



# POLITECNICO DI BARI

## DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELL'INFORMAZIONE

### CORSO DI LAUREA TRIENNALE IN INGEGNERIA INFORMATICA E DELL'AUTOMAZIONE

---

Business Report  
Work Project

Software engineering

# easyBUY

**Docente:**  
Prof.ssa Ing. Marina Mongiello

**Codice Gruppo: 16**

**Componenti gruppo:**

Ettore Quatraro  
Paolo Marchitelli  
Luana Masciulli  
Gabriele Nardomarino  
Daniele Pugliese

---

Anno Accademico 2020/2021

# Indice

|   |           |
|---|-----------|
| <b>Sintesi del progetto.....</b>                        | <b>3</b>  |
| <b>Obbiettivi .....</b>                                 | <b>3</b>  |
| <b>Scopo .....</b>                                      | <b>3</b>  |
| <b>Struttura del progetto .....</b>                     | <b>3</b>  |
| <b>Introduzione.....</b>                                | <b>4</b>  |
| <b>Dominio di interesse.....</b>                        | <b>4</b>  |
| <b>Perché un'applicazione? .....</b>                    | <b>4</b>  |
| <b>Perché una connessione a internet? .....</b>         | <b>5</b>  |
| <b>Piattaforma e Tecnologie .....</b>                   | <b>5</b>  |
| <b>Android Studio .....</b>                             | <b>5</b>  |
| <b>Java .....</b>                                       | <b>5</b>  |
| <b>Altevista e database .....</b>                       | <b>5</b>  |
| <b>Validazione.....</b>                                 | <b>6</b>  |
| <b>Soluzione .....</b>                                  | <b>7</b>  |
| <b>Risultati e discussione.....</b>                     | <b>10</b> |
| <b>Considerazioni conclusive e sviluppi futuri.....</b> | <b>11</b> |
| <b>Bibliografia .....</b>                               | <b>12</b> |

# Sintesi del progetto

## Obbiettivi

Gli obiettivi che il nostro gruppo vuole raggiungere sono:

- *Imparare a lavorare in team utilizzando piattaforme di confronto a distanza che stanno diventando, anche grazie all'emergenza COVID-19, di uso comune nell'ambiente lavorativo*

L'obiettivo è stato raggiunto, infatti, settimanalmente, il gruppo di progetto si è riunito virtualmente sulla piattaforma MS Teams per confrontarsi sui compiti da svolgere e sull'andamento generale del progetto, una volta svolti i compiti ogni singolo componente caricava la parte del proprio lavoro su una cartella condivisa creata su OneDrive.

- *Pensare, studiarne funzioni e fattibilità, progettare e realizzare un App per dispositivi Android pensata come supporto per situazioni di vita quotidiana*

L'obiettivo è stato conseguito.

Il frutto del nostro lavoro è una app in grado di facilitare e velocizzare l'acquisto di generi alimentari o prodotti in generale.

## Scopo

Lo scopo è quello di realizzare un prodotto software in grado di soddisfare un bisogno e aiutare chi lo utilizza nella quotidianità, in particolare l'idea è nata dalla necessità, durante l'emergenza COVID-19, di approvvigionarsi di beni alimentari rapidamente.

Quello che abbiamo notato è che effettuare la spesa in tempi così limitati entrava in contrasto con la ricerca del prezzo e del prodotto migliore, la soluzione a tutto ciò sarebbe stata una app per smartphone in grado di programmare la spesa stando comodamente a casa per poi guidare il cliente tra i vari reparti per raccogliere tutti i prodotti scelti.

## Struttura del progetto

Tra la scelta del cosa realizzare e l'effettiva realizzazione c'è stata una fase di progettazione che riassumiamo brevemente di seguito:

1. Stesura di una presentazione PowerPoint per mostrare al committente (In questo caso i professori del corso) cosa si andrà a realizzare in modo da avere l'approvazione per andare avanti
2. Realizzazione dello studio di fattibilità per prevedere in anticipo i tempi da rispettare per ogni consegna
3. Scelta dei requisiti funzionali e non funzionali che deve avere il prodotto finito
4. Bozza del funzionamento dell'app attraverso il diagramma dei casi d'uso

5. Presentazione del prototipo e del diagramma degli stati
6. Presentazione della beta non testata con la rispettiva documentazione che spiega più nei particolari la fase implementativa
7. Report test e versione finale del software ripulito di tutti i bug venuti fuori dalla fase di test

## Introduzione

**easyBUY** è un'applicazione pensata per dispositivi Android che nasce dall'idea di facilitare e velocizzare l'acquisto di generi alimentari o prodotti in generale.

L'app ti aiuta nella ricerca e nella selezione dei prodotti potendo scegliere tra l'offerta migliore, confrontando i prezzi tra i negozi iscritti all'app.

L'innovazione consiste nel poter creare liste di prodotti da acquistare, creare carrelli dopo aver confrontato il prezzo dello stesso prodotto e calcolare il percorso minore all'interno del centro commerciale così che si possa spendere il minor tempo possibile per acquistare tutti i prodotti nel carrello.

Consente inoltre di condividere le liste o i carrelli con altri utenti iscritti all'app, visionare tutti i prodotti all'interno di un supermercato con la relativa mappa e cercare un determinato prodotto tramite nome o scannerizzando il codice a barre presente su di esso.

## Dominio di interesse

Dopo una fase di brainstorming è stato estrapolato il dominio di interesse ovvero il contesto in cui verrà utilizzata **easyBUY**.

Tale applicazione è pensata per tutte quelle persone che, dotate di uno smartphone connesso a Internet, utilizzano un metodo obsoleto e inefficiente per fare la spesa.

In particolare, l'applicazione è indirizzata alle persone che per motivi lavorativi o personali hanno poco tempo da dedicare agli acquisti finendo quindi per non acquistare tutti i prodotti desiderati o acquistarli in maniera disorganizzata perdendo tempo e senza cogliere il prezzo migliore.

Inoltre, anche coloro che non possiedono temporaneamente una connessione dati possono usufruire del servizio visionando le proprie liste e carrelli di prodotti purché aggiunti in precedenza.

### Perché un'applicazione?

Perché al giorno d'oggi è molto più frequente utilizzare il nostro smartphone piuttosto che carta e penna quindi ci siamo chiesti perché non trasferire il comune gesto di scrivere la lista della spesa sul nostro smartphone aggiungendo funzionalità smart che migliorino il modo classico di fare la spesa.

### **Perché una connessione a internet?**

Per aggiungere delle funzionalità smart come detto in precedenza è nata la necessità di memorizzare i dati della nostra applicazione su un database centrale, al quale tutti gli utenti che utilizzano l'app accedono per scaricare e caricare informazioni.

## **Piattaforma e Tecnologie**

### **Android Studio**

Per lo sviluppo dell'applicazione abbiamo utilizzato l'IDE Android Studio, applicativo disponibile gratuitamente e offerto da Google e pensato specificatamente per programmare applicazioni che verranno usate sul sistema operativo Android, di cui Google stessa si occupa dello sviluppo.

Esso ci ha permesso di poter programmare con comodità e testare ogni parte della nostra applicazione tramite un simulatore o direttamente sul nostro smartphone.

Permette di generare file APK che possono essere utilizzati per installare l'app anche su dispositivi non direttamente collegati all'IDE e offre numerosi strumenti che permettono di aiutarti nello sviluppo e rendere l'app funzionante sul maggior numero di dispositivi possibili.

### **Java**

Il linguaggio di programmazione utilizzato è Java, un linguaggio orientato agli oggetti, che era già nelle nostre conoscenze e funzionale allo scopo, infatti la maggior parte delle applicazioni Android e il sistema operativo Android stesso utilizzano proprio questo linguaggio.

### **Altervista e database**

Per quanto riguarda la gestione dei dati degli utenti, e il reperimento da parte di quest'ultimi delle informazioni relative a prodotti e supermercati, abbiamo deciso di utilizzare un database esterno risiedente su Altervista.

Altervista è una piattaforma web italiana tramite la quale è possibile aprire gratuitamente un sito web e avere accesso a varie funzionalità, nel nostro caso specifico ci interessa la gestione di un database al quale la nostra applicazione si interfaccia tramite dei file php che ci permettono di gestire le richieste che vengono effettuate ad esso.

Su Altervista è possibile accedere ad un gestore dotato di interfaccia grafica chiamato phpMyAdmin, molto diffuso e semplice da utilizzare, tramite il quale si può amministrare il database in ogni sua parte, ad esempio è possibile inserire contenuto al suo interno o modificarne le tabelle.

Chiaramente l'utilizzo di un database esterno obbligherà l'utente quasi sempre ad essere connesso ad internet, questo tuttavia non è un problema poiché al giorno d'oggi il suo utilizzo è ampiamente diffuso se non fondamentale per una enorme fetta della popolazione.

Il database è costruito tramite il modello relazionale, il linguaggio utilizzato per crearlo concretamente invece è sql, ovvero il più diffuso in questo momento storico.

# Validazione

La validazione di un progetto è la verifica che tutto all'interno di esso funzioni correttamente e che tutti gli obiettivi prefissati siano stati implementati. I diversi criteri di verifica prevedono che la stessa validazione venga condotta seguendo i seguenti aspetti:

- a) Affidabilità e resistenza all'uso prolungato;
- b) Completezza;
- c) Leggibilità, coerenza e facilità di utilizzo;
- d) Compatibilità.

Per quanto riguarda l'affidabilità dell'app, essa non richiede che ci sia una elevata sicurezza in quanto i dati personali richiesti per la registrazione degli utenti sono soltanto nome, cognome, sesso, data di nascita e e-mail dell'utente che si sta registrando.

Abbiamo inoltre testato la resistenza dell'app all'uso prolungato e con un accesso elevato di utenti. È stato verificato che l'app continua a funzionare in maniera fluida, senza che si verifichino errori anche se essa viene usata per molto tempo; questo dipende dal fatto che le activity vengono chiuse quando non più visibili così da non occupare memoria RAM. È stato inoltre verificato che il funzionamento dell'app con un accesso elevato di utenti non dipende direttamente dall'app ma dalla qualità del servizio di hosting del database che in questo momento è Altvista.

A proposito della completezza dell'app, ovvero l'essere strutturata e implementata correttamente, durante la stesura del codice si sono presentati diversi bug che abbiamo risolto. Di conseguenza, possiamo affermare che dopo la risoluzione di questi bug, l'app non va in crash e continua a funzionare correttamente.

Per quanto riguarda la leggibilità, la coerenza e la facilità di utilizzo abbiamo strutturato l'intera app intorno a questi requisiti, infatti, per tutta l'app è stata seguita la stessa logica per quanto riguarda i colori, il font delle scritte e la posizione degli oggetti nell'interfaccia. Essi risultano essere intuitivi o eventualmente accompagnati da scritte che ne indicano la funzione, laddove non lo siano. Abbiamo anche verificato la compatibilità della nostra app sui diversi dispositivi ed è emerso che il corretto funzionamento dell'app è garantito dalla API 28 in poi (Android 9.0) pur funzionando anche sulle precedenti.

Concludendo quindi, possiamo affermare che l'app easyBUY risulta essere stata implementata correttamente e che non ci sono bug che ne compromettono l'utilizzo.

## Soluzione

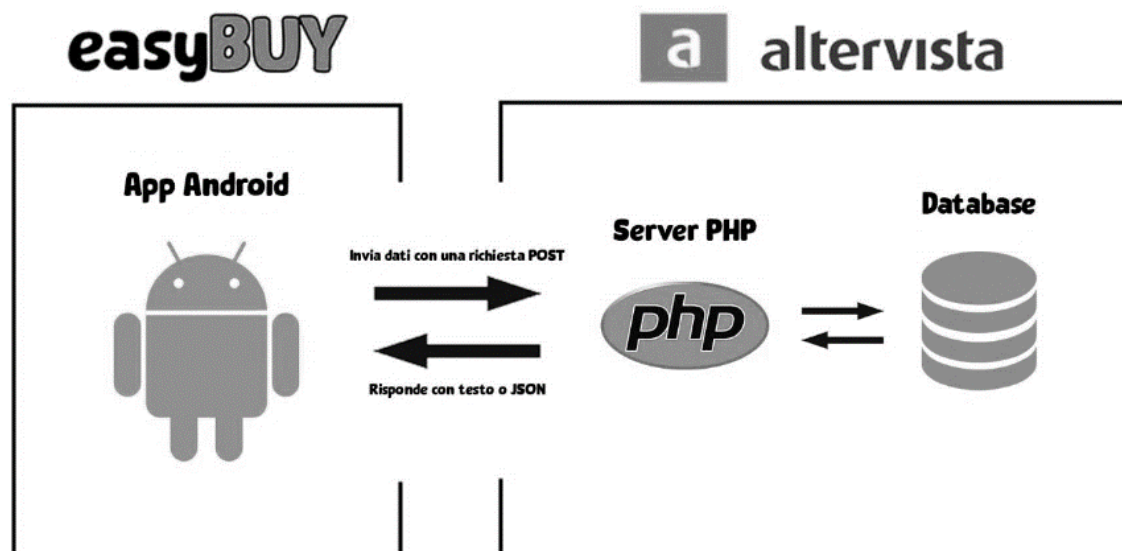
Per funzionare correttamente l'app ha bisogno di accedere alle informazioni contenute nel database.

Da ciò è sorto il primo problema: trovare la disposizione più corretta e ottimizzata per gestire le informazioni.

La soluzione trovata è stata la creazione di due database: uno esterno ed uno interno.

Il database esterno è necessario per manipolare i dati e centralizzare le informazioni così che possano essere elaborate in maniera uguale e indistinta da tutte le app installate.

Abbiamo sfruttato il servizio Altervista, che ci metteva a disposizione un web server, sul quale abbiamo scritto delle API PHP, ed un database. Per recuperare i dati, la nostra app, invia una POST request ad una pagina PHP; una volta recuperati i dati dal database, il server PHP invia indietro un testo semplice o in formato JSON contenente le informazioni richieste.



Durante l'implementazione del database ci siamo imbattuti in alcuni imprevisti che ci hanno fatto optare per alcune soluzioni tecniche a discapito di altre.

L'implementazione di un database esterno all'applicazione è fondamentale per il funzionamento ottimale di essa, basti pensare che senza di esso anche la condivisione dei carrelli e delle liste fra utenti non sarebbe potuta avvenire, motivo per cui abbiamo optato per il servizio di hosting di Altervista come descritto in precedenza. Nonostante ciò quest'ultima piattaforma ci ha imposto alcuni limiti implementativi descritti di seguito:

- Non è possibile accedere direttamente al database esterno, ma solamente alle cartelle o file presenti nel nostro spazio Altervista. Per questo motivo abbiamo provveduto alla

creazione di alcuni file PHP alla quale è possibile passare tramite il metodo POST la query da eseguire nel database ed in seguito stampare le tuple risultanti nel medesimo file.

#### *ESEMPIO FILE PHP*

```
<?php
$DBhost = "localhost";
$DBName = "my_easybuy20";
/* Connettiamoci al database */
mysql_connect($DBhost/*,$DBuser,$DBpass*/) or die("Impossibile collegarsi al server");
mysql_select_db($DBName) or die("Impossibile connettersi al database $DBName");
$sqlquery = $_POST["query"];
*$result = mysql_query($sqlquery);
echo $sqlquery;?>
```

- Impossibilità di recuperare le immagini direttamente dal database, tramite la soluzione precedentemente descritta siamo limitati ad ottenere l'output del database tramite il comando "echo" che traduce le tuple in stringhe. Quindi abbiamo convertito il tipo dell'attributo immagine da BLOB a varchar che quindi conterrà solo il nome del file dell'immagine tramite il quale un metodo apposito nella nostra app scaricherà le immagini richieste in background.
- Non sono implementati i trigger sul database, di conseguenza quando un utente elimina un carrello (lista) sul suo dispositivo viene eliminata la tupla corrispondente nella tabella che associa l'utente al carrello(lista) ma non nella tabella dei carrelli(liste). Tramite l'utilizzo di un trigger, invece, quando anche l'ultimo utente associato a quel carrello (lista) veniva eliminato, veniva eliminato anche il carrello (lista) corrispondente.

Tutti i problemi esposti possono essere risolti tramite l'acquisto di un server e relativo database esterno che non ci limiti queste funzionalità.

Nel database interno, invece, viene salvata una copia speculare del database esterno e vengono memorizzate le credenziali di accesso dell'utente così che l'accesso sia effettuato automaticamente alla riapertura dell'app.

Dal brainstorming teorico iniziale abbiamo pensato a come poter tradurre le idee partorite in aspetti concreti. Siamo giunti a una soluzione per ogni problema che di seguito viene riportato:

Dopo aver scaricato l'applicazione e aperto quest'ultima, all'utente appare la schermata di login. L'utente deve per prima cosa registrarsi al sistema inserendo nell'apposita sezione i suoi dati personali (nome, cognome, data di nascita, sesso, e-mail e password). Il sistema, dopo aver controllato la correttezza dei dati, elabora la richiesta e provvede ad inserire l'utente nel database esterno.



Una volta effettuata la registrazione egli può effettuare l'accesso spuntando o meno la casella "resta collegato" per non effettuare l'accesso le volte successive. Inserite le credenziali per l'accesso, il sistema, verifica la loro veridicità, in caso di convalidazione l'utente accede alla home; in caso contrario l'utente non ha il diritto ad accedere alla piattaforma, pertanto viene invitato ad inserire credenziali valide.

Se il "resta collegato" è stato spuntato, alla riapertura dell'app il sistema riporterà l'utente direttamente all'home page.

Dopo aver fatto il login correttamente, sulla home page appare la ricerca prodotti. Questa funzione permette all'utente di cercare il prodotto desiderato tramite nome o bar code. Una volta digitato o scansionato il prodotto cercato il sistema elabora la richiesta e propone tutti i prodotti presenti nel database contenenti la parola cercata o il bar code scansionato.

Visualizzati i prodotti, l'utente potrà scegliere se aggiungere il prodotto in una delle proprie liste o comparare il prezzo migliore tra tutti i supermercati presenti e aggiungere il prodotto in uno dei propri carrelli. Una volta effettuata questa operazione l'utente viene riportato alla home page.

Cliccando sul menù laterale della home page la sezione "Liste", l'utente può visionare tutte le liste della spesa create. L'utente ha la possibilità di condividerle, eliminarle e crearne di nuove. Cliccando su una lista, l'utente può visualizzare o eliminare i prodotti da ciascuna lista. Tutte le operazioni descritte vengono aggiornate nel database esterno.

Dalla home page l'utente può visionare i carrelli della spesa cliccando dal menù laterale la sezione "Carrelli". Vengono mostrate i carrelli dell'utente con possibilità di condividerli, eliminarli e crearne di nuovi. Cliccando su un carrello, l'utente può visualizzare o eliminare i prodotti di ciascun carrello. Tutte le operazioni descritte vengono aggiornate nel database esterno.

Se i prodotti presenti in un carrello appartengono tutti allo stesso supermercato, all'utente appare un pulsante che gli consente di effettuare la spesa con il percorso più breve per prendere tutti i prodotti presenti nel carrello.

Una volta cliccato il pulsante "inizia spesa", si apre una nuova schermata con la mappa del supermercato corrispondente e un pin che guiderà l'utente verso i successivi reparti da raggiungere e mostrerà i prodotti da prelevare da ciascun reparto ad ogni clic del pulsante "prossimo".

L'algoritmo del calcolo del percorso più breve è stato risolto come segue:

Conosciuto come TSP (Travelling salesman problem o problema del commesso viaggiatore), lo scopo del problema è quello di trovare il percorso più breve partendo da un nodo di partenza, per visitare tutti i nodi di un grafo e ritornare al nodo di partenza. Considerando quindi la mappa del supermercato come un grafo costituito da  $N$  nodi: ogni nodo corrisponde ad un reparto del supermercato (ognuno descritto da coordinate  $x,y$ ); dato l'insieme di nodi e distanze euclidee tra ogni coppia di nodi, e partendo da un nodo che sarà il nodo di partenza (ingresso e uscita), l'algoritmo sviluppato trova il modo più breve per visitare tutti i nodi esattamente una volta e tornare al nodo di partenza. Una volta concluso il percorso per fare la spesa l'utente viene riportato nella home page.

Da qui l'utente può visionare i supermercati presenti cliccando dal menù laterale la sezione "Supermercati".

Una volta cliccato comparirà una schermata con la lista dei supermercati e la possibilità di visionare tutti i prodotti di ciascun supermercato o cliccare sul nome del supermercato e visionare le informazioni relative (mappa del supermercato, nome, città, via, civico, pulsante prodotti).

Una volta cliccato sul pulsante "prodotti" il sistema elabora la richiesta e propone tutte le categorie dei prodotti presenti nel database, all'utente compariranno tutte le categorie in cui sono suddivisi i prodotti (tutti i prodotti, carne, ortofrutta, dolci e colazione, igiene e casa, alcolici, alimentari, surgelati, farinacei, bevande). Cliccando su una delle categorie appaiono all'utente i rispettivi prodotti con la possibilità di aggiungerli ad un carrello una volta comparso il prezzo o di aggiungerlo ad una lista. Effettuata l'operazione l'utente viene rinvio all'homepage.

Dal menù laterale cliccando sulla sezione "Profilo" il sistema elabora la richiesta e propone tutte le informazioni dell'utente presenti nel database esterno. È possibile visualizzare e modificare tutte le credenziali dell'utente esclusa la mail.

Modificati campi l'utente preme il tasto "salva modifiche". I dati, se validi, vengono modificati nel database esterno. Inoltre, l'utente può eliminare il proprio profilo dalla sezione profilo premendo il pulsante "elimina account". I dati relativi all'utente vengono eliminati dal database esterno.

Terminata questa operazione l'utente viene rinvio all'homepage.

Il sistema permette il logout dall'account. Per fare quest'ultima operazione, dal menù laterale, l'utente clicca la sezione "Esci", il sistema ignora la spunta resta collegato precedentemente selezionata e torna alla schermata di login.

## Risultati e discussione

Per risultati di un progetto si intendono i benefici conseguiti attraverso l'implementazione delle azioni richieste dal progetto stesso. Quindi, i risultati sono gli obiettivi che ci siamo proposti di realizzare.

Possiamo affermare che tutti gli obiettivi del nostro progetto sono stati rispettati e quindi l'app garantisce tutte le funzionalità ideate durante la fase di brainstorming iniziale.

Inoltre, noi tutti, avevamo come obiettivo quello di creare un buon prodotto e per fare ciò andavano definiti chiaramente gli obiettivi da portare a termine, conoscere bene il contesto in cui stavamo lavorando e coinvolgere tutti i membri del gruppo definendo bene i compiti di ciascuno. Possiamo affermare che anche questo obiettivo è stato completato grazie al lavoro di squadra.

## Considerazioni conclusive e sviluppi futuri

Gli sviluppi futuri dell'applicazione sono molteplici e cospicui.

Il primo obiettivo futuro è quello di sviluppare l'applicazione per dispositivi ios, così da poter ingrandire il bacino di utenza e far sì che **easyBUY** possa raggiungere chiunque.

Raggiunto quest'ultimo obiettivo, lo step successivo è quello di ampliare la sezione profilo arricchendolo con le statistiche di ciascun utente: l'utente potrà visionare gli acquisti più frequenti, le categorie di prodotti più comperati, istogrammi riguardanti l'importo erogato per ciascuna spesa e creazione di spazi per la gestione degli acquisti in base al budget.

È prevista l'implementazione di una nuova sezione: "Volantini". In questa sezione l'utente potrà visionare tutti i volantini disponibili in quel giorno con le relative offerte e promozioni. In questa sezione l'utente viene anche aggiornato su eventuali coupon o codici promozionali rilasciati da ciascuna attività commerciale.

Un'altra importante implementazione è l'introduzione degli acquisti online.

Sarà possibile infatti acquistare direttamente la propria spesa dal cellulare attraverso pagamento online e sarà data la possibilità all'utente di ritirare la spesa secondo due modalità: consegna a domicilio o ritiro presso l'attività commerciale. In quest'ultima modalità la spesa acquistata dall'utente sarà preparata dagli impiegati dell'attività pronta per il ritiro da parte dell'utente.

Infine, l'ultima implementazione pensata al momento per il futuro dell'applicazione è l'attivazione dei servizi di geolocalizzazione.

Questa funzione consentirà all'utente di poter vedere e di conseguenza considerare particolarmente le attività più vicine alla propria posizione. Inoltre, la geolocalizzazione attiva, nel percorso più breve, la mappa 3D con la possibilità di guidare l'utente, in tempo reale, verso lo scaffale successivo.

# Bibliografia

Android Studio

<https://www.informaweb.it/it/coding-pensiero-computazionale/android-studio-guida-download>

Altervista

<https://it.wikipedia.org/wiki/AlterVista>