MultiProject Analysis with CPM

Course of IT Project Management

BUSINESS INNOVATION AND INFORMATICS

3 ottobre 2017

Author: Michele Palumbo

MultiProject Analysis with CPM

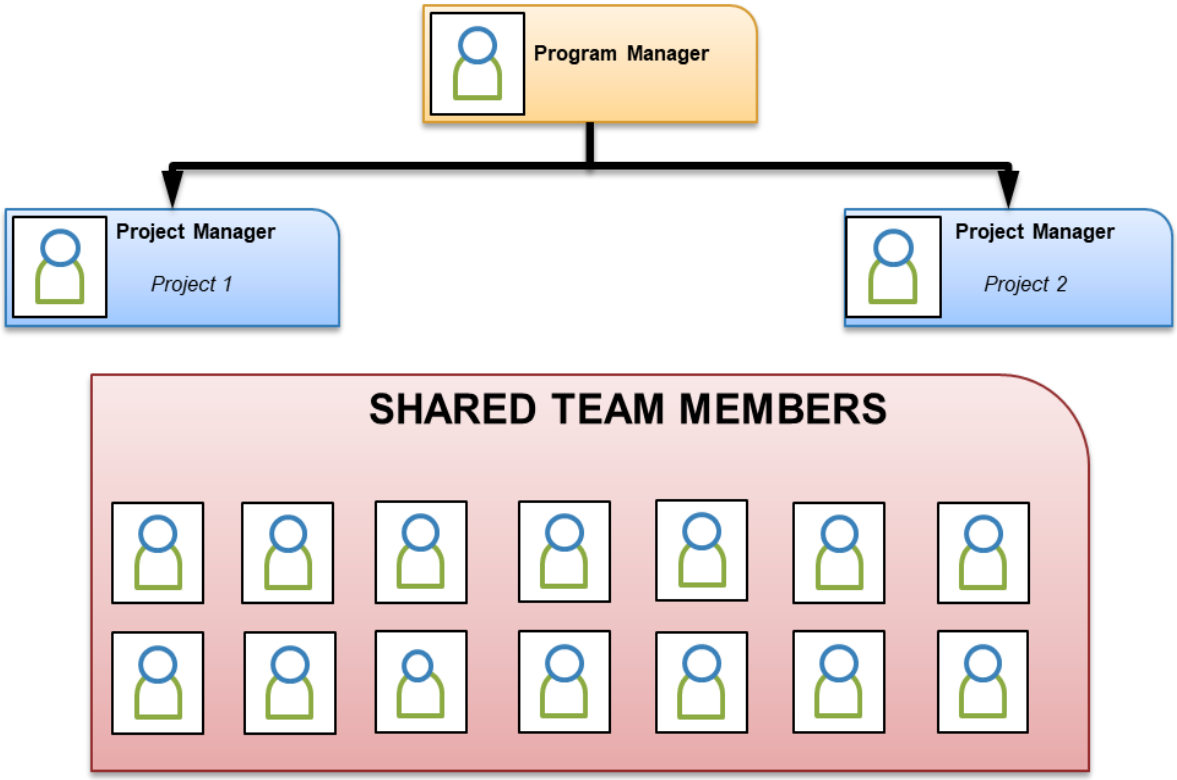
Course of IT Project Management

INDEX

1. CONTEXT

This project has been developed to provide decision support to all **Program managers** who manage multiple projects with shared resources that are, of course, planned by the various project managers assigned. Therefore, there is a vertical communication between the **Program Manager** and the various reference project managers in which the latter give precisely the planning of their project to the program manager.

The stage I decided to focus on is ***post planning***.



2. PROBLEM

One of the most difficult problems to deal with is to manage human resources linked to multiple projects, and then shared resources.
Then, you can analyse whether a given resource may be abnormally allocated across multiple projects, or if you are straddling multiple immediately subsequent critical tasks related to both the single project and the N-projects on which it is allocated.

3. SOLUTION

To try to solve these problems, I decided to develop a software by following the approach of data analysis through the **Critical Path Method (CPM)**

3.1. Critical Path Method (CPM)

It is a technique of analysis of the scheduling lattice that foresees the calculation of the minimum and maximum start and end dates for each scheduled activity. These calculations are purely theoretical because they do not take into account any resource limits. Only at the end of the process will the resources be taken into account, and the network will, if necessary, be rescheduled.
The **Critical Path Method** is a subsequent simplified application of the PERT.
In the CPM the determination of the durations is deterministic (more precise), then disappears the probability function that there is in the PERT (that happens with a 3-factor estimate: **A** = optimistic duration, **M** = Most likely duration, **B** = pessimistic duration, and the average is calculated with this formula → $\frac{A+4M+B}{6}$).
The PERT is indicated in case of projects where the times can be very variable, while the CPM is more consistent in the situation of greater accuracy of dependence between resources and times.
The critical path definition is: "The sequence of scheduled tasks that determines the minimum duration of the project. It is usually the longest path of the project. "
Therefore, the critical path is the longest chain of activities (or chains, if they are more than one) that have a total float equal to zero and that determine the minimum duration of the whole project.
The Total Float (also known as Total Slack) can be defined as the amount of time (→ I referred to *working days*) that can slip the start date of an activity without affecting the end date of the entire project. We will see, later, how it is calculated.
The steps to put into practice the CPM are:

1) To build the reticle, you need to start from the WBS where all the activities foreseen by the project have been listed and their durations have been estimated.
   → I used how to estimate the deterministic one, required by CPM
2) After you identify the project tasks and estimate the durations, you must define the logical constraints, which are the sequential dependencies between the tasks.
   → I used the logical constraint FS – Finish to Start

3) They represent the activities, identified by the WBS, on a graph oriented and acyclic. The possible modes of representation are:

- **AON** (Activity On Node), where the activities are represented by the graph nodes. The first node is always the beginning of the project; The last one is always the end of the project.
- **AOA** (Activity On Arrow) where the activities are represented by the arcs that connect two node of the graph.

  → I used the AON network where I have only one initial node and one final node. For to guarantee the uniqueness of the source and the well of the network I decided to use a "Start" source node without predecessors and a "Finish" well node without successors.

4) At this point, to calculate the "slack" (which will the critical activities and activities that can slip) you need to determine for each task, 4 times, in this way:

- **Earliest start time (ES)**
  Is equal to the earliest finish time (EF) of the previous activity.
  N.B. for activities with more than one predecessor, it is the max time among all the preceding activities.

$$ES_j = \max(EF_i)$$

- **Earliest finish time (EF)**
  Is equal to the earliest start time (ES_j) + the estimated time of duration (d_j)

$$EF_j = EF_i + d_j$$

- **Latest start time (LS)**
  Is equal to the latest finish time (LF_j) of activity - the estimated time of duration (d_j)

$$LS_j = LF_j - d_j$$

- **Latest finish time (LF)**

  Is equal to the latest start time (LS_j) of the next activity.

  N.B. for activities with more than one successor, is the minimum of latest start times.

$$LF_i = \max(LS_j)$$

5) In summary, it is possible to say that the method consists of a repetitive calculation performed before "forward", to obtain the minimum start and end dates (ES and EF), and then "backward" in such a way as to obtain the maximum dates (LS and LF). Depending on the flexibility of the activities, the critical path will be identified. So, at this point, we can say that the total time required to complete the project is the first moment of start of the node "**Finish**"

$$ProjectCompletionTime = ES_{Finish}$$

This is also the longest path of the graph (from start to finish) in terms of cumulative duration, where this cumulative duration is the completion time of the project. This longer path is the **critical path**.

The **critical path**, as we have already said, consists of those activities in which an increase of any of their durations would increase the overall time of completion of the project. These activities have a time $slack = 0$, so there will be no way to increase their durations without increasing the overall length of the project. An activity **j**, instead, it is not on **critical path** if it has a *positive slack*, which is the difference between LS_j and ES_j :

$$slack_j = LS_j - ES_j$$

The activities with *positive slack* can be increased, obviously always on the basis of the slack, without compromising the overall time of completion of the project.

To conclude, we can also say that:

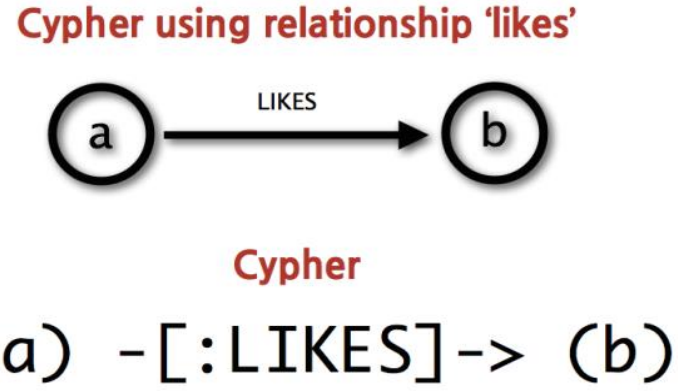
- $ES_{Start} = EF_{Start} = LS_{Start} = LF_{Start} = 0$
- $ES_{Finish} = EF_{Finish} = LS_{Finish} = LF_{Finish} = ProjCompletTime$

3.2. Tools used for the program

The tools used to develop the software are: ***Neo4j*** and ***PyCharm***,
languages: ***Cypher*** and ***Python***,
libraries: *pandas* and *py2neo*

- **Neo4j**

Neo4J is one of the most widely used graph databases in the Big Data field.
The foundational structure for the graph databases is the "node-relationship"; Storing and navigating is done using nodes, their relationships and their properties.
The graphs represent one of the most efficient and natural ways of working with data: they are extremely intuitive and show the interconnection of concepts and ideas.
NEO4J helps in the management of large data docks, for example with issues related to big data and IoT.
Neo4J supports a declarative language called **Cypher**, specifically designed to interrogate the graphs and their components.
The Cypher commands are vaguely based on SQL syntax and are designed for ad hoc graph data queries.



- **PyCharm**

PyCharm is a IDE for Python.
"The intelligent **Python** IDE with unique code assistance and analysis, for productive Python development on all levels"

- **Library: pandas**

Pandas is a software library written for the Python programming language for manipulating and analyzing data.
→ I used this library to read the .csv files

- **Library: py2neo**

Py2neo is a client library and comprehensive toolkit for working with Neo4j from within Python applications and from the command line. The core library has no external dependencies and has been carefully designed to be easy and intuitive to use.

3.3. The steps of the solution

The solution that will be proposed below is based on planning two projects with shared resources. Obviously, the software is generic so you will be able to manage N-projects with as many N-resources.

I. Initial Situation

Initially, I identified the activities of the two projects through WBS. I have, therefore, planned the human resources distributed on the two projects in order to make them compatible with each other. Next, I brought the three tables (two for projects and one for the resources) in format (. csv) to be able to read them easily from pandas and upload them to the database Neo4j.

→ The constraint to make the schedule compatible with the system is to insert a "start" node and a "Finish" node that correspond to the beginning of the project and at the end of the project. And, of course, they must have the same structure.

Here are the project and resource datasets:

Project1.csv

```
df_prog1 = pandas.read_csv('\\Users\\Michele Palumbo\\Documents\\Neo4j\\default.graphdb\\import\\Project1.CSV')
df_prog1
```

	ID	attivita	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	miky1
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	miky1
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				

Project2.csv

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

	ID	attivita	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				

Resources.csv

```
df_res = pandas.read_csv('\\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

II. Load the .csv files on Neo4j and create the graphs

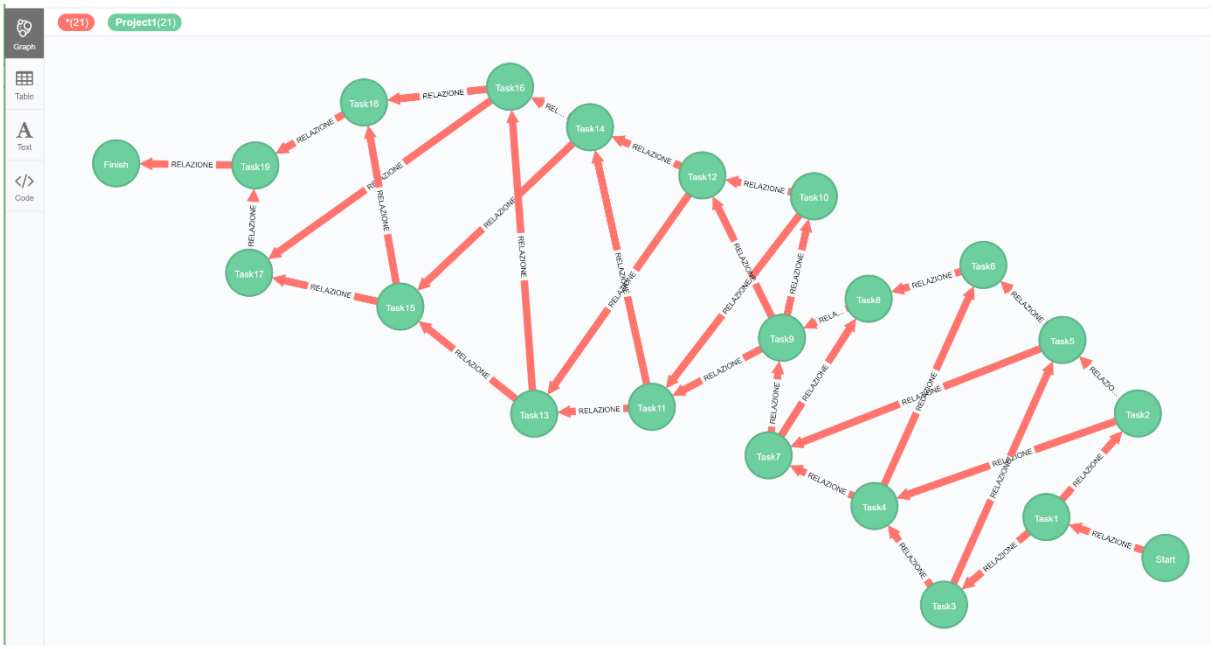
- Load the resources first, so that they can be connected later to the projects.

```
# 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, ';')} )"

sgraph.run(creaRisorse).dump()
```

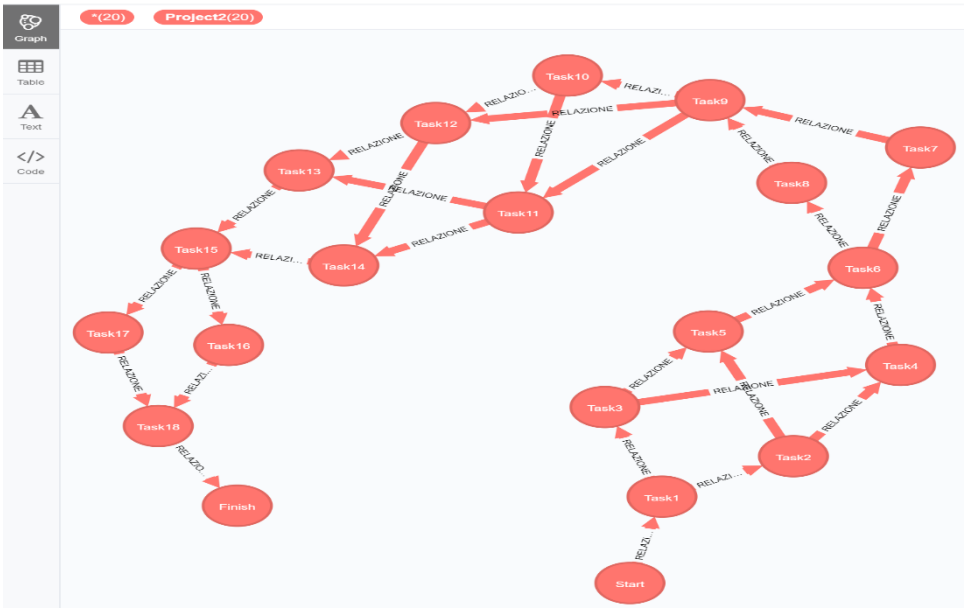
- Load and create a graph for each project. In order to make the system generic and, therefore, a single graph that can go well for each project I made sure to take me from the folder containing the files. csv all those files that begin with "Project".
  → To use the system, you must save the files that relate to projects such as "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 predecessors " \
                                                            "MATCH (p:"+file_name_without_extension+") "\
                                                            "WHERE p.ID IN predecessors " \
                                                            "MERGE (p)-[:RELAZIONE]->(node) "
            sgraph.run(query).dump()
```

The graph of the “Project1”



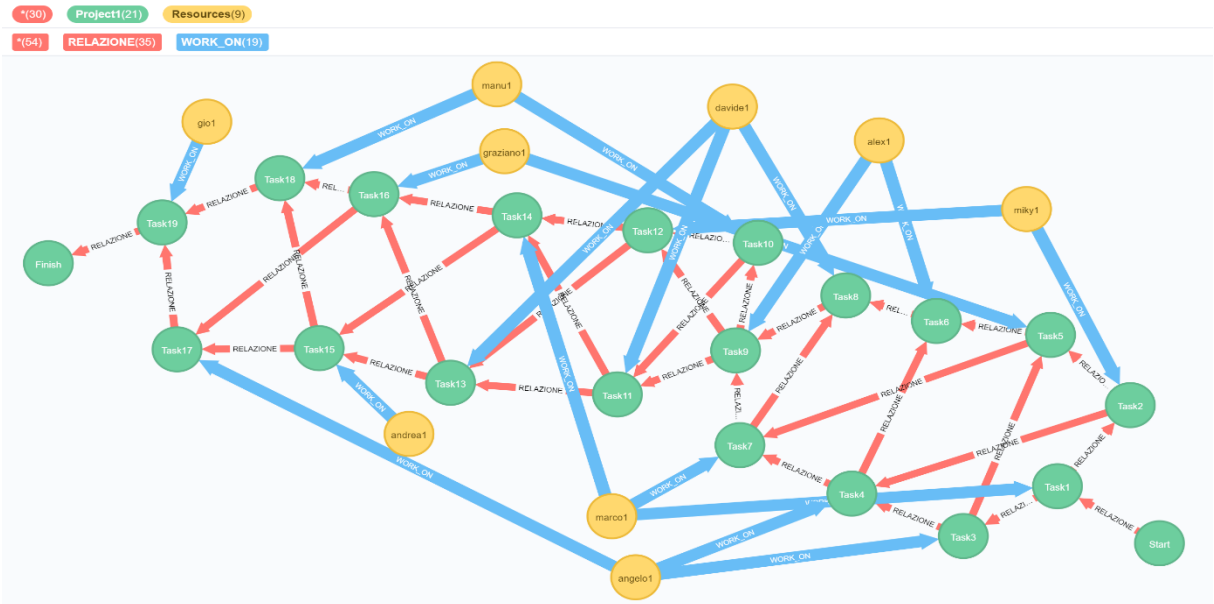
The graph of the “Project2”



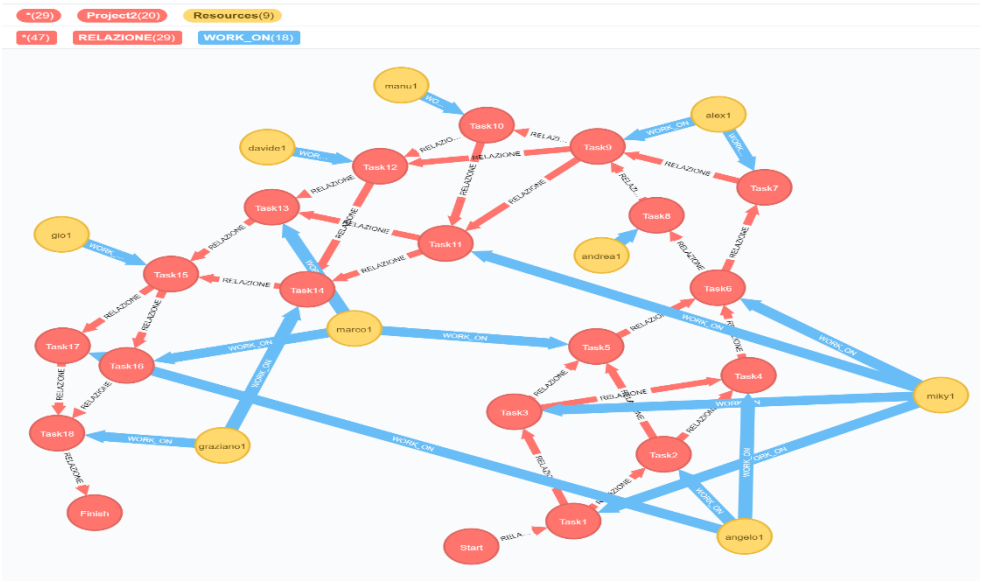
III. I connect resources to each project

```
#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()
```

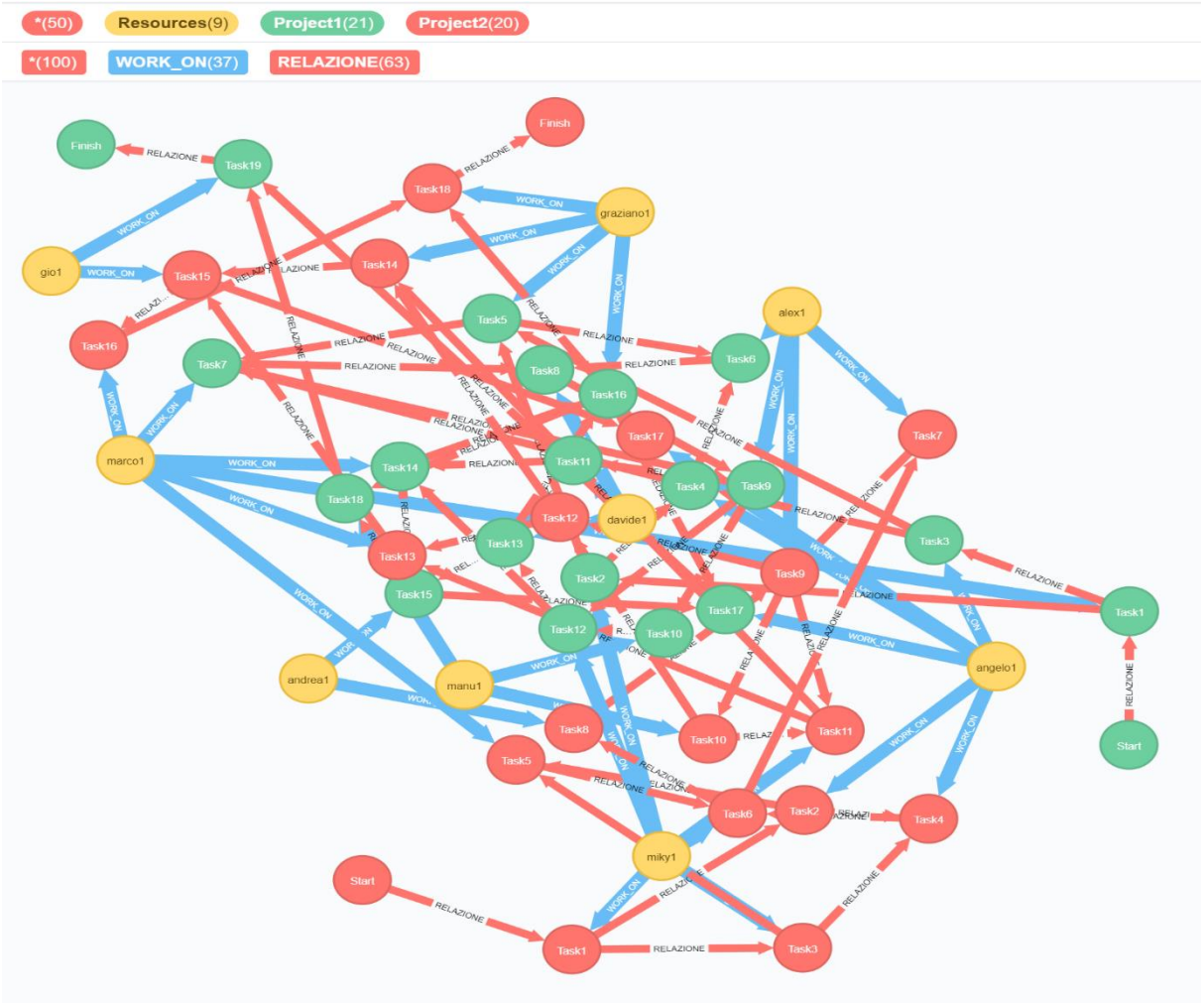
The graph of resources linked to the “Project1”



The graph of resources linked to the “Project2”



The graph of resources linked to both "Project1" and "Project2"



IV. Calculation *ES, EF, LS, LF, Slack* and add them to the graphs.

I perform the calculations I explained above (in section 3.1) to find the critical and flexible activities of projects.

- Add and Set variables **ES**, **EF**, **LS**, **LF = 0** to the "Start" node.

```
# 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 "
```

- Set variables **EF** and **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 "
```

- Update the variables **ES**, **EF**, **LS**, **LF** of the "Finish" node
  I update the properties of the "Finish" node based on the information previously calculated "forward" before calculating the variables LS and 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"
```

- Set the variables **LS** and **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"
```

- Set the variable **slack**

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

- View **ES**, **EF**, **LS**, **LF** and **Slack Times** for each project.

```
# 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"
```

Project1 table:

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

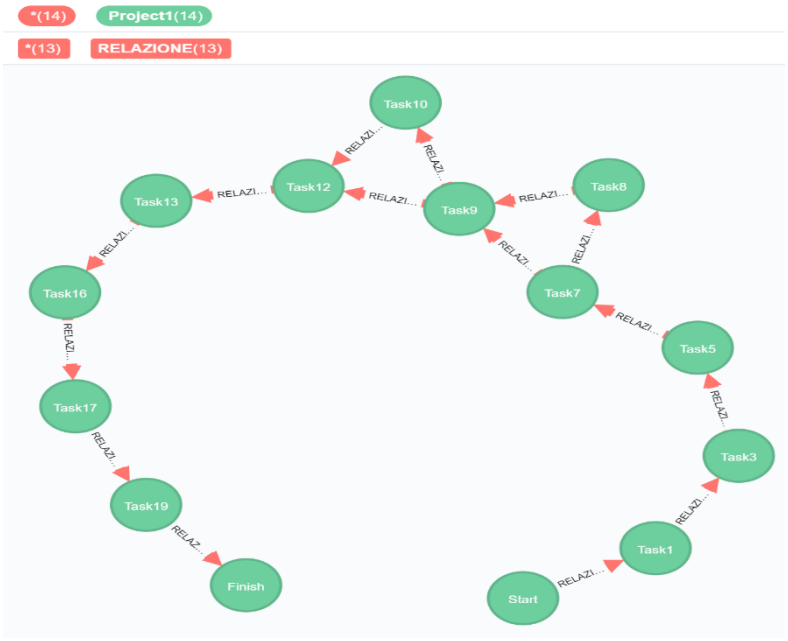
Project2 table:

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

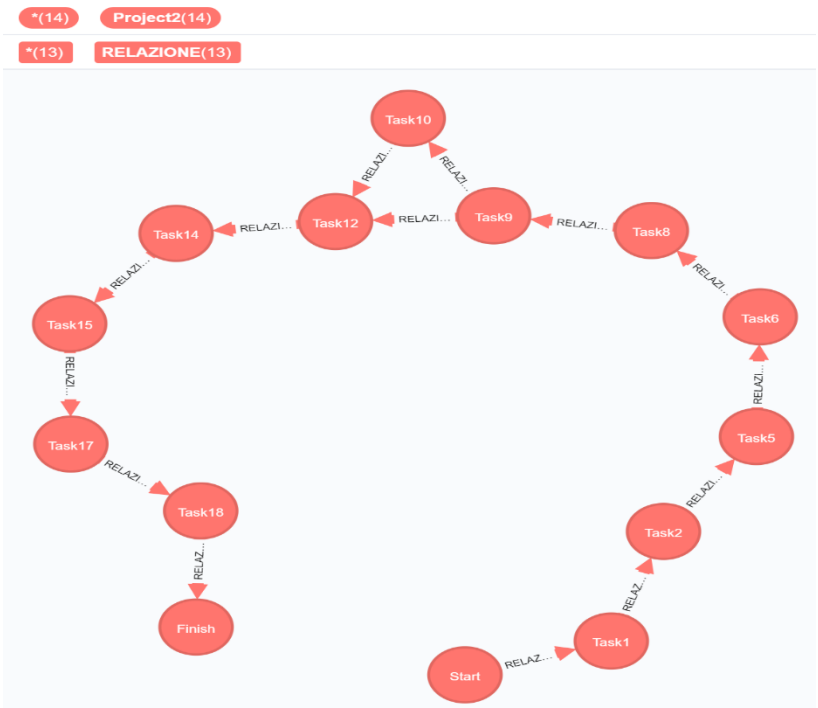
V. Calculate the critical path for each project that is equal to the longest path (which should coincide with the variables *slack* = *0*).

```
# 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:



VI. Comparison if a resource works consecutively on the same critical Path

Now, I have the material to be able to analyze the various critical paths and resources allocated.

Before moving on to compare the critical paths of the various projects between them, I wanted to perform an analysis on the individual critical paths first so as to check for anomalies on the individual critical paths first.

I compared every "previous task" with the "next task" taking as parameters **ID**, **attivita**, **ruolo** and **nick**. Obviously, the parameters "**ID**" and "**attivita**" have served me for convenience, to better visualize the data. The "**ruolo**" parameter served me to understand the associated resource roles at that time to the next incriminating tasks.

The comparison, real, happened with the parameter "**nick**", where as first condition I checked that the nick was not empty (so as not to take the nodes dummy "Start" and "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 CONSECUTIVAMENTE 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 CONSECUTIVAMENTE 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': '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': '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': 'andrea1', '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': ' '}]
```

I think I've planned them too well. Or it will be luck. There are no resources allocated later on the same critical path.

VII. Comparison if a resource is allocated on multiple critical paths at the same time

I take the projects two to two and compare them to each other, for all the N-projects present.

```
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']
            attivita1 = k['attivita']
            nick1 = k['nick']
            ruolo1 = k['ruolo']
            inizio1 = k['start']
            fine1 = k['finish']
```

Then, always using the parameter "**nick**" I'm going to analyse if that resource can be allocated at the same time on more critical path, at least one day.

```
if ((nick == nick1) and (nick != " ") and (nick1 != " ")):
    if ((inizio1 >= inizio) and (fine1 <= inizio)):
        print(name_project + "->" + name_project_2)
        print("Date sovrapposte inizio: " + nick + "\t" + str(inizio1) + "\t" + str(fine1) + "\t" + str(inizio))

    elif ((inizio1 >= fine) and (fine1 <= fine)):
        print(name_project + "->" + name_project_2)
        print("Date sovrapposte fine: " + nick + str(inizio1) + "\t" + str(fine1) + "\t" + str(fine))
        fine = datetime.datetime.strptime(str(fine), '%Y%m%d').date()
        inizio1 = datetime.datetime.strptime(str(inizio1), '%Y%m%d').date()
        fine = fine + datetime.timedelta(days=1)
```

VIII. Comparison if a resource on multiple critical paths is allocated consecutively to several successive tasks of different projects.

Continuing, the code above, I'm going to do another internal comparison to determine whether a resource is allocated on subsequent tasks of different critical paths. Then, comparing the end date task of the $Project_{n-1}$ with the start data of the $Project_{n}$, always using the "**nick**" parameter. And mark the anomaly with tips

```
elif (fine == inizio1):
    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: " + id1 + " * " + "Risorsa: " + nick1 + " * " + "Ruolo: " + ruolo1 + " * " + "Inizio attività: " + st
            inizio1) + " )")
    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 (" + ruolo1 + ")")
```

Result Set

```
Project1
 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': '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': 'andrea1', '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': ' '}]
```

But even in these cases, there are no resources allocated on more critical paths at the same time. There are not even resources that are allocated consecutively to several subsequent critical path tasks of different projects.

IX. Insertion of the "Project3" and "Project4" for a stronger feedback

Considering that I had not been able to report anything between "Project1" and "Project2", I decided to add the planning of two other projects.
We go directly to the Result Set of comparisons.

```
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': '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': 'andrea1', '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': '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': ' '}]
```

As we can see in the comparison in which we go to check if a resource is subsequently allocated to two tasks on the single project's critical path, there is the **davide1** resource allocated later on the **Task7** and **Task8**.

However, while the result sets of other comparisons, there are no resources allocated at the same time on more critical paths. But there are resources that are allocated consecutively to several subsequent critical path tasks of different projects.

```
Project1->Project3
!!!ALERT!!! >>> Risorse su progetti diversi ma con task consecutivi
Project1: (ID: 2 * Risorsa: marco1 * Ruolo: analyst * Fine attività: 20170905 )---> Project3: (ID: 2 * Risorsa: marco1 * Ruolo: analyst * Inizio attività: 20170905 )
SOSTITUIRE RISORSA---> marco1
Mettersi in contatto con i Project Manager del (Project1) e (Project3)
SUGGERIMENTO 1 : Se volete sostituire la risorsa: (marco1) nel Project1 dovete farlo con una risorsa che ha il ruolo (analyst)
SUGGERIMENTO 2 : Se volete sostituire la risorsa: (marco1) 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: davide1 * Ruolo: analyst * Fine attività: 20170922 )---> Project1: (ID: 9 * Risorsa: davide1 * Ruolo: system designer * Inizio attività: 20170922 )
SOSTITUIRE RISORSA---> davide1
Mettersi in contatto con i Project Manager del (Project3) e (Project1)
SUGGERIMENTO 1 : Se volete sostituire la risorsa: (davide1) nel Project3 dovete farlo con una risorsa che ha il ruolo (analyst)
SUGGERIMENTO 2 : Se volete sostituire la risorsa: (davide1) 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: davide1 * Ruolo: analyst * Fine attività: 20170922 )---> Project4: (ID: 9 * Risorsa: davide1 * Ruolo: system designer * Inizio attività: 20170922 )
SOSTITUIRE RISORSA---> davide1
Mettersi in contatto con i Project Manager del (Project3) e (Project4)
SUGGERIMENTO 1 : Se volete sostituire la risorsa: (davide1) nel Project3 dovete farlo con una risorsa che ha il ruolo (analyst)
SUGGERIMENTO 2 : Se volete sostituire la risorsa: (davide1) 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: marco1 * Ruolo: analyst * Fine attività: 20170905 )---> Project3: (ID: 2 * Risorsa: marco1 * Ruolo: analyst * Inizio attività: 20170905 )
SOSTITUIRE RISORSA---> marco1
Mettersi in contatto con i Project Manager del (Project4) e (Project3)
SUGGERIMENTO 1 : Se volete sostituire la risorsa: (marco1) nel Project4 dovete farlo con una risorsa che ha il ruolo (analyst)
SUGGERIMENTO 2 : Se volete sostituire la risorsa: (marco1) nel Project3 dovete farlo con una risorsa che ha il ruolo (analyst)
```