

JavaScript Node.js

Conclusione

Ancora una cosa noiosa

- Il modo piu' semplice per scrivere unit test per Express e' usare il pacchetto supertest (<https://www.npmjs.com/package/supertest>)

```
const request = require("supertest");

const app = require('../..../app')

describe("GET /", function () {

  it("responds with json", function (done) {

    request(app).get("/").set("Accept", "application/json").expect(200, done);

  });

});
```

Disclaimer

Tutto quello che segue e' frutto della mia esperienza personale...

Lavoro step -1: Cosa importa davvero

- L'esperienza (non necessariamente lavorativa) conta piu' di qualunque attestato o certificazione
- L'inglese e' fondamentale
- L'attitudine conta piu' delle competenze tecniche

Lavoro step 0: CV

- Usate un formato di CV standard (possibilmente europeo), possibilmente su 1 sola pagina
- E' piu' importante la qualita' della quantita'
- Esperienze ed educazione con in ordine cronologico inverso (prima la cosa piu' recente)
- Mettete link a Github o progetti passati
- Controllate sempre i refusi

Lavoro step 0.5: Il colloquio tecnico

- I colloqui seri prevedono solitamente una parte di domande tecniche e algoritmi
- Studiate e ripassate alcuni algoritmi e concetti “base”:
 - FizzBuzz
 - big O
 - BFS, DFS, sorting
 - Aspetti specifici del linguaggio/tecnologia per cui state applicando
- Esistono diversi siti su cui e' possibile allenarsi su varie coding challenges:
 - <https://www.codewars.com/>
 - <https://www.hackerrank.com/>

Ruoli

- Business Analyst: formalizza i requisiti
- UX/UI: progetta e disegna l'interfaccia
- Tech Lead/Solution Architect: analizza i requisiti e le soluzioni tecniche
- Team Leader: gestisce un team di sviluppatori
- Developer: sviluppa il codice
- DevOps/Sistemista: si occupa dell'infrastruttura
- QA: testa manualmente e/o scrive test automatici per validare la qualità del prodotto
- Project Manager: gestisce i flussi di lavoro interni ed esterni

Struttura di un team

- La struttura e le dimensioni di un team dipendono moltissimo dal progetto e dell'azienda
- Un team puo' essere formato da 15 persone come da 2 (o anche 1 sola)
- All'interno di un team possono esserci alcune figure specializzate, ma solitamente l'ideale e' che tutti siano in grado di lavorare su ogni aspetto del progetto
- Meeting, chat e git sono strumenti fondamentali all'interno di un team

Alcune metodologie “moderne”

- Oggi sono molto di moda metodologie come Agile e Scrum contrapposte al normale flusso di lavoro chiamato Waterfall
- Nel Waterfall tempi e procedure vengono stabiliti esternamente al team di sviluppo
- Lo scopo generale e' coinvolgere maggiormente e responsabilizzare l'intero team tramite riunioni brevi
- Lo sviluppo di un progetto e' spezzato in periodi brevi ($\frac{1}{3}$ settimane) con obiettivi chiari e quantificabili chiamate Sprint

Un progetto tipo: fase di analisi

- Il BA, parlando con il cliente (anche se e' l'azienda stessa per progetti interni) inizia a definire i dettagli del prodotto
- UX/UI sviluppa prima i mockup e poi i layout che il cliente approvera'
- Il Tech Lead/Solution Architect analizza possibilita' e problematiche tecniche e fa una prima stima generale
- Il BA (spesso con l'aiuto del Tech Lead) formalizza i requisiti scrivendo delle Epiche o delle Storie
- Il QA Lead scrive gli scenari che verranno testati

Un progetto tipo: fase di sviluppo

- **Sprint Planning:**
 - E' un meeting in cui tutto il team analizza, stima e “spezza” in sotto-task le Storie o Epiche che verranno sviluppate durante la sprint
 - Teoricamente e' il team a decidere il numero di task che verranno sviluppati in una Sprint in base alle stime effettuate
- Ogni sviluppatore prende in carico un task, lo completa; il codice viene (in teoria) revisionato e “mergiato” nel progetto.
- Il progetto viene “deployato” spesso in un ambiente per essere testato dal team di QA
- **Sprint Review:** a fine sprint si valuta cosa e' andato bene, cosa no e si cerca di migliorare il processo alla prossima iterazione