

PIANO FORMATIVO

Introduzione alla programmazione - OOP -
Versioning - JS ES6 - Typescript - NodeJS -
Database - Angular/React

Data 22/12/2021

Versione 1.4

Autore Massimiliano Tozzi

INTRODUZIONE

Il programma definito prevede 450 ore totali di formazione di cui 209 di tirocinio, suddivise in 3 Fasi.

Al termine di esse, gli Studenti collaboreranno per lo sviluppo di un **Progetto Finale** che racchiude tutte le nozioni studiate durante il periodo di formazione.



Le Fasi sono così definite:

- **Fase 1:** programma incentrato sulle basi di programmazione Object Oriented, sui sistemi di versionamento del codice (GIT), sulle basi dati relazionali, NoSQL e Cloud. Si approfondiranno quindi i linguaggi di programmazione HTML, CSS, Javascript e Typescript.
- **Fase 2:** contempla nozioni di introduzione al paradigma RESTful, nozioni sulle Rest-API, introduzione alla conoscenza dello sviluppo del Back-end applicativo per mezzo del framework node JS, nozioni sul pattern MVP e acquisiranno le basi dello sviluppo del Front-End applicativo per mezzo dell'apprendimento e dell'utilizzo del framework Angular/React.
- **Fase 3:** i discenti saranno introdotti poi, approfonditamente, allo sviluppo applicativo di Back-end e a tale scopo esamineranno in profondità l'utilizzo del framework node JS.

FASE 1 (208 ore)

L'obiettivo durante questa fase è quello di dare una base di partenza incentrata sulla programmazione OOP che si diramerà poi nell'approccio al linguaggio Javascript ES6 e Typescript in particolare nonché a quello HTML comprensivo della



avvicinamento alla conoscenza dei suoi fogli di stile, CSS. Tutto ciò aiuterà a definire una forte conoscenza di base per gli Studenti.

MATERIA	DETTAGLIO	TEORIA	TIROCINIO
Object Oriented Programming	<ol style="list-style-type: none"> 1. Classi e Oggetti 2. Variabili e Metodi 3. Import e Packages 4. Modificatori di Accesso 5. Incapsulamento 6. Eredità 7. Interfacce e Classi Astratte 8. Polimorfismo 9. Overloading di Metodi 10. Overriding di Metodi 	24 ore	
Git/GitLab	<ol style="list-style-type: none"> 1. Git/Gitlab avanzato 2. Issue tracking 3. Protocollo ssh 	4 ore	4 ore
Javascript	<ol style="list-style-type: none"> 1. Introduzione alla Programmazione <ol style="list-style-type: none"> a. Binary System b. Intro Operatori Logici 2. Valori e Variabili <ol style="list-style-type: none"> a. Esecuzione del codice nella console JS 	20 ore	20 ore



	<ul style="list-style-type: none">b. Introduzione ai numeric. NaN e infinitod. Variabili e Lete. Operatori unarif. Constg. Var		
	<ul style="list-style-type: none">3. Modellare i dati<ul style="list-style-type: none">a. Introduzione variabili booleaneb. stringhec. Indici stringad. Metodi di stringae. Altri metodi per le stringhef. Caratteri di escape delle stringheg. Valori letterali modello stringah. Nullo e non definitoi. tipo di operatorej. parseInt & parseFloat		
	<ul style="list-style-type: none">4. Strutture Dati Basilari (ARRAY)<ul style="list-style-type: none">a. Cos'è un array		
	<ul style="list-style-type: none">5. Utilizzo delle Condizioni, Cicli, Funzioni<ul style="list-style-type: none">a. Prendere decisioni in JSb. Operatori di confrontoc. Doppio uguale (==)d. Triplo uguale (===)		



	<ul style="list-style-type: none">e. Esecuzione di codice da uno scriptf. If Statementg. else ifh. Valori true e falsei. AND logico (&&)j. OR logico ()k. NON Operatore (!)l. La dichiarazione di Switchm. Operatore ternarion. For loopo. Cicli infiniti!p. For loop e arrayq. For loop annidatir. Introduzione ai cicli Whiles. Il ciclo Whilet. Break Keywordu. For...Of e For...Inv. For...Of e For...In con oggettiw. Cosa sono le funzioni e gli argomentix. Funzioni con più argomentiy. La dichiarazione di ritornoz. Maggiori informazioni sui valori di ritorno <p>6. Strutture Dati Basilari 2 (Array, Map, Set)</p> <ul style="list-style-type: none">a. Introduzione ai metodi di callback dell'array		
--	---	--	--



	<ul style="list-style-type: none"> b. Intro ai metodi di Callback dell' Array c. forEach d. Map e. Array.find f. Filter g. Set <ul style="list-style-type: none"> 7. Prototype and Classi 8. Asynch 9. Http Request 10. Manipolare File (Create,Read,Write,Delete) 11. Struttura dati JSON 		
Typescript	<ul style="list-style-type: none"> 1. Cos'è TypeScript 2. Type Examples 3. Class <ul style="list-style-type: none"> a. Classi b. Eredità di base c. Modificatori del metodo di istanza d. Campi nelle classi e. Campi con eredità f. Dove usare le classi 4. Access Modifiers 5. Errors ed Exception 	25 ore	25 ore



	<ul style="list-style-type: none"> a. Catching Errors with Typescript 6. Interface e Abstract Class <ul style="list-style-type: none"> a. Interfaces b. Long Type Annotations c. Fixing Long Annotations with Interfaces d. Syntax Around Interfaces e. Functions in Interfaces 7. Abstract Classes <ul style="list-style-type: none"> a. Why Use Abstract Classes? b. Solving All Issues with Abstract Classes c. Interfaces vs Abstract Classes 8. Typed Arrays 9. Assertions 10. Generics 		
Data Base	<ul style="list-style-type: none"> 1. Cos'è una base dati 2. Disegno di una base dati 3. Creare una tabella 4. Create, Insert, and Select! 5. SQL Statement 6. Filtrare i risultati 	20 ore	20 ore



	<ul style="list-style-type: none">7. Tabelle<ul style="list-style-type: none">a. 1 to 1 e many to manyb. 1 to many e many to 1c. Primary Key e Foreign Keyd. Constraintse. Delete8. Ordinare i risultati9. JOIN10. Comporre Query11. View, Function e Store-Procedure12. Database Relazionali:<ul style="list-style-type: none">a. MySQLb. PostGreSQL<ul style="list-style-type: none">i. Accedere alla base datiii. Initial Setupiii. Building the Usersiv. Connection Poolsv. Credenziali di Connessionevi. Query and Close Connection13. MongoDB<ul style="list-style-type: none">a. Quando utilizzare Mongob. SetUpc. Come MongoDB archivia i datid. Mongoose JS		
--	---	--	--



	<ul style="list-style-type: none"> e. Scrivi query efficienti per la lettura dei dati f. Progetta uno schema NoSQL efficace con sia l'annidamento dei dati che le ricerche <p>14. Cloud DB</p> <ul style="list-style-type: none"> a. Esempio: Azure b. BigQuery 		
HTML/CSS/SCSS	<ol style="list-style-type: none"> 1. HTML 5 2. CSS 3. SCSS 	20 ore	20 ore



FASE 2 (150 ore)

In questa fase gli alunni acquisiranno la conoscenza del paradigma architetturale **RestFul** per lo sviluppo di servizi basati sul protocollo HTTP nonchè saranno introdotti all'utilizzo del framework basato sul linguaggio javascript, **node JS** e del pattern **MVP**. L'approccio e l'approfondimento nel dettaglio dell'utilizzo del framework di front-end **React (Angular in alternativa)** sarà l'ultimo argomento di questa fase del corso.

MATERIA	DETTAGLIO	TEORIA	TIROCINIO
Intro RestFul - Rest-API		8 ore	8 ore
Node JS Intro	<ul style="list-style-type: none">• Cos'è e come utilizzare Node JS• Installare Node JS• NodeJS Module System• File System e Command Line Arguments<ul style="list-style-type: none">◦ Input◦ Yargs◦ Memorizzare dati con JSON• Debug con NodeJS<ul style="list-style-type: none">◦ Debug◦ Messaggi di Errore	22 ore	22 ore



	<ul style="list-style-type: none"> • Asynchronous NodeJS <ul style="list-style-type: none"> ○ Effettuare HTTP requests ○ Gestione degli errori ○ Funzioni di Callback • WebServer <ul style="list-style-type: none"> ○ Express ○ HTML e JSON ○ Risorse statiche 		
React JS	<ol style="list-style-type: none"> 1. Imparare a creare app Web dinamiche. 2. Comprendere le caratteristiche di base di React. 3. Imparare la sintassi di React e come programmare usandola. 4. Imparare a sfruttare JavaScript per raggiungere obiettivi specifici. 5. Imparare a conoscere React Hooks. 6. Implementazione di componenti basati su classi. 7. Imparare alternative a React. 8. Apprendimento e implementazione di single-page o multi-page application 9. Imparare a lavorare con le liste e con i componenti condizionali 10. Imparare a dare uno stile ai vari componenti ed elementi di React. 11. Imparare a eseguire il debug delle app React. 	45 ore	45 ore



	12. Lavorare con il Burger Builder Project per imparare ad accedere ai server.		
Angular (in alternativa a React)	<ol style="list-style-type: none"> 1. Introduzione <ol style="list-style-type: none"> a. Ambiente NodeJS b. Ambiente Angular CLI c. Ambiente Sviluppo V-Studio-Code 2. Component Template 3. Implementare Componenti 4. Direttive in Angular 5. Gestire gli eventi 6. Il Sistema dei Moduli 7. Routing e navigazione tra le pagine 8. Recuperare i dati 9. Validatori 10. Autenticazione 11. Security 12. Angular Front-End - Material 13. Form - Service - Routing in pratica 14. Deploy 	45 ore	45 ore



FASE 3 (92 ore)

La base di quest'ultima fase del corso è incentrata sull'approfondimento della conoscenza del framework **node js**, sulle nozioni di alto livello di architetture di sistemi, dei comandi del sistema operativo UNIX e sulle best practice delle metodologie di approccio e le fasi di un progetto di sviluppo software.

MATERIA	DETTAGLIO	TEORIA	TIROCINIO
Node JS Advanced	<ul style="list-style-type: none">• WebServer<ul style="list-style-type: none">◦ Express◦ HTML e JSON◦ Risorse statiche• REST APIs e base dati (Mongo)• API Authentication• Ordinamento, impaginazione e filtraggio dei dati• Upload di File• Invio di EMAIL• Effettuare TEST con NodeJS<ul style="list-style-type: none">◦ Framework◦ Assertion	30 ore	30 ore



	<ul style="list-style-type: none">• Real Time Web App (Chat)		
Unix	<ul style="list-style-type: none">• Comandi da terminale	4 ore	4 ore
Architetture		8 ore	
Metodologie di Lavoro	<ul style="list-style-type: none">• Best Practice di organizzazione e di gestione di progetti• Come nasce e viene gestito un progetto: dalla raccolta dei requisiti alla realizzazione di un piano di progetto	8 ore	8 ore

