

MULTIPROJECT ANALYSIS WITH CPM

Developed by
Michele Palumbo



MultiProject Analysis with Critical Path Method



CONTEXT

PROBLEM

SOLUTION



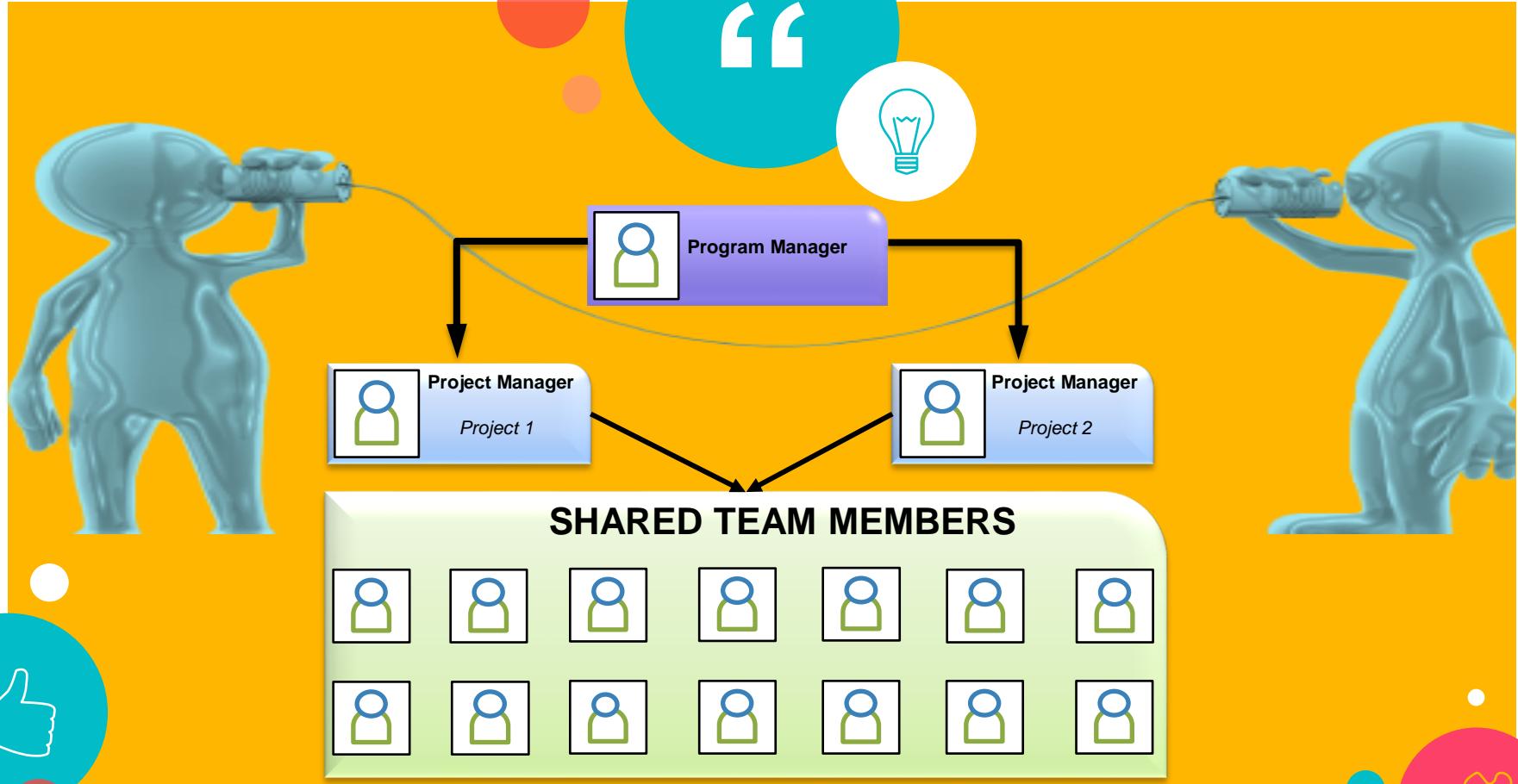


Context

- Questo Progetto è stato sviluppato per fornire un supporto di decisione a tutti i **Program Manager** che gestiscono più progetti con risorse condivise che, ovviamente, sono pianificati dai vari Project Manager assegnati.
- La fase in cui ho deciso di focalizzarmi è quella del *post planning*



“





Problem



PROBLEM

- Uno dei problemi più difficili da affrontare è quello di gestire risorse umane legate a più progetti, quindi risorse condivise.



Riuscire, quindi, ad analizzare se una data risorsa può risultare allocata in modo anomalo su più progetti oppure se si trova a cavallo di più attività critiche immediatamente successive legate sia al progetto singolo sia agli n-progetti su cui è allocata.



In dettaglio ...

PROBLEMI

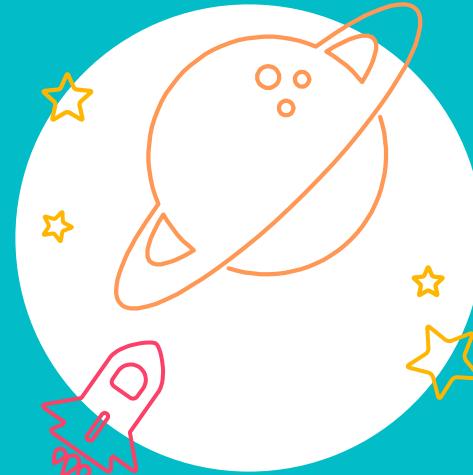
- CONFRONTO SE UNA RISORSA LAVORA CONSECUTIVAMENTE SULLO STESSO CRITICAL PATH
→ Analisi sui critical path singoli per verificare, eventuali, anomalie sui critical path singoli, confrontato ogni “task precedente” con il “task successivo”.
- CONFRONTO SE UNA RISORSA È ALLOCATA SU PIÙ CRITICAL PATH NELLO STESSO MOMENTO
→ Analisi dei progetti a due a due confrontandoli tra di loro, per tutti gli n -progetti presenti. In particolare, si va ad analizzare se tale risorsa può essere allocata contemporaneamente su più critical path, almeno un giorno.
- CONFRONTO SE UNA RISORSA SU PIÙ CRITICAL PATH È ALLOCATA CONSECUTIVAMENTE A PIÙ TASKS SUCCESSIVI DI DIFFERENTI PROGETTI
→ Si effettua un confronto interno per stabilire se una risorsa risulta allocata su task successivi di differenti critical path. Quindi, si confronta la data di fine task del $Project_{n-1}$ con la data di inizio del $Project_n$





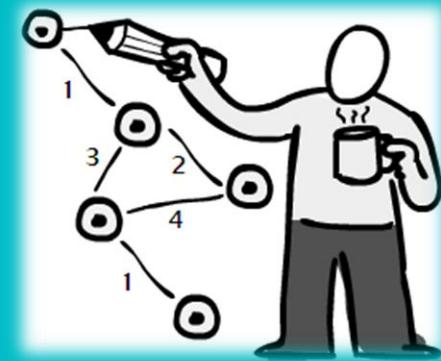
Solution

Per cercare di risolvere questi problemi, ho deciso di sviluppare un software seguendo l'approccio di analisi dati tramite il **Critical Path Method (CPM)**



Critical Path Method (CPM)

*Tecnica reticolare finalizzata
ad individuare il percorso
critico nella realizzazione
di un progetto*



Il CPM in breve

- È una tecnica di analisi del reticolo di schedulazione che prevede il **calcolo delle date di inizio e fine minime e massime** per ciascuna attività **schedulata**.
- Il **Critical Path Method** è una successiva applicazione semplificata del PERT.

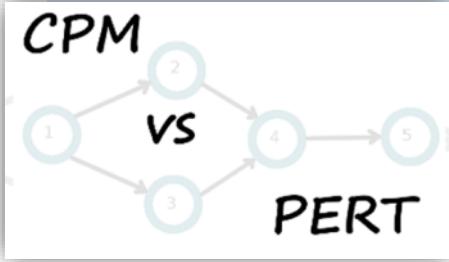
CRITICAL // PATH

CPM vs PERT

- Nel **CPM** la determinazione delle durate è **deterministica**
- Quindi, scompare la funzione di probabilità che vi è nel **PERT** in cui avviene con una stima a 3 fattori:
 A =durata ottimistica, M =durata più probabile, B =durata pessimistica e la media si calcola con questa formula

$$\frac{A + 4M + B}{6}$$

- Il **PERT** è indicato in caso di progetti in cui i tempi possono essere molto variabili
- Il **CPM** è più coerente in situazione di maggior accuratezza di dipendenza tra risorse e tempi.



Il CPM, definizioni



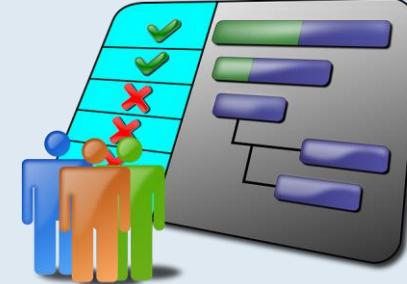
- La definizione di **percorso critico** è:
“la sequenza delle attività scheduled che determina la durata minima del progetto.
Normalmente è il **percorso più lungo** del progetto”.
- Quindi, il percorso critico è la più lunga catena di attività (o catene, se sono più di una) che hanno un **total float uguale a zero**.
- Il **Total Float** (anche noto come Total Slack) si può definire come la quantità di tempo (ho fatto riferimento a giorni lavorativi) di cui può slittare la data di inizio di un’attività senza incidere sulla data di fine di tutto il progetto.



I passi per attuare il Critical Path Method 1/6



1° Passo



Per costruire il reticolo occorre partire dalla **WBS** dove sono state elencate tutte le attività previste dal progetto e sono state stimate le loro durate.

→ Ho usato come **stima quella deterministica**, richiesta dal CPM



I passi per attuare il Critical Path Method

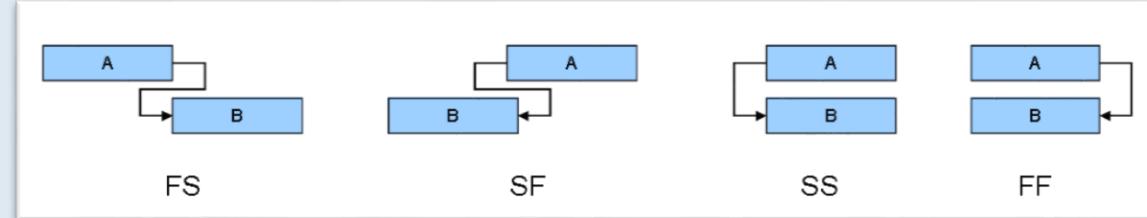
2/6



2° Passo

Dopo aver identificato le attività del progetto e averne stimato le durate, occorre definire i **vincoli logici**, ovvero le dipendenze sequenziali tra le attività.

- **FS – Finish to Start:** l'attività B non può iniziare se l'attività A non è finita.
- **SF – Start to Finish:** l'attività B non può finire se l'attività A non è iniziata
- **SS – Start to Start:** l'attività B non può iniziare se l'attività A non è iniziata
- **FF – Finish to Finish:** l'attività B non può finire se l'attività A non è finita



→ Ho usato il vincolo logico **FS – Finish to Start**

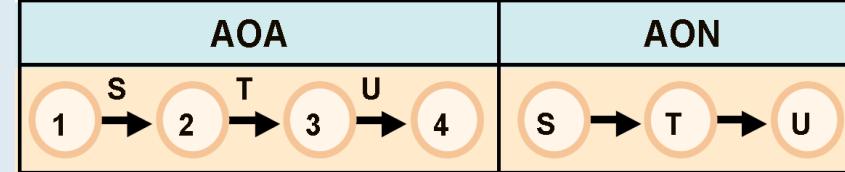


I passi per attuare il Critical Path Method

3/6



3°
Passo



Si rappresentano le attività, individuate mediante la WBS, su un grafo orientato e aciclico.

Le possibili modalità di rappresentazione sono:

- **AON (Activity On Node)** in cui le attività sono rappresentate dai nodi del grafo. Il primo nodo è sempre quello di inizio progetto; l'ultimo è sempre quello di fine progetto.
- **AOA (Activity On Arrow)** in cui le attività sono rappresentate dagli archi che collegano due nodi del grafo.

→ Ho usato la rete **AON** in cui ho un solo nodo iniziale e un solo nodo finale. Per poter garantire l'unicità della sorgente e del pozzo della rete ho deciso di utilizzare un nodo sorgente "Start" senza predecessori ed un nodo pozzo "Finish" senza successori.



I passi per attuare il Critical Path Method

4/6



4° Passo



Per calcolare gli “slack” è necessario determinare per ciascuna attività, **4 tempi**,

- **Earliest start time (ES) - Tempo di inizio al più presto**

È pari a il tempo di fine al più presto (**EF**) dell’attività precedente

Per attività con più di un predecessore, è il tempo **max** tra tutte le attività che precedono.

$$ES_j = \max(EF_i)$$

- **Earliest finish time (EF) - Tempo di fine al più presto**

È pari a il tempo di inizio al più presto (**ES_j**) + il tempo stimato di durata (**d_j**)

$$EF_j = EF_i + d_j$$

- **Latest start time (LS) - Tempo di inizio al più tardi**

È pari a il tempo di fine al più tardi (**LF_j**) dell’attività - la sua durata stimata (**d_j**)

$$LS_j = LF_i - d_j$$

- **Latest finish time (LF) - Tempo di fine al più tardi**

È pari a il tempo di inizio al più tardi (**LS_j**) dell’attività successiva

Per attività con più di un successore, è il **min** dei tempi di inizio al più tardi

$$LF_i = \min(LS_j)$$



I passi per attuare il Critical Path Method 5/6



5° Passo



In sintesi, è possibile dire che il metodo consiste in un calcolo ripetitivo svolto:

- prima “**in avanti**”, per ottenere le date minime di inizio e fine (**ES** and **EF**)
- poi “**all’indietro**” in modo tale da ottenere le date massime (**LS** and **LF**).

In base poi alla flessibilità delle attività si procederà all’individuazione delle criticità con relativo percorso critico.

Quindi, a questo punto, possiamo dire che il tempo totale necessario per completare il progetto è il primo momento di inizio del nodo “Finish”

$$\textbf{ProjectCompletionTime} = \textbf{ES}_{\textbf{Finish}}$$

Questo è anche il percorso più lungo del grafo (dall’inizio alla fine) in termini di durata cumulativa, dove questa durata cumulativa è il tempo di completamento del progetto.

Questo percorso più lungo è il critical path.



I passi per attuare il Critical Path Method

6/6



6° Passo



Il **critical path**, è costituito da quelle attività in cui un aumento di una qualsiasi delle loro durate aumenterebbe il tempo complessivo di completamento del progetto. Queste attività hanno un **slack = 0**, per cui non ci sarà modo di aumentare le loro durate senza far aumentare la durata complessiva del progetto.

Un'attività **j**, invece, non è sul critical path se ha uno slack positivo.

Lo **slack** si calcola come la differenza tra LS_j e ES_j :

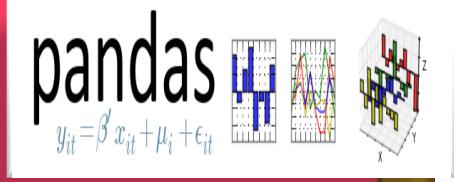
$$slack_j = LS_j - ES_j$$

Le attività con **slack positivo** possono essere aumentate (ovviamente sempre in base allo slack) senza compromettere il tempo complessivo di completamento del progetto.

Per concludere, possiamo dire, inoltre, che:

- $ES_{Start} = EF_{Start} = LS_{Start} = LF_{Start} = 0$
- $ES_{Finish} = EF_{Finish} = LS_{Finish} = LF_{Finish} = ProjCompletTime$





STRUMENTI UTILIZZATI

DB: Neo4j

- La struttura fondante per i database a grafo è la "node-relationship"; la memorizzazione e la navigazione avviene utilizzando dei nodi e loro relazioni che supportano le proprietà (coppia key-value).
- Neo4j supporta un linguaggio dichiarativo chiamato **Cypher**, progettato per interrogare i grafi ed i loro componenti.

Cypher using relationship 'likes'



(a) -[:LIKES]-> (b)

IDE: PyCharm

- PyCharm è un IDE per lo sviluppo in Python.



Librerie : Pandas e Py2neo

Pandas

- Pandas è una libreria software scritta per il linguaggio di programmazione Python per la manipolazione e l'analisi dei dati.

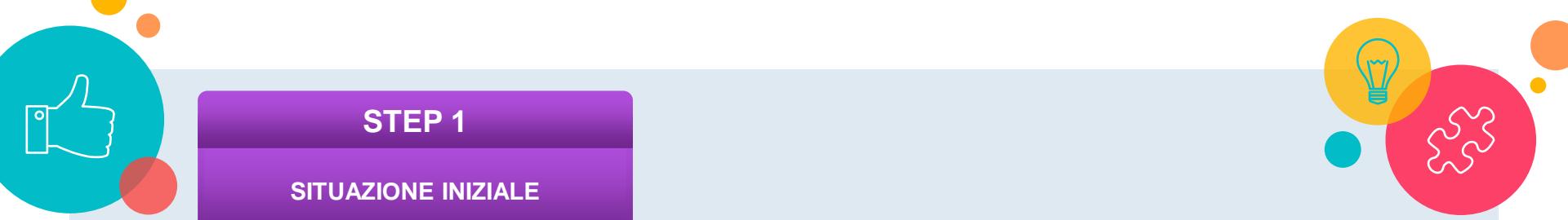
Py2neo

- Py2neo è una libreria client che serve per lavorare con Neo4j dall'interno di applicazioni Python.



Gli step della mia soluzione





STEP 1

SITUAZIONE INIZIALE

- Inizialmente, ho individuato le attività dei due progetti tramite WBS.
- Ho, quindi, pianificato per bene le risorse umane distribuite sui due progetti in modo da renderle compatibili tra di loro.
- Successivamente, ho portato le tre tabelle (due per i progetti e una per le risorse) in formato (.csv) per poterle leggere facilmente da pandas e caricarle sul database Neo4j.

→ Il vincolo per rendere la pianificazione compatibile con il sistema è quello di inserire un nodo “Start” ed un nodo “Finish” che corrispondono all’inizio del progetto e alla fine del progetto. Ed, ovviamente, dovranno avere la stessa struttura.

Il Dataset del Project1

Project1.csv

```
df_progl = pandas.read_csv('..\Users\Michele_Palumbo\Documents\Neo4j\default.graphdb\import\Project1.CSV')
df_progl
```

ID	attività	descrizione	predecessori	durata	inizio	fine	ruolo	nick
0	1	Start		Inizio progetto		0		
1	2	Task1		Actor identification	1	2 2017-09-04	2017-09-05	analyst marco1
2	3	Task2	Requirements functional and no-functional iden...		2	3 2017-09-06	2017-09-08	analyst milky1
3	4	Task3		Rilevant Scenarios identification	2	5 2017-09-06	2017-09-12	analyst angelo1
4	5	Task4		Use case	3;4	2 2017-09-13	2017-09-14	analyst angelo1
5	6	Task5		Mockup	3;4	4 2017-09-13	2017-09-18	graphic designer graziano1
6	7	Task6		Sequence Diagram	6;5	2 2017-09-19	2017-09-20	system designer alex1
7	8	Task7		Class Diagram	5;6	3 2017-09-19	2017-09-21	analyst marco1
8	9	Task8		Design Goals identification	8;7	2 2017-09-22	2017-09-25	system designer davide1
9	10	Task9		Architecture identification	8;9	3 2017-09-26	2017-09-28	system designer alex1
10	11	Task10		Sub-system identification	10	6 2017-09-29	2017-10-06	object designer manu1
11	12	Task11		Package Subdivision	10;11	3 2017-10-09	2017-10-11	object designer davide1
12	13	Task12		Design Pattern identification	10;11	5 2017-10-09	2017-10-13	object designer milky1
13	14	Task13		GUI(Front-End-App) develop	13;12	10 2017-10-16	2017-10-27	developer davide1
14	15	Task14	Component interface Develop DB(Back-end-App)		13;12	5 2017-10-16	2017-10-20	database manager marco1
15	16	Task15		Module GPS Develop	14;15	5 2017-10-30	2017-11-03	developer andrea1
16	17	Task16		Code Develop	14;15	10 2017-10-30	2017-11-10	developer graziano1
17	18	Task17		Unit Test	17;16	7 2017-11-13	2017-11-21	tester angelo1
18	19	Task18		Integration Test	17;16	4 2017-10-13	2017-11-16	tester manu1
19	20	Task19		User Manual	18;19	3 2017-11-22	2017-11-24	graphic designer gio1
20	21	Finish		Fine progetto	20	0		

Il Dataset del Project2

Project2.csv

```
df_prog2 = pandas.read_csv('..\Users\Michele Palumbo\Documents\Neo4j\default.graphdb\import\Project2.csv')
df_prog2
```

ID	attività	descrizione	predecessori	durata	inizio	fine	ruolo	nick
0	1	Start		Inizio progetto		0		
1	2	Task1		Conduct needs analysis	1	5 2017-09-25	2017-09-29	analyst miky1
2	3	Task2		Draft preliminary software specifications	2	4 2017-10-02	2017-10-05	analyst angelo1
3	4	Task3		Requirements elicitation	2	3 2017-10-02	2017-10-04	analyst miky1
4	5	Task4		Requirements analysis	3:4	5 2017-10-06	2017-10-12	analyst angelo1
5	6	Task5		Requirements specification	3:4	6 2017-10-06	2017-10-13	analyst marco1
6	7	Task6		Requirements validation	6:5	5 2017-10-16	2017-10-20	analyst miky1
7	8	Task7		Review preliminary software specifications	7	2 2017-10-23	2017-10-24	system designer alex1
8	9	Task8		Design of functional specifications	7	5 2017-10-23	2017-10-27	system designer andrea1
9	10	Task9		Design of software structure and architecture	8,9	3 2017-10-30	2017-11-01	system designer alex1
10	11	Task10		Software design quality analysis and evaluation	10	3 2017-11-02	2017-11-06	object designer manu1
11	12	Task11		Software design notations	10;11	3 2017-11-07	2017-11-09	object designer miky1
12	13	Task12		Development of functional specifications	10;11	5 2017-11-07	2017-11-13	developer davide1
13	14	Task13		Server side development	13;12	5 2017-11-14	2017-11-20	database manager marco1
14	15	Task14		Develop code	13;12	10 2017-11-14	2017-11-27	developer graziano1
15	16	Task15		Graphic design of the 3D items	14;15	10 2017-11-28	2017-12-11	graphic designer gio1
16	17	Task16		Unit testing	16	4 2017-12-12	2017-12-15	tester marco1
17	18	Task17		Integration testing	16	7 2017-12-12	2017-12-20	tester angelo1
18	19	Task18		User Manual	17;18	2 2017-12-21	2017-12-22	graphic designer graziano1
19	20	Finish		Fine progetto	19	0		

Il Dataset delle Risorse

Resources.csv

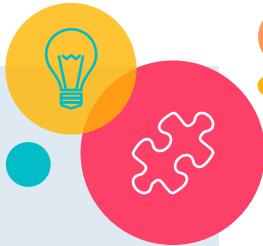
```
df_res = pandas.read_csv('C:\Users\Michele Palumbo\Documents\Neo4j\default.graphdb\import\Resources.csv')  
df_res
```

	ID	nickname	risorsa	ruolo
0	1	miky1	Michele Palumbo	analyst;object designer
1	2	angelo1	Angelo Conte	analyst;tester
2	3	alex1	Alessandro Russo	system designer;object designer;developer
3	4	davide1	Davide Mastricci	system designer;object designer;developer
4	5	graziano1	Graziano Fuccio	graphic designer;developer
5	6	gio1	Giovanni Ippolito	graphic designer;developer
6	7	andrea1	Andrea Fortino	system designer;developer
7	8	manu1	Emanuele Sculco	tester;object designer
8	9	marco1	Marco Parisi	analyst;database manager



STEP 2

CARICO I FILE .csv SU Neo4j E CREO I GRAFI



- Carico prima le risorse, in modo da poterle collegare, successivamente, ai progetti.

```
# CARICO il file Resources.csv su Neo4j
creaRisorse= "LOAD CSV WITH HEADERS FROM 'file:///Resources.csv' AS line "
    "CREATE (node:Resources { ID: line.ID, nickname: line.nickname, risorsa: line.risorsa, ruolo: SPLIT(line.ruolo, ';') })"
graph.run(creaRisorse).dump()
```

- Carico e creo un grafo per ogni progetto.

Per poter rendere il sistema generico e, quindi, un solo grafo che potesse andar bene per ogni progetto ho fatto in modo di prendermi dalla cartella contenenti i file .csv tutti quei file che iniziano con “Project”.

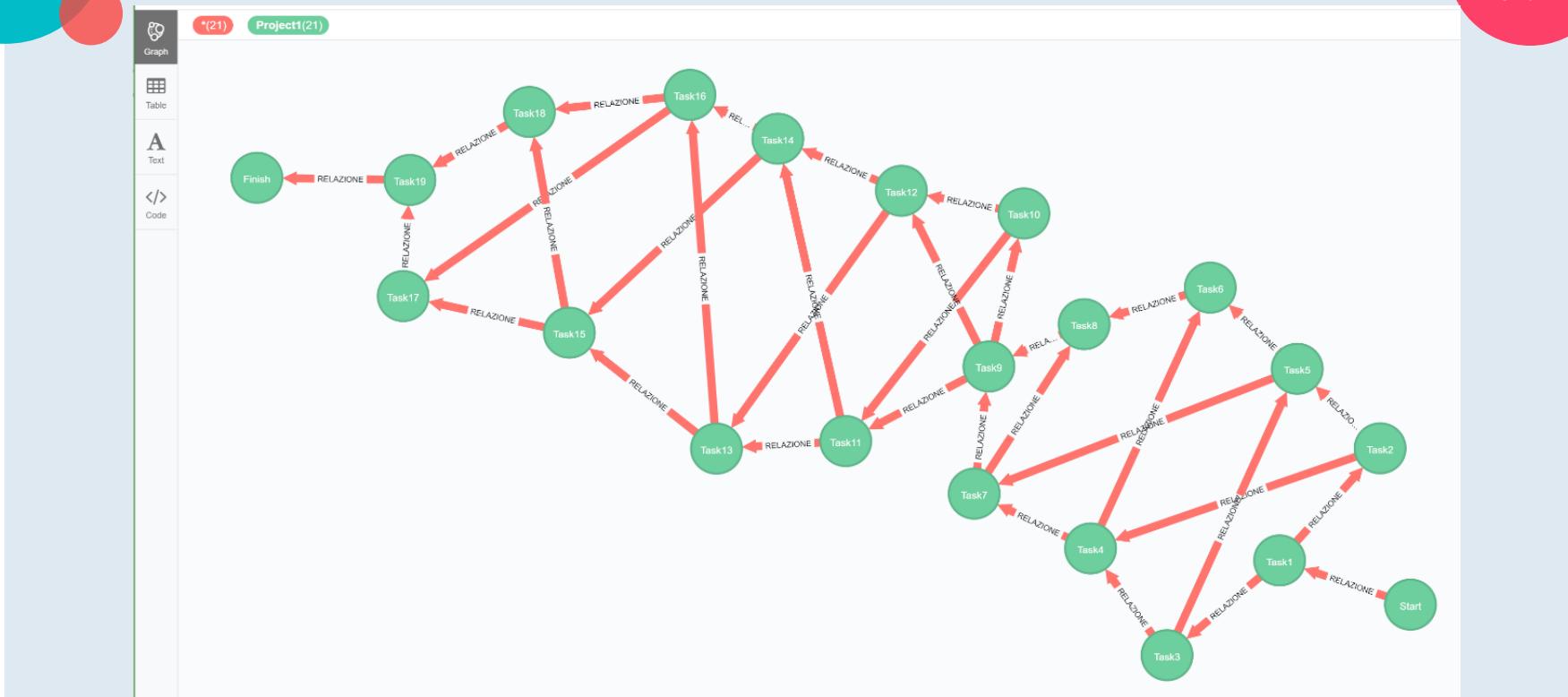
→ Per poter usare, quindi, il sistema bisogna salvare i file che riguardano i progetti come “Project*****.csv”

```
# CARICO i files .csv dei progetti contenenti "Project"
content_list = []

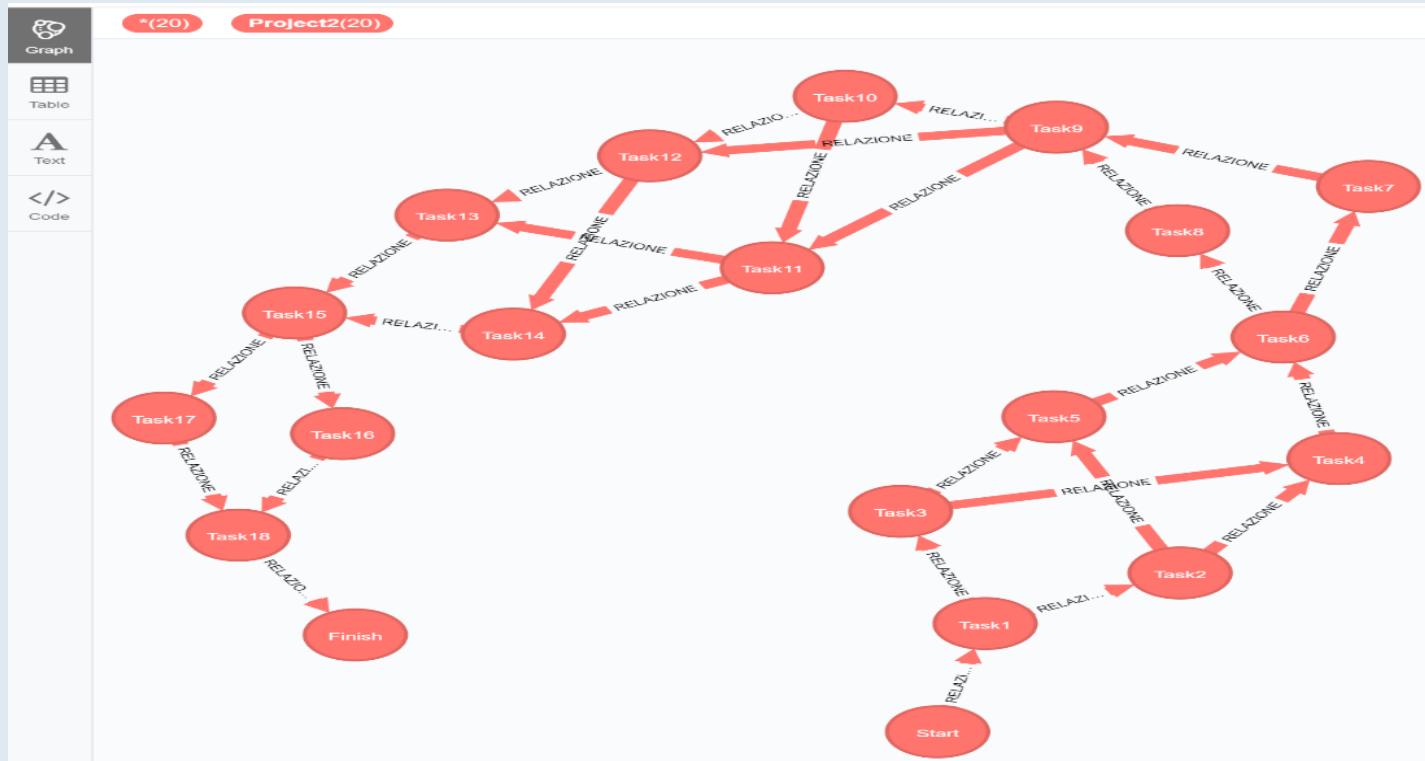
for content in os.listdir("\Users\\\Michele Palumbo\\Documents\\Neo4j\\default.graphdb\\import"):

    if content.endswith(".csv"):
        if "Project" in content:
            index_of_dot = content.index('.')
            file_name_without_extension = content[:index_of_dot]
            content_list.append(content)
            print("\n")
            print(file_name_without_extension)
            query = "LOAD CSV WITH HEADERS FROM 'file:///\" + content + "\" AS line "
                "CREATE (node:" + file_name_without_extension + "
                    { ID: line.ID, attivita: line.attivita, descrizione: line.descrizione, predecessori: SPLIT(line.predecessori, ','), "
                    "durata:toInt(line.durata), inizio: line.inizio, fine: line.fine, nick: line.nick, ruolo: line.ruolo, "
                    "start:toInt(REPLACE(line.inizio,'-','')), finish:toInt(REPLACE(line.fine,'-','')) }) "
                    "WITH node, SPLIT(line.predecessori, ',') AS predecessori "
                    "MATCH (p:\"+file_name_without_extension+)\"
                    "WHERE p.ID IN predecessori "
                    "MERGE (p)-[:RELAZIONE]->(node) "
graph.run(query).dump()
```

Il grafo del Project1



Il grafo del Project2



STEP 3

COLLEGO LE RISORSE AD OGNI PROGETTO

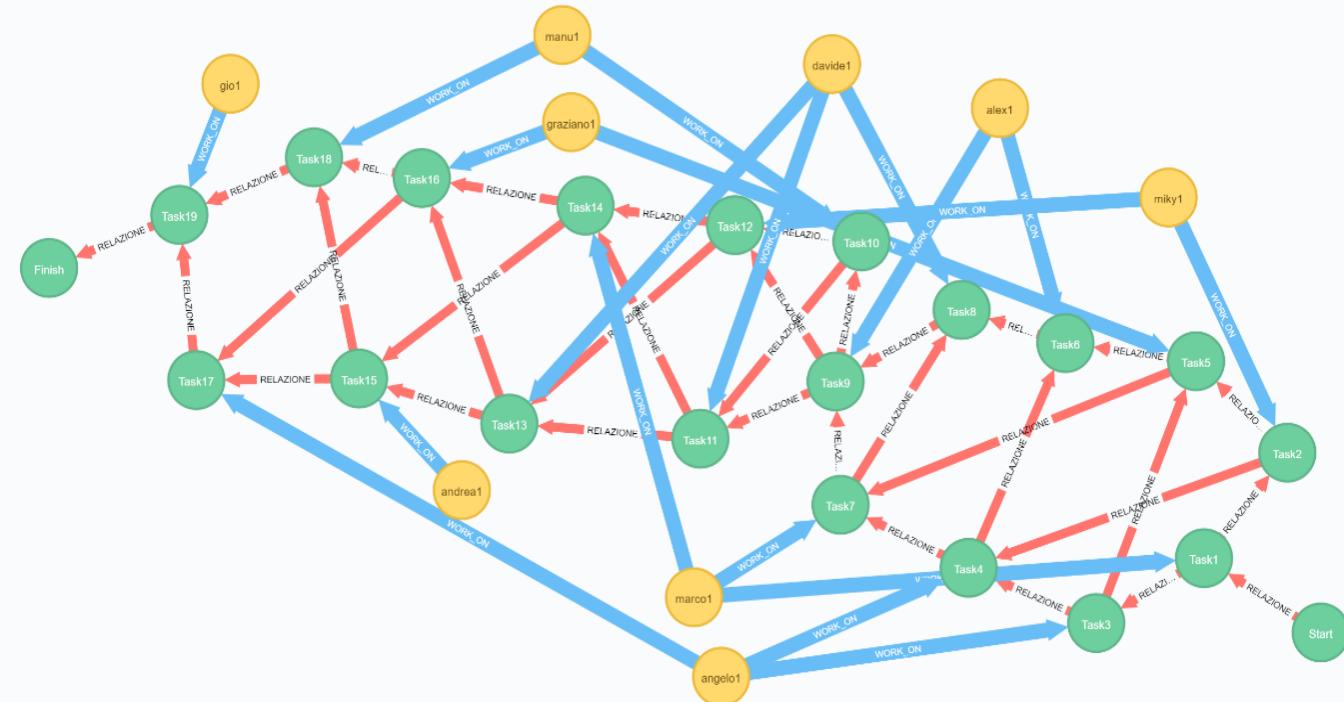
```
#COLLEGO RISORSE AD OGNI PROGETTO
collegaRisorse= "MATCH(u: Resources), (r: "+ file_name_without_extension +")" \
    "WHERE r.nick = u.nickname " \
    "CREATE (u)-[: WORK_ON]->(r)"

sgraph.run(collegaRisorse).dump()

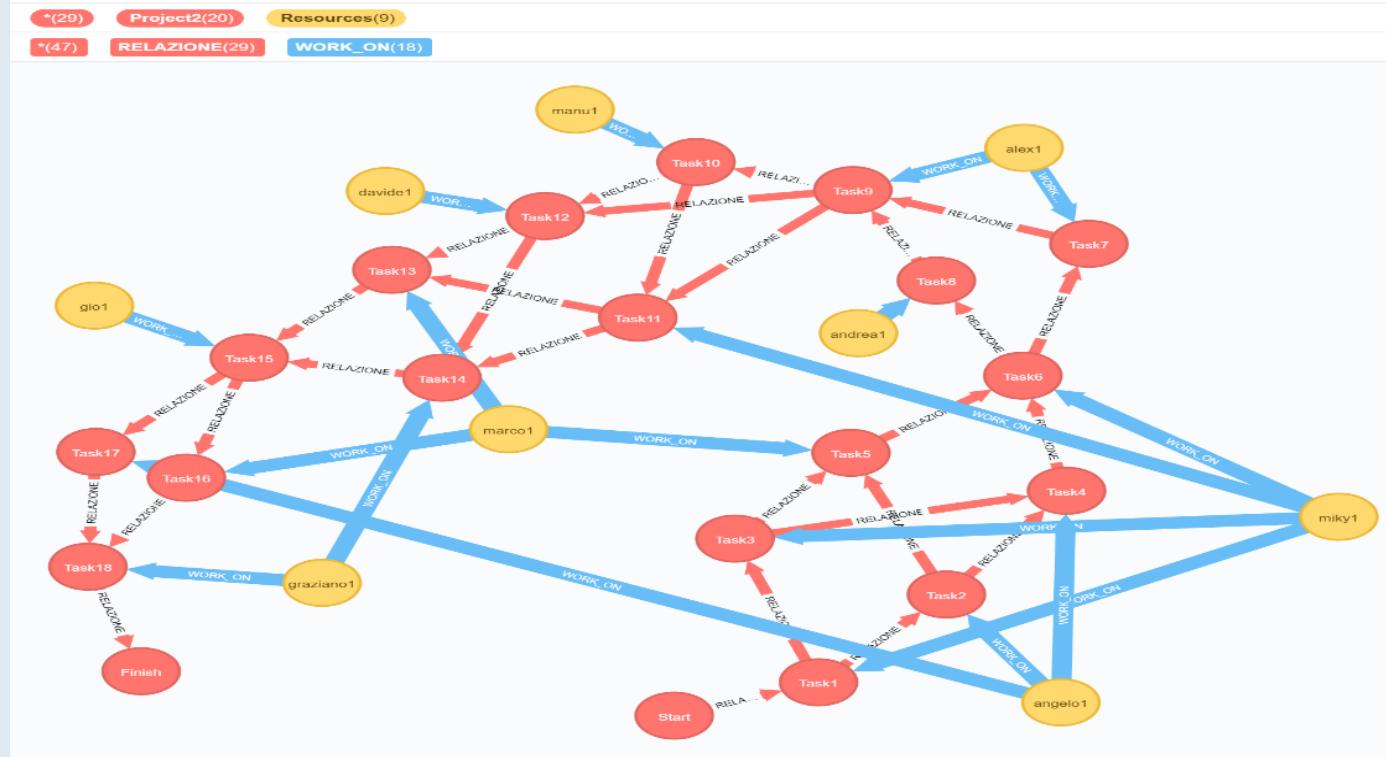
indiceRisorse = "CREATE INDEX ON :Resources(nick)"
sgraph.run(indiceRisorse).dump()
```

Il grafo delle risorse collegate al Project1

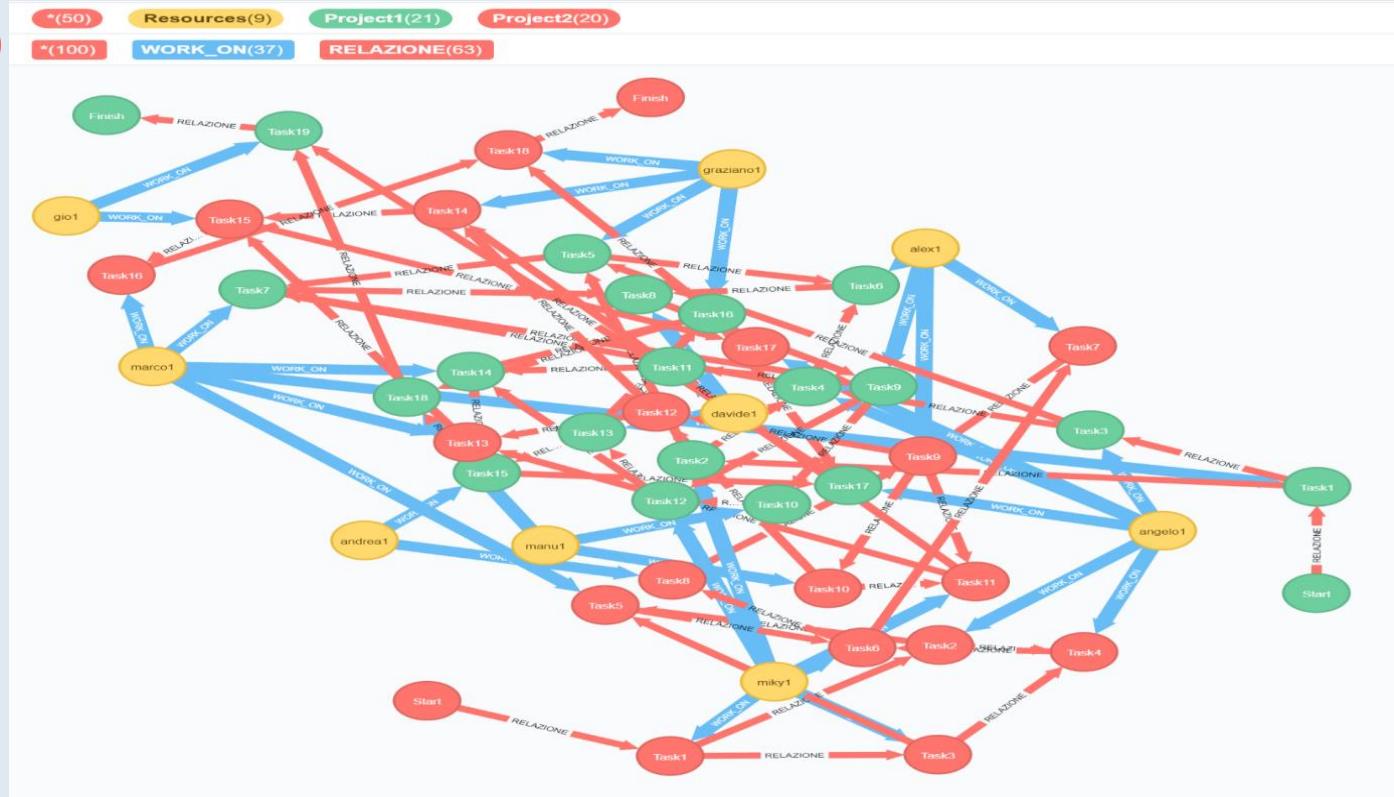
(30) Project1(21) Resources(9)
(54) RELAZIONE(35) WORK_ON(19)



Il grafo delle risorse collegate al Project2



Il grafo delle risorse collegate al Project1 e Project2





STEP 4

Calcolo ES, EF, LS, LF, slack e le aggiungo ai grafi

1/3



- Aggiungo e setto le variabili **ES, EF, LS, LF = 0** al nodo “Start”

```
# SET Variabili ES, EF, LS, LF del nodo START a 0
query = "MATCH (f:" + file_name_without_extension +" {attivita:'Start'}) ) " \
        "SET f.earliest_start = 0 " \
        "SET f.earliest_finish = 0 " \
        "SET f.latest_start = 0 " \
        "SET f.latest_finish = 0 "
```

- Setto le variabili **EF e ES**

```
# SET EARLIEST FINISH (EF) per i successivi nodi ---> EFj = ESj + dj
query2 = "MATCH p = (" + file_name_without_extension +" {attivita:'Start'})-[:RELAZIONE*]->(j:" + file_name_without_extension +") WITH j, " \
         "MAX REDUCE(x = 0, a IN NODES(p) | x + a.durata) ) " \
         "AS ef SET j.earliest_finish = ef "

# SET EARLIEST START (ES) per i successivi nodi ---> ESj = max(EFi)
query3 = "MATCH (i:" + file_name_without_extension +")-[:RELAZIONE]->(j:" + file_name_without_extension +") WITH j, " \
         "MAX(i.earliest_finish) AS max_ef " \
         "SET j.earliest_start = max_ef "
```



STEP 4

Calcolo ES, EF, LS, LF, slack e le aggiungo ai grafi

2/3



- Aggiorno le variabili **ES, EF, LS, LF** del nodo “Finish”
Aggiorno le proprietà del nodo “Finish” in base alle informazioni precedentemente calcolate “in avanti”, prima di calcolare “all’indietro” le variabili **LS** e **LF**.

```
# SET Variabili ES, EF, LS, LF del nodo Finish al valore finale di completamento
query5 = "MATCH (f:" + file_name_without_extension +" {attivita:'Finish'}) " \
    "SET f.earliest_finish = f.earliest_start " \
    "SET f.latest_start = f.earliest_start " \
    "SET f.latest_finish = f.earliest_start"
```

- Setto le variabili **LS** e **LF**

```
# SET LATEST START (LS) ---> LSj = LFj - dj
query6 = "MATCH p = (j:" + file_name_without_extension +")-[:RELAZIONE*]->(f:" + file_name_without_extension +" {attivita:'Finish'}) WITH j, " \
    "MIN(REDUCE(x = f.earliest_start, a IN NODES(p) | x - a.durata)) AS ls " \
    "SET j.latest_start = ls"

# SET LATEST FINISH (LF) ---> LFi = min(LSj)
query7 = "MATCH (i:" + file_name_without_extension +")-[:RELAZIONE]->(j:" + file_name_without_extension +") WITH i, " \
    "MIN(j.latest_start) AS min_ls " \
    "SET i.latest_finish = min_ls"
```



STEP 4

Calcolo ES, EF, LS, LF, slack e le aggiungo ai grafi

3/3



- Setto la variabile **slack**

```
# SET SLACK ---> LSj - ESj
query8 = "MATCH (a:" + file_name_without_extension +") SET a.slack = a.latest_start - a.earliest_start"
```

- Visualizzo **ES, EF, LS, LF and Slack Times** per ogni progetto

```
# View ES, EF, LS, LF, & Slack Times
query9 = "MATCH (a:" + file_name_without_extension +") RETURN a.ID AS ID, a.attivita AS Progetto , " \
    "a.earliest_start AS Earliest_START_Time , " \
    "a.earliest_finish AS Earliest_Finish_Time , " \
    "a.latest_start AS Latest_START_Time , " \
    "a.latest_finish AS Latest_Finish_Time , " \
    "a.slack AS Slack ORDER BY a.id"
```

Tabella Project1

Project1	Earliest START Time	Earliest Finish Time	Latest START Time	Latest Finish Time	Slack
"Start"	0	0	0	0	0
"Task1"	0	2	0	2	0
"Task2"	2	5	4	7	2
"Task3"	2	7	2	7	0
"Task4"	7	9	9	11	2
"Task5"	7	11	7	11	0
"Task6"	11	13	12	14	1
"Task7"	11	14	11	14	0
"Task8"	14	16	14	16	0
"Task9"	16	19	16	19	0
"Task10"	19	25	19	25	0
"Task11"	25	28	27	30	2
"Task12"	25	30	25	30	0
"Task13"	30	40	30	40	0
"Task14"	30	35	35	40	5
"Task15"	40	45	45	50	5
"Task16"	40	50	40	50	0
"Task17"	50	57	50	57	0
"Task18"	50	54	53	57	3
"Task19"	57	60	57	60	0
"Finish"	60	60	60	60	0

Tabella Project2

Project2	Earliest START Time	Earliest Finish Time	Latest START Time	Latest Finish Time	Slack
"Start"	0	0	0	0	0
"Task1"	0	5	0	5	0
"Task2"	5	9	5	9	0
"Task3"	5	8	6	9	1
"Task4"	9	14	10	15	1
"Task5"	9	15	9	15	0
"Task6"	15	20	15	20	0
"Task7"	20	22	23	25	3
"Task8"	20	25	20	25	0
"Task9"	25	28	25	28	0
"Task10"	28	31	28	31	0
"Task11"	31	34	33	36	2
"Task12"	31	36	31	36	0
"Task13"	36	41	41	46	5
"Task14"	36	46	36	46	0
"Task15"	46	56	46	56	0
"Task16"	56	60	59	63	3
"Task17"	56	63	56	63	0
"Task18"	63	65	63	65	0
"Finish"	65	65	65	65	0



STEP 5

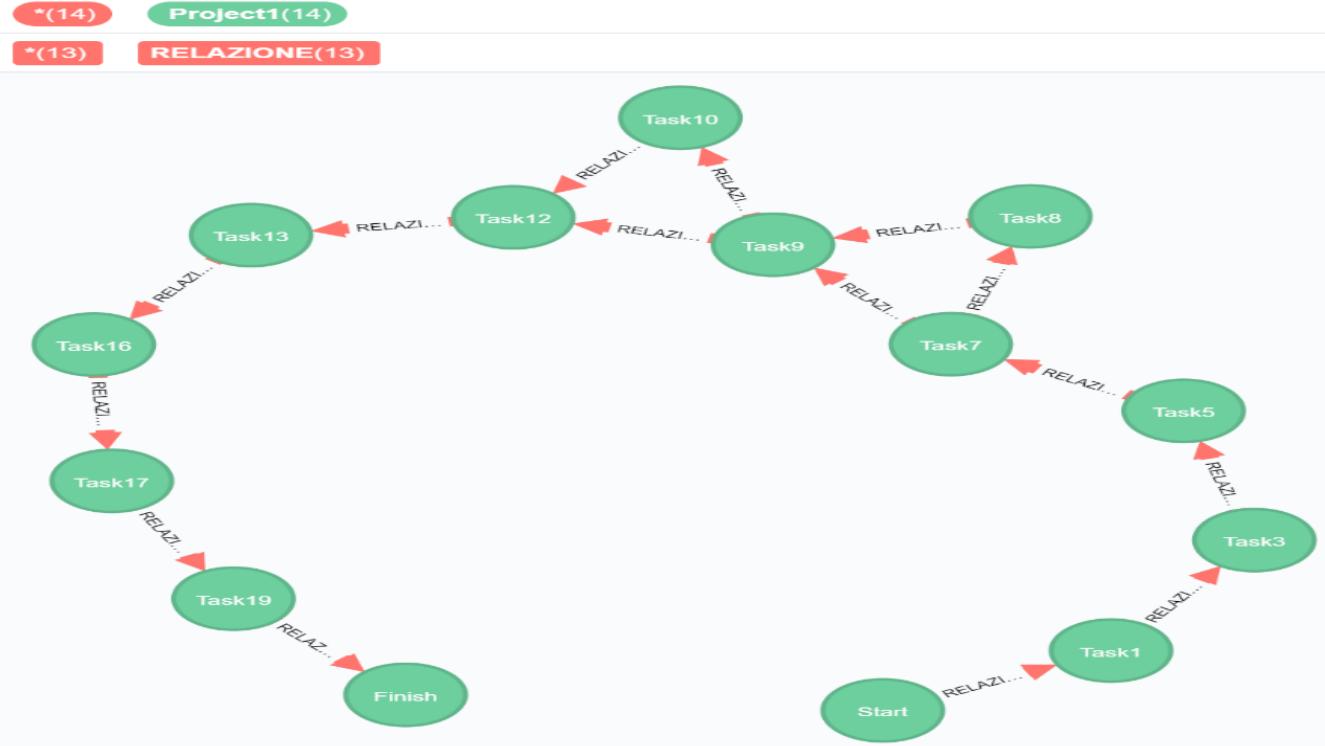
Calcolo il critical path per ogni progetto



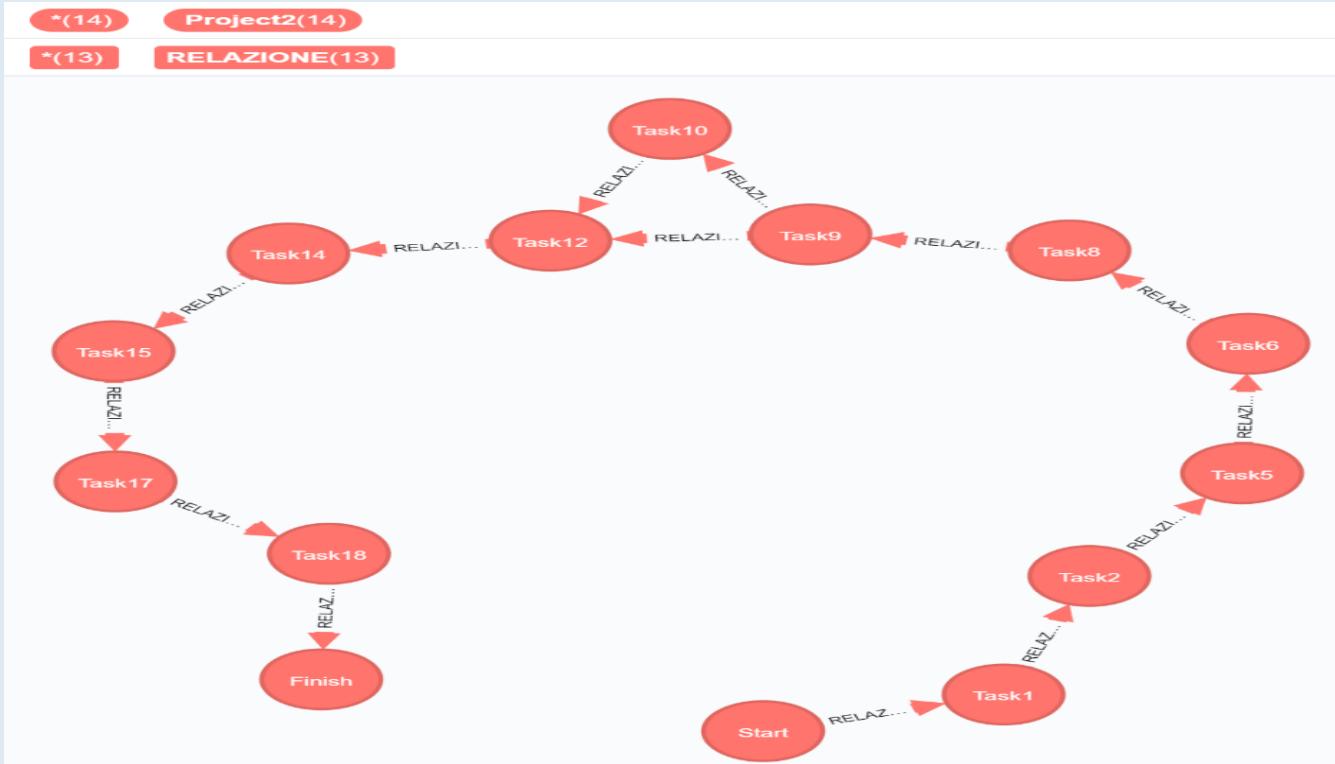
- Il **critical path** per ogni progetto è uguale al percorso più lungo (coincide con le variabili **slack=0** viste nelle tabelle precedenti)

```
# Calcolo Critical Path
query10 = "MATCH p= (: " + file_name_without_extension + \
          "{attivita: 'Start'})-[: RELAZIONE *]->(: " + file_name_without_extension + "{attivita: 'Finish'}) " \
          "WITH p , REDUCE(x=0, a IN NODES(p) | x + a.durata) AS cum_duration " \
          "ORDER BY cum_duration DESC LIMIT 1 " \
          "RETURN [node in nodes(p) | node {.ID, .nick, .attivita}] as critPath "
```

Critical Path Project1



Critical Path Project2





STEP 6

Confronto se una risorsa lavora consecutivamente sullo stesso critical Path

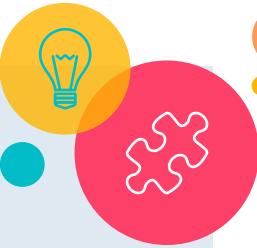
- Ho effettuato una prima un'analisi sui critical path singoli per verificare prima eventuali anomalie suicritical path singoli.
- Ho confrontato ogni “**task precedente**” con il “**task successivo**” prendendo come parametri ID, attività, ruolo e nick.
Ovviamente, i parametri “ID” e “attività” mi sono serviti per comodità, per visualizzare meglio i dati. Il parametro “ruolo” mi è servito per poter capire i ruoli della risorsa associati in quel momento ai task incriminati successivi.
- Il confronto, vero e proprio, è avvenuto con il parametro “**nick**”, in cui come prima condizione ho controllato che il nick non fosse vuoto (in modo da non prendere i nodi fittizi “Start” e “Finish”)

```
precNick = "" #nick del task precedente consecutivo
precRuolo = "" #ruolo del task precedente consecutivo
precAttivita="" #attivita del task precedente consecutivo
id = 0 #ID DEL TASK PRECEDENTE CONSECUTIVO

# CONFRONTA UNA RISORSA SE LAVORA CONSEGUENTIVAMENTE SULLO STESSO CRITICAL PATH
for l in stampa:
    if l['nick'] != '':
        if precNick == l['nick']:

            print("ATTENZIONE!!! La risorsa (" + l['nick'] + ") LAVORA PIU VOLTE SULLO STESSO CRITICAL PATH CONSEGUENTIVAMENTE SU : [ID: " + str(id)+")
            print("SOSTITUIRE la risorsa (" + l['nick']+")")
            print("Se volete sostituirla nel (" + precAttivita +") dovete farlo con una risorsa con ruolo [" + precRuolo+"]")
            print("Se volete sostituirla nel (" + l['attivita'] +") dovete farlo con una risorsa con ruolo [" + l['ruolo'] + "]")
            print("\n")
            occorrenzeCriticalPath.append(l['nick'])

    id = l['ID']
    precRuolo= l['ruolo']
    precAttivita = l['attivita']
else:
    precNick = l['nick']
    precRuolo = l['ruolo']
    precAttivita = l['attivita']
    id = l['ID']
```



Result Set

Project1
critPath

```
[{"ID": "1", "attivita": "Start", "nick": " ", "ruolo": " "}, {"ID": "2", "attivita": "Task1", "nick": "marco1", "ruolo": "analyst"}, {"ID": "4", "attivita": "Task3", "nick": "angeloi", "ruolo": "analyst"}, {"ID": "6", "attivita": "Task5", "nick": "graziano1", "ruolo": "graphic designer"}, {"ID": "8", "attivita": "Task7", "nick": "marco1", "ruolo": "analyst"}, {"ID": "9", "attivita": "Task8", "nick": "davide1", "ruolo": "system designer"}, {"ID": "10", "attivita": "Task9", "nick": "alex1", "ruolo": "system designer"}, {"ID": "11", "attivita": "Task10", "nick": "manu1", "ruolo": "object designer"}, {"ID": "13", "attivita": "Task12", "nick": "miky1", "ruolo": "object designer"}, {"ID": "14", "attivita": "Task13", "nick": "davide1", "ruolo": "developer"}, {"ID": "17", "attivita": "Task16", "nick": "graziano1", "ruolo": "developer"}, {"ID": "18", "attivita": "Task17", "nick": "angelo1", "ruolo": "tester"}, {"ID": "20", "attivita": "Task19", "nick": "gio1", "ruolo": "graphic designer"}, {"ID": "21", "attivita": "Finish", "nick": " ", "ruolo": " "}]
```

Project2
critPath

```
-----  
[{"ID": "1", "attivita": "Start", "nick": " ", "ruolo": " "}, {"ID": "2", "attivita": "Task1", "nick": "miky1", "ruolo": "analyst"}, {"ID": "3", "attivita": "Task2", "nick": "angelo1", "ruolo": "analyst"}, {"ID": "6", "attivita": "Task5", "nick": "marco1", "ruolo": "analyst"}, {"ID": "7", "attivita": "Task6", "nick": "miky1", "ruolo": "analyst"}, {"ID": "9", "attivita": "Task8", "nick": "andreal", "ruolo": "system designer"}, {"ID": "10", "attivita": "Task9", "nick": "alex1", "ruolo": "system designer"}, {"ID": "11", "attivita": "Task10", "nick": "manu1", "ruolo": "object designer"}, {"ID": "13", "attivita": "Task12", "nick": "davide1", "ruolo": "developer"}, {"ID": "15", "attivita": "Task14", "nick": "graziano1", "ruolo": "developer"}, {"ID": "16", "attivita": "Task15", "nick": "gio1", "ruolo": "graphic designer"}, {"ID": "18", "attivita": "Task17", "nick": "angelo1", "ruolo": "tester"}, {"ID": "19", "attivita": "Task18", "nick": "graziano1", "ruolo": "graphic designer"}, {"ID": "20", "attivita": "Finish", "nick": " ", "ruolo": " "}]
```

- Non risultano risorse allocate successivamente sullo stesso critical path.



STEP 7

Confronto se una risorsa è allocata su più critical path nello stesso momento

- Mi prendo i progetti a due a due e li confronto tra di loro, per tutti gli n-progetti presenti.
- Poi, sempre utilizzando il parametro “nick” vado ad analizzare se tale risorsa può essere allocata contemporaneamente su più critical path, almeno un giorno.

```
if ((nick == nick1) and (nick != " ") and (nick1 != " ")):  
    if ((iniziol >= inizio) and (finel <= inizio)):  
        print(name_project + "->" + name_project_2)  
        print("Date sovrapposte inizio: " + nick + "\t" + str(iniziol) + "\t" + str(finel) + "\t" + str(inizio))  
  
    elif ((iniziol >= fine) and (finel <= fine)):  
        print(name_project + "->" + name_project_2)  
        print("Date sovrapposte fine: " + nick + str(iniziol) + "\t" + str(finel) + "\t" + str(fine))  
        fine = datetime.datetime.strptime(str(fine), '%Y%m%d').date()  
        iniziol = datetime.datetime.strptime(str(iniziol), '%Y%m%d').date()  
        fine = fine + datetime.timedelta(days=1)
```

```
if (name_project != name_project_2):  
    for e in project:  
        id = e['ID']  
        attivita = e['attivita']  
        nick = e['nick']  
        ruolo = e['ruolo']  
        inizio = e['start']  
        fine = e['finish']  
  
    for k in project_2:  
        id1 = k['ID']  
        attivital = k['attivita']  
        nick1 = k['nick']  
        ruolo1 = k['ruolo']  
        iniziol = k['start']  
        finel = k['finish']
```

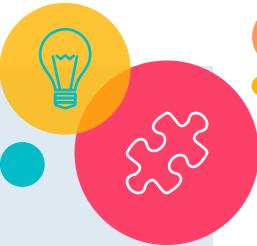


STEP 8

Confronto se una risorsa su più critical path è allocata consecutivamente a più tasks successivi di differenti progetti.

- Continuando, vado a fare un altro confronto interno per stabilire se una risorsa risulta allocata su task successivi di differenti critical path.
- Quindi, confronto la data di fine task del **Project_{n-1}** con la data di inizio del **Project_n**, sempre utilizzando il parametro “nick”. E segnalo l'anomalia con i suggerimenti.

```
elif (fine == iniziol):
    print("\n")
    print(name_project + "->" + name_project_2)
    print("!!!!ALERT!!! >>> Risorse su progetti diversi ma con task consecutivi ")
    print(
        name_project + ": (" + "ID: " + id + " * " + "Risorsa: " + nick + " * " + "Ruolo: " + ruolo + " * " + "Fine attività: " + str(
            fine) + " ) " + "-->" + name_project_2 + ": (" + "ID: " + idl + " * " + "Risorsa: " + nick1 + " * " + "Ruolo: " + ruolol + " * " + "Inizio attività: " + str(
            iniziol) + " )")
    print("SOSTITUIRE RISORSA--> " + nick)
    print(
        "Mettersi in contatto con i Project Manager del (" + name_project + ") e (" + name_project_2 + ")")
    print(
        "SUGGERIMENTO 1 : Se volete sostituire la risorsa: (" + nick + ") nel " + name_project + " dovete farlo con una risorsa che ha il ruolo (" + ruolo + ")")
    print(
        "SUGGERIMENTO 2 : Se volete sostituire la risorsa: (" + nick1 + ") nel " + name_project_2 + " dovete farlo con una risorsa che ha il ruolo (" + ruolol + ")")
```



Result Set

Project1
critPath

```
[{'ID': '1', 'attivita': 'Start', 'nick': ' ', 'ruolo': ' '}, {'ID': '2', 'attivita': 'Task1', 'nick': 'marco1', 'ruolo': 'analyst'}, {"ID": "4", "attivita": "Task3", "nick": "angeloi", "ruolo": "analyst"}, {"ID": "6", "attivita": "Task5", "nick": "graziano1", "ruolo": "graphic designer"}, {"ID": "8", "attivita": "Task7", "nick": "marco1", "ruolo": "analyst"}, {"ID": "9", "attivita": "Task8", "nick": "davide1", "ruolo": "system designer"}, {"ID": "10", "attivita": "Task9", "nick": "alex1", "ruolo": "system designer"}, {"ID": "11", "attivita": "Task10", "nick": "manu1", "ruolo": "object designer"}, {"ID": "13", "attivita": "Task12", "nick": "miky1", "ruolo": "object designer"}, {"ID": "14", "attivita": "Task13", "nick": "davide1", "ruolo": "developer"}, {"ID": "17", "attivita": "Task16", "nick": "graziano1", "ruolo": "developer"}, {"ID": "18", "attivita": "Task17", "nick": "angelo1", "ruolo": "tester"}, {"ID": "20", "attivita": "Task19", "nick": "gio1", "ruolo": "graphic designer"}, {"ID": "21", "attivita": "Finish", "nick": " ", "ruolo": " '}]
```

Project2
critPath

```
-----  
[{'ID': '1', 'attivita': 'Start', 'nick': ' ', 'ruolo': ' '}, {"ID": "2", "attivita": "Task1", "nick": "miky1", "ruolo": "analyst"}, {"ID": "3", "attivita": "Task2", "nick": "angelo1", "ruolo": "analyst"}, {"ID": "6", "attivita": "Task5", "nick": "marco1", "ruolo": "analyst"}, {"ID": "7", "attivita": "Task6", "nick": "miky1", "ruolo": "analyst"}, {"ID": "9", "attivita": "Task8", "nick": "andreal", "ruolo": "system designer"}, {"ID": "10", "attivita": "Task9", "nick": "alex1", "ruolo": "system designer"}, {"ID": "11", "attivita": "Task10", "nick": "manu1", "ruolo": "object designer"}, {"ID": "13", "attivita": "Task12", "nick": "davide1", "ruolo": "developer"}, {"ID": "15", "attivita": "Task14", "nick": "graziano1", "ruolo": "developer"}, {"ID": "16", "attivita": "Task15", "nick": "gio1", "ruolo": "graphic designer"}, {"ID": "18", "attivita": "Task17", "nick": "angelo1", "ruolo": "tester"}, {"ID": "19", "attivita": "Task18", "nick": "graziano1", "ruolo": "graphic designer"}, {"ID": "20", "attivita": "Finish", "nick": " ", "ruolo": " '}]
```

- Ma anche in questi casi, non risultano risorse allocate su più critical path nello stesso momento.
- Non risultano nemmeno risorse allocate consecutivamente a più task successivi di critical path di differenti progetti.



STEP AGGIUNTIVO

Inserimento del “Project3” e “Project4”

1/2

- Come possiamo notare nel confronto in cui andiamo a verificare se una risorsa è allocata successivamente a due task sul critical path del singolo progetto, vi è la risorsa **davide1** allocata successivamente sul **task7** e **task8**

Project3
critPath

```
[{"ID": "1", "attivita": "Start", "nick": " ", "ruolo": " "}, {"ID": "2", "attivita": "Task1", "nick": "marco1", "ruolo": "analyst"}, {"ID": "4", "attivita": "Task3", "nick": "angelo1", "ruolo": "analyst"}, {"ID": "5", "attivita": "Task4", "nick": "davide1", "ruolo": "analyst"}, {"ID": "3", "attivita": "Task2", "nick": "miky1", "ruolo": "analyst"}, {"ID": "6", "attivita": "Task5", "nick": "graziano1", "ruolo": "graphic designer"}, {"ID": "8", "attivita": "Task7", "nick": "davide1", "ruolo": "analyst"}, {"ID": "9", "attivita": "Task8", "nick": "davide1", "ruolo": "graphic designer"}, {"ID": "10", "attivita": "Task9", "nick": "alexi1", "ruolo": "system designer"}, {"ID": "11", "attivita": "Task10", "nick": "manui", "ruolo": "object designer"}, {"ID": "13", "attivita": "Task12", "nick": "miky1", "ruolo": "object designer"}, {"ID": "14", "attivita": "Task13", "nick": "davide1", "ruolo": "developer"}, {"ID": "17", "attivita": "Task16", "nick": "andreal1", "ruolo": "developer"}, {"ID": "18", "attivita": "Task17", "nick": "davide1", "ruolo": "tester"}, {"ID": "20", "attivita": "Task19", "nick": "gio1", "ruolo": "graphic designer"}, {"ID": "21", "attivita": "Finish", "nick": " ", "ruolo": " "}]
```

ATTENZIONE!!! La risorsa (davide1) LAVORA PIU VOLTE SULLO STESSO CRITICAL PATH CONSECUTIVAMENTE SU :
[ID: 8 - Task7 - RUOLO: (analyst)] E [ID: 9 - Task8 - RUOLO: (graphic designer)]

SOSTITUIRE la risorsa (davide1)

Se volete sostituirla nel (Task7) dovete farlo con una risorsa con ruolo [analyst]
Se volete sostituirla nel (Task8) dovete farlo con una risorsa con ruolo [graphic designer]

Project4
critPath

```
[{"ID": "1", "attivita": "Start", "nick": " ", "ruolo": " "}, {"ID": "2", "attivita": "Task1", "nick": "marco1", "ruolo": "analyst"}, {"ID": "4", "attivita": "Task3", "nick": "angelo1", "ruolo": "analyst"}, {"ID": "6", "attivita": "Task5", "nick": "graziano1", "ruolo": "graphic designer"}, {"ID": "8", "attivita": "Task7", "nick": "marco1", "ruolo": "analyst"}, {"ID": "9", "attivita": "Task8", "nick": "davide1", "ruolo": "system designer"}, {"ID": "10", "attivita": "Task9", "nick": "alexi1", "ruolo": "system designer"}, {"ID": "11", "attivita": "Task10", "nick": "manui", "ruolo": "object designer"}, {"ID": "13", "attivita": "Task12", "nick": "miky1", "ruolo": "object designer"}, {"ID": "14", "attivita": "Task13", "nick": "davide1", "ruolo": "developer"}, {"ID": "17", "attivita": "Task16", "nick": "graziano1", "ruolo": "developer"}, {"ID": "18", "attivita": "Task17", "nick": "angelo1", "ruolo": "tester"}, {"ID": "20", "attivita": "Task19", "nick": "gio1", "ruolo": "graphic designer"}, {"ID": "21", "attivita": "Finish", "nick": " ", "ruolo": " "}]
```

Result Set 1



STEP AGGIUNTIVO

Inserimento del “Project3” e “Project4”

2/2

- Mentre, invece, nei result set degli altri confronti risultano non esserci risorse allocate nello stesso momento su più critical path.
- Ma ci sono risorse allocate consecutivamente a più task successivi di critical path di differenti progetti.

```
Project1->Project3
!!!ALERT!! >>> Risorse su progetti diversi ma con task consecutivi
Project1: (ID: 2 * Risorsa: marcol * Ruolo: analyst * Fine attività: 20170905 )---> Project3: (ID: 2 * Risorsa: marcol * Ruolo: analyst * Inizio attività: 20170905 )
SOSTituIRE RISORSA--> marcol
Mettersi in contatto con i Project Manager del (Project1) e (Project3)
SUGGERIMENTO 1 : Se volete sostituire la risorsa: (marcol) nel Project1 dovete farlo con una risorsa che ha il ruolo (analyst)
SUGGERIMENTO 2 : Se volete sostituire la risorsa: (marcol) nel Project3 dovete farlo con una risorsa che ha il ruolo (analyst)
```

```
Project3->Project1
!!!ALERT!! >>> Risorse su progetti diversi ma con task consecutivi
Project3: (ID: 8 * Risorsa: davidel * Ruolo: analyst * Fine attività: 20170922 )---> Project1: (ID: 9 * Risorsa: davidel * Ruolo: system designer * Inizio attività: 20170922 )
SOSTITUIRE RISORSA--> davidel
Mettersi in contatto con i Project Manager del (Project3) e (Project1)
SUGGERIMENTO 1 : Se volete sostituire la risorsa: (davidel) nel Project3 dovete farlo con una risorsa che ha il ruolo (analyst)
SUGGERIMENTO 2 : Se volete sostituire la risorsa: (davidel) nel Project1 dovete farlo con una risorsa che ha il ruolo (system designer)
```

```
Project3->Project4
!!!ALERT!! >>> Risorse su progetti diversi ma con task consecutivi
Project3: (ID: 8 * Risorsa: davidel * Ruolo: analyst * Fine attività: 20170922 )---> Project4: (ID: 9 * Risorsa: davidel * Ruolo: system designer * Inizio attività: 20170922 )
SOSTITUIRE RISORSA--> davidel
Mettersi in contatto con i Project Manager del (Project3) e (Project4)
SUGGERIMENTO 1 : Se volete sostituire la risorsa: (davidel) nel Project3 dovete farlo con una risorsa che ha il ruolo (analyst)
SUGGERIMENTO 2 : Se volete sostituire la risorsa: (davidel) nel Project4 dovete farlo con una risorsa che ha il ruolo (system designer)
```

```
Project4->Project3
!!!ALERT!! >>> Risorse su progetti diversi ma con task consecutivi
Project4: (ID: 2 * Risorsa: marcol * Ruolo: analyst * Fine attività: 20170905 )---> Project3: (ID: 2 * Risorsa: marcol * Ruolo: analyst * Inizio attività: 20170905 )
SOSTITUIRE RISORSA--> marcol
Mettersi in contatto con i Project Manager del (Project4) e (Project3)
SUGGERIMENTO 1 : Se volete sostituire la risorsa: (marcol) nel Project4 dovete farlo con una risorsa che ha il ruolo (analyst)
SUGGERIMENTO 2 : Se volete sostituire la risorsa: (marcol) nel Project3 dovete farlo con una risorsa che ha il ruolo (analyst)
```

Result Set 2



THE END
THANK YOU