

---

## Introduzione

Dalla scoperta dell'effetto transistor (nel 1947) fino ad oggi, il campo dell'Elettronica ha visto moltissimi cambiamenti, grazie agli sviluppi tecnologici apportati, che hanno permesso di realizzare dispositivi sempre più piccoli con prestazioni sempre migliori.

Il Metal-Oxide-Semiconductor Field-Effect Transistor (MOSFET) è stato, senza dubbio, una delle scoperte più importanti del XX secolo. Grazie alle sue proprietà, infatti, si è arrivati ad uno sviluppo tecnologico e ad una potenza di calcolo che, fino alla metà del Novecento, sarebbero stati impensabili. Per rendersi conto di questo progresso basta pensare che la prima macchina da calcolo, l'ENIAC (Electronic Numerical Integrator And Computer), prodotta nel 1946, era composta da circa 18000 valvole termoioniche, occupava un'intera stanza, dissipava una potenza di circa 200 kW ed eseguiva un'istruzione ogni 0.5 ms. Oggigiorno si possono realizzare sistemi in grado di eseguire un'istruzione in meno di 1 ns, che dissipano potenze inferiori a 100 W.

Nella società moderna, un numero notevole e sempre crescente di prodotti incorpora tecnologie elettroniche digitali, cioè basate sulla logica binaria; e, quella CMOS<sup>1</sup> (il cui mattone fondamentale è il MOSFET) ha soppiantato quasi ovunque tutte le precedenti tecnologie. Microprocessori, personal computer, telefoni cellulari e dispositivi interconnessi, sono solo alcuni dei molteplici esempi di applicazioni che hanno cambiato la vita quotidiana delle persone nell'ultimo ventennio.

In particolare, i dispositivi mobili hanno conquistato il mercato degli ultimi anni, grazie alla diffusione di nuovi modelli sempre più efficienti e innovativi (si pensi, ad esempio, agli smartwatch o agli occhiali per la realtà aumentata, i famosi "Google Glass"). Inoltre, la diffusione degli smartphone e le possibilità offerte dalla rete Internet hanno incoraggiato notevolmente lo sviluppo di applicazioni sempre più complesse e colme di funzionalità (qualche volta di dubbia utilità). Pertanto, la presenza stessa delle aziende sul web non può, ormai, prescindere dalle caratteristiche e dalle modalità di utilizzo di questi dispositivi.

In conseguenza di ciò, negli ultimi anni, si è verificata una vera e propria esplosione di nuovi linguaggi di programmazione e *framework*<sup>2</sup> in grado di rispondere ad una sempre maggiore richiesta di interattività e *look&feel* da parte degli utenti; ciò

---

<sup>1</sup> *Complementary Metal Oxide Semiconductor*.

<sup>2</sup> Ambiente all'interno del quale è possibile sviluppare applicazioni web.

ha portato i vari team specializzati alla creazione di strumenti, tra cui il *Responsive Web Design* (RWD), in grado di aiutare gli sviluppatori nella realizzazione di applicativi web sempre più avanzati.

L'RWD è un importante elemento dell'accessibilità e tiene conto di numerosi fattori incentrati non solo sui dispositivi, ma anche sulle caratteristiche dell'utente. Non si tratta soltanto di una metodologia per migliorare l'esperienza dell'utente che la utilizza, ma rappresenta anche uno strumento per i progettisti software per poter sviluppare sistemi compatibili su dispositivi diversi [22].

Fu proprio con l'introduzione dell'iPhone (nel 2007) che si verificò il punto di svolta; prima di allora, infatti, i siti web dovevano essere adattati alla visualizzazione per schermi desktop, i quali variavano sì per dimensioni, ma in modo irrilevante. Inizialmente, si pensò di realizzare versioni secondarie dei siti, appositamente per cellulari; tuttavia, con la diffusione di smartphone e tablet dalle dimensioni sempre più diversificate, si pensò di implementare una metodologia che consentisse di realizzare *layout*<sup>3</sup> funzionali, indipendentemente dalle dimensioni dei display [26]; fu, così, che si entrò nell'era del design responsivo.

A fronte della diffusione di dispositivi sempre più piccoli ed anche meno capienti, l'obiettivo degli sviluppatori fu, altresì, quello di adottare delle soluzioni che consentissero l'utilizzo delle applicazioni, senza necessità di installazione delle stesse, garantendo, così, una totale compatibilità con tutti i dispositivi. Iniziarono a diffondersi, quindi, le cosiddette "web application" (dette, più comunemente, "web app"), ovvero applicazioni accessibili attraverso la rete Internet, direttamente dal browser del proprio dispositivo, indipendentemente dal sistema operativo installato. Si prevede che, in un futuro non troppo lontano, PC, smartphone, smartwatch, smartglass, etc. saranno interconnessi in maniera massiva e, a loro volta, saranno allacciati a potenti server che daranno, agli utenti finali, tutti i servizi possibili, senza dover più installare o possedere nulla sul proprio dispositivo; i programmi stessi non dovranno più essere installati, visto che risiederanno sulla rete.

Già da qualche tempo, Google fornisce dei servizi online molto efficienti, che sostituiscono, in maniera egregia, buona parte della suite *Office* di Microsoft. L'utilità di un sistema in remoto risiede, anche, nel fatto che ci si deve preoccupare solo di usare il software e non della sua manutenzione, nè del suo backup (operazioni demandate ai server su cui risiede). Oltre a ciò, una web app può essere usata da qualunque postazione al mondo: di fatto, in ogni angolo del pianeta, con una macchina e con un browser installato, sarà come essere nel proprio ufficio.

Al giorno d'oggi, sono molte le aziende che assumono ingegneri elettronici ed informatici appositamente per creare delle app utili a rendere l'attività lavorativa presente e attiva sul web; quella dello sviluppatore software è una delle professioni più richieste in assoluto nel settore dell'ICT<sup>4</sup>. Il termine "sviluppatore", tuttavia, è ormai diventato molto generico, in quanto, sotto questa dicitura, coesistono, in realtà, compiti e funzioni ben diversi. In particolare, la figura del *full-stack developer* è una delle più importanti nell'ambito dell'Ingegneria del Software, in quanto identifica uno sviluppatore con competenze sia lato front-end (relativo, cioè, alla programmazione lato-client) che lato back-end (ovvero tutto ciò che riguarda la programmazione e la configurazione lato-server).

<sup>3</sup> Disposizione degli elementi grafici nell'area visibile.

<sup>4</sup> *Information and Communications Technology*.

Con la diffusione dei dispositivi mobili, disporre di un'applicazione gestionale personalizzata, rappresenta, anche per le associazioni no-profit (siano esse di natura sportiva, culturale, ideologica, etc.), un'opportunità che consente di promuovere progetti sociali che devono essere comunicati per poter crescere e coinvolgere sempre più persone.

Scopo di questo lavoro di tesi è quello di informatizzare una realtà che opera nel Terzo Settore, ovvero l'Associazione culturale "Verdidee Giovani". Ciò che ci ha spinto ad intraprendere questo progetto è stata la voglia di approfondire le tematiche relative allo sviluppo di software sempre più orientati, come detto in precedenza, al mondo mobile, nonché la richiesta del mercato del lavoro, oggi più che mai, alla ricerca di analisti e progettisti con competenze non solo in ambito hardware, ma anche, e soprattutto, in ambito software.

Il sistema progettato prevede la realizzazione di un sito web e di un pannello di amministrazione. Il sito web, tecnicamente indicato come "front-end", è la parte visibile ed utilizzabile da qualsiasi visitatore. Esso è costituito da due pagine; la prima pagina è relativa alla presentazione dell'Associazione (attraverso le sezioni "Chi Siamo", "Attività", "Photogallery" e "Contatti"), mentre la seconda riguarda la possibilità di iscriversi agli eventi e di ottenere informazioni sugli stessi. Il pannello di amministrazione ("back-end"), invece, rappresenta la parte "nascosta" al visitatore, ovvero quella accessibile solo agli amministratori, dalla quale, gli stessi potranno gestire tutti gli aspetti inerenti le attività dell'Associazione (dall'amministrazione contabile alla gestione dei tesseramenti, dall'organizzazione degli eventi alla gestione delle iscrizioni agli stessi, etc.).

Con l'obiettivo di realizzare un software gestionale accessibile da qualsiasi dispositivo (sia esso desktop o mobile), indipendentemente dalle dimensioni del display e dal sistema operativo installato, si è pensato di realizzare un'applicazione di tipo *responsive*, fruibile via Internet (una "web app" appunto), attraverso l'utilizzo (per quanto riguarda l'interfaccia grafica) delle tecnologie HTML5, CSS3 e JavaScript, strumenti ormai indispensabili per gli sviluppatori in ambito di progettazione di *layout* web; l'implementazione delle funzionalità del sistema, invece, è stata realizzata utilizzando il linguaggio di programmazione PHP (acronimo ricorsivo di "PHP Hypertext Preprocessor"), un linguaggio di *scripting* general-purpose open source, molto utilizzato e, soprattutto, indicato per lo sviluppo web, visto che può essere integrato nel codice HTML<sup>5</sup> delle pagine. Ciò che distingue PHP da altri linguaggi di *scripting* del tipo "client-side" (lato-client) è che il codice viene eseguito sul server, il quale genera il codice HTML che viene, successivamente, inviato al client; pertanto, quest'ultimo riceve i risultati dell'esecuzione dello *script* senza conoscere qual è il codice eseguito.

Per la rappresentazione e la manipolazione dei dati è stato utilizzato il DBMS<sup>6</sup> MySQL, un software gratuito, anch'esso open source, molto diffuso in ambito di gestione di database e basato sulla sintassi del linguaggio SQL<sup>7</sup>. Attraverso MySQL si è potuto rappresentare, in termini di tabelle, le entità e le relazioni ricavate dalle progettazioni concettuale e logica della realtà informatizzata.

<sup>5</sup> *HyperText Markup Language*.

<sup>6</sup> *DataBase Management System*.

<sup>7</sup> *Structured Query Language*.

Il presente elaborato è strutturato come di seguito specificato:

- Nel *Capitolo 1* si pone l'attenzione sulle tecnologie che hanno spinto gli esperti del settore ad orientare la ricerca verso lo sviluppo di software di tipo mobile; in particolare, vengono illustrati i principali approcci e gli ambienti di programmazione necessari per implementare software di tipo *responsive*.
- Nel *Capitolo 2* vengono illustrate le tecnologie impiegate per la realizzazione del presente elaborato di tesi. In particolare, si descriveranno i concetti generali sui linguaggi di programmazione utilizzati (lato-client e lato-server), sul DBMS MySQL e sul linguaggio SQL.
- Il *Capitolo 3* si occupa dell'analisi dei requisiti relativi alla realizzazione della web app creata e del comportamento del sistema attraverso una descrizione dei casi d'uso.
- Nel *Capitolo 4* viene trattata la fase di progettazione dei dati. La prima parte riguarda la progettazione concettuale ed ha come oggetto il modello entità-relazione; nella seconda si fa riferimento al modello relazionale come modello logico.
- Il *Capitolo 5* si occupa di descrivere la fase di progettazione della componente applicativa, che ha lo scopo di progettare le funzionalità offerte dal sistema, attraverso la definizione della struttura della web app, l'illustrazione dei *process flow* di alcuni casi d'uso ed, infine, di alcuni *mockup*.
- Nel *Capitolo 6* vengono presentati e commentati alcuni listati che hanno consentito la realizzazione del sistema (dalla creazione delle tabelle del database all'implementazione delle funzioni principali del sistema); successivamente, viene presentato il manuale utente.
- Nel *Capitolo 7* viene effettuata l'Analisi SWOT, che mette in evidenza i punti di forza, i punti di debolezza, le opportunità e le minacce a cui potrebbe essere esposto il sistema realizzato.
- Il *Capitolo 8* riguarda il confronto con alcuni sistemi correlati, ovvero altre applicazioni di associazioni simili a quella della realtà informatizzata nel presente lavoro di tesi, mettendo in evidenza gli aspetti comuni e quelli peculiari di ciascun sistema.
- Infine, il *Capitolo 9* riguarda la stima dei costi di realizzazione e le conclusioni finali, con uno sguardo ai possibili sviluppi futuri.