



Il programma del Corso App Developer

210 ore
dal 17 ottobre 2022

UI DESIGN E GOOGLE MATERIAL

Ricerca e Analisi del progetto

- Individuazione del briefing e del problema da risolvere
- Concept
- Architettura delle informazioni - IA
- Tree test per validazione Architettura Informazioni
- Analisi funzionale basica
- Concetto di UCD (user centered design process)
- Eventuale Card sort e lavoro con penna e matita per un brainstorming
- Prime bozze / idee realizzate su carta

Design

- Rendere reale il concept ideato
- Studio e definizione dei componenti funzionali delle App
- Studio e posizionamento dei contenuti e dei componenti all'interno dell'applicativo
- Censimento dei componenti esistenti, approfondendone le caratteristiche, le dimensioni, l'aspetto e la funzionalità
- Definire quali di questi sono idonei alle esigenze funzionali dell'applicativo
- Google Material Design: studio approfondito della guida di google material e dei suoi componenti
- Creazione di un Wireframe
- Concetto di User Interface - UI
- Comunicazione Visuale dell'applicativo
- Colori (teoria del colore per guidare la scelta della palette dei colori dell'applicativo)
- Creazione di un'interfaccia grafica pixel perfect, personalizzazione del wireframe approvato
- Typography

Gerarchia degli elementi grafici

- Testing
- Usabilità e Accessibilità
- SPLIT TEST + AB test + PREFERENCE TEST
- Concetto di testing sull'utente finale. Come migliorare l'applicativo tramite feedback
- Consegna della User Interface al cliente finale
- Simulazione della consegna dell'interfaccia dell'applicativo a un ipotetico cliente

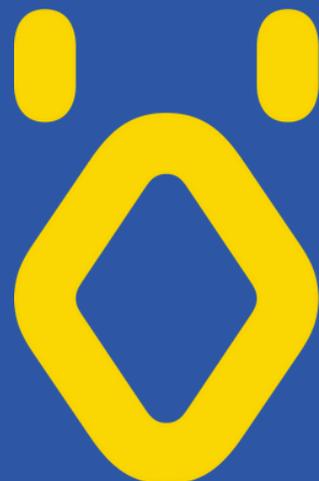
JAVA

Introduzione a Java e le sue tecnologie

- Compilazione e JVM
- Tipi primitivi e tipizzazione
- Variabili e i loro tipi, casting
- Compilazione ed esecuzione di applicazioni
- Problem solving (diagrammi funzionali)
- Classi Wrapper: boxing, unboxing e autoboxing
- Operatori e manipolazione di dati
- Diagrammi funzionali e controllo del flusso
- If, else, elseif, switch case
- Cicli
- Eccezioni ed errori in Java
- Array

Programmazione OOP e astrazione

- Le classi Java e le istanze di classi
- Gestione della memoria: inizializzazione, distruzione, duplicazione di oggetti
- Metodi e attributi (campi)
- Eccezioni
- Package Java e organizzazione del codice
- Incapsulamento
- Estendere le classi: ereditarietà
- Overriding
- Polimorfismo
- Interfacce
- Oggetti, attributi e metodi statici
- Classi astratte
- Overloading
- Il controllo dell'errore
- Collection framework



FLUTTER BASE

Introduzione e installazione di Flutter

- Che cos'è Flutter, a cosa serve e perché utilizzarlo
- Anatomia di un'applicazione Flutter
- Installazione del Flutter SDK
- Installazione di Android Studio e VS Code
- Installazione di Android Emulator

Creare applicazioni Flutter da zero

- Creazione di un nuovo progetto
- Strutturare un'applicazione Flutter
- Lavorare con gli Assets e il Pubspec file
- Lanciare e fare il deploy dell'applicazione sulla macchina in locale
- Fare il deploy dell'applicazione Flutter su un telefono Android
- Fare il deploy dell'applicazione Flutter su un telefono IOS/iPad

Introduzione a Dart Programming Language

- Dart e Dart Pad - L'editor Online
- Variabili e tipi variabili in Dart
- Condizioni If else, operatori logici e ciclo For in Dart
- While, Do-While, Break e Switch Cases: cosa sono e come utilizzarli
- Le Funzioni in Dart: gli argomenti, i return types e l'Arrow function
- Cosa sono i Parametri opzionali
- Introduzione alle Collections in Dart
- La Null Safety

Creare UI utilizzando i Widgets di flutter

- Come utilizzare i Container Widgets
- Come utilizzare i Column e Row Widgets per il Layout
- Come incorporare dei font custom all'interno dell'applicazione Flutter
- Flutter Card e ListTile Widgets

FLUTTER AVANZATO

Flutter Animations

- Introduzione alle animazioni in Flutter
- Ticker, Animation e AnimationController
- Classe Tween: che cos'è e come funziona
- Chain Animation e Animation State: come gestire più animazioni

Flutter Navigation

- Che cos'è la navigation in Flutter e come funziona
- Concetto di Rotta e Navigazione
- Come spostarsi e passare informazione da una rotta all'altra

Flutter DB

- Introduzione ai database relazionali
- Utilizzo di SQLite

Flutter State Management

- ListView e BottomSheet Widget
- Che cos'è lo state e come fare a controllarlo
- ListView Builder, introduzione al Provider Package
- Utilizzo del Provider e del ChangeNotifier

Networking in Flutter

- Introduzione al HTTP e ai JSON
- Async e Future: Richieste HTTP
- Come aggiungere il package HTTP al nostro progetto e come utilizzarlo
- Come fare una chiamata HTTP e come intercettare la sua response (JSON data)
- Catturare la response utilizzando il FutureBuilder Widget di Flutter

Pubblicare l'applicazione Flutter sull'Appstore e il Playstore

- Creazione di un'icona per la nostra app
- Aggiungere uno Splash Screen su Android
- Aggiungere una Launcher Image su IOS
- Registrazione dell'app Android e generazione della KeyStore e dell'AppBundle
- Pubblicazione dell'applicazione sul PlayStore
- Registrazione dell'app IOS e pubblicazione sull'AppStore

FIREBASE E DATABASE noSQL

Introduzione a FlutterFire per il backend

- Firestore Database
- Firebase Authentication
- Flutterfire UI
- Notifiche

MACHINE LEARNING E FIREBASE ML

I pacchetti indispensabili

- font_awesome_flutter
- image_picker
- flutter_form_builder
- google_maps_flutter
- Altri pacchetti concordati



+20 ore di VIDEO CORSI ON DEMAND

Algoritmi e strutture dati

- La programmazione del calcolatore: modalità logiche per la risoluzione di problemi software
- Differenza tra algoritmi e programmi
- I diagrammi di flusso (flow charts): traduzione visiva della logica per la risoluzione di un programma
- Tipi di dati: numerici, stringhe, booleani, array, oggetti
- La programmazione strutturata e le strutture dei dati
- La programmazione procedurale: risoluzione dei problemi "dal generale al particolare"
- Il concetto di funzione e il passaggio dei parametri
- Strutture dinamiche complesse: Alberi, Grafi e reti di dati complesse
- Dalla programmazione strutturata procedurale alla programmazione ad oggetti (OOP - Object Oriented Programming): cenni
- I "pattern" della programmazione nelle web application

GIT

- Introduzione al versioning: lavorare in team in una web application
- Software per installazione/ utilizzo di web application
- Composer e Vagrant
- Git Hub
- Introduzione a Git e ai sistemi di versioning
- Repository esterni (NPM - Node Package Manager)
- CLI (Command Line Interface)
- Installazione di GIT
- I progetti GIT
- Inizializzazione e configurazione di un progetto Git
- Clonare un progetto GIT ed aggiungere componenti
- Repository remoti: GIT HUB e suo utilizzo
- Branches: ramificare un progetto GIT
- Creare un nuovo branch
- Operazioni sui branches: navigazione, merge, commit



+20 ore di WORKSHOP IN DIRETTA

Intro a UX/UI Design

Teoria del colore

Marketing & Comunicazione Digitale

Trovare il lavoro è un lavoro: il Curriculum Vitae

Trovare il lavoro è un lavoro: il profilo LinkedIn

Trovare il lavoro è un lavoro: il colloquio di lavoro

Introduzione alla SEO

Il formato SVG: la grafica vettoriale per il web





Corso App Developer

210 ore

dal 17 ottobre 2022

In Live Streaming o Video On Demand
Part-time in orario serale

Per tutte le info



www.labfortraining.it



info@labfortraining.it



06.58205135



3519392372

