Idee progetto Large-Scale:

1) App per la ricerca di lavoro (tipo LinkedIn)

Riferimenti utili: <https://engineering.linkedin.com/architecture/brief-history-scaling-linkedin>

<https://www.geeksforgeeks.org/use-of-nosql-in-industry/>

Tipi di database:

* MongoDB per conservare le offerte di lavoro e le relative informazioni
* Neo4J per le relazioni tra gli utenti

Query/Analytics: Possibilità di confrontarle con quelle ufficiali di LinkedIn

1) Lavori suggeriti

2) Utenti suggeriti da seguire

3) Lavori più indicati in base al profilo

4) Lavori più pagati

<https://blog.hootsuite.com/linkedin-analytics/>

<https://blog.hootsuite.com/how-the-linkedin-algorithm-works-hacks/>

Web Scraping: <https://medium.com/federicohaag/linkedin-scraping-with-python-d8d14519602d>

Si può fare anche con qualsiasi altro sito di offerte di lavoro

2) Sito per raccolta di ebook gratuiti (sembra che non c’è moltissima roba simile) (bello)

Tipi di database:

* + MongoDB per conservare le informazioni sui libri/autori
  + Redis / qualsiasi altro key value per la cache

Query:

1) Libri più letti

2) Autori più prolifici

3) Libri con valutazioni migliori

Web scraping:

<https://github.com/pablotrinidad/oreilly-api>

3) App per il meteo (brutto)

Tipi di database: non saprei sinceramente da pensarci

Si può pensare tipo di integrare un social Network dove commentare il meteo, tipo l’affidabilità delle previsioni o anche eventi atmosferici straordinari

Query:

1) Si potrebbe pensare di guardare i giorni precedenti e provare a fare previsioni sui giorni futuri oppure fare solo scraping

2) Località più piovose della zona

3) Giorni migliori per uscire all’aria aperta

Web Scraping:

<https://www.visualcrossing.com/resources/blog/how-to-load-historical-weather-data-using-python-without-scraping/#:~:text=Scraping%20weather%20data%20means%20we,and%20then%20difficult%20to%20maintain>

<https://towardsdatascience.com/weather-data-collection-web-scraping-using-python-a4189e7a2ee6>

4) App per suggerimenti su giochi da tavolo

Tipi di database:

* + MongoDB per informazioni relative ai giochi
  + Neo4J ( o qualsiasi altro a grafi) per relazioni tra utenti

Query:

1) Giochi più giocati

2) Giochi che costano meno

3) Vari filtri (numero giocatori, genere ecc)

4) Giochi meglio recensiti

Web Scraping:

Siti:

1) <http://www.igiochideigrandi.it/>

2) <https://boardgamesinvasion.com/>

3) <https://www.giochinscatola.it/it/>

Scraper:

1) <https://towardsdatascience.com/web-scraping-the-list-of-greatest-video-games-from-wikipedia-pt-1-230c64d93e8c>

2) <https://github.com/recommend-games/board-game-scraper> **!!!!!!!!**

3) <https://boardgamegeek.com/thread/2287371/boardgamegeek-games-and-ratings-datasets> (dataset)

4) <https://www.kaggle.com/mshepherd/board-games> (dataset)

5) <https://nycdatascience.com/blog/student-works/scrapping-boardgames/>

6) <https://sdsawtelle.github.io/blog/output/boardgamegeek-data-scraping.html>