenti.

L' articolo Scientific American considera il web semantico come *"un'estensione del web attuale dove alle informazioni viene dato un significato ben definito, consentendo ai computer e alle persone di lavorare in cooperazione."*

### 3.1 Ontologia

#### 3.1.1 lom

#### 3.1.2 accessible

### 3.2 Knowledge base

### 3.3 Linked Data

### 3.4 Rule base model

# Capitolo 4

## Analisi

### 4.1 Scopo del progetto

Pathadora ha come obbiettivo finale la generazione dei learning paths dinamici ed efficienti basandosi sugli parametri iniziali specificati dall'utente e dalle relazioni generate tra le istanze del sistema.

#### 4.1.1 Modellazione ontologica

Pathadora deve modellare tramite gli strumenti del web semantico diversi domini tra i quali:

- **l'istituzione accademica:** l'organizzazione dell'università in scuole, dipartimenti, facoltà e corsi. La modellazione dell'istituzione tramite una gerarchia permetterebbe di generare una sequenza di dipartimenti, facoltà e corsi tendono in considerazione le associazioni tra tali componenti. Le informazioni dichiarate dallo studente, permettono al sistema di iniziare la ricerca in cima della gerarchia, ovvero le scuole per poi scendere step-by-step fino ai corsi.
- **le informazioni dello studente:** tale informazioni possono essere:
  - generali: informazioni che specificano i dati anagrafici, le passioni, le lingue che parla, etc;
  - personali: informazioni sulle disabilità, sugli obbiettivi, sul metodo di apprendimento, etc;
  - accademici: informazioni che enfatizzano gli obbiettivi raggiunti durante il suo percorso accademico.



- le risorse didattiche: modellazione dei materiali didattici focalizzando sulle informazioni che riguardano l'accessibilità della risorsa e il topic del corso associato.
- l'accessibilità delle risorse; rappresenta l'accessibilità come un macro concetto e si focalizza sulla modellazione di informazioni sulla accessibilità che possono essere utilizzati per usufruire diversi tipi di risorse con diverse caratteristiche. L'ontologia mappa metodi di accessibilità di una risorsa e metodi di conversione di una risorsa in modo da offrire un'altra soluzione di accessibilità.

### 4.1.2 Rule and query base model

La logica del Pathadora si dovrebbe basare sulle regole in modo da creare un sistema *rule-based*. Tale sistema sarebbe in grado di memorizzare e manipolare la conoscenza e interpretare le informazioni in diversi modi. Pathadora utilizzerà questo sistema per fare deduzioni o scelte durante la generazione dei learning path basandosi sulle informazioni inferite dall'applicazione delle regole.

### 4.1.3 Pathadora engine

Pathadora-engine sarà una engine sempre in esecuzione che servirà le richieste ricevute da parte dell'utente. Tale engine offrirà i meccanismi necessari per interrogare la knowledge dell'ontologia.

## 4.2 Analisi dei requisiti

In questa sezione verranno spiegati i requisiti funzionali e non funzionali del progetto.

### 4.2.1 Requisiti funzionali

Il sistema finale del progetto dovrà soddisfare i seguenti requisiti funzionali:

- le ontologie:
  - pathadora-ontology: L'ontologia che modellerà tutto il dominio del sistema, racchiudendo i concetti dell'organizzazione dell'università, dei studenti, delle risorse e la loro accessibilità;

- accessibile: Sincronizzare l'ontologia dell'accessibilità con l'ontologia del sistema, pathadora-ontology;
  - lom: Sincronizzare l'ontologia che modella i metadati per le risorse didattiche con l'ontologia del sistema.
- gestire le richieste: L'utente potrà interagire con il sistema mandando delle richieste dal interfaccia web e il sistema dovrà essere in grado di accettare e servire queste richieste;
- manipolazione delle ontologie: Il sistema dovrà offrire meccanismi di manipolazione delle ontologie con l'inserimento di nuove istanze oppure i loro aggiornamento;
- manipolazione delle regole: L'utente deve essere in grado di personalizzare e parametrizzare le regole da applicare sulle ontologie presenti;
- manipolare le interrogazioni: L'utente deve essere in grado di manipolare le interrogazioni che verranno fatte al sistema di persistenza;
- Pathadora Recommender: Progettazione dell'engine che racchiude tutti i componenti del sistema.

#### 4.2.2 Requisiti non funzionali

Il sistema finale del progetto dovrà soddisfare i seguenti requisiti non funzionali:

- la prestazione: Il sistema deve essere in grado di gestire la richiesta e produrre una risposta nel minor tempo possibile, sfruttando al massimo le risorse computazionali messe a disposizione;
- la scalabilità: A livello di codice il sistema deve essere facilmente estendibile se sarebbe necessario l'introduzione di nuovi componenti;
- la disponibilità: Pathadora deve essere sempre in ascolto e servire le richieste, definendo tecniche di fault-tolerance;
- la consistenza: Il sistema deve essere in grado di mantenere la consistenza nel caso di inserimento o aggiornamento della knowledge, verificando lo stato degli strumenti utilizzati per la persistenza.

Maybe Glossario dei termini [here](#)

# Capitolo 5

## Progettazione



*Pathadora* è un sistema di raccomandazione di facoltà e corsi da seguire da uno studente in base alle caratteristiche specificate. Il sistema oltre alle facoltà e corsi, genera una sequenza di risorse didattiche da consigliare per un determinato corso scelto. Il learning path delle risorse dipenderà principalmente dal livello di accessibilità delle risorse e dalle eventuali disabilità dichiarate dello studente.

Pathadora gestisce le caratteristiche che riguardano gli studenti, le facoltà, i corsi e le risorse tramite la modellazione ontologica utilizzando strumenti e meccanismi del web semantico. Sfruttando tali strumenti, Pathadora è in grado di produrre un risultato dinamico e aggiornabile con nuove entità o caratteristiche che vengono aggiunte nel sistema. La generazione di relazioni tra le entità del sistema offre la possibilità di ottenere percorsi non uniformi e uguali per tutti, aumentando l'efficacia del sistema.

Gli utenti possono interagire con il sistema, tramite l'interfaccia web che Pathadora offre in modo da catturare e servire le richieste. *Pathadora-engine* si occupa della gestione degli strumenti ontologici del sistema e la produzione dei learning paths e la verifica di tali risultati.

## 5.1 Architettura

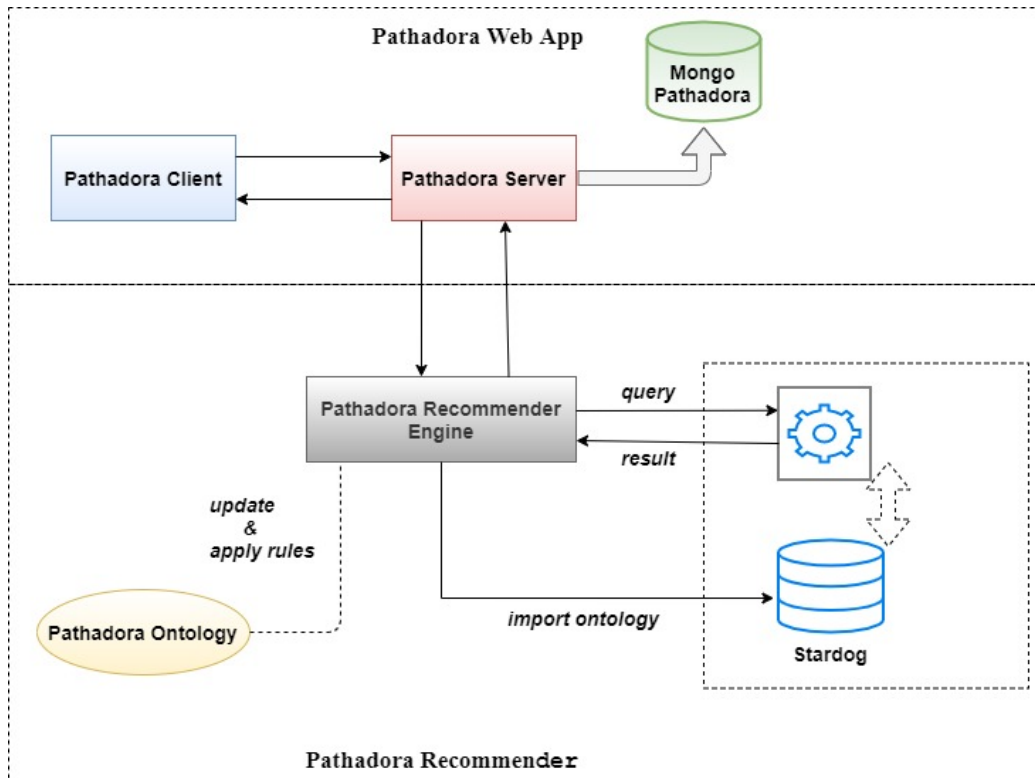


Figura 5.1: Architettura di Pathadora

Il sistema Pathadora consiste in due macro concetti principali, Pathadora Web App e Pathadora Recommender. Pathadora Web App offre una interfaccia web permettendo all'utente di interagire con il sistema. L'applicazione web è composta da tre blocchi importanti: il client, il server e il database.

Pathadora Ontology è un componente del sistema che ha come obiettivo di gestire e manipolare l'ontologia utilizzata per rappresentare il dominio del progetto. Tale componente interagisce con l'engine in caso diversi aggiornamenti o interrogazioni vengono richieste.

Pathadora Rule Model ha il compito di parametrizzare le regole ed effettuare la loro applicazione, notificando l'engine sull'andamento di tale operazione. Questa attività viene eseguita con l'inizializzazione del Pathadora Recommender e in quel momento il sistema è pronto ad ricevere richieste di interrogazione.

Per connettere le informazioni a livello di elaborazione e non di archiviazione, viene utilizzata una piattaforma esterna, Stardog. Tale componente ha l'obbligo di ospitare il knowledge graph dell'ontologia e preparare le risposte delle richieste fatte.

Pathadora Recommender racchiude tutta la modellazione e la logica dietro la generazione dei learning path. Questa engine rimane sempre in esecuzione ad intercettare le richieste e inoltrare al componente che ha il compito di gestirle.

## 5.2 Progettazione del knowledge graph

Per poter organizzare ed esprimere in modo logico le relazioni semantiche tra gli elementi del dominio, è stato progettato un knowledge graph. Tramite il knowledge graph vengono aggregate e dedotte le relazioni tra i concetti inseriti nella grafo. L'informazione verrà memorizzato in un database a grafo e visualizzato come una struttura a grafi.

Il grafo è composto da nodi e archi, dove i nodi modellano un elemento e l'arco modella la relazione di tale elemento con un altro. Qualsiasi componente del sistema potrebbe essere un nodo. La modellazione dell'ontologia Pathadora definisce un formato di rappresentazione delle istanze del knowledge graph.

La conoscenza viene rappresentata da un knowledge framework multidimensionale. Ogni concetto del dominio è stato modellato tramite le classi e le sottoclassi per le informazioni più dettagliate. La modellazione multidimensionale permette di aggregare le informazioni a diversi livelli di dettaglio.

### 5.2.1 L'università

L'università come istituzione è stata modellata tramite una gerarchia a tre livelli. In cima di questa gerarchia sono le scuole, per poi scendere fino ai corsi. Utilizzando la gerarchica e una organizzazione a livelli, è più facile mappare le preferenze di uno studente con determinato learning path. In questo modo, se si ottiene una scuola che soddisfa alcune caratteristiche dello studente, allora si assicura che scendendo sono questo ramo, i risultati ottenuti continuano a soddisfare le sue preferenze.

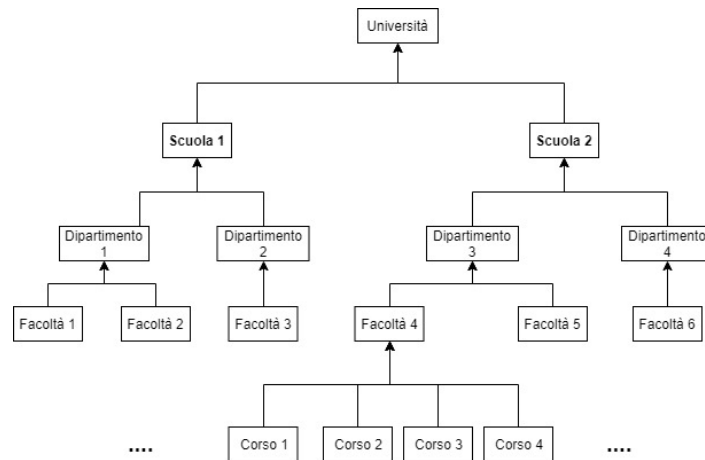


Figura 5.2: Organizzazione dell'università

### 5.2.2 Lo studente

La modellazione delle informazioni dello studente si basa su una knowledge graph. Per rappresentare le relazioni tra i diversi concetti in base al livello di dettaglio, le informazioni dell'utente sono state divise in tre classi: General, Personal e Accademic.

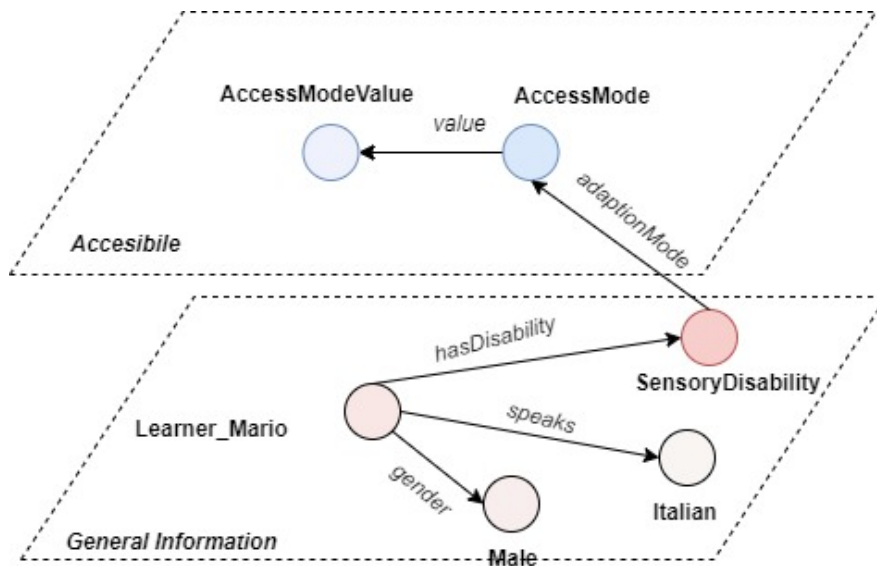


Figura 5.3: Knowledge graph sulle disabilità dello studente

## 5.3 Gestione delle iterazioni e richieste

Il sistema dovrà gestire le richieste che arrivano da parte del cliente e istanziare il componente adeguato a computare la risposta.

### 5.3.1 Inserimento

La prima tipologia di richiesta da gestire è l'inserimento di nuove istanze nell'ontologia. In questa richiesta dovranno essere specificate tutte le informazioni che serviranno all'engine per aggiornare la knowledge graph. La richiesta potrebbe dichiarare relazioni tra elementi che non sono presenti e in tal caso, l'engine prima creerà questi elementi per poi definire la relazione tra loro. Nel caso della richiesta di inserimento tutti gli elementi verranno aggiunti nell'ontologia, indipendentemente se esistono, se rispettano il formato giusto oppure se le relazioni sono semanticamente corrette.

### 5.3.2 Generazione di facoltà

La seconda tipologia di richiesta consiste nella generazione del percorso delle facoltà. Lo studente nella richiesta di generazione dovrà specificare il tipo di diploma che vuole conferire come parametro per selezionare le facoltà da consigliare. L'aggregazione delle facoltà verrà fatta applicando una regola, parametrizzata con il tipo di diploma da conferire. Oltre al tipo di diploma, il reasoner utilizzerà tutte le informazioni dichiarate dallo studente durante la registrazione per aggregare una lista di facoltà.

### 5.3.3 Generazione di corsi

La generazione dei corsi, viene fatta sempre dopo la generazione delle facoltà. L'utente dovrà interagire con il sistema per fare delle scelte che serviranno per generare il risultato del prossimo step. La generazione del percorso di studio, step-by-step, necessita che lo studente scegliesse la facoltà preferita dall'elenco delle facoltà raccomandate. Fatta tale scelta, lo studente dovrà specificare l'anno dei corsi che vorrebbe ricevere come raccomandazione. Il sistema Pathadora, una volta ricevuto tale scelta, inizierà il knowledge graph database e lo interrogherà parametrizzando la query con le informazioni dello studente.

### 5.3.4 Generazione di risorse

L'ultima richiesta da gestire consiste nella raccomandazione delle risorse, focalizzandosi sulla accessibilità delle risorse e sulle disabilità dello studente. Per ogni risorsa verranno definite informazioni che rappresentano il livello di accessibilità utilizzando coefficienti che specificano tale livello. Lo studente durante la registrazione definirà le disabilità, specificando un livello a scala numerica. Pathadora-engine dovrebbe mappare la correlazione tra questi coefficienti di disabilità e di accessibilità per aggregare una lista di risorse da raccomandare allo studente. Oltre alla accessibilità, la raccomandazione delle risorse si focalizzerà sulle caratteristiche dello studente come stile di apprendimento, obiettivi, etc. Anche in questo caso, la richiesta della generazione delle risorse, dovrà essere fatta dopo la generazione dei corsi, in modo che lo studente scegliesse il corso per il quale vorrebbe ricevere una lista di risorse.

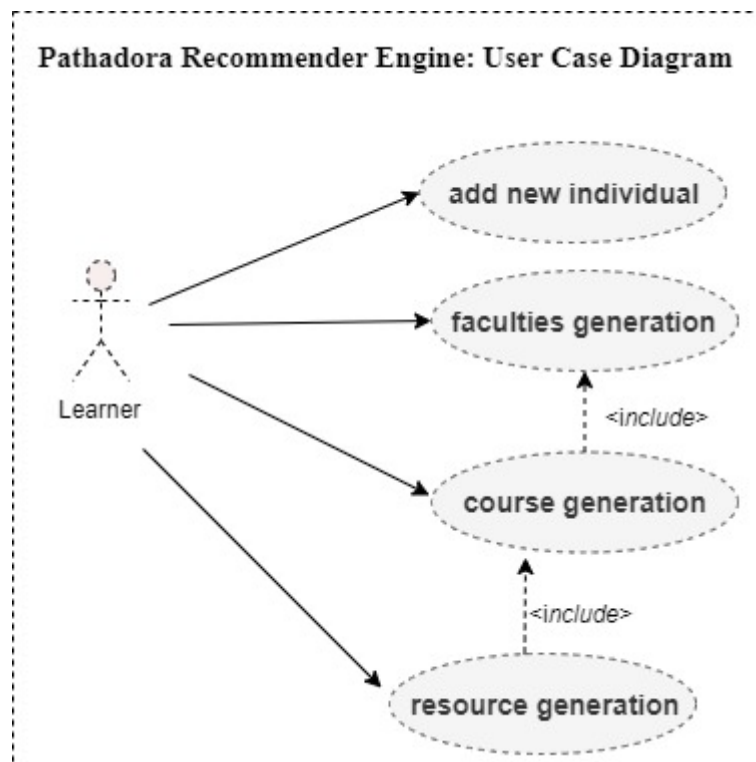


Figura 5.4: Pathadora Recommender Engine: User case diagram



# Capitolo 6

## Implementazione

### 6.1 Fase di sviluppo

### 6.2 Tecnologie

#### 6.2.1 RDF

#### 6.2.2 OWL

#### 6.2.3 SWRL

#### 6.2.4 Web Scrapping

### 6.3 Tools

#### 6.3.1 SWRL & OWL API

#### 6.3.2 Protege

#### 6.3.3 Stardog

##### 6.3.3.1 Docker

## Capitolo 7

### Risultati

## Capitolo 8

## Conclusioni

Conclusioni to be completed

# References

- [1] A learning path recommendation model based on a multidimensional knowledge graph framework for e-learning, <https://www.sciencedirect.com/science/article/pii/S095070512030085X?via%3Dihub>
- [2] <Name of the reference here>, <urlhere>

# Elenco delle figure

2.1	Curva caratteristica dell'elemento. <i>La forma del grafico consiste in una S (Sigmoide/Ogiva). La probabilità di ottenere una risposta corretta dipende dalla abilità del intervistato la quale nelle applicazioni rispetta il range -3 and +3.</i> . . . . .	9
5.1	Architettura di Pathadora . . . . .	17
5.2	Organizzazione dell'università . . . . .	19
5.3	Knowledge graph sulle disabilità dello studente . . . . .	19
5.4	Pathadora Recommender Engine: User case diagram . . . . .	21