

# Università Politecnica delle Marche

Facoltà di Ingegneria

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea in Ingegneria Informatica e dell'Automazione

---



**Tesi di Laurea**

**Progettazione e implementazione di un'app Xamarin relativa  
all'Università Politecnica delle Marche**

**Design and implementation of a Xamarin app relative to the  
Polytechnic University of Marche**

Relatore

Prof. Domenico Ursino

Candidato

Davide Giannubilo

---

**Anno Accademico 2018-2019**



---

# Indice

<b>Introduzione</b>	3
<b>1 Introduzione a Xamarin</b>	7
1.1 Web app, ibride o native?	7
1.1.1 Web app	7
1.1.2 App ibride	7
1.1.3 App native	9
1.2 Cos'è Xamarin?	9
1.3 Com'è fatto Xamarin?	10
1.4 Xamarin.Forms o Xamarin.Android?	11
1.4.1 Xamarin.Forms	11
1.4.2 Xamarin.Android	12
1.5 Perché usare Xamarin?	12
<b>2 Analisi dei requisiti</b>	15
2.1 Descrizione della realtà	15
2.2 Requisiti funzionali e non funzionali	15
2.2.1 Requisiti funzionali	15
2.2.2 Requisiti non funzionali	16
2.2.3 Glossario dei termini	16
2.3 Use Case	17
2.3.1 Descrizione dei casi d'uso	17
<b>3 Progettazione della componente dati</b>	21
3.1 Progettazione concettuale	21
3.1.1 Schema E-R	21
3.1.2 Dizionari delle Entità e delle Relazioni	21
3.2 Progettazione logica	22
3.2.1 Ristrutturazione dello schema E-R	22
3.2.2 Traduzione dello schema E-R ristrutturato	24

<b>4</b>	<b>Progettazione della componente applicativa</b>	25
4.1	Architettura del sistema	25
4.2	Process Flow	26
4.2.1	Visualizzare un evento	26
4.2.2	Chiamata o invio di un'e-mail ad un docente	26
4.2.3	Visualizzare una planimetria/mappa	27
4.2.4	Visualizzare le informazioni di una segreteria	28
4.2.5	Visualizzare un'offerta formativa o un orario	29
4.3	Mockup	30
4.3.1	Mockup di livello 1	31
<b>5</b>	<b>Implementazione</b>	35
5.1	Strumenti e linguaggi utilizzati	35
5.1.1	Visual Studio	35
5.1.2	Android Device Manager	36
5.1.3	C#	36
5.1.4	XML	37
5.2	HomePage	37
5.2.1	Implementazione del layout della HomePage	37
5.2.2	Implementazione del codice C# della HomePage	39
5.3	Pagina relativa alle offerte formative e agli orari	44
5.3.1	Implementazione del layout delle pagine relative alle offerte formative e agli orari	44
5.3.2	Implementazione del codice C# della pagina relativa alle offerte formative e agli orari	56
5.4	Pagina relativa ai docenti	58
5.4.1	Implementazione del layout della pagina relativa ai docenti	58
5.4.2	Implementazione del codice C# della pagina relativa ai docenti	61
5.5	Pagina relativa alle mappe	64
5.5.1	Implementazione del layout della pagina relativa alle mappe	64
5.5.2	Implementazione del codice C# della pagina relativa alle mappe	65
5.6	Pagina relativa alle segreterie	67
5.6.1	Implementazione del layout della pagina relativa alle segreterie	67
5.6.2	Implementazione del codice C# della pagina relativa alle segreterie	70
5.7	Manuale del sistema	70
<b>6</b>	<b>Conclusioni</b>	77
	<b>Riferimenti bibliografici</b>	79

---

## Elenco delle figure

1.1	Com'è fatta una Web app	8
1.2	Esempio di fruibilità di un'app ibrida	8
1.3	Vantaggi di un'app nativa	9
1.4	Logo di Xamarin	10
1.5	Com'è fatto Xamarin	11
1.6	Xamarin.Forms	11
1.7	Xamarin.Android	12
2.1	Use Case di uno studente/utente generico	17
3.1	Schema E-R della realtà d'interesse	22
4.1	Architettura dell'app oggetto della presente tesi	25
4.2	Process Flow relativo alla visualizzazione degli eventi	26
4.3	Process Flow relativo alla possibilità di invio di una e-mail e di chiamata ad un docente	27
4.4	Process Flow relativo alla possibilità di visualizzare una planimetria o mappa di un edificio	28
4.5	Process Flow relativo alla possibilità di visualizzare le informazioni di una segreteria	29
4.6	Process Flow relativo alla possibilità di visualizzare un'offerta formativa o un orario	30
4.7	Mockup di livello 1 relativo alla "HomePage"	31
4.8	Mockup di livello 1 relativo al menù a scomparsa laterale	32
4.9	Mockup di livello 1 relativo alla pagina dei "Docenti"	32
4.10	Mockup di livello 1 relativo alla pagina delle "Offerte formative"	33
4.11	Mockup di livello 1 relativo alla pagina degli "Orari"	33
4.12	Mockup di livello 1 relativo alla pagina delle "Mappe"	34
4.13	Mockup di livello 1 relativo alla pagina delle "Segreterie"	34
5.1	Logo di Visual Studio	35
5.2	Schermata iniziale di Visual Studio	36
5.3	Home Page ed elenco degli eventi	71
5.4	Menù a scomparsa laterale	72

## VI      **Elenco delle figure**

5.5	Offerte Formative della Facoltà di Agraria . . . . .	72
5.6	Orari della Facoltà di Agraria . . . . .	73
5.7	Visualizzazione di un docente . . . . .	73
5.8	Chiamata ad un docente cercato . . . . .	74
5.9	Planimetrie della Facoltà di Economia . . . . .	74
5.10	Informazioni della segreteria della Facoltà di Agraria . . . . .	75
5.11	Invio di una e-mail all'indirizzo della segreteria . . . . .	75

---

## Elenco delle tabelle

2.1	Glossario dei termini relativo all'applicazione di interesse . . . . .	16
3.1	Dizionario delle Entità . . . . .	23
3.2	Dizionario delle Relazioni . . . . .	23
3.3	Dizionario dei Vincoli . . . . .	24





---

## Elenco dei listati

5.1	Layout della HomePage .....	38
5.2	Template <b>ListView</b> per gli eventi.....	39
5.3	Classe <b>Eventi</b> .....	40
5.4	Activity della HomePage .....	40
5.5	Classe <b>Database</b> .....	42
5.6	Classe <b>ListViewAdapter</b> .....	43
5.7	Layout della pagina relativa alle offerte formative della Facoltà di Agraria .....	44
5.8	Layout della pagina relativa agli orari della Facoltà di Agraria .....	50
5.9	Implementazione della <b>BottomNavigationView</b> per le facoltà.....	56
5.10	Implementazione della <b>BottomNavigationView</b> per le offerte formative e gli orari .....	56
5.11	Activity della pagina relativa alle offerte formative.....	57
5.12	Activity della pagina relativa agli orari .....	57
5.13	Layout della pagina relativa ai docenti .....	59
5.14	Template <b>ListView</b> per i docenti .....	60
5.15	Classe <b>Docenti</b> .....	61
5.16	Activity della pagina relativa ai docenti .....	61
5.17	Classe <b>Database</b> per i docenti.....	62
5.18	Classe <b>ListViewAdapterDoc</b> .....	63
5.19	Layout della pagina relativa alle planimetrie della Facoltà di Economia	64
5.20	Activity della pagina relativa alle mappe della Facoltà di Economia ..	65
5.21	Classe <b>ImageSliderEco</b> .....	66
5.22	Layout della pagina relativa alla segreteria della Facoltà di Agraria ..	67
5.23	Activity della pagina relativa alla segreteria della Facoltà di Agraria .	70



---

## Introduzione

Durante l'ultimo ventennio c'è stata una grande diffusione dei dispositivi mobili. Ciò è dovuto all'avanzamento tecnologico e, anche, all'esigenza di poter usufruire di servizi ed informazioni ovunque noi siamo, in qualsiasi momento della giornata.

A tal proposito, anche l'università è tenuta a rimanere al passo con i tempi e con le esigenze dei propri studenti, in modo tale da poter offrire loro un servizio adeguato, sfruttando servizi e funzionalità disponibili, sempre e ovunque, grazie ad un'app. Infatti, quest'ultima risponde all'esigenza di avere, a portata di "click", qualsiasi cosa l'utente voglia cercare o fare.

Le applicazioni mobile nascono negli anni '70, ma soltanto dal 2008 si assiste ad una radicale diffusione di esse. Ciò è dovuto al fatto che, a partire dagli anni 2000, sono migliorate drasticamente le caratteristiche hardware dei dispositivi mobili. Ciò è stato possibile grazie alla miniaturizzazione del transistor e al miglioramento dell'utilizzo del silicio, semiconduttore per eccellenza per le CPU.

Questa evoluzione è stata decisiva per favorire l'avvento dei dispositivi mobili come telefoni cellulari e tablet. Ultimi in ordine cronologico, sono gli *smartwatch*, ovvero orologi che implementano le stesse funzionalità di un telefono.

Ora, cosa ha spinto aziende come Apple, Google o Microsoft a scommettere in questo nuovo mercato, dato che loro avevano già le loro risorse inserite in altri mercati? La loro lungimiranza li ha portati ad investire in questo nuovo ambito per potersi assicurare una grossa fetta di esso e, di conseguenza, all'inizio c'è stato un vero e proprio "scontro" tra iOS mobile, Android e Windows Phone.

Tuttora sono i maggiori sistemi operativi diffusi per dispositivi mobili ma, di recente, molte altre case hanno lanciato i propri sistemi operativi, in particolare Mozilla con il suo FirefoxOS, Samsung con Tizen, Ubuntu con UbuntuOS mobile e, probabilmente, anche Huawei lascerà Android per usare un proprio sistema operativo. Si è passati, dunque, dall'avere un numero preciso e ridotto di piattaforme, ad una quantità di sistemi operativi che sembra destinata a crescere ulteriormente.

Ad oggi, quindi, il mobile racchiude tutti quei servizi che, fino a poco tempo fa, potevano essere offerti solo da un PC: navigazione Internet, e-mail, messaggistica istantanea, etc.

L'utente cerca di avere le informazioni richieste nel modo più semplice e veloce possibile; per la semplicità sarebbe meglio un sito web, invece, per la velocità, meglio un'app.

A tal proposito, possiamo suddividere le app mobile in tre tipologie, in base alle tecnologie usate. La prima è quella relativa alle app native, cioè sviluppate con codice e librerie proprietari. La seconda è rappresentata dalle web app, ovvero normali applicazioni web che permettono di simulare l'aspetto nativo, utilizzando le funzionalità tipiche delle tecnologie web. L'anello mancante tra le due tipologie di app appena descritte è l'app ibrida, che ingloba in sé gli aspetti positivi delle altre due tipologie, cercando di limitare al massimo gli aspetti negativi.

Per l'implementazione dell'app oggetto della presente tesi, si è scelto di utilizzare il framework Xamarin. Si tratta di un ambiente di sviluppo innovativo, in quanto riesce a trovare il miglior compromesso tra i due principali metodi di sviluppo per dispositivi mobili, ovvero quello ibrido e quello nativo. In particolare, Xamarin riesce a garantire la condivisibilità del codice tra diverse piattaforme e, contemporaneamente, permette di avere performance elevate, tipiche di un'applicazione sviluppata nativamente.

Xamarin permette di sviluppare applicazioni mobile mediante l'utilizzo dell'IDE (Integrated Development Environment) Visual Studio e un codice condiviso su diverse piattaforme basato su C#. Questo grazie a Microsoft che, poco dopo la nascita di Xamarin, lo ha acquistato e reso open-source.

Come accennato in precedenza, l'obiettivo della presente tesi è quello di realizzare, mediante Xamarin, un'app mobile, per Android, a supporto degli studenti dell'Università Politecnica delle Marche.

Nello specifico, l'app ha l'obiettivo di mostrare all'utente le informazioni più utili e ricercate. In particolare, sarà presente la possibilità di visualizzare l'elenco degli eventi dell'Ateneo, l'offerta didattica erogata e l'orario delle lezioni di ogni corso di laurea di ogni facoltà, le informazioni relative ad un docente, le informazioni delle segreterie e le planimetrie delle varie facoltà. A queste, si affiancano dei servizi aggiuntivi, quali la possibilità di contattare un docente o di inviare una e-mail alla segreteria.

Il presente elaborato approfondisce tutti questi aspetti. Esso è strutturato come di seguito specificato:

- Nel Capitolo 1 viene presentata un'introduzione a Xamarin, specificando la differenza tra le tre tipologie di app; inoltre, vengono trattate anche tutte le caratteristiche del framework, mettendo in evidenza pregi e difetti di esso.
- Nel Capitolo 2 viene fatta l'analisi dei requisiti dell'app; in primo luogo, viene analizzata la realtà di riferimento e, in seguito, vengono trattate nel dettaglio le varie funzionalità che dovranno essere garantite da essa.
- Nel Capitolo 3 viene presentata la progettazione della componente dati, divisa in due sezioni, una per la progettazione concettuale e una per la progettazione logica. Nella progettazione concettuale, viene fatta una descrizione della realtà ad un alto livello di astrazione, mentre, nella progettazione logica viene fatta una descrizione più vicina al livello macchina, in modo tale da ottenere il modello relazionale che poi verrà usato dagli RDBMS (Relational Database Management System) per la gestione dei dati.
- Il Capitolo 4 tratta la progettazione della componente applicativa, attraverso la definizione dell'architettura di sistema, i *process flow* e i *mockup* di livello 1.
- Il Capitolo 5 illustra l'implementazione dell'app, analizzando e spiegando le varie porzioni di codice che implementano le varie funzionalità.

- Infine, nel Capitolo 6, vengono tratte le conclusioni del progetto e viene dato uno sguardo ai possibili sviluppi futuri.



## Introduzione a Xamarin

*In questo capitolo si introduce il framework Xamarin, mettendo in risalto le differenze tra le varie tipologie di app mobile e illustrando i pregi e le caratteristiche di esso.*

### 1.1 Web app, ibride o native?

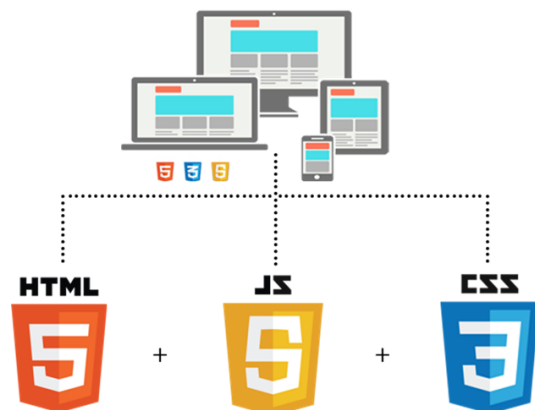
#### 1.1.1 Web app

Una *Web application* è un'applicazione o sito web che gira su un server web e che viene utilizzata attraverso un browser web, a differenza una tradizionale applicazione desktop (Figura 1.1). Ciò sta ad indicare che è possibile utilizzare una Web app da qualsiasi computer o dispositivo mobile: è sufficiente che quest'ultimo abbia un browser installato ed una connessione Internet. Di conseguenza, una Web app non necessita di alcuna installazione e di nessun tempo di download. Altra caratteristica è quella di essere “responsive” cioè, la capacità di adattarsi alla dimensione dello schermo del dispositivo su cui viene avviata.

Questo tipo di app presenta anche alcuni svantaggi. Il primo è quello legato al fatto che interagendo con un server è molto facile incappare in tempi di attesa più lunghi rispetto alle altre app; in alcuni casi, viene, inoltre, sovraccaricato il server. Il secondo è legato al fatto che, per farla funzionare, è necessario avere accesso ad Internet, cosa che non accade con altri tipi di app; infine, il terzo riguarda le API native dei vari sistemi operativi; queste, infatti, non vengono sfruttate, ma ci si limita ad utilizzare solo quelle disponibili da i vari browser.

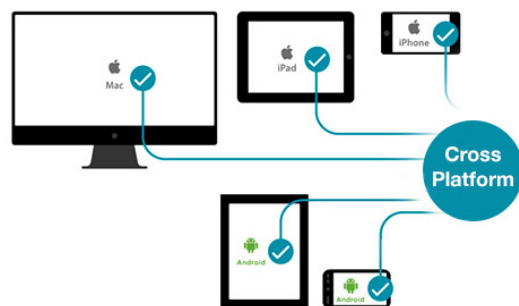
#### 1.1.2 App ibride

Un'*app ibrida* è un'app cross-platform fruibile su diversi dispositivi che permette la coesistenza ed integrazione di funzionalità native con caratteristiche proprie delle web application. I più semplici esempi di app ibride consistono nell'uso di un componente nativo in grado di visualizzare pagine web (ad esempio, nel caso di Android, una WebView), e di una web application che viene automaticamente visualizzata



**Figura 1.1.** Com'è fatta una Web app

tramite tale componente nativo. La web application può essere sia embedded (ovvero inclusa nell'app, in forma di sito statico e in locale), oppure remota, cioè installata su un web server, ed accessibile solo mediante una connessione web (Figura 1.2). Quest'ultimo caso permette, inoltre, di visualizzare la stessa web application di volta in volta su piattaforme diverse, dal momento che un'app “contenitore” (sia essa su Android o su iOS) può accedervi facilmente via HTTP.



**Figura 1.2.** Esempio di fruibilità di un'app ibrida

L'approccio ibrido ha il vantaggio di limitare al minimo la necessità di manutenzione dei componenti nativi (dal momento che tutta la logica è implementata nella web application), mentre il grosso della manutenzione è, idealmente, unico per entrambe le piattaforme. Inoltre, tale approccio facilita le fasi di implementazione, manutenzione e riusabilità del codice dal momento che gli sviluppatori potranno trascurare la complessità dei vari SDK e concentrarsi solo sulla logica di business.

D'altro canto, le performance di un'applicazione ibrida non possono essere paragonate a quelle di un'app nativa, poiché quest'ultima non richiede l'esecuzione

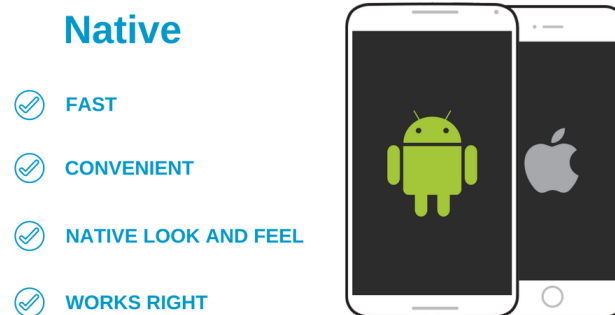


di alcun interprete JavaScript o di alcuni motori di rendering HTML/CSS, che invece rappresentano i middleware aggiuntivi (il cui onere computazionale non è trascurabile), incorporati in qualunque componente per la visualizzazione di pagine web.

### 1.1.3 App native

Un'app nativa è un'app mobile che viene sviluppata usando i vari SDK e framework offerti dai produttori stessi del sistema operativo per cui deve operare. Se decidiamo di creare un'app per Android, potremo utilizzare Java o Kotlin, ed i relativi SDK. Allo stesso modo, i linguaggi Swift ed Objective-C ci permetteranno di creare app per iOS, mentre, i linguaggi C# e XAML ci consentiranno di realizzare app per Windows Phone. In questi casi, quello che facciamo solitamente è realizzare un'app nativa, che sfrutta cioè le caratteristiche messe a disposizione direttamente dagli SDK, potendo, quindi, interfacciarsi con le varie funzionalità del sistema operativo ospitante.

Un'app nativa, generalmente, ha il vantaggio di poter accedere direttamente alle funzionalità del sistema operativo, e non necessita di livelli di astrazione o middleware aggiuntivi (Figura 1.3). Per questo motivo, tale soluzione è preferita da molti sviluppatori che abbiano a cuore le performance dell'app.



**Figura 1.3.** Vantaggi di un'app nativa

Uno svantaggio è quello dovuto ai dispendiosi costi che un'azienda deve sostenere per avere due team di sviluppatori per la creazione di app Android ed iOS; un altro è quello dovuto al fatto di non poter riutilizzare il codice visto che le due versioni sono completamente diverse.

## 1.2 Cos'è Xamarin?

*Xamarin* (Figura 1.4) nato nel Maggio 2011, era una società informatica creata da N. Friedman e M. de Icaza (noto per aver coordinato la realizzazione di *Mono*,

framework open source per l'implementazione del .NET Framework di Microsoft su Linux).

L'idea di Xamarin è quella di offrire agli sviluppatori un modo semplice e veloce per creare app cross-platform basate su un unico linguaggio orientato agli oggetti attraverso ambienti di sviluppo prestanti. Per raggiungere tale obiettivo, è stato creato il framework Xamarin, basato su Mono e quindi su C#.



**Figura 1.4.** Logo di Xamarin

Nato come progetto indipendente, nel corso degli anni è stato oggetto d'interesse per gli sviluppatori, ma anche per le aziende, tra cui Microsoft, tanto che, nel 2016, quest'ultima ha acquisito Xamarin dando inizio ad una nuova era per lo sviluppo di app cross-platform. A partire da Marzo 2016, Xamarin è, inoltre, ufficialmente open source e gratuito per sviluppatori indipendenti e per i piccoli team che possono utilizzarlo nei propri progetti.

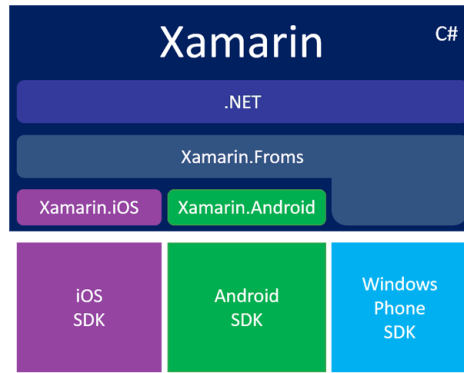
### 1.3 Com'è fatto Xamarin?

*Xamarin* è un framework per lo sviluppo di app native e cross-platform con C#. Esso si basa sul progetto open source Mono offrendo pieno supporto non solo alle piattaforme Android e iOS ma anche a Windows Phone.

Come mostrato nella Figura 1.5, Xamarin è costituito da tre componenti principali:

- *Xamarin.Android*, basato su Mono, permette lo sviluppo di app Android tramite un wrapping delle API native in C#;
- *Xamarin.iOS*, basato sulla libreria proprietaria MonoTouch, offre pieno supporto allo sviluppo di app iOS tramite un wrapping delle API native in C#;
- *Xamarin.Forms*, al di sopra dei precedenti componenti, permette la piena condivisione del codice tra i tre OS.

Grazie a queste componenti è possibile gestire in C# tutte le caratteristiche di Android, iOS e Windows Phone, dall'interfaccia utente alle risorse hardware del dispositivo.



**Figura 1.5.** Com'è fatto Xamarin

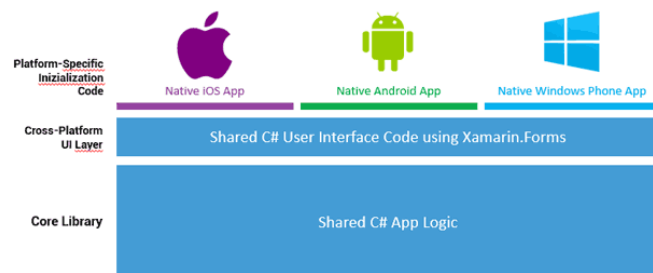
## 1.4 Xamarin.Forms o Xamarin.Android?

### 1.4.1 Xamarin.Forms

*Xamarin.Forms* è uno dei principali benefici di Xamarin, visto che permette la realizzazione di app cross-platform.

Come si evince dalla Figura 1.6, *Forms* è composto da tre strati:

- *Core Library*; che offre la logica dell'applicazione e di accesso ai dati definita in C#;
- *Cross-Platform UI Layer*; è lo strato contenente tutta la UI realizzata con Xamarin.Forms definita in C#;
- *Platform-Specific Initialization Code*; ha una piccola porzione di codice in C# nei progetti di ogni piattaforma ed è necessario per inizializzare ed eseguire l'app su ogni OS nativo.



**Figura 1.6.** Xamarin.Forms

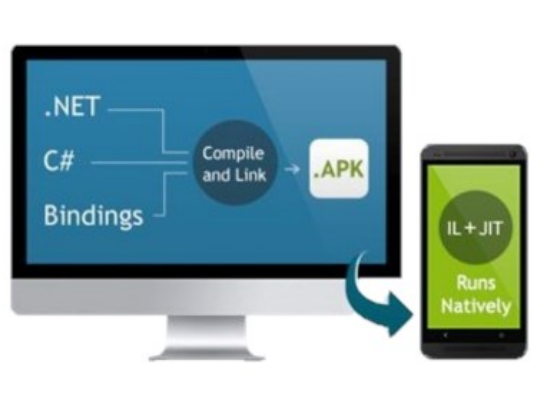
Il *vantaggio* di *Xamarin.Forms* è quello di utilizzare un solo linguaggio per poi sfruttare l'app su dispositivi con diversi sistemi operativi e garantire, quindi, quasi

le stesse prestazioni di app native. Ciò è stato possibile creando dei *wrapper* (ovvero dei particolari traduttori di linguaggio), i quali, in buona sostanza, fanno sì che il codice scritto in C# sia tradotto in codice Java (per Android) o Objective-C (per iOS) al momento della compilazione, distribuendo, così, un'applicazione che, agli occhi del dispositivo dove viene eseguita, è scritta nel linguaggio presunto per esso.

Lo *svantaggio* è quello di non poter sfruttare a pieno le API native dei vari sistemi operativi; ciò rende Xamarin Forms inutilizzabile quando c'è bisogno di un'app che sfrutti una particolare funzionalità di un determinato sistema.

### 1.4.2 Xamarin.Android

*Xamarin.Android* (Figura 1.7) consente di creare applicazioni Android native con la stessa efficienza di un'applicazione scritta in Java e con la possibilità di utilizzare tutti i controlli nativi dell'interfaccia grafica.



**Figura 1.7.** Xamarin.Android

Lo svantaggio nell'utilizzo di Xamarin.Android sta nello sviluppo dell'interfaccia grafica XML per Android. Nonostante ciò, poichè la logica di business è la stessa ed è implementata utilizzando C#, si riesce a riciclare circa il 60% del codice; tale percentuale aumenta nelle applicazioni con interfacce grafiche meno elaborate.

## 1.5 Perchè usare Xamarin?

Sia lo sviluppo nativo che quello ibrido presentano diversi vantaggi; Xamarin cerca di trarre il meglio da entrambe le soluzioni.

A differenza di quanto permettono di fare altre soluzioni di sviluppo multiplatforma (PhoneGap e simili), che utilizzano come linguaggio HTML, CSS e Javascript ed eseguono il codice in un controllo WebView, *Xamarin* permette di sviluppare applicazioni native utilizzando come linguaggio di sviluppo C# e l'IDE Visual Studio.

Un vantaggio da non sottovalutare avendo a disposizione app native (oltre, naturalmente, alle performance, all'utilizzo di memoria e all'integrazione con il sistema) è quello che la nostra app utilizzerà la UI della piattaforma su cui gira; quindi uno stesso controllo sarà renderizzato in modo coerente con la piattaforma su cui l'app sta girando.

Xamarin permette, anche, di sviluppare app per Android Wear, Android Tv ed Apple Watch permettendo agli sviluppatori di essere produttivi praticamente su tutte le piattaforme.



## Analisi dei requisiti

*In questo capitolo verranno analizzati i requisiti necessari alla realizzazione di un'app Android per l'Università Politecnica delle Marche. Verranno, inoltre, illustrate le funzionalità che saranno garantite all'utente.*

### 2.1 Descrizione della realtà

Il lavoro di questa tesi propone la realizzazione di un'app Android per gli studenti dell'Università Politecnica delle Marche mediante l'utilizzo di Xamarin.

L'applicazione garantirà le seguenti funzionalità:

- visualizzare gli eventi dell'ateneo;
- consultare le offerte formative;
- consultare gli orari;
- visualizzare i vari contatti delle segreterie di ogni facoltà;
- cercare un docente per visualizzare i suoi contatti;
- consultare le planimetrie dell'ateneo.

Per la visualizzazione degli eventi e dei docenti verrà effettuata un'operazione di estrapolazione di dati da un database, gestito mediante il DBMS SQLite.

### 2.2 Requisiti funzionali e non funzionali

Un requisito è una funzionalità che deve essere implementata dall'applicazione. I requisiti si dividono in requisiti funzionali e requisiti non funzionali.

#### 2.2.1 Requisiti funzionali

L'applicazione dovrà rispettare i seguenti requisiti funzionali:

- *RF1* - L'applicazione dovrà permettere l'attività di read sugli eventi dell'ateneo.
- *RF2* - L'applicazione dovrà permettere l'attività di read sui contatti dei docenti.

- *RF3* - L'applicazione dovrà permettere la chiamata rapida dai contatti dei docenti.
- *RF4* - L'applicazione dovrà permettere l'invio di e-mail al docente.
- *RF5* - L'applicazione dovrà permettere la chiamata rapida dai contatti delle segreterie.
- *RF6* - L'applicazione dovrà permettere l'invio di e-mail alla segreteria.
- *RF7* - L'applicazione dovrà permettere la visualizzazione delle planimetrie, ove presenti, dei vari edifici dell'ateneo, suddivisi per facoltà.
- *RF8* - L'applicazione dovrà permettere la visualizzazione delle offerte formative dei vari corsi di laurea di ogni facoltà.
- *RF9* - L'applicazione dovrà permettere la visualizzazione degli orari dei vari corsi di laurea di ogni facoltà.

### 2.2.2 Requisiti non funzionali

I requisiti non funzionali da realizzare sono i seguenti:

- *RNF1* - L'applicazione dovrà essere sviluppata mediante l'utilizzo di Xamarin.
- *RNF2* - L'applicazione dovrà essere implementata utilizzando C# e XML.

### 2.2.3 Glossario dei termini

Il glossario dei termini (Tabella 2.1) descrive il significato dei principali termini utilizzati nel progetto.

TERMINI	DESCRIZIONE
Utente	Studente, o utente interessato, che utilizzerà l'applicazione.
Docente	Professore appartenente all'ateneo.
Evento	Avvenimento organizzato dall'ateneo.
Planimetria	Mappa dettagliata di ogni piano su cui sono indicate le aule.
Offerta formativa	Elenco dettagliato dei vari esami da sostenere negli anni di studio.
Orario	File contenente l'orario dettagliato di un semestre di un determinato corso di laurea.
Segreteria	Luogo in cui lo studente, o una persona interessata a chiedere informazioni sull'ateneo, può rivolgersi in caso di necessità

**Tabella 2.1.** Glossario dei termini relativo all'applicazione di interesse

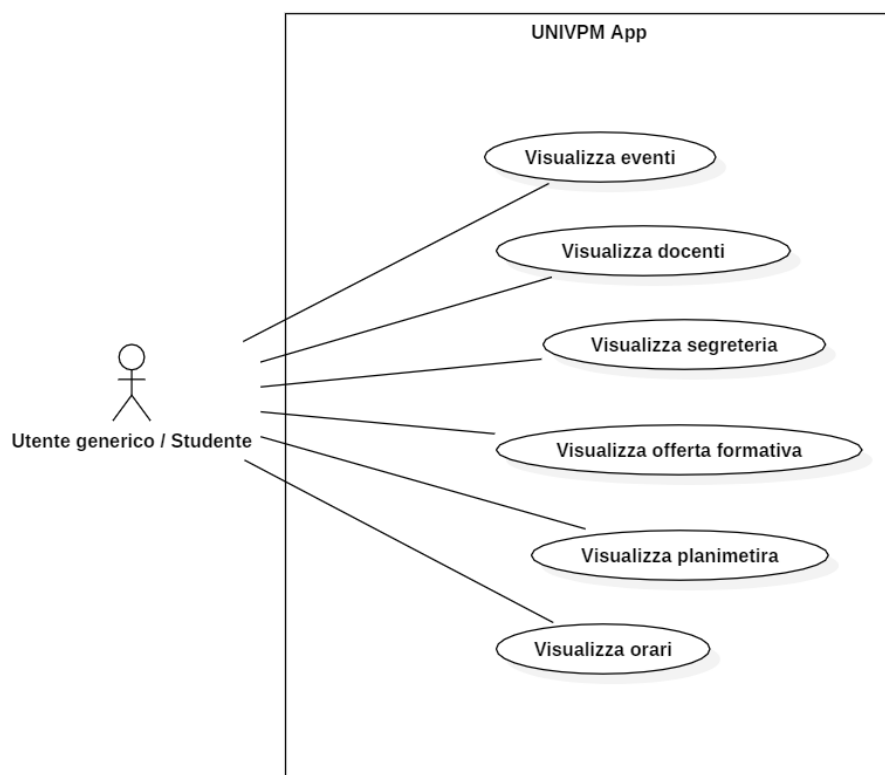


## 2.3 Use Case

Un caso d'uso è un'azione che un attore può fare tramite l'applicazione. L'attore può essere uno studente o un utente generico che sta cercando informazioni al suo interno.

Il diagramma dei casi d'uso descrive le funzionalità offerte dall'applicazione e come gli attori interagiscono con esse.

In Figura 2.1 viene illustrato il caso d'uso relativo ad un utente generico ed ad uno studente, dato che possono effettuare le stesse azioni.



**Figura 2.1.** Use Case di uno studente/utente generico

Come si evince dalla figura, un attore può visualizzare gli eventi, cercare e visualizzare i contatti dei docenti, visualizzare gli orari, le offerte formative, i contatti delle segreterie e le planimetrie degli edifici delle facoltà.

### 2.3.1 Descrizione dei casi d'uso

In questa sezione verranno spiegate in dettaglio tutte le azioni previste dai casi d'uso menzionati in Figura 2.1.

### **CU1 - Visualizza eventi**

Permette all'utente di visualizzare, sulla sua HomePage, gli eventi, in programma, e quelli passati, relativi all'ateneo, con tutte le informazioni necessarie per sapere dove e quando si svolgeranno.

### **CU2 - Visualizza docenti**

Permette all'attore di cercare un docente; se quest'ultimo è presente verranno visualizzati sullo schermo i suoi contatti con la possibilità di effettuare una chiamata e/o di inviargli una e-mail.

Il caso d'uso inizia quando l'attore apre il menù a scomparsa laterale e clicca sulla voce "Docenti". Una volta aperta la pagina, l'attore potrà scrivere nell'apposita barra il nome del docente. Se quest'ultimo è presente verranno visualizzate sullo schermo tutte le informazioni a lui relative.

### **CU3 - Visualizza segreteria**

Permette all'attore di cercare la segreteria della facoltà desiderata.

Il caso d'uso inizia quando l'attore apre il menù a scomparsa laterale e clicca sulla voce "Segreteria". A questo punto potrà scegliere la facoltà di riferimento. A seguito di ciò verranno visualizzate tutte le informazioni ad essa relativa, come, per esempio, l'indirizzo e-mail e/o gli orari di apertura.

### **CU4 - Visualizza offerta formativa**

Permette all'attore di visualizzare un elenco con tutte le offerte formative suddivise per facoltà e per anno di immatricolazione.

Il caso d'uso inizia quando l'attore apre il menù a scomparsa laterale e clicca sulla voce "Corsi di laurea". A quel punto potrà scegliere la facoltà. Una volta effettuata tale scelta, sullo schermo saranno visualizzate, tramite tabella, tutte le offerte formative, relative alla facoltà prescelta, triennali e magistrali.

### **CU5 - Visualizza planimetria**

Permette all'attore di visualizzare tutte le planimetrie, ove disponibili, degli edifici delle varie facoltà.

Il caso d'uso inizia quando l'attore apre il menù a scomparsa laterale e clicca sulla voce "Mappe". A questo punto potrà scegliere la facoltà. A seguito di ciò, in alcuni casi verrà visualizzata una tabella dove si possono aprire i vari pdf delle planimetrie; in altri casi, verrà presentato uno slider di immagini.

### CU6 - Visualizza orari

Permette all'attore di visualizzare un elenco con tutti gli orari suddivisi per facoltà, per semestre e per anno di corso (ad esempio, Primo Anno Primo Semestre).

Il caso d'uso inizia quando l'attore apre il menù a scomparsa laterale e clicca sulla voce "Corsi di laurea". A quel punto potrà scegliere la facoltà. Cliccando successivamente "Orari" saranno visualizzati sullo schermo, tramite tabella, tutti gli orari relativi ai corsi di studio attivi presso la Facoltà selezionata.



## Progettazione della componente dati

*Questo capitolo tratta la fase di progettazione della componente dati. Esso si divide in due parti, ovvero progettazione concettuale e progettazione logica.*

### 3.1 Progettazione concettuale

La progettazione concettuale rappresenta i dati della realtà di interesse in un modello concettuale, sotto forma di concetti e di relazioni tra questi ultimi. In questa fase individuiamo le entità della realtà di riferimento e le relazioni che esistono tra di esse.

Un'entità è una classe o un concetto con un'esistenza autonoma.

Un'istanza di un'entità è un oggetto della classe rappresentata dall'entità.

Lo scopo della progettazione concettuale è quello di effettuare una descrizione formale della realtà; uno schema concettuale è un modello più vicino all'uomo che alla macchina. Quello utilizzato nella presente tesi è il modello Entità-Relazione (detto anche modello E-R o schema E-R).

#### 3.1.1 Schema E-R

Lo schema E-R rappresenta il contenuto informativo di una base di dati. Tale diagramma può essere corredato da una documentazione di supporto in modo tale da rendere la progettazione concettuale più leggibile.

Una descrizione più approfondita verrà fornita attraverso il *Dizionario delle Entità* e il *Dizionario delle Relazioni*.

In Figura 3.1 viene mostrato il diagramma E-R della realtà di interesse.

#### 3.1.2 Dizionari delle Entità e delle Relazioni

I dizionari ci permettono di approfondire i concetti rappresentati dallo schema E-R. Nella Tabella 3.1 viene rappresentato il *Dizionario delle Entità* relativo allo schema E-R, mentre nella Tabella 3.2 viene evidenziato il *Dizionario delle Relazioni*. Infine, nella Tabella 3.3, viene rappresentato il *Dizionario dei Vincoli*.

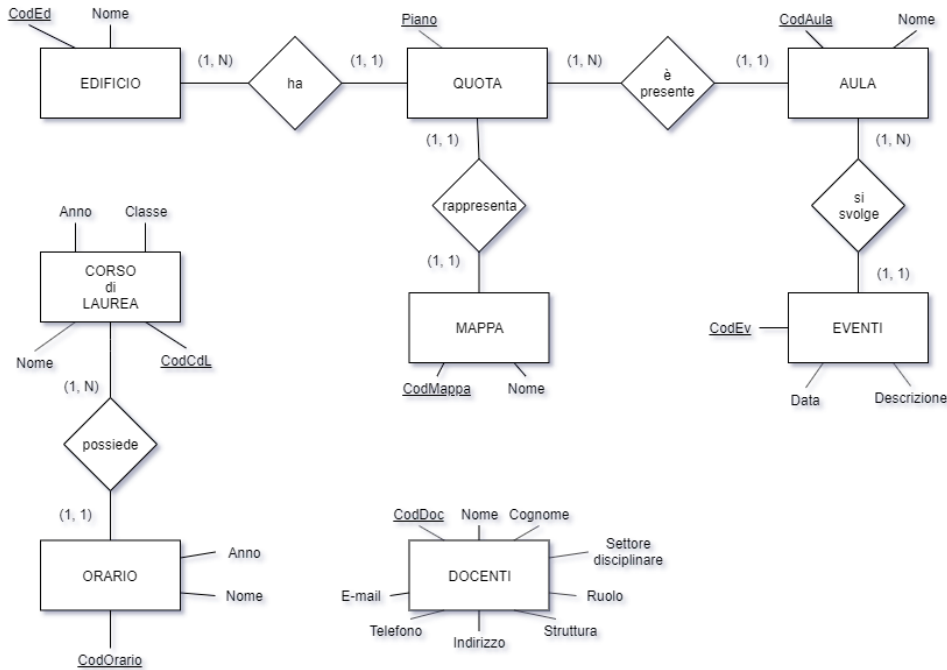


Figura 3.1. Schema E-R della realtà d'interesse

## 3.2 Progettazione logica

La seconda fase della progettazione della componente dati riguarda la progettazione logica. Durante questa fase viene effettuata la traduzione dallo schema concettuale allo schema logico.

Il modello logico ha lo scopo di rappresentare le informazioni contenute nel modello concettuale in un formato più vicino alla macchina.

La progettazione logica si suddivide in due fasi, ovvero la ristrutturazione dello schema E-R e la traduzione dello schema E-R ristrutturato nel modello relazionale.

### 3.2.1 Ristrutturazione dello schema E-R

La ristrutturazione ha l'obiettivo di eliminare le generalizzazioni e gli attributi multivalore. I passi fondamentali della ristrutturazione sono i seguenti:

- *Analisi delle ridondanze*; nello schema E-R della presente tesi non sono stati rilevati attributi che derivano da altri della stessa entità o da altre entità, per cui possiamo affermare che non sono presenti ridondanze.
- *Eliminazione delle generalizzazioni*; nello schema E-R in esame non sono presenti generalizzazioni da eliminare.
- *Partizionamento/Accorpamento di entità*; nello schema E-R in esame non ci sono entità che necessitano di essere partizionate o accorpate.

ENTITÀ	DESCRIZIONE	ATTRIBUTI	ID
Aula	Un'aula dell'ateneo	CodAula, Nome	CodAula
Corso di laurea	Corsi di studi dell'ateneo	CodCdL, Nome, Anno, Classe	CodCdL
Edificio	Un edificio dell'ateneo	CodEd, Nome	CodEd
Evento	Un evento dell'ateneo	CodEv, Nome, Descrizione	CodEv
Orario	Orario di un corso di laurea	CodOrario, Nome, Anno	CodOrario
Mappa	Una planimetria di una quota	CodMappa, Nome	CodMappa
Quota	Un piano di un edificio dell'ateneo	Piano	Piano
Docente	Un professore dell'ateneo	CodDoc, Nome, Cognome, Settore disciplinare, Ruolo, Struttura, Indirizzo, Telefono, E-mail	CodDoc

Tabella 3.1. Dizionario delle Entità

RELAZIONE	DESCRIZIONE	ENTITÀ	ATTRIBUTI
Ha	Permette di visualizzare i piani di un edificio	Edificio - Quota	-
Possiede	Estrae gli orari di un corso di laurea	Orario - CdL	-
Rappresenta	Estrae la mappa di una quota	Mappa - Quota	-
È presente	Permette di visualizzare le aule di un piano (quota)	Quota - Aula	-
Si svolge	Permette di visualizzare tutti gli eventi che si svolgono in quell'aula	Eventi - Aula	-

Tabella 3.2. Dizionario delle Relazioni

- *Eliminazione degli attributi composti*; nello schema E-R in esame non sono presenti attributi composti.
- *Eliminazione degli attributi multivalore*; nello schema E-R in esame non sono state utilizzate cardinalità per gli attributi.
- *Scelta degli identificatori primari*; come identificatori primari sono stati scelti codici alfanumerici o solo numerici per tutte le entità.

ANNOTAZIONE	REGOLA
V1	Tutti gli attributi chiave devono essere distinti
V2	Il campo “Anno” deve avere un valore maggiore di 0
V3	Il campo “Quota” deve avere un valore maggiore di 0
V4	Il campo “Data” deve essere di tipo <i>date</i>

**Tabella 3.3.** Dizionario dei Vincoli

### 3.2.2 Traduzione dello schema E-R ristrutturato

Poiché non è stato necessario effettuare alcuna ristrutturazione, per la traduzione verso il modello relazionale verrà utilizzato lo schema E-R ricavato al termine della progettazione concettuale. A questo punto, occorre applicare le regole della teoria dei database per ottenere lo schema relazionale.

Tutte le associazioni sono di tipo 1-N eccetto “rappresenta” che è di tipo 1-1; quindi, non c’è bisogno di creare ulteriori tabelle per tradurre le relazioni dello schema E-R.

A questo punto otteniamo il seguente modello relazionale:

- CORSO DI LAUREA(CodCdL, Nome, Anno, Classe)
- ORARIO(CodOrario, Nome, Anno) + (CodCdl) chiave esterna
- EDIFICIO(CodEd, Nome)
- QUOTA(Piano) + (CodEd) chiave esterna
- MAPPA(CodMappa, Nome) + (Piano) chiave esterna
- AULA(CodAula, Nome) + (Piano) chiave esterna
- EVENTI(CodEv, Data, Descrizione) + (CodAula) chiave esterna
- DOCENTI(CodDoc, Nome, Cognome, Settore disciplinare, Ruolo, Struttura, Indirizzo, Telefono, E-mail)



## Progettazione della componente applicativa

*La progettazione della componente applicativa ha lo scopo di progettare le funzionalità software offerte dall'applicazione. Questa fase viene suddivisa in tre passi, ovvero la progettazione dell'architettura del sistema, la progettazione dei mockup e il process flow.*

### 4.1 Architettura del sistema

L'*architettura del sistema* specifica come sono organizzate le informazioni all'interno dell'applicazione e, quindi, quali sono gli "step" per raggiungere una determinata pagina. In Figura 4.1 viene rappresentata l'architettura dell'applicazione oggetto della presente tesi.

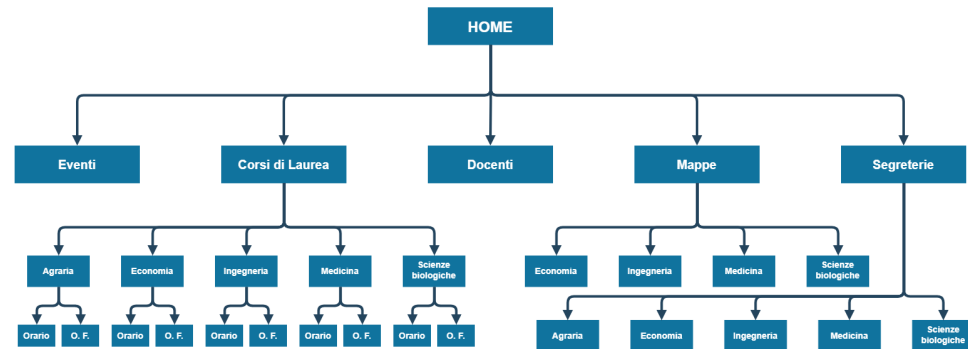


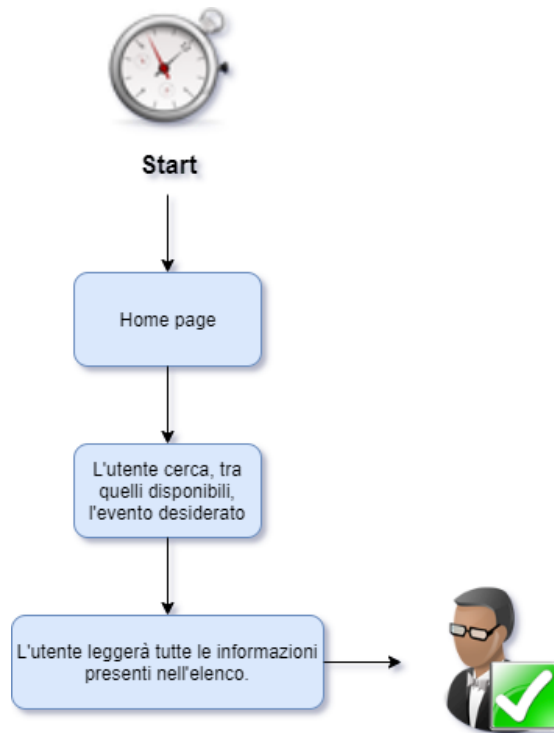
Figura 4.1. Architettura dell'app oggetto della presente tesi

## 4.2 Process Flow

Chiamato anche *Flow Chart*, un *Process Flow* raffigura gli “step” che l’attore deve seguire per usufruire di una funzionalità. Le funzionalità del sistema sono quelle definite dagli Use Case.

### 4.2.1 Visualizzare un evento

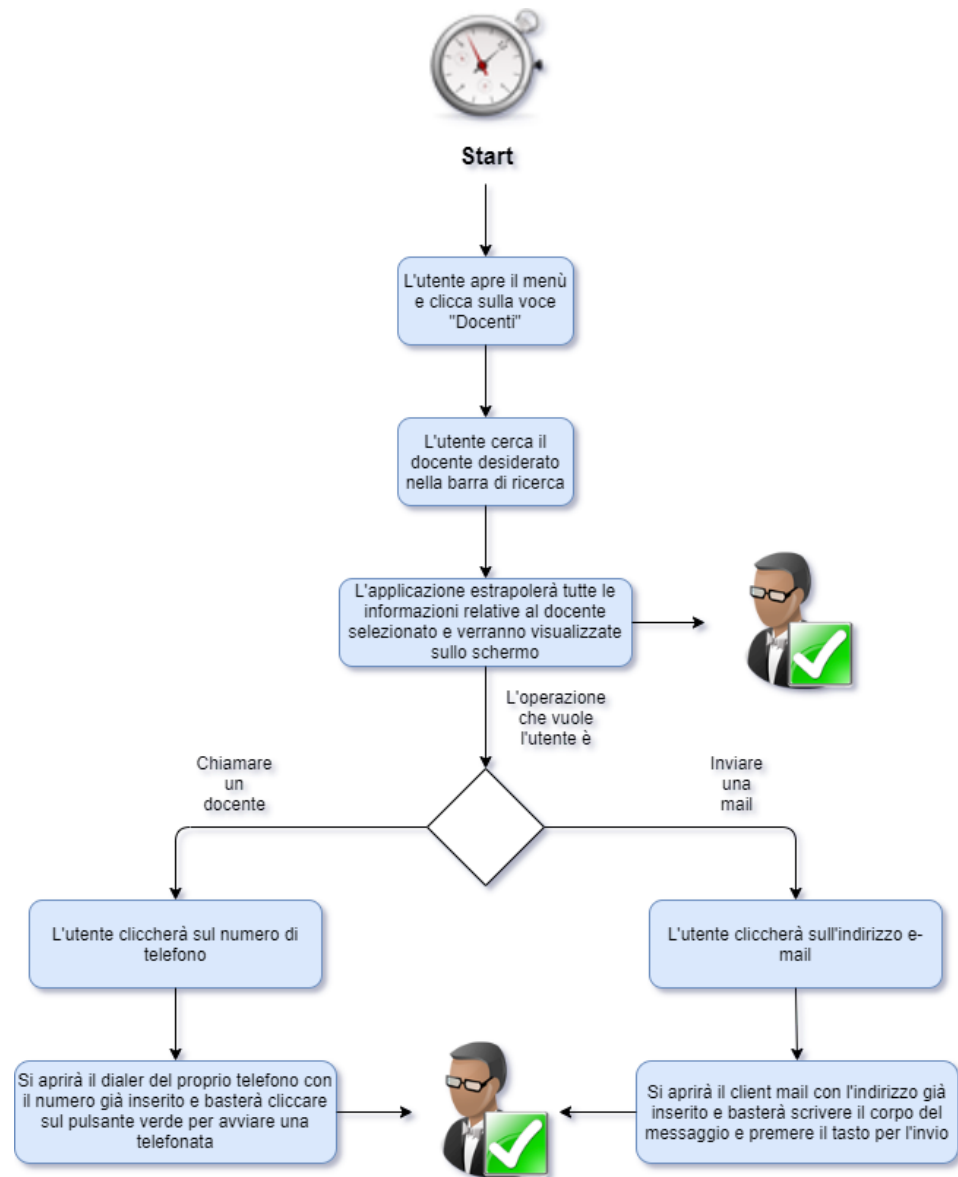
L’attore ha la possibilità di visualizzare un elenco di tutti gli eventi in programma e di tutti quelli passati. Saranno presenti tutte le informazioni utili per sapere dove e quando si è svolto e/o si svolgerà un evento. In Figura 4.2 possiamo vedere il *Process Flow* relativo a questa attività.



**Figura 4.2.** Process Flow relativo alla visualizzazione degli eventi

### 4.2.2 Chiamata o invio di un’e-mail ad un docente

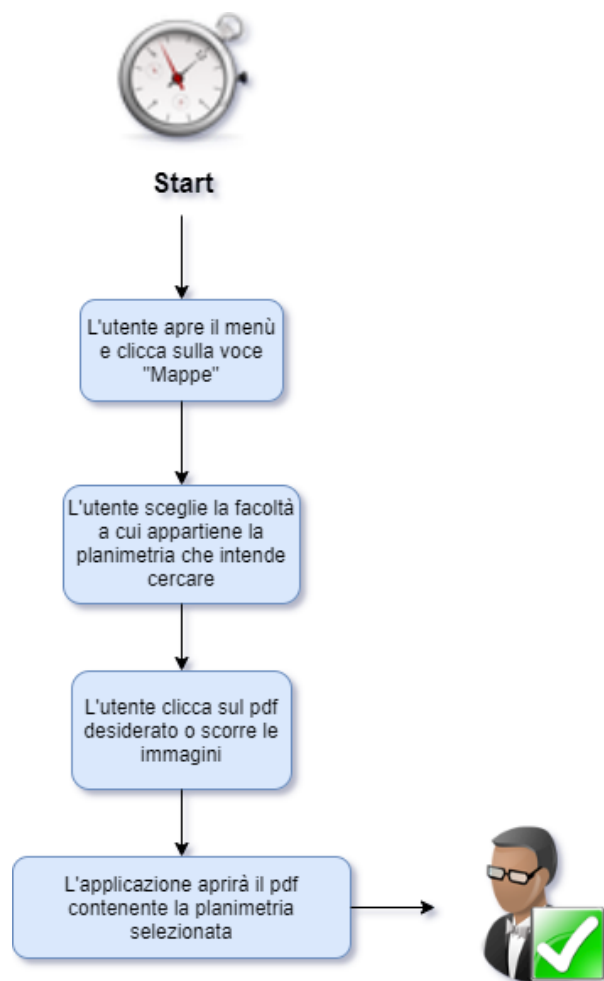
L’attore ha la possibilità di cercare un docente e, nel caso in cui il docente specificato è effettivamente presente nel database, di visualizzare tutte le informazioni a lui relative, in modo tale che sia possibile inviargli una e-mail o effettuare una chiamata. In Figura 4.3 possiamo vedere il *Process Flow* relativo a questa attività.



**Figura 4.3.** Process Flow relativo alla possibilità di invio di una e-mail e di chiamata ad un docente

### 4.2.3 Visualizzare una planimetria/mappa

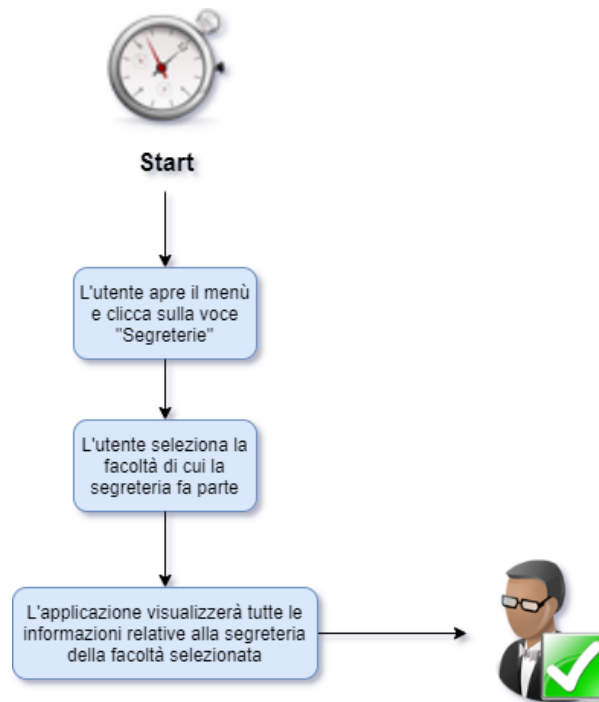
L'attore ha la possibilità di scegliere la facoltà e, poi, di visualizzare la planimetria o la mappa dell'edificio di suo interesse, se presente. In Figura 4.4 possiamo vedere il *Process Flow* relativo a questa attività.



**Figura 4.4.** Process Flow relativo alla possibilità di visualizzare una planimentria o mappa di un edificio

#### 4.2.4 Visualizzare le informazioni di una segreteria

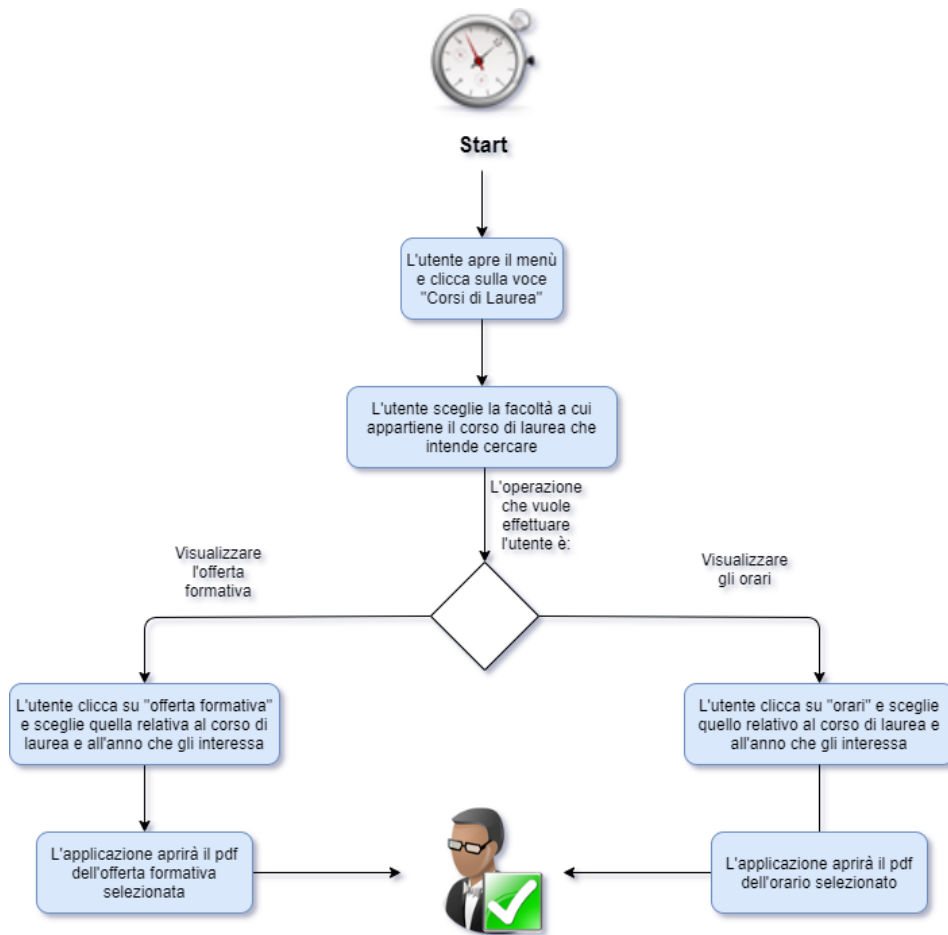
L'attore ha la possibilità di scegliere una delle cinque facoltà e, sullo schermo, verranno visualizzate tutte le informazioni relative alla segreteria studenti relativa alla facoltà scelta. In Figura 4.5 possiamo vedere il *Process Flow* corrispondente a questa attività.



**Figura 4.5.** Process Flow relativo alla possibilità di visualizzare le informazioni di una segreteria

#### 4.2.5 Visualizzare un'offerta formativa o un orario

L'attore può scegliere la facoltà di suo interesse. Una volta fatto ciò, egli può scegliere tra Offerta Formativa e Orario; in base alla sua scelta, verranno visualizzate sullo schermo tutte le offerte formative o tutti gli orari di quella facoltà. In Figura 4.6 possiamo vedere il *Process Flow* relativo a questa attività.



**Figura 4.6.** Process Flow relativo alla possibilità di visualizzare un'offerta formativa o un orario

### 4.3 Mockup

I Mockup giocano un ruolo molto importante nella progettazione della componente applicativa poiché sono l'unico strumento che permette di connettere il prodotto finale con il suo progetto. Quindi, essi sono la rappresentazione grafica di come dovrebbe essere il prodotto finale. Nel nostro caso, essi rappresentano le *GUI* (*Graphic User Interface*) dell'app oggetto della presente tesi.

Esistono tre livelli di mockup:

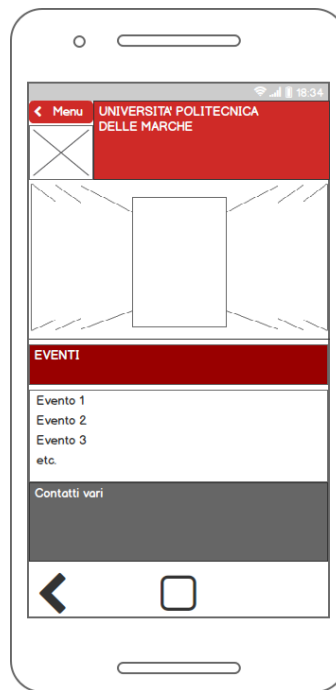
- *Wireframe*, o *mockup di livello 0*, rappresentano semplicemente uno "schizzo" di come sarà l'interfaccia grafica;
- *Mockup di livello 1*, rappresentano un raffinamento dei wireframe in cui si tiene conto delle proporzioni e della disposizione dei vari elementi, tuttavia la grafica non è ancora quella definitiva;

- *Mockup di livello 2*, si ottengono dai mockup di livello 1 migliorando la grafica in tutti i dettagli.

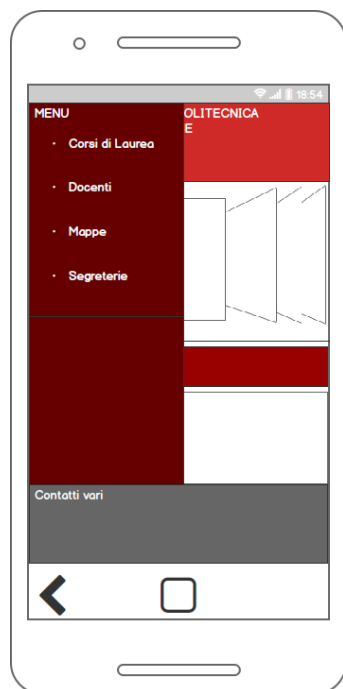
### 4.3.1 Mockup di livello 1

Di seguito possiamo vedere i mockup di livello 1 realizzati. Più specificatamente:

- in Figura 4.7 abbiamo il prototipo della “*HomePage*”;
- in Figura 4.8 viene rappresentato quello relativo al menù a scomparsa laterale;
- in Figura 4.9 abbiamo il mockup di livello 1 relativo alla pagina dei “*Docenti*”;
- in Figura 4.10 mostriamo il mockup di livello 1 relativo alla pagina “*Offerte formative*”;
- in Figura 4.11 illustriamo il mockup di livello 1 della pagina relativa agli “*Orari*”;
- in Figura 4.12 abbiamo il mockup di livello 1 della pagina “*Mappe*”;
- infine, in Figura 4.13 illustriamo il mockup di livello 1 relativo alla pagina “*Segreteria*”



**Figura 4.7.** Mockup di livello 1 relativo alla “HomePage”

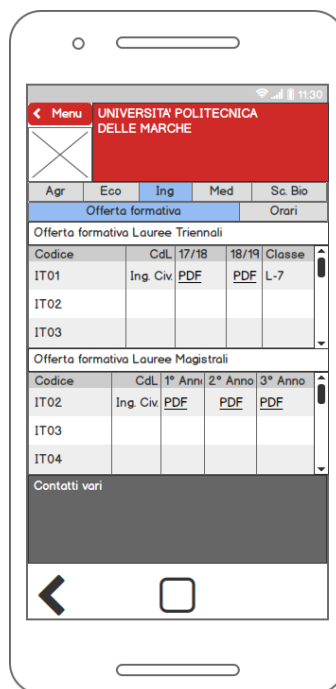


**Figura 4.8.** Mockup di livello 1 relativo al menù a scomparsa laterale

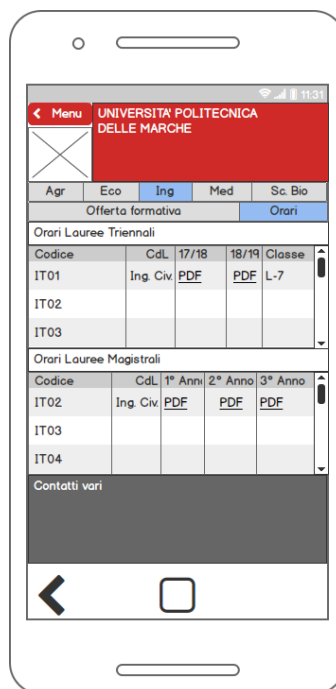


**Figura 4.9.** Mockup di livello 1 relativo alla pagina dei "Docenti"

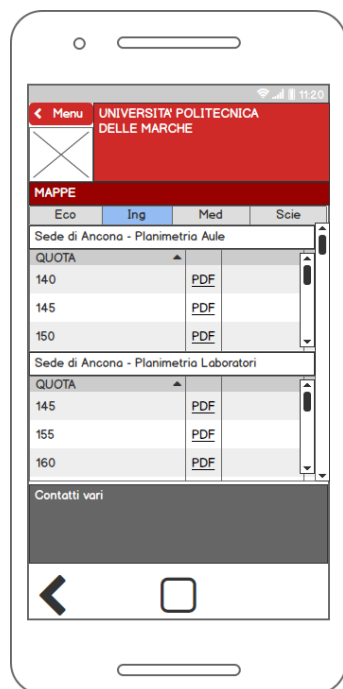




**Figura 4.10.** Mockup di livello 1 relativo alla pagina delle "Offerte formative"



**Figura 4.11.** Mockup di livello 1 relativo alla pagina degli "Orari"



**Figura 4.12.** Mockup di livello 1 relativo alla pagina delle "Mappe"



**Figura 4.13.** Mockup di livello 1 relativo alla pagina delle "Segreterie"

## Implementazione

*In questo capitolo ci soffermeremo sull'implementazione dell'app. Inizialmente effettueremo una presentazione degli strumenti e dei linguaggi utilizzati per il suo sviluppo. Successivamente, analizzeremo le principali funzionalità, esaminandone il codice.*

### 5.1 Strumenti e linguaggi utilizzati

#### 5.1.1 Visual Studio

L'*ambiente di sviluppo integrato (Integrated Development Environment IDE)* di *Visual Studio* è un'area di avvio creativa che consente di modificare, eseguire il debug, compilare il codice e quindi pubblicare un'app. In Figura 5.1 mostriamo il suo logo.

Un *ambiente di sviluppo integrato (IDE)* è un programma con numerose funzionalità che può essere usato per molti aspetti dello sviluppo del software. A differenza dell'editor e del debugger standard disponibili nella maggior parte degli ambienti *IDE*, *Visual Studio* utilizza la tecnologia "IntelliSense", che permette di correggere errori sintattici senza compilare l'applicazione ed utilizza un particolare debugger per il rilevamento di errori logici a tempo d'esecuzione.



**Figura 5.1.** Logo di Visual Studio

Inoltre, *Visual Studio* mette a disposizione anche una finestra di progettazione, che facilita gli sviluppatori nella costruzione dell'interfaccia grafica, consentendo il posizionamento di alcuni elementi tramite un semplice "drag and drop". In Figura 5.2 viene mostrata la schermata iniziale di Visual Studio.

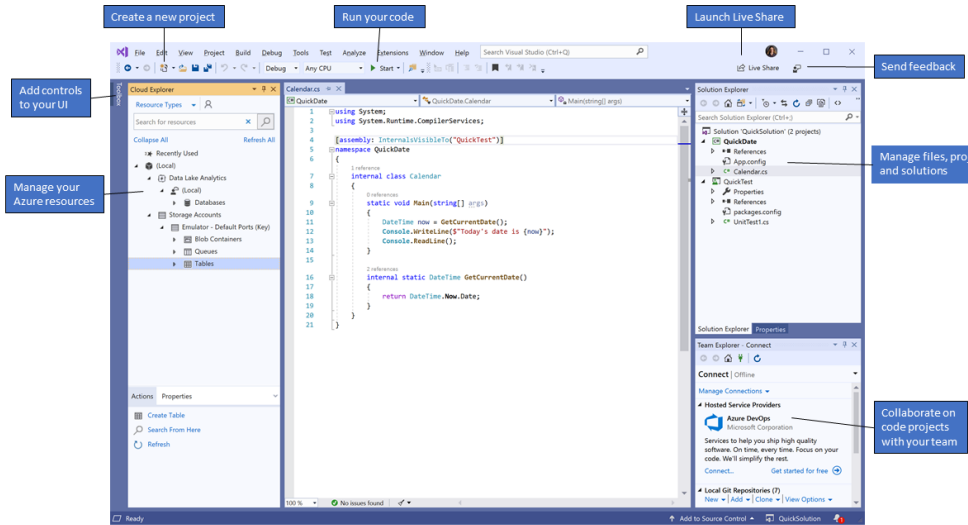


Figura 5.2. Schermata iniziale di Visual Studio

### 5.1.2 Android Device Manager

*Android Device Manager* è un tool presente in Xamarin che permette di emulare un dispositivo mobile per effettuare test e debug dell'applicazione che si sta sviluppando. È possibile anche collegare il proprio dispositivo Android, tramite USB, per effettuare il debug dell'app.

Per lo sviluppo della presente tesi è stato emulato un dispositivo configurato nel seguente modo:

- *Android Version*: Marshmallow 6.0 - API 23
- *CPU*: x86;
- *Screen Size*: 1080x1920 pixel/420 dpi;
- *RAM*: 1024 MB;
- *Data Storage*: 775 MB.

Oltre ad un dispositivo emulato, è stato utilizzato un dispositivo Android reale con le seguenti caratteristiche:

- *Android Version*: Pie 9.0 - API 28
- *CPU*: x64;
- *Screen Size*: 1080x1920 pixel/401 dpi;
- *RAM*: 6144 MB;
- *Data Storage*: 64 GB.

### 5.1.3 C#

C# può essere considerato il linguaggio di programmazione per eccellenza del *Framework .NET*; diversamente dagli altri linguaggi, come *Visual Basic* o *C++*, esso

è nato espressamente per la nuova piattaforma. In questo senso, è significativo il fatto che Microsoft stessa si sia servita di C# per scrivere gran parte delle librerie di .NET. Uno degli slogan che hanno accompagnato C# fin dalla sua nascita lo presenta come “un linguaggio facile come Java, potente come il C++”.

C# è un linguaggio di programmazione orientato agli oggetti adatto allo sviluppo di una vasta gamma di software: desktop, web e mobile.

Di seguito verranno illustrate alcune caratteristiche del linguaggio:

- i nomi delle variabili, delle funzioni, delle classi e di altri elementi sono “case sensitive”;
- ogni istruzione deve essere chiusa da un punto e virgola;
- si utilizzano le parentesi graffe per raggruppare porzioni di codice specifico (ad esempio, relative ad una classe o ad un metodo);
- seguendo le consuetudini dei linguaggi di programmazione orientati agli oggetti, i metodi sono raggruppati in classi e le classi sono raggruppate nei namespace.

Un *namespace* è un meccanismo per organizzare le classi C# in gruppi logici; in particolare, ha lo scopo di riunire classi correlate tra di loro. Si tratta dello stesso concetto che in *Java* viene chiamato *package*.

### 5.1.4 XML

*XML* (eXtensible Markup Language) è un metalinguaggio per la definizione di un linguaggio di markup. Esso, analogamente ad HTML, utilizza dei marcatori, denominati tag, per assegnare una semantica al testo. Ogni tag inizia e finisce con delle parentesi angolari; i tag possono contenere al loro interno diversi parametri che consentono di modificare l'aspetto degli elementi.

In *Xamarin.Android*, viene utilizzato XML insieme al linguaggio *XAML* (eXtensible Application Markup Language), che deriva dal precedente, per definire il layout dell'applicazione e per implementare l'interfaccia grafica, definendo i controlli grafici che si vogliono utilizzare e le loro caratteristiche, quali la posizione all'interno del display, la dimensione, il colore, l'id, etc.

## 5.2 HomePage

La *HomePage* è la pagina principale dell'app; essa è formata da un'immagine che raffigura uno dei loghi dell'Università Politecnica delle Marche, nonché da un pulsante in alto a sinistra che permette l'apertura di un menù dal quale è possibile cliccare per attivare una delle varie funzionalità introdotte precedentemente.

La *HomePage* è divisa in due parti: una pagina per la definizione del layout, attraverso il linguaggio XML/XAML, e una pagina in C#, per la logica di business.

### 5.2.1 Implementazione del layout della HomePage

Per la costruzione del layout abbiamo implementato una pagina XAML denominata `page_home`; essa rappresenta la view della *HomePage*. Il layout delle pagine dell'app è

stato definito attraverso il tag `LinearLayout`; quest'ultimo permette di visualizzare gli elementi della pagina in maniera lineare, con una disposizione che può essere orizzontale o verticale. Nel Listato 5.1 viene mostrato il codice relativo alla pagina "HomePage" per lo sviluppo della parte grafica.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#ffffff">
    <ImageView
        android:src="@drawable/img_page_home_univpm"
        android:layout_width="match_parent"
        android:layout_height="180dp"
        android:minWidth="25dp"
        android:minHeight="25dp"
        android:paddingTop="50dp"
        android:id="@+id/logoHome"
        android:background="#00264B" />
    <TextView
        android:text="EVENTI"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_width="match_parent"
        android:layout_height="40dp"
        android:id="@+id/textHomeEventTitle"
        android:background="#8b0000"
        android:textColor="#ffffff"
        android:editable="false"
        android:clickable="false"
        android:gravity="center_vertical"
        android:paddingLeft="10dp" />
    <RelativeLayout
        android:orientation="vertical"
        android:id="@+id/relativeLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <ListView
            android:minWidth="25px"
            android:minHeight="25px"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_above="@+id/textView1"
            android:id="@+id/listViewHome" />
        <TextView
            android:text="UNIVERSITA' POLITECNICA DELLE MARCHE"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:minWidth="25dp"
            android:minHeight="25dp"
            android:id="@+id/textView1"
            android:gravity="center_horizontal"
            android:textColor="#ffffff"
            android:background="#ff616161"
            android:layout_above="@+id/textView2"
            android:paddingTop="5dp" />
        <TextView
            android:text="Piazza Roma 21, 60121 Ancona"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:minWidth="25dp"
            android:minHeight="25dp"
            android:id="@+id/textView2"
            android:gravity="center_horizontal"
            android:textColor="#ffffff"
            android:background="#ff616161"
            android:layout_above="@+id/textView3"
            android:paddingTop="5dp" />
        <TextView
            android:text="tel: (+39) 071 220 21"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:minWidth="25dp"
            android:minHeight="25dp"
            android:id="@+id/textView3"
            android:gravity="center_horizontal"
            android:textColor="#ffffff"
            android:background="#ff616161"
            android:layout_alignParentBottom="true"
            android:paddingTop="5dp"
            android:autoLink="phone"
            android:textColorLink="#ffffff" />
    </RelativeLayout>
</LinearLayout>
```

**Listato 5.1.** Layout della HomePage

La pagina è formata da un gruppo di tag; ciascun tag rappresenta un elemento grafico che è composto da una serie di attributi; ogni attributo ha un valore. L'attributo `android:orientation` indica in che modo devono essere disposti gli elementi della pagina, in questo caso in maniera verticale.

Il valore `match_parent`, associato all'altezza e alla larghezza del layout, indica che il corrispettivo componente occuperà l'intera dimensione del display. Il valore `wrap_content`, invece, indica che occuperà tutto lo spazio sufficiente a mostrare il proprio contenuto.

Il tag `ImageView` rappresenta un'immagine; ad essa si associano diverse proprietà, come l'id e l'src; quest'ultima riceve come valore il percorso in cui si trova l'immagine che deve essere visualizzata.

Il tag `RelativeLayout`, a differenza delle caratteristiche del `LinearLayout`, ci permette di specificare la posizione degli "elementi figlio" rispetto a quella dell'"elemento padre" tramite l'id. Questo avviene mediante l'utilizzo degli attributi `android:layout_alignParentBottom` e `android:layout_above`. Il primo ci consente di ancorare un elemento alla parte bassa del display; invece, il secondo ci consente di posizionare gli elementi uno sopra all'altro.

L'attributo `android:autoLink` nell'ultima `TextView` ci dà la possibilità di effettuare una chiamata cliccando sul numero indicato.

Il tag `ListView` ci permette la creazione di una lista di elementi che verrà usata per la visualizzazione degli eventi. Questo elemento ha bisogno di un "template" specifico per definire come verranno visualizzati gli "item" al proprio interno. Nel Listato 5.2 viene mostrato il codice per la definizione del template.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:text="Descrizione"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:minWidth="25px"
        android:minHeight="25px"
        android:id="@+id/listText1"
        android:textColor="@android:color/black" />
    <TextView
        android:text="Data"
        android:textAppearance="?android:attr/textAppearanceSmall"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/listText2" />
    <!--TextView
        android:text="CodEv"
        android:textAppearance="?android:attr/textAppearanceSmall"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/listText3" /-->
</LinearLayout>
```

**Listato 5.2.** Template `ListView` per gli eventi

Grazie all'uso di questo "template", i cui valori delle `TextView` verranno sostituiti a tempo di esecuzione da un `adapter` che verrà analizzato in seguito.

### 5.2.2 Implementazione del codice C# della HomePage

Per l'implementazione dell'activity della HomePage abbiamo, innanzitutto, creato una classe, il cui codice viene mostrato nel Listato 5.3.

```

namespace UNIVPM.Resources.Model
{
    public class Eventi
    {
        [PrimaryKey, AutoIncrement]
        public int CodEv { get; set; }
        public string Descrizione { get; set; }
        public string Data { get; set; }
    }
}

```

**Listato 5.3.** Classe Eventi

Ogni variabile corrisponde ad un attributo della classe `Eventi`. Attraverso il costrutto `AutoIncrement`, incrementiamo automaticamente la chiave primaria `CodEV` alla creazione di un nuovo evento. In questo modo si evitano possibili errori riguardanti la chiave primaria.

Dopodiché, nel Listato 5.4, mostriamo il codice relativo alla `MainActivity.cs` dove viene implementata tutta la Home Page, e non solo.

```

namespace UNIVPM
{
    [Activity(Label = "@string/app_name", Theme = "@style/AppTheme.NoActionBar", MainLauncher = true,
        Icon = "@drawable/logo_app")]
    public class MainActivity : AppCompatActivity, NavigationView.IOnNavigationItemSelectedListener
    {
        ListView lstData;
        List<Eventi> lstSource = new List<Eventi>();
        Database db;

        protected override void OnCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);
            setContentView(Resource.Layout.activity_main);
            Android.Support.V7.Widget.Toolbar toolbar = FindViewById<Android.Support.V7.Widget.Toolbar>
                (Resource.Id.toolbar);
            SetSupportActionBar(toolbar);

            DrawerLayout drawer = FindViewById<DrawerLayout>(Resource.Id.drawer_layout);
            ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(this, drawer, toolbar,
                Resource.String.navigation_drawer_open, Resource.String.navigation_drawer_close);
            drawer.AddDrawerListener(toggle);
            toggle.SyncState();

            NavigationView navigationView = FindViewById<NavigationView>(Resource.Id.nav_view);
            navigationView.SetNavigationItemSelectedListener(this);

            //Create DataBase
            db = new Database();
            //Delete all previous events
            db.deleteTableEventi();
            //Create a new empty table "Eventi"
            db.createDatabase();
            string folder = System.Environment.GetFolderPath(System.Environment.SpecialFolder.Personal);
            Log.Info("DB_PATH", folder);

            lstData = FindViewById<ListView>(Resource.Id.listViewHome);

            Eventi evento1 = new Eventi()
            {
                Descrizione = "Sessione di laurea triennale Autunnale, Aula Magna - Polo Monte Dago,
                    60121 Ancona",
                Data = "28-10-2019",
            };
            db.insertIntoTableEventi(evento1);

            Eventi evento2 = new Eventi()
            {
                Descrizione = "Sessione di laurea triennale Invernale, Aula Magna - Polo Monte Dago,
                    60121 Ancona",
                Data = "18/12/2019",
            };
            db.insertIntoTableEventi(evento2);

            Eventi evento3 = new Eventi()
            {
                Descrizione = "Corso di perfezionamento in Computer Music Production & Sound Design,
                    scadenza iscrizioni 11/10/2019",
                Data = "",
            };
            db.insertIntoTableEventi(evento3);

            Eventi evento4 = new Eventi()

```



```

        {
            Descrizione = "Corso di Perfezionamento in Ingegneria dei materiali compositi,
            scadenza iscrizioni 14/10/2019",
            Data = "",
        };
        db.insertIntoTableEventi(evento4);

        Eventi evento5 = new Eventi()
        {
            Descrizione = "La condizione delle persone con DSA nel 2019 - Incontri sul tema
            dislessia ad Ancona e Osimo, presso Facolta di Medicina e Chirurgia, Torrette di
            Ancona (AN) e Osimo - Scuola Marta Russo",
            Data = "10 e 11/10/2019",
        };
        db.insertIntoTableEventi(evento5);

        LoadData();

        db.deleteTableEventi();

    }

    private void LoadData()
    {
        lstSource = db.selectTableEventi();
        var adapter = new ListViewAdapter(this, lstSource);
        lstData.Adapter = adapter;
    }

    public override void OnBackPressed()
    {
        DrawerLayout drawer = FindViewById<DrawerLayout>(Resource.Id.drawer_layout);
        if (drawer.IsDrawerOpen(GravityCompat.Start))
        {
            drawer.CloseDrawer(GravityCompat.Start);
        }
        else
        {
            base.OnBackPressed();
        }
    }

    public override bool OnCreateOptionsMenu(IMenu menu)
    {
        MenuInflater.Inflate(Resource.Menu.menu_main, menu);
        return true;
    }

    public override bool OnOptionsItemSelected(IMenuItem item)
    {
        int id = item.ItemId;
        if (id == Resource.Id.action_home)
        {
            var activity_home = new Intent(this, typeof(MainActivity));
            StartActivity(activity_home);
        }
        return base.OnOptionsItemSelected(item);
    }

    public bool OnNavigationItemSelected(IMenuItem item)
    {
        int id = item.ItemId;
        if (id == Resource.Id.nav_cd1)
        {
            var activity_agr = new Intent(this, typeof(OdL_Agr));
            StartActivity(activity_agr);
        }
        else if (id == Resource.Id.nav_doc)
        {
            var activity_doc = new Intent(this, typeof(SearchDoc));
            StartActivity(activity_doc);
        }
        else if (id == Resource.Id.nav_map)
        {
            var activity_map = new Intent(this, typeof(Map_Eco));
            StartActivity(activity_map);
        }
        else if (id == Resource.Id.nav_seg)
        {
            var activity_seg = new Intent(this, typeof(Seg_Agr));
            StartActivity(activity_seg);
        }
    }

    DrawerLayout drawer = FindViewById<DrawerLayout>(Resource.Id.drawer_layout);
    drawer.CloseDrawer(GravityCompat.Start);
    return true;
}
}
}

```

Listato 5.4. Activity della HomePage

In questo listato vengono definiti molti elementi chiave dell'app oggetto di questa tesi. Partendo dall'inizio abbiamo:

- La dichiarazione di una `ListView` e di una lista di tipo `Eventi`, su cui verranno inserite le informazioni ad essa relative, nonché la definizione di un `Database`.
- Il metodo `SetContentView`; esso prende come argomento il layout che vogliamo visualizzare come prima pagina e che sarà quello chiamato `activity_main` in cui verrà incluso il layout `page_home`.
- L'elemento `drawer`, che serve per la creazione del menù a scomparsa.
- L'elemento `navigationview`, che viene utilizzato per permettere all'utente di potersi muovere tra le varie pagine dell'app, anche grazie al metodo `FindViewById`, che effettua il collegamento all'elemento dell'interfaccia grafica.
- Un elemento della classe `Database` che, successivamente, viene creato e gestito grazie al file chiamato `Database.cs`, il cui codice viene mostrato nel Listato 5.5. In questo listato vengono definite le operazioni CRUD che, nel nostro caso, sono: creazione di un database e di una tabella tramite `createDatabase`, inserimento con `insertIntoTableEventi`, e cancellazione di una tabella con `deleteTableEventi`.
- Il metodo `LoadData`, che serve per prendere gli elementi presenti nella tabella e poi visualizzarli sulla pagina tramite un elenco. Questo viene fatto anche grazie alla classe `ListViewAdapter`, il cui codice viene mostrato nel Listato 5.6.
- Il metodo `OnBackPressed`, che serve per chiudere il menù, nel caso in cui sia aperto.
- Il metodo `OnCreateOptionsMenu` viene utilizzato per la creazione di un "menù" sulla barra principale; grazie al metodo `OnOptionsItemSelected`, è possibile assegnare un'activity quando viene premuto l'elemento presente nel menù. Nel nostro caso, viene richiamata la `main.activity`.
- Il metodo, `OnNavigationItemSelected`, che consente di muoversi tra le varie pagine mediante l'utilizzo di un `if` e dell'`id`. In base alla pagina desiderata dell'utente, l'app provvede al richiamo dell'activity ad essa associata, tramite il metodo `StartActivity`.

```
namespace UNIVPM.Resources.DataHelper
{
    class Database
    {
        string folder = System.Environment.GetFolderPath(System.Environment.SpecialFolder.Personal);
        public bool createDatabase()
        {
            try
            {
                using (SQLiteConnection connection = new SQLiteConnection(System.IO.Path.Combine(
                    folder, "SQLite.db3")))
                {
                    connection.CreateTable<UNIVPM.Resources.Model.Evento>();
                    return true;
                }
            }
            catch (SQLiteException ex)
            {
                Log.Info("SQLiteEx", ex.Message);
                return false;
            }
        }

        public bool insertIntoTableEventi(Model.Evento evento)
        {
            try
            {
                using (var connection = new SQLiteConnection(System.IO.Path.Combine(folder,
                    "SQLite.db3")))
                {
```

```

        connection.Insert(evento);
        return true;
    }
}

catch (SQLiteException ex)
{
    Log.Info("SQLiteEx", ex.Message);
    return false;
}

}

public List<Model.Evento> selectTableEventi()
{
    try
    {
        using (var connection = new SQLiteConnection(System.IO.Path.Combine(folder,
            "SQLite.db3")))
        {
            return connection.Table<Model.Evento>().ToList();
        }
    }
    catch (SQLiteException ex)
    {
        Log.Info("SQLiteEx", ex.Message);
        return null;
    }
}

public bool deleteTableEventi()
{
    try
    {
        using (var connection = new SQLiteConnection(System.IO.Path.Combine(folder,
            "SQLite.db3")))
        {
            connection.DropTable<UNIVPM.Resources.Model.Evento>();
            return true;
        }
    }
    catch (SQLiteException ex)
    {
        Log.Info("SQLiteEx", ex.Message);
        return false;
    }
}

}
}
}

```

Listato 5.5. Classe Database

Alcune funzionalità utilizzate nel listato precedente, come, `Insert`, `DropTable` o `CreateTable` provengono dalla libreria `SQLite`.

```
namespace UNIVPM.Resources
{
    public class ViewHolder : Java.Lang.Object
    {
        public TextView txtDescr { get; set; }
        public TextView txtData { get; set; }
    }

    public class ListViewAdapter : BaseAdapter
    {
        private Activity activity;
        private List<UNIVPM.Resources.Model.Eventi> lstEventi;
        public ListViewAdapter(Activity activity, List<UNIVPM.Resources.Model.Eventi> lstEventi)
        {
            this.activity = activity;
            this.lstEventi = lstEventi;
        }

        public override int Count
        {
            get
            {
                return lstEventi.Count;
            }
        }

        public override Java.Lang.Object GetItem(int position)
        {
            return null;
        }

        public override long GetItemId(int position)
        {
            return lstEventi[position].CodEv;
        }
    }
}
```

```

    public override View GetView(int position, View convertView, ViewGroup parent)
    {
        var view = convertView ?? activity.LayoutInflater.Inflate(Resource.Layout.list_view_template,
            parent, false);

        var txtDescr = view.FindViewById<TextView>(Resource.Id.listText1);
        var txtData = view.FindViewById<TextView>(Resource.Id.listText2);
        //var txtCod = view.FindViewById<TextView>(Resource.Id.listText3);

        txtDescr.Text = lstEventi[position].Descrizione;
        txtData.Text = lstEventi[position].Data;
        //txtCod.Text = "" + lstEventi[position].CodEv;

        return view;
    }
}

```

Listato 5.6. Classe ListViewAdapter

Parte fondamentale di questa classe è il metodo `public override View GetView` dove viene selezionato il template della `ListView` che viene, successivamente, riempito con i valori associati alla lista chiamata `lstEventi`, dichiarata all'inizio del listato.

### 5.3 Pagina relativa alle offerte formative e agli orari

Quando l'utente clicca su *Corsi di laurea* dal menù a scomparsa, sul display verrà visualizzata la pagina relativa alle varie offerte formative della Facoltà di Agraria; cliccando su "Orari" saranno visualizzati sullo schermo l'elenco degli orari dei corsi di laurea della stessa facoltà. Questo viene ripetuto per tutte le altre facoltà dell'Ateneo. Per implementare la pagina relativa ai corsi di laurea abbiamo utilizzato varie activity e vari layout associati tramite il metodo `SetContentView`.

#### 5.3.1 Implementazione del layout delle pagine relative alle offerte formative e agli orari

Per implementare il layout, abbiamo creato una pagina XML per le offerte formative e una per gli orari, chiamate, rispettivamente, `page_cdl_agr_of` e `page_cdl_agr_or`. Nel Listato 5.7 viene mostrato il codice relativo alla prima, e nel Listato 5.8 quello relativo alla seconda.

```

<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#ffffff">
    <android.support.design.widget.BottomNavigationView
        android:id="@+id/cdl_nav_agr"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="?android:attr/windowBackground"
        app:menu="@menu/page_cdl_menu" />
    <android.support.design.widget.BottomNavigationView
        android:id="@+id/navigation2"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:background="?android:attr/windowBackground"
        app:menu="@menu/page_cdl_top2_nav"/>
    <TextView
        android:text="CORSI DI LAUREA TRIENNALI"
        android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content"
        android:id="@+id/tableTitleCdL"
        android:textColor="#8b0000"
        android:textStyle="bold"
        android:paddingLeft="5dp"
        android:paddingTop="5dp"
        android:paddingBottom="5dp" />
<TableLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/table"
    android:background="#ff9e9e">
    <TableRow
        android:id="@+id/Row1"
        android:background="#ff9e9e">
        <TextView
            p4:placeholder="true"
            android:minWidth="30dp"
            android:minHeight="30dp"
            xmlns:p4="http://xamarin.com/mono/android/designer"
            android:id="@+id/Col11"
            android:text="CDD"
            android:textColor="#000000"
            android:paddingBottom="10dp"
            android:paddingTop="10dp"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:gravity="center_horizontal"
            android:layout_width="84.0dp" />
        <TextView
            p4:placeholder="true"
            android:minWidth="30dp"
            android:minHeight="30dp"
            xmlns:p4="http://xamarin.com/mono/android/designer"
            android:id="@+id/Col12"
            android:text="CdL TRIENNALE"
            android:textColor="#000000"
            android:paddingBottom="10dp"
            android:paddingTop="10dp"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:gravity="center_horizontal"
            android:layout_width="154.0dp" />
        <TextView
            p4:placeholder="true"
            android:minWidth="30dp"
            android:minHeight="30dp"
            xmlns:p4="http://xamarin.com/mono/android/designer"
            android:id="@+id/Col13"
            android:text="2019"
            android:textColor="#000000"
            android:paddingBottom="10dp"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:paddingTop="10dp"
            android:gravity="center_horizontal"
            android:layout_width="88.0dp" />
        <TextView
            p4:placeholder="true"
            android:minWidth="30dp"
            android:minHeight="30dp"
            xmlns:p4="http://xamarin.com/mono/android/designer"
            android:id="@+id/Col14"
            android:text="CLASSE"
            android:textColor="#000000"
            android:paddingBottom="10dp"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:paddingTop="10dp"
            android:gravity="center_horizontal"
            android:layout_width="94.0dp" />
    </TableRow>
</TableLayout>
<android.support.v4.widget.NestedScrollView
    android:minWidth="25dp"
    android:minHeight="25dp"
    android:layout_width="match_parent"
    android:layout_height="140dp"
    android:id="@+id/nestedScrollView1"
    android:scrollIndicators="right"
    android:scrollbars="vertical">
    <TableLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/tableCdL">
        <TableRow
            android:id="@+id/Row2">
            <TextView
                p4:placeholder="true"
                android:minWidth="30dp"
                android:minHeight="30dp"
                xmlns:p4="http://xamarin.com/mono/android/designer"
                android:id="@+id/Col21"
                android:text="A701"

```

```

        android:paddingBottom="10dp"
        android:paddingTop="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:gravity="center_horizontal"
        android:layout_width="84.0dp" />
<TextView
    p4:placeholder="true"
    android:minWidth="30dp"
    android:minHeight="30dp"
    xmlns:p4="http://xamarin.com/mono/android/designer"
    android:id="@+id/Col22"
    android:text="Scienze e Tecnologie Agrarie"
    android:paddingBottom="10dp"
    android:paddingTop="10dp"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:gravity="center_horizontal"
    android:layout_width="154.0dp" />
<TextView
    p4:placeholder="true"
    android:minWidth="30dp"
    android:minHeight="30dp"
    xmlns:p4="http://xamarin.com/mono/android/designer"
    android:id="@+id/Col23"
    android:text="@string/prova"
    android:paddingBottom="10dp"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:paddingTop="10dp"
    android:layout_width="88.0dp"
    android:gravity="center_horizontal" />
<TextView
    p4:placeholder="true"
    android:minWidth="30dp"
    android:minHeight="30dp"
    xmlns:p4="http://xamarin.com/mono/android/designer"
    android:id="@+id/Col24"
    android:text="25"
    android:paddingBottom="10dp"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:paddingTop="10dp"
    android:gravity="center_horizontal"
    android:layout_width="94.0dp" />
</TableRow>
<TableRow
    android:id="@+id/Row3">
    <TextView
        p4:placeholder="true"
        android:minWidth="30dp"
        android:minHeight="30dp"
        xmlns:p4="http://xamarin.com/mono/android/designer"
        android:id="@+id/Col31"
        android:text="AT03"
        android:paddingBottom="10dp"
        android:paddingTop="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:gravity="center_horizontal"
        android:layout_width="84.0dp" />
    <TextView
        p4:placeholder="true"
        android:minWidth="30dp"
        android:minHeight="30dp"
        xmlns:p4="http://xamarin.com/mono/android/designer"
        android:id="@+id/Col32"
        android:text="Scienze e Tecnologie Alimentari"
        android:paddingBottom="10dp"
        android:paddingTop="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:gravity="center_horizontal"
        android:layout_width="154.0dp" />
    <TextView
        p4:placeholder="true"
        android:minWidth="30dp"
        android:minHeight="30dp"
        xmlns:p4="http://xamarin.com/mono/android/designer"
        android:id="@+id/Col33"
        android:text="@string/prova"
        android:paddingBottom="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:paddingTop="10dp"
        android:layout_width="88.0dp"
        android:gravity="center_horizontal" />
    <TextView
        p4:placeholder="true"
        android:minWidth="30dp"
        android:minHeight="30dp"
        xmlns:p4="http://xamarin.com/mono/android/designer"
        android:id="@+id/Col34"
        android:text="26"

```

```

        android:paddingBottom="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:paddingTop="10dp"
        android:gravity="center_horizontal"
        android:layout_width="94.0dp" />
    </TableRow>
    <TableRow
        android:id="@+id/Row4">
        <TextView
            p4:placeholder="true"
            android:minWidth="30dp"
            android:minHeight="30dp"
            xmlns:p4="http://xamarin.com/mono/android/designer"
            android:id="@+id/Col41"
            android:text="A702"
            android:paddingBottom="10dp"
            android:paddingTop="10dp"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:gravity="center_horizontal"
            android:layout_width="84.0dp" />
        <TextView
            p4:placeholder="true"
            android:minWidth="30dp"
            android:minHeight="30dp"
            xmlns:p4="http://xamarin.com/mono/android/designer"
            android:id="@+id/Col42"
            android:text="Scienze Forestali ed Ambientali"
            android:paddingBottom="10dp"
            android:paddingTop="10dp"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:gravity="center_horizontal"
            android:layout_width="154.0dp" />
        <TextView
            p4:placeholder="true"
            android:minWidth="30dp"
            android:minHeight="30dp"
            xmlns:p4="http://xamarin.com/mono/android/designer"
            android:id="@+id/Col43"
            android:text="@string/prova"
            android:paddingBottom="10dp"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:paddingTop="10dp"
            android:layout_width="88.0dp"
            android:gravity="center_horizontal" />
        <TextView
            p4:placeholder="true"
            android:minWidth="30dp"
            android:minHeight="30dp"
            xmlns:p4="http://xamarin.com/mono/android/designer"
            android:id="@+id/Col44"
            android:text="25"
            android:paddingBottom="10dp"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:paddingTop="10dp"
            android:gravity="center_horizontal"
            android:layout_width="94.0dp" />
    </TableRow>
</android.support.v4.widget.NestedScrollView>
<TextView
    android:text="CORSI DI LAUREA MAGISTRALE"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/table2TitleCdl"
    android:textColor="#8b0000"
    android:textStyle="bold"
    android:paddingTop="5dp"
    android:paddingLeft="5dp"
    android:paddingBottom="5dp" />
<TableLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/table"
    android:background="#ff9e9e">
    <TableRow
        android:id="@+id/Row1"
        android:background="#ff9e9e">
        <TextView
            p4:placeholder="true"
            android:minWidth="30dp"
            android:minHeight="30dp"
            xmlns:p4="http://xamarin.com/mono/android/designer"
            android:id="@+id/Col11"
            android:text="CDD"
            android:textColor="#000000"
            android:paddingBottom="10dp"
            android:paddingTop="10dp"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"

```

```

        android:gravity="center_horizontal"
        android:layout_width="84.0dp" />
    <TextView
        p4:placeholder="true"
        android:minWidth="30dp"
        android:minHeight="30dp"
        xmlns:p4="http://xamarin.com/mono/android/designer"
        android:id="@+id/Col12"
        android:text="CdL MAGISTRALE"
        android:textColor="#000000"
        android:paddingBottom="10dp"
        android:paddingTop="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:gravity="center_horizontal"
        android:layout_width="154.0dp" />
    <TextView
        p4:placeholder="true"
        android:minWidth="30dp"
        android:minHeight="30dp"
        xmlns:p4="http://xamarin.com/mono/android/designer"
        android:id="@+id/Col13"
        android:text="2019"
        android:textColor="#000000"
        android:paddingBottom="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:paddingTop="10dp"
        android:gravity="center_horizontal"
        android:layout_width="88.0dp" />
    <TextView
        p4:placeholder="true"
        android:minWidth="30dp"
        android:minHeight="30dp"
        xmlns:p4="http://xamarin.com/mono/android/designer"
        android:id="@+id/Col14"
        android:text="CLASSE"
        android:textColor="#000000"
        android:paddingBottom="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:paddingTop="10dp"
        android:gravity="center_horizontal"
        android:layout_width="94.0dp" />
</TableRow>
</TableLayout>
<android.support.v4.widget.NestedScrollView
    android:minWidth="25dp"
    android:minHeight="25dp"
    android:layout_width="match_parent"
    android:layout_height="145dp"
    android:id="@+id/nestedScrollView1"
    android:scrollIndicators="right"
    android:scrollbars="vertical">
    <TableLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/tableCdL">
        <TableRow
            android:id="@+id/Row2">
            <TextView
                p4:placeholder="true"
                android:minWidth="30dp"
                android:minHeight="30dp"
                xmlns:p4="http://xamarin.com/mono/android/designer"
                android:id="@+id/Col21"
                android:text="AM01"
                android:paddingBottom="10dp"
                android:paddingTop="10dp"
                android:paddingLeft="10dp"
                android:paddingRight="10dp"
                android:gravity="center_horizontal"
                android:layout_width="84.0dp" />
            <TextView
                p4:placeholder="true"
                android:minWidth="30dp"
                android:minHeight="30dp"
                xmlns:p4="http://xamarin.com/mono/android/designer"
                android:id="@+id/Col22"
                android:text="Scienze Agrarie e Del Territorio"
                android:paddingBottom="10dp"
                android:paddingTop="10dp"
                android:paddingLeft="10dp"
                android:paddingRight="10dp"
                android:gravity="center_horizontal"
                android:layout_width="154.0dp" />
            <TextView
                p4:placeholder="true"
                android:minWidth="30dp"
                android:minHeight="30dp"
                xmlns:p4="http://xamarin.com/mono/android/designer"
                android:id="@+id/Col23"
                android:text="@string/prova"
                android:paddingBottom="10dp"

```



```

        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:paddingTop="10dp"
        android:layout_width="88.0dp"
        android:gravity="center_horizontal" />
<TextView
    p4:placeholder="true"
    android:minWidth="30dp"
    android:minHeight="30dp"
    xmlns:p4="http://xamarin.com/mono/android/designer"
    android:id="@+id/Col24"
    android:text="69"
    android:paddingBottom="10dp"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:paddingTop="10dp"
    android:gravity="center_horizontal"
    android:layout_width="94.0dp" />
</TableRow>
<TableRow
    android:id="@+id/Row3">
    <TextView
        p4:placeholder="true"
        android:minWidth="30dp"
        android:minHeight="30dp"
        xmlns:p4="http://xamarin.com/mono/android/designer"
        android:id="@+id/Col31"
        android:text="AM04"
        android:paddingBottom="10dp"
        android:paddingTop="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:gravity="center_horizontal"
        android:layout_width="84.0dp" />
    <TextView
        p4:placeholder="true"
        android:minWidth="30dp"
        android:minHeight="30dp"
        xmlns:p4="http://xamarin.com/mono/android/designer"
        android:id="@+id/Col32"
        android:text="Food and Beverage Innovation and Management"
        android:paddingBottom="10dp"
        android:paddingTop="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:gravity="center_horizontal"
        android:layout_width="154.0dp" />
    <TextView
        p4:placeholder="true"
        android:minWidth="30dp"
        android:minHeight="30dp"
        xmlns:p4="http://xamarin.com/mono/android/designer"
        android:id="@+id/Col33"
        android:text="@string/prova"
        android:paddingBottom="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:paddingTop="10dp"
        android:layout_width="88.0dp"
        android:gravity="center_horizontal" />
    <TextView
        p4:placeholder="true"
        android:minWidth="30dp"
        android:minHeight="30dp"
        xmlns:p4="http://xamarin.com/mono/android/designer"
        android:id="@+id/Col34"
        android:text="70"
        android:paddingBottom="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:paddingTop="10dp"
        android:gravity="center_horizontal"
        android:layout_width="94.0dp" />
</TableRow>
<TableRow
    android:id="@+id/Row4">
    <TextView
        p4:placeholder="true"
        android:minWidth="30dp"
        android:minHeight="30dp"
        xmlns:p4="http://xamarin.com/mono/android/designer"
        android:id="@+id/Col41"
        android:text="AM03"
        android:paddingBottom="10dp"
        android:paddingTop="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:gravity="center_horizontal"
        android:layout_width="84.0dp" />
    <TextView
        p4:placeholder="true"
        android:minWidth="30dp"
        android:minHeight="30dp"
        xmlns:p4="http://xamarin.com/mono/android/designer"

```

```

        android:id="@+id/Col42"
        android:text="Scienze Forestali, dei Suoli e del Paesaggio"
        android:paddingBottom="10dp"
        android:paddingTop="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:gravity="center_horizontal"
        android:layout_width="154.0dp" />

<TextView
    p4:placeholder="true"
    android:minWidth="30dp"
    android:minHeight="30dp"
    xmlns:p4="http://xamarin.com/mono/android/designer"
    android:id="@+id/Col43"
    android:text="@string/prova"
    android:paddingBottom="10dp"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:paddingTop="10dp"
    android:layout_width="88.0dp"
    android:gravity="center_horizontal" />

<TextView
    p4:placeholder="true"
    android:minWidth="30dp"
    android:minHeight="30dp"
    xmlns:p4="http://xamarin.com/mono/android/designer"
    android:id="@+id/Col44"
    android:text="73"
    android:paddingBottom="10dp"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:paddingTop="10dp"
    android:gravity="center_horizontal"
    android:layout_width="94.0dp" />

</TableRow>
</TableLayout>
</android.support.v4.widget.NestedScrollView>
<RelativeLayout
    android:orientation="vertical"
    android:id="@+id/relativeLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <TextView
        android:text="UNIVERSITA' POLITECNICA DELLE MARCHE"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:minWidth="25dp"
        android:minHeight="25dp"
        android:id="@+id/textView1"
        android:gravity="center_horizontal"
        android:textColor="#ffffff"
        android:background="#ff616161"
        android:layout_above="@+id/textView2"
        android:paddingTop="5dp" />
    <TextView
        android:text="Piazza Roma 21, 60121 Ancona"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:minWidth="25dp"
        android:minHeight="25dp"
        android:id="@+id/textView2"
        android:gravity="center_horizontal"
        android:textColor="#ffffff"
        android:background="#ff616161"
        android:layout_above="@+id/textView3"
        android:paddingTop="5dp" />
    <TextView
        android:text="tel: (+39) 071 220 21"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:minWidth="25dp"
        android:minHeight="25dp"
        android:id="@+id/textView3"
        android:gravity="center_horizontal"
        android:textColor="#ffffff"
        android:background="#ff616161"
        android:layout_alignParentBottom="true"
        android:paddingTop="5dp"
        android:autoLink="phone"
        android:textColorLink="#ffffff" />
</RelativeLayout>
</LinearLayout>

```

**Listato 5.7.** Layout della pagina relativa alle offerte formative della Facoltà di Agraria

```

<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"

```

```

android:orientation="vertical"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="#ffffff">
<android.support.design.widget.BottomNavigationView
    android:id="@+id/cdl_nav_agr"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="?android:attr/windowBackground"
    app:menu="@menu/page_cdl_menu" />
<android.support.design.widget.BottomNavigationView
    android:id="@+id/navigation2"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:background="?android:attr/windowBackground"
    app:menu="@menu/page_cdl_top2_nav"/>
<TextView
    android:text="CORSI DI LAUREA TRIENNALI"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/tableTitleCdl"
    android:textColor="#8b0000"
    android:textStyle="bold"
    android:paddingLeft="5dp"
    android:paddingTop="5dp"
    android:paddingBottom="5dp" />
<TableLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/table"
    android:background="#ff9e9e">
    <TableRow
        android:id="@+id/Row1"
        android:background="#ff9e9e">
        <TextView
            p4:placeholder="true"
            android:minWidth="30dp"
            android:minHeight="30dp"
            xmlns:p4="http://xamarin.com/mono/android/designer"
            android:id="@+id/Col11"
            android:text="CDD"
            android:textColor="#000000"
            android:paddingBottom="10dp"
            android:paddingTop="10dp"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:gravity="center_horizontal"
            android:layout_width="84.0dp" />
        <TextView
            p4:placeholder="true"
            android:minWidth="30dp"
            android:minHeight="30dp"
            xmlns:p4="http://xamarin.com/mono/android/designer"
            android:id="@+id/Col12"
            android:text="CdL TRIENNALE"
            android:textColor="#000000"
            android:paddingBottom="10dp"
            android:paddingTop="10dp"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:gravity="center_horizontal"
            android:layout_width="154.0dp" />
        <TextView
            p4:placeholder="true"
            android:minWidth="30dp"
            android:minHeight="30dp"
            xmlns:p4="http://xamarin.com/mono/android/designer"
            android:id="@+id/Col13"
            android:text="19/20 - 1 Semestre"
            android:textColor="#000000"
            android:paddingBottom="10dp"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:paddingTop="10dp"
            android:gravity="center_horizontal"
            android:layout_width="182.0dp" />
        </TableRow>
    </TableLayout>
<android.support.v4.widget.NestedScrollView
    android:minWidth="25dp"
    android:minHeight="25dp"
    android:layout_width="match_parent"
    android:layout_height="140dp"
    android:id="@+id/nestedScrollView1"
    android:scrollIndicators="right"
    android:scrollbars="vertical">
    <TableLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/tableCdl">
        <TableRow
            android:id="@+id/Row2">
            <TextView
                p4:placeholder="true"

```

```

        android:minWidth="30dp"
        android:minHeight="30dp"
        xmlns:p4="http://xamarin.com/mono/android/designer"
        android:id="@+id/Col21"
        android:text="AT01"
        android:paddingBottom="10dp"
        android:paddingTop="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:gravity="center_horizontal"
        android:layout_width="84.0dp" />
<TextView
    p4:placeholder="true"
    android:minWidth="30dp"
    android:minHeight="30dp"
    xmlns:p4="http://xamarin.com/mono/android/designer"
    android:id="@+id/Col22"
    android:text="Scienze e Tecnologie Agrarie"
    android:paddingBottom="10dp"
    android:paddingTop="10dp"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:gravity="center_horizontal"
    android:layout_width="154.0dp" />
<TextView
    p4:placeholder="true"
    android:minWidth="30dp"
    android:minHeight="30dp"
    xmlns:p4="http://xamarin.com/mono/android/designer"
    android:id="@+id/Col23"
    android:text="@string/prova"
    android:paddingBottom="10dp"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:paddingTop="10dp"
    android:layout_width="182.0dp"
    android:gravity="center_horizontal" />
</TableRow>
<TableRow
    android:id="@+id/Row3">
    <TextView
        p4:placeholder="true"
        android:minWidth="30dp"
        android:minHeight="30dp"
        xmlns:p4="http://xamarin.com/mono/android/designer"
        android:id="@+id/Col31"
        android:text="AT03"
        android:paddingBottom="10dp"
        android:paddingTop="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:gravity="center_horizontal"
        android:layout_width="84.0dp" />
    <TextView
        p4:placeholder="true"
        android:minWidth="30dp"
        android:minHeight="30dp"
        xmlns:p4="http://xamarin.com/mono/android/designer"
        android:id="@+id/Col32"
        android:text="Scienze e Tecnologie Alimentari"
        android:paddingBottom="10dp"
        android:paddingTop="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:gravity="center_horizontal"
        android:layout_width="154.0dp" />
    <TextView
        p4:placeholder="true"
        android:minWidth="30dp"
        android:minHeight="30dp"
        xmlns:p4="http://xamarin.com/mono/android/designer"
        android:id="@+id/Col33"
        android:text="@string/prova"
        android:paddingBottom="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:paddingTop="10dp"
        android:layout_width="182.0dp"
        android:gravity="center_horizontal" />
</TableRow>
<TableRow
    android:id="@+id/Row4">
    <TextView
        p4:placeholder="true"
        android:minWidth="30dp"
        android:minHeight="30dp"
        xmlns:p4="http://xamarin.com/mono/android/designer"
        android:id="@+id/Col41"
        android:text="AT02"
        android:paddingBottom="10dp"
        android:paddingTop="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:gravity="center_horizontal"

```

```

        android:layout_width="84.0dp" />
<TextView
    p4:placeholder="true"
    android:minWidth="30dp"
    android:minHeight="30dp"
    xmlns:p4="http://xamarin.com/mono/android/designer"
    android:id="@+id/Col42"
    android:text="Scienze Forestali ed Ambientali"
    android:paddingBottom="10dp"
    android:paddingTop="10dp"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:gravity="center_horizontal"
    android:layout_width="154.0dp" />
<TextView
    p4:placeholder="true"
    android:minWidth="30dp"
    android:minHeight="30dp"
    xmlns:p4="http://xamarin.com/mono/android/designer"
    android:id="@+id/Col43"
    android:text="@string/prova"
    android:paddingBottom="10dp"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:paddingTop="10dp"
    android:layout_width="182.0dp"
    android:gravity="center_horizontal" />
</TableRow>
</TableLayout>
</android.support.v4.widget.NestedScrollView>
<TextView
    android:text="CORSI DI LAUREA MAGISTRALE"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/table2TitleCdL"
    android:textColor="#8b0000"
    android:textStyle="bold"
    android:paddingTop="5dp"
    android:paddingLeft="5dp"
    android:paddingBottom="5dp" />
<TableLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/table"
    android:background="#ff9e9e">
<TableRow
    android:id="@+id/Row1"
    android:background="#ff9e9e">
<TextView
    p4:placeholder="true"
    android:minWidth="30dp"
    android:minHeight="30dp"
    xmlns:p4="http://xamarin.com/mono/android/designer"
    android:id="@+id/Col11"
    android:text="CDD"
    android:textColor="#000000"
    android:paddingBottom="10dp"
    android:paddingTop="10dp"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:gravity="center_horizontal"
    android:layout_width="84.0dp" />
<TextView
    p4:placeholder="true"
    android:minWidth="30dp"
    android:minHeight="30dp"
    xmlns:p4="http://xamarin.com/mono/android/designer"
    android:id="@+id/Col12"
    android:text="CdL MAGISTRALE"
    android:textColor="#000000"
    android:paddingBottom="10dp"
    android:paddingTop="10dp"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:gravity="center_horizontal"
    android:layout_width="154.0dp" />
<TextView
    p4:placeholder="true"
    android:minWidth="30dp"
    android:minHeight="30dp"
    xmlns:p4="http://xamarin.com/mono/android/designer"
    android:id="@+id/Col13"
    android:text="19/20 - 1 Semestre"
    android:textColor="#000000"
    android:paddingBottom="10dp"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:paddingTop="10dp"
    android:gravity="center_horizontal"
    android:layout_width="182.0dp" />
</TableRow>
</TableLayout>
<android.support.v4.widget.NestedScrollView
    android:minWidth="25dp"

```

```

android:minHeight="25dp"
android:layout_width="match_parent"
android:layout_height="145dp"
android:id="@+id/nestedScrollView1"
android:scrollIndicators="right"
android:scrollbars="vertical">
<TableLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/tableCdl">
    <TableRow
        android:id="@+id/Row2">
        <TextView
            p4:placeholder="true"
            android:minWidth="30dp"
            android:minHeight="30dp"
            xmlns:p4="http://xamarin.com/mono/android/designer"
            android:id="@+id/Col21"
            android:text="AM01"
            android:paddingBottom="10dp"
            android:paddingTop="10dp"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:gravity="center_horizontal"
            android:layout_width="84.0dp" />
        <TextView
            p4:placeholder="true"
            android:minWidth="30dp"
            android:minHeight="30dp"
            xmlns:p4="http://xamarin.com/mono/android/designer"
            android:id="@+id/Col22"
            android:text="Scienze Agrarie e Del Territorio"
            android:paddingBottom="10dp"
            android:paddingTop="10dp"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:gravity="center_horizontal"
            android:layout_width="154.0dp" />
        <TextView
            p4:placeholder="true"
            android:minWidth="30dp"
            android:minHeight="30dp"
            xmlns:p4="http://xamarin.com/mono/android/designer"
            android:id="@+id/Col23"
            android:text="@string/prova"
            android:paddingBottom="10dp"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:paddingTop="10dp"
            android:layout_width="182.0dp"
            android:gravity="center_horizontal" />
    </TableRow>
    <TableRow
        android:id="@+id/Row3">
        <TextView
            p4:placeholder="true"
            android:minWidth="30dp"
            android:minHeight="30dp"
            xmlns:p4="http://xamarin.com/mono/android/designer"
            android:id="@+id/Col31"
            android:text="AM04"
            android:paddingBottom="10dp"
            android:paddingTop="10dp"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:gravity="center_horizontal"
            android:layout_width="84.0dp" />
        <TextView
            p4:placeholder="true"
            android:minWidth="30dp"
            android:minHeight="30dp"
            xmlns:p4="http://xamarin.com/mono/android/designer"
            android:id="@+id/Col32"
            android:text="Food and Beverage Innovation and Management"
            android:paddingBottom="10dp"
            android:paddingTop="10dp"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:gravity="center_horizontal"
            android:layout_width="154.0dp" />
        <TextView
            p4:placeholder="true"
            android:minWidth="30dp"
            android:minHeight="30dp"
            xmlns:p4="http://xamarin.com/mono/android/designer"
            android:id="@+id/Col33"
            android:text="@string/prova"
            android:paddingBottom="10dp"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:paddingTop="10dp"
            android:layout_width="182.0dp"
            android:gravity="center_horizontal" />
    </TableRow>

```

```

<TableRow
    android:id="@+id/Row4">
    <TextView
        p4:placeholder="true"
        android:minWidth="30dp"
        android:minHeight="30dp"
        xmlns:p4="http://xamarin.com/mono/android/designer"
        android:id="@+id/Col41"
        android:text="AM03"
        android:paddingBottom="10dp"
        android:paddingTop="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:gravity="center_horizontal"
        android:layout_width="84.0dp" />
    <TextView
        p4:placeholder="true"
        android:minWidth="30dp"
        android:minHeight="30dp"
        xmlns:p4="http://xamarin.com/mono/android/designer"
        android:id="@+id/Col42"
        android:text="Scienze Forestali, dei Suoli e del Paesaggio"
        android:paddingBottom="10dp"
        android:paddingTop="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:gravity="center_horizontal"
        android:layout_width="154.0dp" />
    <TextView
        p4:placeholder="true"
        android:minWidth="30dp"
        android:minHeight="30dp"
        xmlns:p4="http://xamarin.com/mono/android/designer"
        android:id="@+id/Col43"
        android:text="@string/prova"
        android:paddingBottom="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:paddingTop="10dp"
        android:layout_width="182.0dp"
        android:gravity="center_horizontal" />
    </TableRow>
</TableLayout>
</android.support.v4.widget.NestedScrollView>
<RelativeLayout
    android:orientation="vertical"
    android:id="@+id/relativeLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <TextView
        android:text="UNIVERSITA' POLITECNICA DELLE MARCHE"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:minWidth="25dp"
        android:minHeight="25dp"
        android:id="@+id/textView1"
        android:gravity="center_horizontal"
        android:textColor="#ffffff"
        android:background="#ff616161"
        android:layout_above="@+id/textView2"
        android:paddingTop="5dp" />
    <TextView
        android:text="Piazza Roma 21, 60121 Ancona"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:minWidth="25dp"
        android:minHeight="25dp"
        android:id="@+id/textView2"
        android:gravity="center_horizontal"
        android:textColor="#ffffff"
        android:background="#ff616161"
        android:layout_above="@+id/textView3"
        android:paddingTop="5dp" />
    <TextView
        android:text="tel: (+39) 071 220 21"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:minWidth="25dp"
        android:minHeight="25dp"
        android:id="@+id/textView3"
        android:gravity="center_horizontal"
        android:textColor="#ffffff"
        android:background="#ff616161"
        android:layout_alignParentBottom="true"
        android:paddingTop="5dp"
        android:autoLink="phone"
        android:textColorLink="#ffffff" />
    </RelativeLayout>
</LinearLayout>

```

Listato 5.8. Layout della pagina relativa agli orari della Facoltà di Agraria

Anche questi due layout sono stati implementati con l'utilizzo del tag `LinearLayout`, eccetto per la parte finale dove è stato usato un `RelativeLayout`. Da sottolineare l'utilizzo dei tag `TableLayout` e `TableRow`, che ci permettono, rispettivamente, di creare una tabella e le righe che la compongono. In questi layout abbiamo utilizzato il tag `android.support.v4.widget.NestedScrollView`; esso ci consente di creare una *ScrollView* all'interno della tabella.

Per potersi muovere tra le varie facoltà e tra le offerte formative e gli orari, abbiamo utilizzato due `BottomNavigationView`, i cui elementi vengono definiti nei Listati 5.9 e 5.10. Il primo ci permette di spostarci tra le varie facoltà dell'ateneo, mentre il secondo ci permette di spostarci tra offerte formative e orari.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <group android:checkableBehavior="single">
    <item
      android:id="@+id/nav_agr"
      android:icon="@drawable/ic_nav_cdl_agr"
      android:title="AGRARIA" />
    <item
      android:id="@+id/nav_eco"
      android:icon="@drawable/ic_nav_cdl_eco"
      android:title="ECONOMIA" />
    <item
      android:id="@+id/nav_ing"
      android:icon="@drawable/ic_nav_cdl_ing"
      android:title="INGEGNERIA" />
    <item
      android:id="@+id/nav_med"
      android:icon="@drawable/ic_nav_cdl_med"
      android:title="MEDICINA" />
    <item
      android:id="@+id/nav_sci"
      android:icon="@drawable/ic_nav_cdl_sci"
      android:title="SCIENZE" />
  </group>
</menu>
```

**Listato 5.9.** Implementazione della `BottomNavigationView` per le facoltà

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item
    android:id="@+id/nav_of"
    android:title="OFFERTA FORMATIVA" />
  <item
    android:id="@+id/nav_orari"
    android:title="ORARI" />
</menu>
```

**Listato 5.10.** Implementazione della `BottomNavigationView` per le offerte formative e gli orari

In entrambi vengono creati degli `item` che saranno gli elementi cliccabili sulla barra di navigazione, in aggiunta, nel primo vengono inserite, anche, delle icone grazie all'attributo `android:icon` che prende come valore il path dell'icona da inserire.

### 5.3.2 Implementazione del codice C# della pagina relativa alle offerte formative e agli orari

Per implementare l'activity associata al click, da parte dell'utente, nel menù, su Corsi di Laurea, sono state create due activity chiamate `CdL_Agr.cs` e `CdL_Agr_Orari.cs`. Queste ultime vengono mostrate nei Listati 5.11 e 5.12.



```

namespace UNIVPM
{
    [Activity(Label = "Corsi di Laurea - Agraria")]
    public class CdL_Agr : AppCompatActivity, BottomNavigationView.INavigationItemSelectedListener
    {
        protected override void onCreate(Bundle savedInstanceState)
        {
            base.onCreate(savedInstanceState);
            setContentView(Resource.Layout.page_cdL_agr_of);
            BottomNavigationView nav = findViewById<BottomNavigationView>(Resource.Id.cdL_nav_agr);
            nav.SetOnNavigationItemSelectedListener(this);
            BottomNavigationView nav2 = findViewById<BottomNavigationView>(Resource.Id.navigation2);
            nav2.SetOnNavigationItemSelectedListener(this);
        }

        public override bool onCreateOptionsMenu(IMenu menu)
        {
            MenuInflater.Inflate(Resource.Menu.menu_main, menu);
            return true;
        }

        public override bool onOptionsItemSelected(IMenuItem item)
        {
            int id = item.ItemId;
            if (id == Resource.Id.action_home)
            {
                var activity_home = new Intent(this, typeof(MainActivity));
                StartActivity(activity_home);
            }
            return base.onOptionsItemSelected(item);
        }

        public bool OnNavigationItemSelectedListener(IMenuItem item)
        {
            switch (item.ItemId)
            {
                case Resource.Id.nav_orari:
                    var activity_agr_orari = new Intent(this, typeof(CdL_Agr_Orari));
                    StartActivity(activity_agr_orari);
                    return true;
                case Resource.Id.nav_eco:
                    var activity_eco = new Intent(this, typeof(CdL_Eco));
                    StartActivity(activity_eco);
                    return true;
                case Resource.Id.nav_ing:
                    var activity_ing = new Intent(this, typeof(CdL_Ing));
                    StartActivity(activity_ing);
                    return true;
                case Resource.Id.nav_med:
                    var activity_med = new Intent(this, typeof(CdL_Med));
                    StartActivity(activity_med);
                    return true;
                case Resource.Id.nav_sci:
                    var activity_sci = new Intent(this, typeof(CdL_Scie));
                    StartActivity(activity_sci);
                    return true;
            }
            return false;
        }
    }
}

```

Listato 5.11. Activity della pagina relativa alle offerte formative

```

namespace UNIVPM
{
    [Activity(Label = "Corsi di Laurea - Agraria")]
    public class CdL_Agr_Orari : AppCompatActivity, BottomNavigationView.INavigationItemSelectedListener
    {
        protected override void onCreate(Bundle savedInstanceState)
        {
            base.onCreate(savedInstanceState);
            setContentView(Resource.Layout.page_cdL_agr_or);
            BottomNavigationView nav = findViewById<BottomNavigationView>(Resource.Id.cdL_nav_agr);
            nav.SetOnNavigationItemSelectedListener(this);
            BottomNavigationView nav2 = findViewById<BottomNavigationView>(Resource.Id.navigation2);
            nav2.SetOnNavigationItemSelectedListener(this);
        }

        public override bool onCreateOptionsMenu(IMenu menu)
        {
            MenuInflater.Inflate(Resource.Menu.menu_main, menu);
            return true;
        }

        public override bool onOptionsItemSelected(IMenuItem item)
        {
            int id = item.ItemId;
            if (id == Resource.Id.action_home)
            {
                var activity_home = new Intent(this, typeof(MainActivity));
                StartActivity(activity_home);
            }
            return base.onOptionsItemSelected(item);
        }

        public bool OnNavigationItemSelectedListener(IMenuItem item)
        {
            switch (item.ItemId)
            {
                case Resource.Id.nav_orari:
                    var activity_agr_orari = new Intent(this, typeof(CdL_Agr_Orari));
                    StartActivity(activity_agr_orari);
                    return true;
                case Resource.Id.nav_eco:
                    var activity_eco = new Intent(this, typeof(CdL_Eco));
                    StartActivity(activity_eco);
                    return true;
                case Resource.Id.nav_ing:
                    var activity_ing = new Intent(this, typeof(CdL_Ing));
                    StartActivity(activity_ing);
                    return true;
                case Resource.Id.nav_med:
                    var activity_med = new Intent(this, typeof(CdL_Med));
                    StartActivity(activity_med);
                    return true;
                case Resource.Id.nav_sci:
                    var activity_sci = new Intent(this, typeof(CdL_Scie));
                    StartActivity(activity_sci);
                    return true;
            }
            return false;
        }
    }
}

```

```

        {
            var activity_home = new Intent(this, typeof(MainActivity));
            StartActivity(activity_home);
        }
        return base.OnOptionsItemSelected(item);
    }

    public bool OnNavigationItemSelected(IMenuItem item)
    {
        switch (item.ItemId)
        {
            case Resource.Id.nav_of:
                var activity_agr = new Intent(this, typeof(Cdl_Agr));
                StartActivity(activity_agr);
                return true;
            case Resource.Id.nav_eco:
                var activity_eco = new Intent(this, typeof(Cdl_Eco));
                StartActivity(activity_eco);
                return true;
            case Resource.Id.nav_ing:
                var activity_ing = new Intent(this, typeof(Cdl_Ing));
                StartActivity(activity_ing);
                return true;
            case Resource.Id.nav_med:
                var activity_med = new Intent(this, typeof(Cdl_Med));
                StartActivity(activity_med);
                return true;
            case Resource.Id.nav_sci:
                var activity_sci = new Intent(this, typeof(Cdl_Scie));
                StartActivity(activity_sci);
                return true;
        }
        return false;
    }
}

```

**Listato 5.12.** Activity della pagina relativa agli orari

Grazie al metodo `SetContentView` viene chiamato il layout da visualizzare sullo schermo al momento del click dell'utente.

Come si vede, i due listati sono molto simili tra loro. In entrambi i casi, vengono dichiarate le due `BottomNavigationView` e vengono associate ai corrispettivi file dell'interfaccia grafica grazie al metodo `FindViewById`. Dopodiché vengono implementati i due metodi per la creazione di un menù sulla barra principale, per consentire all'utente di tornare alla Home Page in qualsiasi momento.

Infine, abbiamo implementato il metodo `OnNavigationItemSelected` che, grazie all'id di ogni `item`, e grazie ad uno `switch`, ci permette di muoverci tra le facoltà creando un'activity ad hoc per ognuna di esse.

## 5.4 Pagina relativa ai docenti

Quanto l'utente clicca su “*Docenti*” nel menù a scomparsa, verrà visualizzata sullo schermo la pagina relativa alla ricerca di un docente, nel caso in cui esso sia presente nel database dell'app. A seguito di ciò vengono creati alcuni layout e alcune classi che vedremo in dettaglio.

### 5.4.1 Implementazione del layout della pagina relativa ai docenti

Per implementare il layout grafico abbiamo creato una pagina XML il cui codice viene mostrato nel Listato 5.13.

```

<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#ffffff">
    <RelativeLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/editText1"
            android:hint="Inserisci il cognome del docente" />

        <Button
            android:text="CERCA"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@+id/editText1"
            android:id="@+id/button1" />

        <ListView
            android:minWidth="25px"
            android:minHeight="25px"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/listView"
            android:layout_below="@+id/button1" />

    </RelativeLayout>
    <RelativeLayout
        android:orientation="vertical"
        android:id="@+id/relativeLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:text="UNIVERSITA' POLITECNICA DELLE MARCHE"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:minWidth="25dp"
            android:minHeight="25dp"
            android:id="@+id/textView1"
            android:gravity="center_horizontal"
            android:textColor="#ffffff"
            android:background="#ff616161"
            android:layout_above="@+id/textView2"
            android:paddingTop="5dp" />
        <TextView
            android:text="Piazza Roma 21, 60121 Ancona"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:minWidth="25dp"
            android:minHeight="25dp"
            android:id="@+id/textView2"
            android:gravity="center_horizontal"
            android:textColor="#ffffff"
            android:background="#ff616161"
            android:layout_above="@+id/textView3"
            android:paddingTop="5dp" />
        <TextView
            android:text="tel: (+39) 071 220 21"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:minWidth="25dp"
            android:minHeight="25dp"
            android:id="@+id/textView3"
            android:gravity="center_horizontal"
            android:textColor="#ffffff"
            android:background="#ff616161"
            android:layout_alignParentBottom="true"
            android:paddingTop="5dp"
            android:autoLink="phone"
            android:textColorLink="#ffffff" />
    </RelativeLayout>
</LinearLayout>

```

Listato 5.13. Layout della pagina relativa ai docenti

Anche in questo caso è stato creato un layout utilizzando il tag `LinearLayout`. Il `RelativeLayout` viene, invece, utilizzato per posizionare nel modo corretto gli elementi `EditText`, `Button` e `ListView`. Il primo viene utilizzato per permettere all'utente di scrivere il cognome del docente da ricercare all'interno del database; il secondo viene utilizzato per dare il via alla ricerca all'interno di esso ed, infine, la

`ListView` viene utilizzata per presentare all'utente il risultato della ricerca. Come abbiamo visto nel caso degli eventi, anche in questo caso, abbiamo bisogno di un "template", il quale viene mostrato nel Listato 5.14.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:text="Nome"
            android:textAppearance="?android:attr/textAppearanceMedium"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/textView1"
            android:textColor="@android:color/black"
            android:layout_weight="1"
            android:layout_marginTop="60dp"
            android:textStyle="bold" />
        <TextView
            android:text="Cognome"
            android:textAppearance="?android:attr/textAppearanceMedium"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/textView2"
            android:layout_marginTop="60dp"
            android:textColor="@android:color/black"
            android:layout_weight="1"
            android:textStyle="bold" />
    </LinearLayout>
    <TextView
        android:text="Settore"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="@android:color/black"
        android:id="@+id/textView3" />
    <TextView
        android:text="Ruolo"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/textView4"
        android:textColor="@android:color/black" />
    <TextView
        android:text="Struttura"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/textView5"
        android:textColor="@android:color/black" />
    <TextView
        android:text="Indirizzo"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/textView6"
        android:textColor="@android:color/black" />
    <TextView
        android:text="Telefono"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/textView7"
        android:textColor="@android:color/black"
        android:autoLink="phone" />
    <TextView
        android:text="E-mail"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/textView8"
        android:textColor="@android:color/black"
        android:autoLink="email" />
</LinearLayout>
```

Listato 5.14. Template `ListView` per i docenti

### 5.4.2 Implementazione del codice C# della pagina relativa ai docenti

Innanzitutto abbiamo bisogno di definire una classe che rappresenti il “docente” all’interno del database. Questo viene mostrato nel Listato 5.15.

```
namespace UNIVPM.Resources.Model
{
    public class Docenti
    {
        [PrimaryKey, AutoIncrement]
        public int CodDoc { get; set; }
        public string Nome { get; set; }
        public string Cognome { get; set; }
        public string Settore { get; set; }
        public string Ruolo { get; set; }
        public string Struttura { get; set; }
        public string Indirizzo { get; set; }
        public string Telefono { get; set; }
        public string Email { get; set; }
    }
}
```

Listato 5.15. Classe Docenti

Ogni variabile di tipo **string** rappresenta un attributo all’interno del database. Per implementare l’activity associata alla pagina abbiamo creato il file `SearchDoc.cs` mostrato nel Listato 5.16.

```
namespace UNIVPM
{
    [Activity(Label = "Ricerca docenti")]
    public class SearchDoc : AppCompatActivity
    {
        ListView lstData;
        List<Docenti> lstSource = new List<Docenti>();
        Database2 db2;
        protected override void OnCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);
            SetContentView(Resource.Layout.page_doc);

            //Create DataBase
            db2 = new Database2();
            db2.createDatabase();
            string folder = System.Environment.GetFolderPath(System.Environment.SpecialFolder.Personal);
            Log.Info("DB_PATH", folder);

            lstData = FindViewById<ListView>(Resource.Id.listView);

            Docenti docente = new Docenti()
            {
                Nome = "Gianni",
                Cognome = "Albertini",
                Settore = "FIS/01- Fisica sperimentale",
                Ruolo = "Professore associato",
                Struttura = "Dip. di Scienze e Ingegneria della Materia dell'Ambiente ed Urbanistica (SIMAU)",
                Indirizzo = "Via Brece Bianche - 60131 Ancona",
                Telefono = "+39 3387099058",
                Email = "g.albertini@univpm.it"
            };
            db2.insertIntoTableDocenti(docente);

            var edtName = FindViewById<EditText>(Resource.Id.editText1);
            var btnSearch = FindViewById<Button>(Resource.Id.button1);

            btnSearch.Click += delegate
            {
                string value = edtName.Text;
                LoadData(value);
            };

            private void LoadData(string value)
            {
                lstSource = db2.selectTableDocenti(value);
                var adapter = new ListViewAdapterDoc(this, lstSource);
                lstData.Adapter = adapter;
            }

            public override bool OnOptionsItemSelected(IMenu menu)
        }
    }
}
```

```

    {
        MenuInflater.Inflate(Resource.Menu.menu_main, menu);
        return true;
    }

    public override bool OnOptionsItemSelected(IMenuItem item)
    {
        int id = item.ItemId;
        if (id == Resource.Id.action_home)
        {
            var activity_home = new Intent(this, typeof(MainActivity));
            StartActivity(activity_home);
        }
        return base.OnOptionsItemSelected(item);
    }
}

```

**Listato 5.16.** Activity della pagina relativa ai docenti

In questa activity viene creato il database e viene inserito un docente di prova. Le istruzioni per creare il database e per inserire il docente possono essere visualizzate nel Listato 5.17. Ora, in base a quello che l'utente digita all'interno dell'elemento **EditText**, l'activity lo cattura in una variabile di tipo **string** e lo passa come parametro alla funzione **LoadData** solo dopo aver premuto il **Button**. Se il docente è presente all'interno del database, allora verranno visualizzate sullo schermo tutte le informazioni a lui relative ove presenti, altrimenti non viene visualizzato nulla e la pagina rimane vuota. Nella funzione **LoadData** viene utilizzata un'altra classe chiamata **ListViewAdapterDoc** che provvede alla visualizzazione delle informazioni. Questa classe viene mostrata nel Listato 5.18.

Come di consuetudine, sono presenti i due metodi **OnOptionsItemSelected** e **OnOptionsItemSelected** per la creazione del menù sulla barra principale.

```

namespace UNIVPM.Resources.DataHelper
{
    class Database2
    {
        string folder = System.Environment.GetFolderPath(System.Environment.SpecialFolder.Personal);
        public bool createDatabase()
        {
            try
            {
                using (SQLiteConnection connection = new SQLiteConnection(System.IO.Path.Combine(
                    folder, "SQLite1.db3")))
                {
                    connection.CreateTable<UNIVPM.Resources.Model.Docenti>();
                    return true;
                }
            }
            catch (SQLiteException ex)
            {
                Log.Info("SQLiteEx", ex.Message);
                return false;
            }
        }

        public bool insertIntoTableDocenti(Model.Docenti docente)
        {
            try
            {
                using (var connection = new SQLiteConnection(System.IO.Path.Combine(folder,
                    "SQLite1.db3")))
                {
                    connection.Insert(docente);
                    return true;
                }
            }
            catch (SQLiteException ex)
            {
                Log.Info("SQLiteEx", ex.Message);
                return false;
            }
        }

        public List<Model.Docenti> selectTableDocenti(string value)
        {
            try
            {

```

```

        using (var connection = new SQLiteConnection(System.IO.Path.Combine(folder,
            "SQLite1.db3")))
        {
            return connection.Query<Model.Docenti>("SELECT * FROM Docenti Where Cognome=?",
                value).ToList();
        }
    }
    catch (SQLiteException ex)
    {
        Log.Info("SQLiteEx", ex.Message);
        return null;
    }
}

public bool deleteTableDocenti()
{
    try
    {
        using (var connection = new SQLiteConnection(System.IO.Path.Combine(folder,
            "SQLite1.db3")))
        {
            connection.DropTable<UNIVPM.Resources.Model.Docenti>();
            return true;
        }
    }
    catch (SQLiteException ex)
    {
        Log.Info("SQLiteEx", ex.Message);
        return false;
    }
}
}
}
}

```

Listato 5.17. Classe Database per i docenti

```

namespace UNIVPM.Resources
{
    public class ViewHolder : Java.Lang.Object
    {
        public TextView txtNome { get; set; }
        public TextView txtCognome { get; set; }
        public TextView txtSettore { get; set; }
        public TextView txtRuolo { get; set; }
        public TextView txtStruttura { get; set; }
        public TextView txtIndirizzo { get; set; }
        public TextView txtTelefono { get; set; }
        public TextView txtEmail { get; set; }
    }

    public class ListViewAdapterDoc : BaseAdapter
    {
        private Activity activity;
        private List<Model.Docenti> lstDocenti;
        public ListViewAdapterDoc(Activity activity, List<UNIVPM.Resources.Model.Docenti> lstDocenti)
        {
            this.activity = activity;
            this.lstDocenti = lstDocenti;
        }

        public override int Count
        {
            get
            {
                return lstDocenti.Count;
            }
        }

        public override Java.Lang.Object GetItem(int position)
        {
            return null;
        }

        public override long GetItemId(int position)
        {
            return lstDocenti[position].CodDoc;
        }

        public override View GetView(int position, View convertView, ViewGroup parent)
        {
            var view = convertView ?? activity.LayoutInflater.Inflate(
                (Resource.Layout.list_view_template_doc, parent, false);

            var txtNome = view.FindViewById<TextView>(Resource.Id.textView1);
            var txtCognome = view.FindViewById<TextView>(Resource.Id.textView2);
            var txtSettore = view.FindViewById<TextView>(Resource.Id.textView3);
            var txtRuolo = view.FindViewById<TextView>(Resource.Id.textView4);
            var txtStruttura = view.FindViewById<TextView>(Resource.Id.textView5);
            var txtIndirizzo = view.FindViewById<TextView>(Resource.Id.textView6);
            var txtTelefono = view.FindViewById<TextView>(Resource.Id.textView7);

```

```

        var txtEmail = view.FindViewById<TextView>(Resource.Id.textView8);

        txtNome.Text = lstDocenti[position].Nome;
        txtCognome.Text = lstDocenti[position].Cognome;
        txtSettore.Text = lstDocenti[position].Settore;
        txtRuolo.Text = lstDocenti[position].Ruolo;
        txtStruttura.Text = lstDocenti[position].Struttura;
        txtIndirizzo.Text = lstDocenti[position].Indirizzo;
        txtTelefono.Text = lstDocenti[position].Telefono;
        txtEmail.Text = lstDocenti[position].Email;

        return view;
    }
}

```

Listato 5.18. Classe ListViewAdapterDoc

## 5.5 Pagina relativa alle mappe

Per quanto riguarda le mappe sono stati implementati due tipi di layout; il primo comprende uno *slider* di immagini, quindi permette la visualizzazione della planimetria; il secondo comprende un elenco con i link per poter scaricare il pdf contenente la planimetria scelta.

In questo caso, abbiamo quattro layout e quattro activity poichè, per la Facoltà di Agraria, le planimetrie sono le stesse della Facoltà di Ingegneria.

### 5.5.1 Implementazione del layout della pagina relativa alle mappe

Nel Listato 5.19 viene mostrato il codice relativo alla parte grafica.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <android.support.design.widget.BottomNavigationView
        android:id="@+id/maps_eco_top_nav"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="?android:attr/windowBackground"
        app:menu="@menu/page_maps_menu" />
    <RelativeLayout
        android:orientation="vertical"
        android:id="@+id/relativeLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#ffffff">
        <android.support.v4.view.ViewPager
            android:minWidth="25dp"
            android:minHeight="25dp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_above="@id/imageView1"
            android:id="@+id/viewPager" />
        <ImageView
            android:src="@drawable/img_eco_legend"
            android:layout_width="wrap_content"
            android:layout_height="120.0dp"
            android:id="@+id/imageView1"
            android:layout_above="@id/textView1" />
        <TextView
            android:text="UNIVERSITA' POLITECNICA DELLE MARCHE"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:minWidth="25dp"

```



```

        android:minHeight="25dp"
        android:id="@+id/textView1"
        android:gravity="center_horizontal"
        android:textColor="#ffffff"
        android:background="#ff616161"
        android:layout_above="@+id/textView2"
        android:paddingTop="5dp" />
<TextView
    android:text="Piazza Roma 21, 60121 Ancona"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:minWidth="25dp"
    android:minHeight="25dp"
    android:id="@+id/textView2"
    android:gravity="center_horizontal"
    android:textColor="#ffffff"
    android:background="#ff616161"
    android:layout_above="@+id/textView3"
    android:paddingTop="5dp" />
<TextView
    android:text="tel: (+39) 071 220 21"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:minWidth="25dp"
    android:minHeight="25dp"
    android:id="@+id/textView3"
    android:paddingBottom="5dp"
    android:gravity="center_horizontal"
    android:textColor="#ffffff"
    android:background="#ff616161"
    android:layout_alignParentBottom="true"
    android:paddingTop="5dp"
    android:autoLink="phone"
    android:textColorLink="#ffffff" />
</RelativeLayout>
</LinearLayout>

```

**Listato 5.19.** Layout della pagina relativa alle planimetrie della Facoltà di Economia

Per la parte grafica abbiamo utilizzato un **LinearLayout** con attributo **match\_parent** su altezza e larghezza in modo da occupare tutto il display a disposizione. Al suo interno è stato inserito un **RelativeLayout**, che ci consente il giusto posizionamento dei vari elementi nella pagina.

Lo slider di immagini, menzionato poco fa, è stato realizzato grazie all'utilizzo del tag **android.support.v4.view.ViewPager**, mentre la sua leggenda è stata inserita grazie al tag **ImageView**. Infine, abbiamo il solito “footer” al termine della pagina.

### 5.5.2 Implementazione del codice C# della pagina relativa alle mappe

Nel Listato 5.20 viene mostrato il codice relativo all'activity con le planimetrie della Facoltà di Economia.

```

namespace UNIVPM
{
    [Activity(Label = "Mappe Economia")]
    public class Map_Eco : AppCompatActivity, BottomNavigationView.IOnNavigationItemSelectedListener
    {
        protected override void onCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);
            SetContentView(Resource.Layout.page_maps_eco);

            BottomNavigationView nav_eco = FindViewById<BottomNavigationView>
                (Resource.Id.maps_eco_top_nav);
            nav_eco.SetOnNavigationItemSelectedListener(this);
            var viewPager = FindViewById<ViewPager>(Resource.Id.viewPager);
            ImageSliderEco slider = new ImageSliderEco(this);
            viewPager.Adapter = slider;
        }
        public bool OnNavigationItemSelectedListener(IMenuItem item)
        {
            switch (item.ItemId)
            {
                case Resource.Id.nav_ing:
                    var activity_ing = new Intent(this, typeof(Seg_Inng));

```

```

        StartActivity(activity_ing);
        return true;
    case Resource.Id.nav_med:
        var activity_med = new Intent(this, typeof(Map_Med));
        StartActivity(activity_med);
        return true;
    case Resource.Id.nav_sci:
        var activity_sci = new Intent(this, typeof(Map_Scie));
        StartActivity(activity_sci);
        return true;
    }
    return false;
}
}
}

```

**Listato 5.20.** Activity della pagina relativa alle mappe della Facoltà di Economia

L'activity sopra riportata è costituita da due importanti elementi:

1. la **BottomNavigationView**, che ci consente lo spostamento tra i vari layout delle segreterie;
2. lo "slider"; creato mediante la dichiarazione di un oggetto di tipo **ImageSliderEco** da noi realizzato.

Nel Listato 5.21 viene mostrato il codice C# della classe **ImageSliderEco**.

```

namespace UNIVPM
{
    public class ImageSliderEco : PagerAdapter
    {
        private Context context;
        private int[] imageList =
        {
            Resource.Drawable.img_maps_eco_semint,
            Resource.Drawable.img_maps_eco_0,
            Resource.Drawable.img_maps_eco_0d,
            Resource.Drawable.img_maps_eco_1,
            Resource.Drawable.img_maps_eco_2,
            Resource.Drawable.img_maps_eco_3,
        };

        public ImageSliderEco (Context context)
        {
            this.context = context;
        }

        public override int Count
        {
            get
            {
                return imageList.Length;
            }
        }

        public override bool IsViewFromObject(View view, Java.Lang.Object @object)
        {
            return view == ((ImageView)@object);
        }

        public override Java.Lang.Object InstantiateItem(View container, int position)
        {
            ImageView imageView = new ImageView(context);
            imageView.SetScaleType(ImageView.ScaleType.FitCenter);
            imageView.SetImageResource(imageList[position]);
            ((ViewPager)container).AddView(imageView, 0);
            return imageView;
        }

        public override void DestroyItem(View container, int position, Java.Lang.Object @object)
        {
            ((ViewPager)container).RemoveView((ImageView)@object);
        }
    }
}

```

**Listato 5.21.** Classe **ImageSliderEco**

Per prima cosa, abbiamo definito un vettore, chiamato **imageList**, contenente le immagini da visualizzare. Successivamente, abbiamo una funzione per il **context**

e una che restituisce la lunghezza del vettore dichiarato precedentemente. Infine, viene assegnato il vettore delle immagini allo slider e viene specificato come e dove visualizzare le immagini stesse, utilizzando `SetScaleType` e `SetImageResource`.

## 5.6 Pagina relativa alle segreterie

Per implementare le pagine relative alle segreterie abbiamo utilizzato un'activity ed un layout per ciascuna di essa; tramite il metodo `SetContentView`, abbiamo richiamato il layout corrispondente alla segreteria scelta. Quindi, in totale, avremo cinque layout, uguali tra loro, e cinque activity, necessarie per la visualizzazione e lo spostamento tra le varie segreterie.

### 5.6.1 Implementazione del layout della pagina relativa alle segreterie

Nel Listato 5.22 viene mostrato il codice relativo alla pagina “Segreterie” per lo sviluppo della parte grafica.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >
    <android.support.design.widget.BottomNavigationView
        android:id="@+id/nav_seg_agr"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="?android:attr/windowBackground"
        app:menu="@menu/page_seg_menu" />
    <TextView
        android:text="Contatti segreteria di Agraria - Ancona"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="5dp"
        android:paddingBottom="5dp"
        android:paddingLeft="5dp"
        android:id="@+id/textView1"
        android:gravity="center_vertical"
        android:background="#8b0000"
        android:textStyle="bold"
        android:textColor="#ffffff"
        android:textSize="20dp" />
    <RelativeLayout
        android:orientation="vertical"
        android:id="@+id/relativeLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#d9d9d9">
        <TextView
            android:text="Ufficio Corsi di Studio dell'Area di Agraria e Scienze"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:minWidth="25dp"
            android:minHeight="25dp"
            android:id="@+id/text1"
            android:textColor="#000000"
            android:paddingTop="5dp"
            android:paddingLeft="5dp"
            android:textSize="20dp" />
        <TextView
            android:text="Via Breccie Bianche - 60131 Ancona"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:minWidth="25dp"
            android:minHeight="25dp"
            android:id="@+id/text2"
            android:textColor="#000000"
            android:paddingTop="5dp"
            android:paddingLeft="5dp"
            android:textSize="16dp" />
    </RelativeLayout>
</LinearLayout>
```

```

        android:layout_below="@+id/text1" />
<TextView
    android:text="Orario di apertura degli Uffici"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:minWidth="25dp"
    android:minHeight="25dp"
    android:id="@+id/text3"
    android:textColor="#000000"
    android:paddingTop="15dp"
    android:paddingBottom="15dp"
    android:paddingLeft="5dp"
    android:textSize="18dp"
    android:textStyle="bold"
    android:layout_below="@+id/text2" />
</RelativeLayout>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:id="@+id/relativeLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#d9d9d9" >
    <TextView
        android:text="da gennaio ad agosto"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/textView4"
        android:layout_weight="1"
        android:textColor="#000000"
        android:textStyle="bold"
        android:paddingLeft="5dp"
        android:paddingBottom="5dp" />
    <TextView
        android:text="da settembre a dicembre"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/textView5"
        android:layout_weight="1"
        android:textColor="#000000"
        android:textStyle="bold"
        android:paddingBottom="5dp" />
</LinearLayout>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:id="@+id/relativeLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#d9d9d9">
    <TextView
        android:text="11:00 - 13:00 Lunedì e Giovedì"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/textView4"
        android:layout_weight="1"
        android:textColor="#000000"
        android:paddingLeft="5dp"
        android:paddingBottom="5dp" />
    <TextView
        android:text="10:00 - 13:00 Lunedì, Martedì, Giovedì, Venerdì"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/textView5"
        android:layout_weight="1"
        android:textColor="#000000"
        android:paddingBottom="5dp" />
</LinearLayout>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:id="@+id/relativeLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#d9d9d9">
    <TextView
        android:text="15:00 - 16:30 Mercoledì"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/textView4"
        android:layout_weight="1"
        android:textColor="#000000"
        android:paddingLeft="5dp" />
    <TextView
        android:text="15:00 - 16:30 Mercoledì"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/textView5"
        android:layout_weight="1"
        android:textColor="#000000" />
</LinearLayout>
<TextView
    android:text="Tel.: +39 0712204341"
    android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content"
        android:id="@+id/textView5"
        android:textColor="#000000"
        android:paddingTop="35dp"
        android:background="#d9d9d9"
        android:paddingLeft="5dp"
        android:autoLink="phone" />
<TextView
    android:text="Fax: +39 0712204133"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/textView5"
    android:textColor="#000000"
    android:paddingTop="10dp"
    android:background="#d9d9d9"
    android:paddingLeft="5dp"
    android:autoLink="phone" />
<TextView
    android:text="e-mail: segreteria.agraria-scienze@univpm.it"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/textView5"
    android:textColor="#000000"
    android:paddingTop="10dp"
    android:background="#d9d9d9"
    android:paddingLeft="5dp"
    android:autoLink="email" />
<RelativeLayout
    android:orientation="vertical"
    android:id="@+id/relativeLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#d9d9d9" >
    <TextView
        android:text="UNIVERSITA' POLITECNICA DELLE MARCHE"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:minWidth="25dp"
        android:minHeight="25dp"
        android:id="@+id/textView1"
        android:gravity="center_horizontal"
        android:textColor="#ffffff"
        android:background="#ff616161"
        android:layout_above="@+id/textView2"
        android:paddingTop="5dp" />
    <TextView
        android:text="Piazza Roma 21, 60121 Ancona"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:minWidth="25dp"
        android:minHeight="25dp"
        android:id="@+id/textView2"
        android:gravity="center_horizontal"
        android:textColor="#ffffff"
        android:background="#ff616161"
        android:layout_above="@+id/textView3"
        android:paddingTop="5dp" />
    <TextView
        android:text="tel: (+39) 071 220 21"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:minWidth="25dp"
        android:minHeight="25dp"
        android:id="@+id/textView3"
        android:gravity="center_horizontal"
        android:textColor="#ffffff"
        android:background="#ff616161"
        android:layout_alignParentBottom="true"
        android:paddingTop="5dp"
        android:textColorLink="#ffffff"
        android:autoLink="phone" />
</RelativeLayout>
</LinearLayout>

```

**Listato 5.22.** Layout della pagina relativa alla segreteria della Facoltà di Agraria

Le `TextView` rappresentano le varie caselle di testo. Alcune di loro hanno un tag chiamato `android:layout_weight`; quest'ultimo permette di dare un "peso" a quella `TextView` in modo da poterla disporre in modo equo, insieme alle altre, in orizzontale o in verticale, all'interno della pagina. Anche in questa pagina è presente una `BottomNavigationView` che permette la connessione dei vari layout delle segreterie.

### 5.6.2 Implementazione del codice C# della pagina relativa alle segreterie

Per poter usufruire di questa funzionalità è stata creata un'activity per ogni segreteria. Nel Listato 5.23 viene mostrato il codice relativo alla segreteria della Facoltà di Agraria.

```
namespace UNIVPM
{
    [Activity(Label = "Segreteria Agraria")]
    public class Seg_Agr : AppCompatActivity, BottomNavigationView.INavigationItemSelectedListener
    {
        protected override void onCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);
            SetContentView(Resource.Layout.page_seg_agr);

            BottomNavigationView navigation = FindViewById<BottomNavigationView>(Resource.Id.nav_seg_agr);
            navigation.SetOnNavigationItemSelectedListener(this);
        }

        public bool OnNavigationItemSelectedListener(IMenuItem item)
        {
            switch (item.ItemId)
            {
                case Resource.Id.nav_eco:
                    var activity_eco = new Intent(this, typeof(Seg_Eco));
                    StartActivity(activity_eco);
                    return true;
                case Resource.Id.nav_ing:
                    var activity_ing = new Intent(this, typeof(Seg_Ing));
                    StartActivity(activity_ing);
                    return true;
                case Resource.Id.nav_med:
                    var activity_med = new Intent(this, typeof(Seg_Med));
                    StartActivity(activity_med);
                    return true;
                case Resource.Id.nav_sci:
                    var activity_sci = new Intent(this, typeof(Seg_Scie));
                    StartActivity(activity_sci);
                    return true;
            }
            return false;
        }
    }
}
```

**Listato 5.23.** Activity della pagina relativa alla segreteria della Facoltà di Agraria

Per visualizzare le informazioni della segreteria è stata utilizzata l'activity sopra riportata. È stata adottata una **BottomNavigationView** che permette di spostarsi tra le varie segreterie; essa, grazie al metodo **OnNavigationItemSelectedListener**, tramite l'id ed uno **switch**, ci consente lo spostamento tra i vari layout collegati.

## 5.7 Manuale del sistema

In questa sezione, descriviamo alcune funzionalità del nostro sistema. In particolare:

- In Figura 5.3 vediamo la schermata relativa alla Home Page con l'elenco degli eventi; su di essi non è possibile cliccare poiché tutte le informazioni vengono già riportate nell'elenco.
- In Figura 5.4 mostriamo la schermata relativa al menù a scomparsa laterale che permette lo spostamento tra le varie funzionalità.
- Nelle Figure 5.5 e 5.6 vengono mostrate le schermate relative alla visualizzazione delle offerte formative e degli orari della Facoltà di Agraria.

- In Figura 5.7 mostriamo la possibilità di ricercare un docente all'interno del database; in Figura 5.8 mostriamo la possibilità di effettuare una chiamata. Si può, anche, inviare una e-mail ma questa funzionalità verrà esposta in seguito.
- In Figura 5.9 mostriamo la schermata relativa alla visualizzazione delle planimetrie della Facoltà di Economia, con la possibilità di poter scorrere le immagini da sinistra a destra e viceversa, grazie allo *slider* introdotto in precedenza.
- In Figura 5.10 mostriamo la schermata relativa alla visualizzazione delle informazioni della segreteria di Agraria; infine, in Figura 5.11, mostriamo la possibilità di inviare direttamente una mail all'indirizzo della segreteria tramite il proprio client di e-mail.



**Figura 5.3.** Home Page ed elenco degli eventi

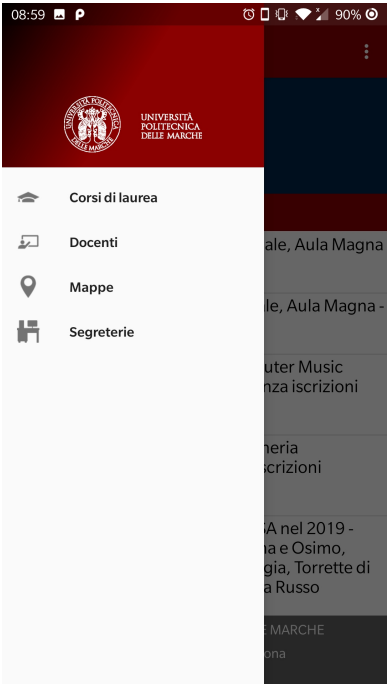


Figura 5.4. Menù a scomparsa laterale

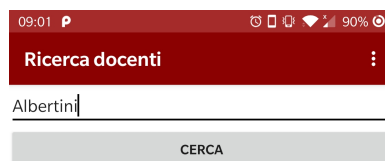


Figura 5.5. Offerte Formative della Facoltà di Agraria





Figura 5.6. Orari della Facoltà di Agraria



**Gianni Albertini**  
 FIS/01- Fisica sperimentale  
 Professore associato  
 Dip. di Scienze e Ingegneria della Materia  
 dell'Ambiente ed Urbanistica (SIMAU)  
 Via Brecce Bianche - 60131 Ancona  
 +39 3387099058  
 g.albertini@univpm.it

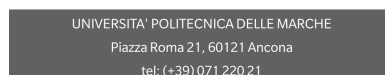


Figura 5.7. Visualizzazione di un docente

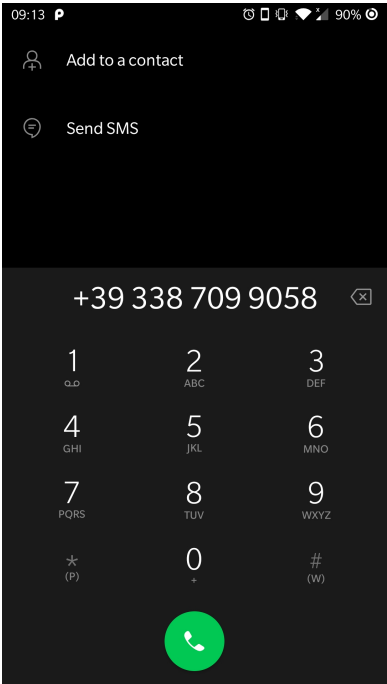


Figura 5.8. Chiamata ad un docente cercato

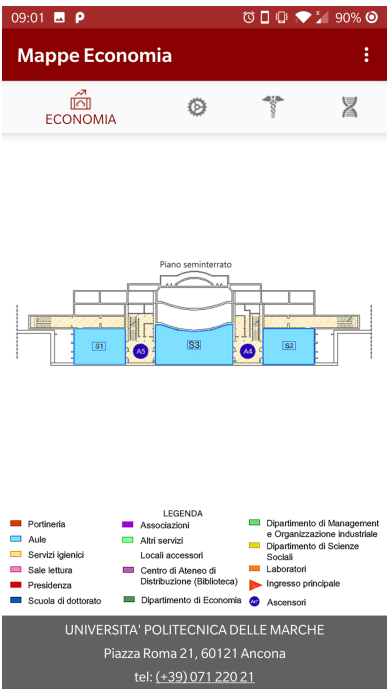
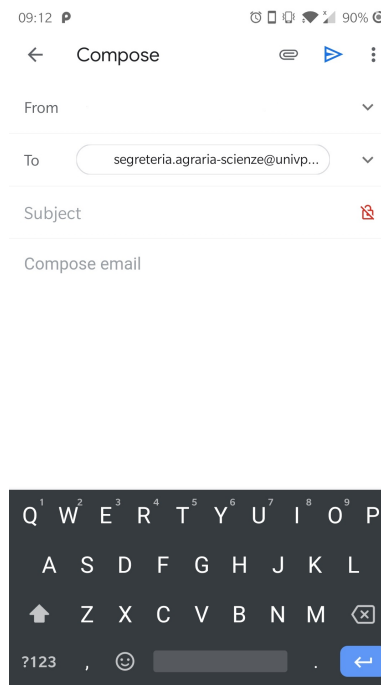


Figura 5.9. Planimetrie della Facoltà di Economia



**Figura 5.10.** Informazioni della segreteria della Facoltà di Agraria



**Figura 5.11.** Invio di una e-mail all'indirizzo della segreteria



## Conclusioni

Il progetto descritto nella presente tesi ci ha portato a realizzare un'app mobile, per Android, a supporto degli studenti, e non, dell'Università Politecnica delle Marche.

Per garantire una maggiore qualità, come in ogni progetto software, è stato necessario effettuare una prima parte dedicata alla progettazione. In particolare, abbiamo effettuato un'analisi dei requisiti, per individuare le funzionalità che dovevano essere implementate nel sistema; successivamente, si è passati alla progettazione della componente dati ed, infine, alla progettazione della componente applicativa.

Al termine della progettazione, si è passati all'implementazione dell'applicazione. Per questa fase abbiamo utilizzato il framework Xamarin, attraverso l'ambiente di sviluppo Visual Studio.

La prima scelta che abbiamo dovuto effettuare ha riguardato quale strumento di Xamarin utilizzare. La scelta è ricaduta su *Xamarin.Android*, in quanto, rispetto a *Xamarin.Forms*, consente di implementare funzionalità specifiche per il sistema operativo preso in considerazione, come, ad esempio, l'utilizzo di diverse *NavigationView*, o la possibilità di poter creare e modificare un database interno all'app per la realizzazione delle varie funzionalità illustrate nella parte riguardante la progettazione.

Questa scelta ha innescato una problematica, ovvero quella riguardante la non possibilità di riutilizzare il codice per le altre piattaforme.

Infine, durante la fase di implementazione, abbiamo descritto, ed analizzato, il funzionamento del codice delle varie pagine dell'applicazione.

Per quanto riguarda i possibili sviluppi futuri di questo lavoro di tesi, si potrebbe, in primo luogo, si potrebbe implementare una parte dedicata al login dello studente utilizzando i database di Esse3; in questo modo, si avrebbe la possibilità di consultare il proprio libretto elettronico, visualizzare la media aritmetica e ponderata del proprio percorso universitario e, cosa molto importante, visualizzare le date degli esami ed iscriversi ad essi senza l'utilizzo del browser.

In secondo luogo, bisogna considerare che questa app è stata sviluppata solo per Android. In futuro, potremmo sviluppare anche delle versioni per Apple e Windows Phone, coprendo, quindi, i tre principali sistemi operativi presenti sul mercato. Il raggiungimento di tale obiettivo viene facilitato da Xamarin, poiché, la logica di business, è completamente condivisibile con le altre piattaforme. Per questo motivo, l'unico "grande" lavoro da effettuare riguarda la realizzazione delle interfacce

grafiche per Apple e Windows Phone, che vanno implementate, rispettivamente, in Storyboards e in XAML.

---

## Riferimenti bibliografici

1. Cos'è una Web Mobile App. <https://glue-labs.com/articoli/cosa-e-fa-una-mobile-web-app>.
2. Definition of Use Case. <https://www.techopedia.com/definition/25813/use-case>.
3. Il modello Entità-Relazione. <http://www.dacrema.com/Informatica/er.htm>.
4. Qual è la differenza tra Wireframe, Prototipo e Mockup? <http://www.fabiomarasco.it/la-differenza-tra-wireframe-prototipo-e-mockup/>.
5. What is a Flow Chart? <https://asq.org/quality-resources/flowchart>.
6. Xamarin – Cos'è, come funziona, i vantaggi. <https://www.eliformit.it/2017/11/15/xamarin/>.
7. Xamarin.Android. <https://docs.microsoft.com/it-it/xamarin/android/>.
8. Introduzione a Xamarin. <https://social.technet.microsoft.com/wiki/contents/articles/33618-introduzione-a-xamarin-it-it.aspx>, 2016.
9. Introduzione a Xamarin Framework. <https://www.html.it/pag/60443/introduzione-a-xamarin-framework/>, 2017.
10. Benvenuti all'IDE di Visual Studio. <https://docs.microsoft.com/it-it/visualstudio/get-started/visual-basic/visual-studio-ide?view=vs-2019#customize-visual-studio>, 2018.
11. App mobile ibride, native o web: le differenze. <https://www.html.it/faq/app-mobile-ibride-native-o-web-le-differenze/>, 2019.