
# GreenHabits

## Progetto di Mobile Programming e Multimedia

*Bryan Lucchetta* (1237584)
*Mariano Sciacco* (2007692)

2020 — 2021

*Corso di Laurea Magistrale in Informatica, Dipartimento di Matematica, UniPD*

# Indice

# Elenco delle figure

# 1 Introduzione

## 1.1 Abstract

Per il progetto di Mobile Programming e Multimedia, abbiamo voluto realizzare un'applicazione che consentisse agli utenti di tenere traccia del proprio stile di vita, seguendo obiettivi che possano essere significativi in ottica di impatto ambientale. Il focus dell'app si basa sul mettere a conoscenza l'utente sul risparmio di anidride carbonica che esso può raggiungere attraverso attività mirate a tale scopo. Per raggiungere tale obiettivo, abbiamo inoltre pensato di integrare nella nostra app il concetto di **gamification**, cosicché l'utente possa provare emozioni positive e sentirsi coinvolto quando segue e porta a compimento determinate attività "green". In questa app, l'utente può decidere quali attività seguire, dalle più semplici alle più complicate, tracciandole in modo giornaliero e in modo settimanale, guadagnando nel frattempo esperienza e premi. Alla conclusione di ogni settimana l'utente avrà un riscontro del suo andamento, mentre giornalmente può vedere come si sta comportando in relazione ai suoi obiettivi giornalieri.

## 1.2 Requisiti tecnici

Dal momento che per la nostra applicazione abbiamo deciso di utilizzare un framework che utilizza l'approccio cross-compiled secondo la classificazione di Raj e Tolety, ciò ci ha permesso di sviluppare per i due principali sistemi operativi per smartphone attualmente in commercio: *Android* e *iOS* ottenendo performance simil-native. Di seguito riportiamo le specifiche richieste per il funzionamento dell'app.

- **Sistema operativo:** Android 6.0 (Marshmallow) o superiore, iOS 9.0 o superiore
- **Dimensione dello schermo:** 4 - 7 pollici
- **Memoria RAM:** 512MB o più di ram
- **Spazio richiesto:** 30MB o più disponibile

## 1.3 Terminologia

All'interno della seguente relazione utilizzeremo i seguenti termini per indicare:

- **Attività**: azione eco-sostenibile rivolta all'utente, atta a migliorare la salute del nostro pianeta diminuendo in modo consistente la CO2 emessa. L'utente potrà decidere o meno di impegnarsi per portare l'azione a termine e di conseguenza completarla.
  Un'attività a sua volta si divide in:

  - **Attività settimanale**: attività che per essere portata a termine deve essere svolta in una settimana e che di conseguenza non può essere fatta giorno per giorno. Per conseguire tale attività all'utente verrà chiesto alla fine di ogni settimana se è stata completata o meno nella settimana appena trascorsa.
  - **Attività giornaliera**: attività che determina un'azione che può essere portata a termine ogni giorno e pertanto può essere contrassegnata dall'utente come completata giorno per giorno.

- **Attività di fitness**: attività che è stata inizialmente inserita per poter rilevare attraverso i sensori dello smartphone i movimenti dell'utente che determinano esercizi di fitness, quali camminata, bicicletta, salita di scale, etc. Questo tipo di attività è direttamente coinvolta con la parte di machine learning ed è una feature sperimentale.

- **Abitudine**: con questo termine indichiamo un'attività che l'utente ha portato a termine un discreto numero di volte in modo consecutivo nel tempo e che attraverso un nostro algoritmo è stata segnata come tale.

- **Plugin**: estensione (pacchetto) del linguaggio Flutter, che permette l'aggiunta di nuove classi e funzionalità a quest'ultimo. Tutti i plugin usati in questa applicazione provengono dal gestore dei pacchetti di Flutter chiamato pub.dev.

# 2 Mock-up e progettazione

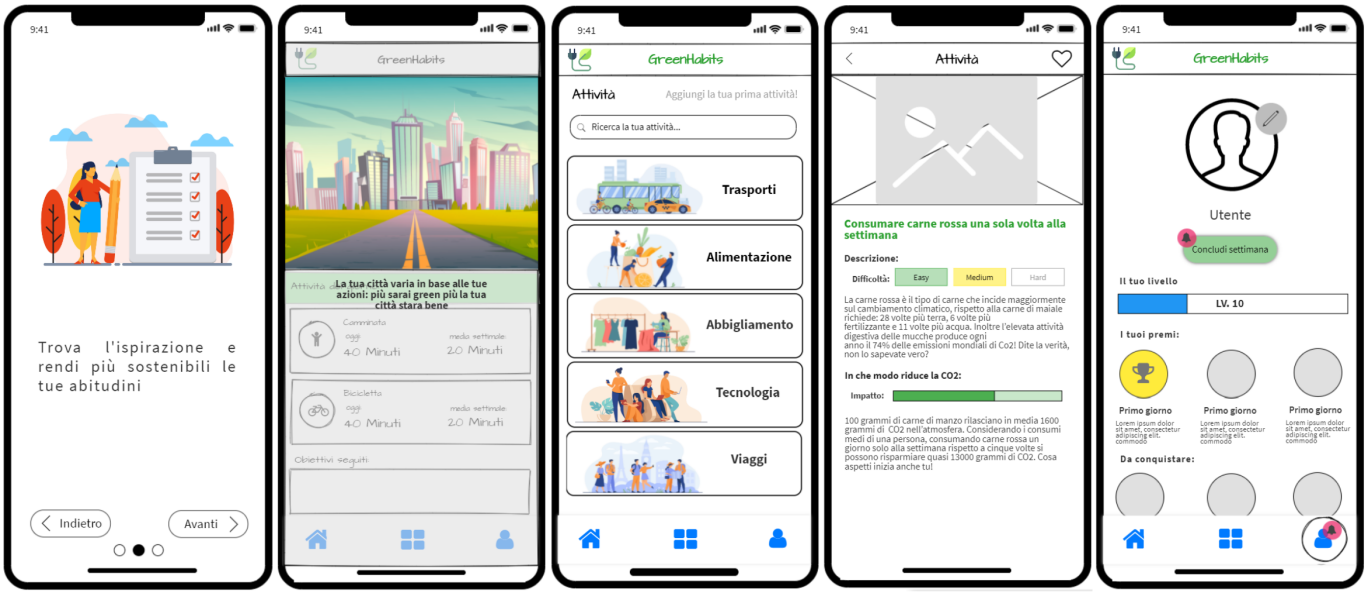



Figura 1: Da sinistra verso destra: - Onboarding in iniziale - Uso del coaching tramite tutorial in overlay - Pagina categorie - Pagina di un'attività - Pagina del profilo utente

Una delle prassi più importante quando un'app deve essere progettata è buona regola fare uno "mock", per chiarire le proprie idee, esporre in maniera visuale le stesse e mettere in atto in modo semplice e veloce le regole di usabilità/interfaccia apprese. Questo permette di avere una rappresentazione approssimativa visuale di quello che sarà il prodotto finale. Per tali motivi abbiamo dunque scelto di utilizzare Mockflow come editor, in quanto da noi considerato come strumento molto semplice ed intuitivo per la progettazione di wireframes o mock-up di interfacce grafiche per Android o iOS.

Riportiamo in Figura 1 alcuni screen delle interfacce realizzate con questo strumento, dove è possibile notare alcuni dei concetti chiave dell'applicazione che spieghiamo di seguito.

- **Onboarding**: Con l'onboarding vogliamo cercare di introdurre l'utente allo scopo e alla vision dell'app in questione. In particolare vogliamo porre l'attenzione dell'utente sul fatto che il mondo può essere ancora salvato dai cambiamenti climatici causati dal nostro impatto ambientale, e che se tutti ci impegnassimo (anche con poco) a seguire alcune delle abitudini green proposte, si potrebbe sicuramente contribuire alla sua salvaguardia. Per questi motivi la scelta è ricaduta sullo sviluppo di 3 pagine introduttive, in cui cerchiamo di spiegare in maniera veloce e concisa la "mission" dell'applicazione.

- **Home**: Affinché l'utente sia consapevole del proprio impatto ambientale, abbiamo deciso di progettare una home che lo tenesse informato delle proprie attività seguite e portate a termine di giorno in giorno, nonché avere una panoramica generale del suo andamento. Per questo motivo possiamo trovare le attività raggruppate in sezione distinte a seconda della loro tipologia e completamento. Non volevamo che l'app risultasse troppo severa e rivolta a un pubblico esperto interessato esclusivamente a questo tema. Infatti, la nostra missione è quella di coinvolgere più persone possibili, in maniera semplice e divertente, per sensibilizzarle a tale tematica. Per fare questo ci siamo avvalsi dunque della **gamification**, e una delle prime sue applicazioni e stata l'aggiunta di una città "virtuale" isometrica, per cercare di far credere all'utente di prendersi cura della propria città attraverso le proprie azioni. Le attività green che l'utente avrebbe deciso di intraprendere e completare avrebbero così contribuito a rendere la città più verde e bella, coinvolgendo l'utente emotivamente nel perseguire tale scopo.

- **Ricerca e categorizzazione delle attività**: In questa sezione vogliamo che l'utente possa ricercare in maniera semplice e veloce le attività presenti all'interno dell'applicazione. A questo scopo le attività pertanto sono state suddivise inizialmente all'interno di categorie prestabilite, che l'utente poi ritroverà in questa pagina. Inoltre è stato scelto di introdurre anche una barra di ricerca con il quale l'utente possa cercare una qualunque attività, facilitando così la procedura di ricerca. Dopo aver cercato un'attività o filtrato per categoria, viene fornito all'utente un elenco contenente per ogni attività il titolo dell'attività, l'impatto che essa ha sul nostro pianeta e il tipo di questa attività (settimanale o giornaliera).

- **Pagina attività**: La pagina attività deve riassumere in modo dettagliato e preciso tutte le informazioni inerenti all'attività in sé. All'interno della pagina l'utente troverà: il livello di difficoltà stimata, una sua descrizione dettagliata, il suo impatto sul pianeta e una descrizione sul modo in cui questa attività riduce la CO2 emessa. A questo punto l'utente sempre da

questa pagina può decidere di seguire o meno l'attività, oppure di indicare tale attività come completata se è di tipo giornaliera ed è seguita dall'utente.

- **Profilo utente**: Con questa pagina vogliamo riassumere in maniera sintetica tutte le informazioni riguardanti l'utente. In particolare un'utente può decidere di visualizzare o modificare il proprio profilo (nome e avatar) e di tenere sott'occhio tutti i premi e l'esperienza guadagnata tramite le proprie attività green completate, introducendo ancora una volta concetti della **gamification**.

# 3 Sezioni e viste dell'applicazione

In questo paragrafo ci dedicheremo ora all'analisi di tutte le scelte fatte per quanto riguarda l'usabilità dell'applicazione e il design delle interfacce grafiche.

## 3.1 Onboarding

Come spiegato nel paragrafo precedente abbiamo usato l'onboarding per introdurre l'utente allo scopo e alla "mission" di tale applicazione con 3 semplici schede. Per implementare l'onboarding ci siamo avvalsi di un plugin chiamato introduction_screen, che permette di creare una serie di pagine introduttive personalizzate per introdurre l'utente al primo avvio dell'app. I punti a favore per cui abbiamo deciso di adottare il seguente plugin sono i seguenti:

- La possibilità di scorrere tra una scheda e l'altra dell'onboarding attraverso una gesture di swipe. Questa funzionalità è molto utile in quanto permette all'utente di muoversi in maniera agevole tramite l'uso delle gesture, evitando di tradire la metafora visiva dove ci si aspetta di poter scorrere tra le schede come se fosse un "libro".
- Delle animazioni molto curate tra una scheda e l'altra, che accompagnano l'utente nella navigazione all'interno dell'onboarding.
- E' presente nella parte inferiore una barra di controllo in cui è possibile visualizzare in quale delle schede ci trova, saltare l'intero onboarding o proseguire con la scheda successiva nel caso in cui non si vogliano usare le gesture. Inoltre tale barra risulta variare i propri controlli in base al contesto corrente in cui si trova l'utente: infatti alla fine dell'onboarding per esempio scompare il bottone per saltare l'onboarding, e il bottone dedicato alla pagina successiva si modifica in un bottone di fine onboarding.
- La completa personalizzazione delle schede dell'onboarding, essendo possibile inserire testi e/o immagini a proprio piacimento.

Al termine dell'onboarding, in una prima versione dell'applicazione veniva richiesto all'utente di compilare (attraverso la visualizzazione di una lista con tutte le attività presenti nell'applicazione) le abitudini "iniziali" che l'utente possedeva come riportato in Figura 2(b). Osservando i primi utenti a cui abbiamo fatto provare il primo prototipo, abbiamo riscontrato che di fronte ad una lista molto lunga l'utente faceva spesso ricadere il proprio sguardo su di essa, senza prestare particolarmente attenzione alla domanda e alle istruzioni che gli venivano fornite in cima alla pagina. Così facendo, tale pagina creava una sensazione di smarrimento a un utente neofita che non capiva cosa gli venisse chiesto, e soprattutto che potesse tenere premuto su un'attività per avere una descrizione più dettagliata nel caso fosse necessario. Per tali motivi, abbracciando inoltre la politica del **"less is more"**, è stato deciso di eliminare la pagina in questione e di tracciare le abitudini dell'utente man mano che esso utilizza l'applicazione. In questo modo, anche se non è possibile conoscere le abitudini green che un'utente possiede già, è comunque possibile tenerne traccia in un secondo momento, risolvendo così un problema di usabilità importante.


(a) Due delle tre schede dell'onboarding in cui è possibile notare inoltre come la barra di navigazione sotto cambia in base al contesto in cui l'utente si trova


(b) Pagina per conoscere le abitudini iniziali dell'utente

Figura 2: Alcune delle pagine che l'utente ritrova/ritrovava al primo avvio dell'applicazione

## 3.2 Navigazione

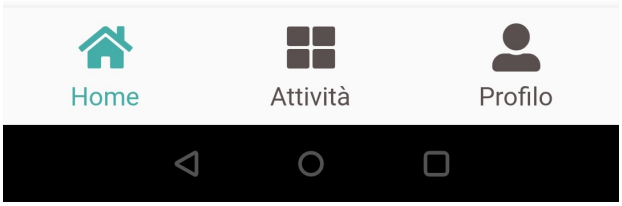
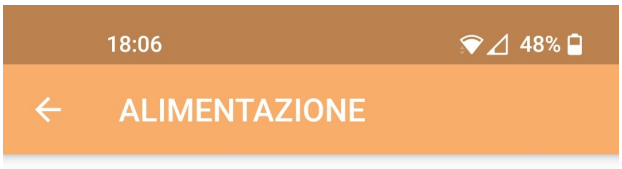


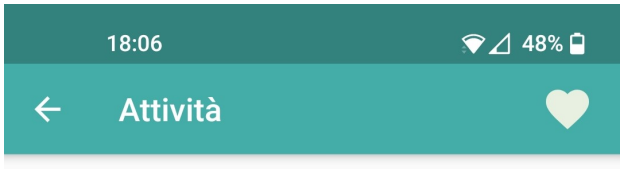

Figura 3: Barra di navigazione principale della nostra app affiancata alla barra di navigazione del sistema Android

Per la navigazione all'interno dell'applicazione ci siamo affidati principalmente alla navigazione offerta direttamente dal linguaggio Flutter. In particolare per la navigazione principale, ossia quella tra le 3 pagine principali dell'app (home, attività e profilo) abbiamo usato la classe BottomNavigationBar che permette la visualizzazione di una barra di navigazione in fondo all'applicazione. Questa decisione non è corretta per la versione Android, in quanto la barra di navigazione dell'applicazione si andrebbe a trovare troppo vicino ai controlli della barra di navigazione di sistema, potendo così causare tap involontari da parte dell'utente sulla medesima. Nonostante nelle ultime versioni del sistema operativo Android sia stata introdotta la possibilità di usare le gesture per effettuare le stesse operazioni previste dalla barra di navigazione di sistema, come fa la controparte iOS da anni, abbiamo comunque deciso di risolvere il problema nel caso in cui l'utente usi il vecchio sistema di navigazione Android, rendendo l'area di tab della nostra barra di navigazione più grande.

Al di fuori delle tre pagine principali dell'applicazione, che sono la pagina Home, la pagina delle Attività e la pagina del Profilo, l'utente non vedrà più la barra di navigazione principale, ma vedrà un barra in alto di navigazione. Quest'ultima è offerta nativamente da Flutter nel momento in cui si va ad aggiungere una nuova pagina nel Navigator. Questa risulta essere molto comoda, perché oltre a offrire un titolo per dare il contesto in cui si trova l'utente, ha già integrato un pulsante di "back" a sinistra che permette all'utente di tornare comodamente indietro. Inoltre questa barra può essere ulteriormente personalizzata attraverso l'inserimento di altri bottoni/funzionalità nella parte destra. Nel nostro caso si è rilevata particolarmente utile quest'ultima opzione per inserire il bottone per seguire un'attività, quando l'utente si trova all'interno di una di queste.




(a) Barra di navigazione in alto all'ingresso su una specifica categoria di attività




(b) Barra di navigazione in alto all'interno di una specifica attività

Figura 4: Barre di navigazione in alto al di fuori delle tre pagine principali dell'applicazione

Questo meccanismo di navigazione messo a disposizione da Flutter rispetta la regola del **content always on top**, in quanto tutti questi controlli rimangono ai lati superiori e inferiori dello schermo per permettere la visualizzazione in primo piano del contenuto da parte dell'utente.

## 3.3 Coaching e tutorial

Al fine di favorire una migliore comprensione di tutte le funzionalità della nostra applicazione e di istruire l'utente all'uso corretto, abbiamo deciso di integrare una serie di tutorial **just-in-time** che vengono quindi mostrati all'utente nel momento opportuno. Questi si presentano con un titolo e un'apposita didascalia per spiegarne il funzionamento nel contesto in cui vengono mostrati. In particolare i tutorial proposti sono i seguenti:

- Tutorial per la homepage, in cui riportiamo le forme che la città può assumere e il funzionamento della barra di navigazione.
- Tutorial per seguire un'attività, in cui spieghiamo e indichiamo il bottone da utilizzare quando l'utente vuole seguire una determinata attività.
- Tutorial per il profilo, dove viene illustrato l'uso del bottone concludi settimana, i livelli e i premi proposti.
- Tutorial del bottone *Storico* in homepage, che comparirà solo alla conclusione della prima settimana.
- Tutorial dello storico settimanale, che spiegherà all'utente i vari elementi presenti nella pagina alla prima conclusione della settimana.


Figura 5: Da sinistra verso destra: Tutorial homepage (1 di 2), Tutorial homepage (2 di 2), Tutorial dettaglio attività, Tutorial profilo (1 di 2), Tutorial storico delle città

L'insieme di questi tutorial può essere riassunto in un sistema di **coaching** che guidano l'utente all'uso dell'applicazione nelle nuove funzionalità e schermate che gli vengono mostrate per la prima volta. Inoltre, sfruttando la conclusione della prima settimana, nella nostra applicazione adottiamo anche il principio di **leveling-up**, per cui non appena vi è un progresso effettivo dell'utente (per

l'appunto la prima settimana conclusa), facciamo scoprire una nuova sezione denominata "storico" che sarà quindi accessibile dalla homepage. Come nota di merito, riportiamo inoltre che in una prima versione dell'app era stato inserito anche un tutorial la prima volta che l'utente portava a termine un'attività di fitness, tuttavia questa parte è stata rimossa a seguito dei problemi riscontrati per il rilevamento dell'attività di fitness dell'utente.

Ciascuno di questi tutorial lo abbiamo reso tale da non essere troppo invasivo e per tale motivo è sempre stato inserito un bottone di *Fine* o *Salta*, per permettere all'utente di saltare il tutorial all'occorrenza. Nello specifico, per i tutorial con singole didascalie abbiamo fornito solamente il bottone di *Fine* sempre presente in basso a destra, mentre per i tutorial con più didascalie abbiamo inserito un bottone *Salta* in basso a sinistra dello schermo e un bottone *Avanti / Fine* in basso a destra dello schermo. Questa miglioria è stata implementata a seguito della limitazione di questo plugin, denominato tutorial_coach_mark, che non ci ha permesso nativamente l'utilizzo di un doppio bottone o, più banalmente, di un bottone *Avanti* nell'interazione con questi tutorial, così da non sacrificare l'usabilità dell'applicazione.

Una peculiarità di questo plugin è l'uso delle animazioni e dello sfondo nero semi-trasparente per focalizzare l'attenzione dell'utente sull'elemento interessato. Nella scelta dell'intervallo di animazione per l'espansione e la contrazione dell'animazione sull'elemento evidenziato abbiamo deciso di ispirarci al tempo che intercorre tra i respiri dell'essere umano, prendendo spunto dal brevetto Apple utilizzato per indicare lo stato di caricamento della batteria dei MacBook tramite led fisici.

## 3.4 Homepage


Figura 6: Da sinistra verso destra: Homepage di giorno senza attività completate, Homepage con il cambio di città, Homepage di notte, Titoli rimanenti nella homepage, Homepage con avviso.

Per la pagina principale dell'app abbiamo pensato di realizzare un'interfaccia che non si distaccasse troppo da quanto pensato in fase di progettazione. Nello specifico, abbiamo integrato una prima parte in cui è presente l'immagine di una città isometrica che può assumere ambientazioni diverse in base alle azioni giornaliere dell'utente, ispirandoci così ai principi della **gamification**. Infatti, se l'utente registrerà delle attività nel corso della giornata ne conseguirà una diminuzione della CO2 emessa e la città rappresentata migliorerà; nel caso contrario, la città peggiorerà assumendo toni più grigi. Questo quindi dovrebbe spronare l'utente a seguire e registrare delle attività che gradualmente potranno divenire delle abitudini. La città inoltre ha uno sfondo basato sull'ora corrente in cui l'utente usa l'applicazione e mostrerà la luna o il sole di conseguenza, aumentando così il senso di coinvolgimento dell'utente nell'app.

Nella seconda parte della homepage sono presenti 3 sezioni: le attività giornaliere completate, le attività giornaliere non ancora completate e le attività settimanali che l'utente sta seguendo. Queste sezioni permettono all'utente di tenere traccia di tutte le attività che ha deciso di seguire, siano esse giornaliere o settimanali. Infine con un semplice tap su una delle attività è possibile entrare nella pagina dedicata ai dettagli dell'attività.

Essendo dunque la homepage una pagina principale, nonché elemento fondamentale e più importante della nostra applicazione, abbiamo effettuato diverse rivisitazioni e ri-progettazioni. In un primo momento abbiamo pensato di inserire una parte con le attività di fitness, che sono state poi oscurate per essere in futuro meglio collaudate e perfezionate anche in relazione alla parte di *machine learning*. Inoltre, inizialmente venivamo riportate solo le attività seguite e le attività abitudinarie ma questa idea è stata rivista in modo che la homepage contenesse nel dettaglio le attività che l'utente sta seguendo suddivise in "completate" e "rimanenti", così da mostrare con maggior rilievo una visione d'insieme delle attività svolte e seguite.

Sempre nella homepage è stato inserito un avviso che segnala all'utente la disponibilità del concludi settimana nella sezione profilo. Questo avviso viene inserito **just-in-time** e permette di essere usato come scorciatoia per spostarsi nella sezione profilo. Abbiamo preferito questo tipo di approccio rispetto al mandare direttamente l'utente alla pagina di conclusione settimanale, per favorire maggiormente l'uso della sezione relativa al profilo, anche in vista di futuri aggiornamenti che potrebbero integrare dinamiche più "social".

## 3.5 Storico delle città

Nel momento in cui l'utente conclude una settimana vengono salvate delle statistiche (*snapshot*) sull'andamento settimanale in un'apposita sezione. Questa sezione è possibile raggiungerla tramite la pagina Home attraverso il bottone *Storico* che comparirà una volta conclusa la prima settimana.

Nella sezione viene riportata una scheda riassuntiva della città nel corso della settimana, in cui sono presenti tre componenti fondamentali:

- La città dinamica, che in base alla quantità di CO2 risparmiata prenderà un aspetto diverso.
- Un messaggio di testo per spronare l'utente a migliorare in base all'andamento.
- L'andamento statistico della settimana con l'impatto totale, le attività registrate e le attività seguite per quella settimana.

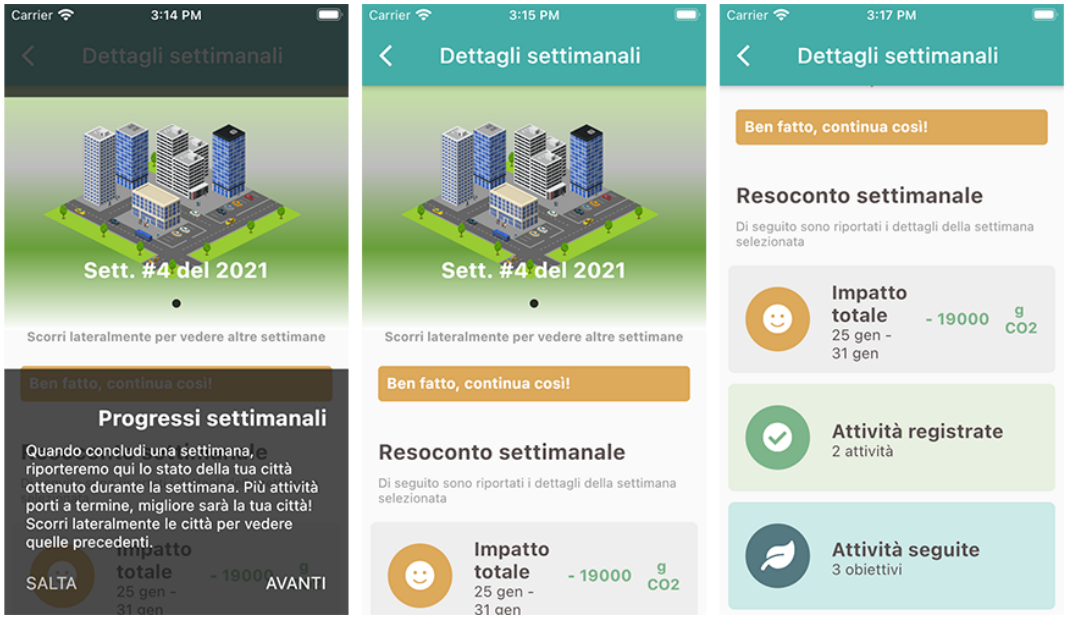



Figura 7: Da sinistra verso destra (storico delle città): Tutorial iniziale, Città dinamica, Dettagli sul resoconto settimanale

In questa sezione è possibile spostarsi da una settimana più recente a una più vecchia sfruttando il carosello delle città presente nella prima parte della sezione, integrato con il plugin carousel_slider. Come visto a lezione, i caroselli sono molto utili per mostrare dati lineari, dove l'utente sà già cosa aspettarsi tra una scheda e l'altra. Infatti, il nostro carosello riporta i dettagli di ciascuna settimana trascorsa e il numero della settimana nell'anno corrente. La scelta di riportare il solo numero della settimana è stata fatta per rendere la navigazione progressiva e il contenuto più chiaro e sintetico, evitando all'utente di perdere la cognizione temporale se avessimo riportato solo i giorni della settimana. Solamente per l'ultima settimana conclusa, viene riportato anche un messaggio di sostegno o di consiglio per l'utente, al fine di migliorarsi o di continuare nel modo in cui sta portando a termine le settimane. Inoltre con l'utilizzo di diversi tipi di "smile" nella prima *tile* delle statistiche e di colori (rosso, arancione, verde) che variano in base all'andamento che l'utente tiene, vengono ripresi ancora una volta i concetti della **gamification**.

## 3.6 Attività e Categorie


Figura 8: Da sinistra verso destra: - Pagina ricerca attività - Ricerca di una attività - Pagina categoria attività - Pagina di un'attività

Nella seconda pagina più importante dell'app, abbiamo deciso di integrare la ricerca e visualizzazione delle attività come riportato in Figura 8. L'utente ha a disposizione due modi per ricercare le attività: il classico modo tramite barra di ricerca e un secondo modo basato sulla suddivisione delle attività in base alla categorie in cui sono state da noi raggruppate.

Per quanto riguarda la parte di ricerca è stato deciso di utilizzare un plugin di Flutter molto versatile chiamato flappy_search_bar. Come è possibile notare il plugin renderizza a schermo un barra di ricerca molto semplice e intuitiva con anche un'icona di ricerca personalizzabile e una serie di controlli per gestirla. Il plugin va poi configurato per poter svolgere la ricerca delle attività all'interno del nostro database, definendo le funzioni del linguaggio opportune. Come è possibile notare dalla Figura 8, una volta inserita la parola da ricercare all'interno dei titoli di tutte le attività, è possibile cancellare la propria ricerca tramite un apposito bottone che compare sulla destra. La sua implementazione e integrazione con la visualizzazione delle categorie non è risultata tuttavia molto semplice, in quanto si venivano a creare problemi di renderizzazione con le categorie che venivano distaccate parecchio dalla barra di ricerca a causa dei margini troppo elevati. Per risolvere tale problema è stato deciso di utilizzare uno Stack per poter così posizionare le categorie al posto giusto.

Per accedere alla lista di attività di una categoria, è sufficiente fare un tap sulla categoria desiderata. Una volta visualizzato l'elenco delle attività per categoria, abbiamo deciso di usare il principio del **progressive disclosure**. Infatti per non affollare troppo l'interfaccia utente con tutte le informazioni che possono essere presentate per ogni attività, mostriamo solo una lista di attività

con le informazioni più importanti e utili all'utente. In questo modo, oltre che visualizzare il titolo e l'immagine dell'attività, possiamo vederne l'impatto e la tipologia. Per capire quali attività l'utente già segue, abbiamo riportato su di essa un'icona a forma di cuore posizionata in basso a destra, come riportato in Figura 8.

Una volta individuata un'attività di proprio interesse, l'utente facendo tap su di essa può entrare nella pagina dedicata. Tale pagina è composta da diverse sezioni informative: oltre all'immagine e al titolo dell'attività, ora l'utente può visualizzare anche una descrizione dettagliata dell'attività e una spiegazione sul come tale attività riduce i consumi di CO2. Quest'ultima ha la funzione di convincere e istruire l'utente sull'efficacia di questa attività. Ad entrambe le sezioni è dedicata un'apposita barra che indica nel primo caso il livello di difficoltà stimato e il livello di impatto che tale attività ha sull'ambiente. Ovviamente tali parametri sono stati stimati in base alla nostra esperienza soggettiva, pertanto non rispecchiano effettivamente le attitudine dell'utente ma possono in ogni caso fornire una stima indicativa di tali parametri con il quale l'utente possa confrontarsi in un primo momento. In questo caso abbiamo deciso di rispettare la regola **content always on top** in quanto i comandi sono messi nella parte superiore (bottone a forma di cuore per seguire o meno l'attività) o inferiore (slider per completare l'attività), lasciando il contenuto dell'attività sempre in primo piano. Inoltre l'utente ha sempre la piena consapevolezza se si tratta o meno di un'attività settimanale o giornaliera, grazie all'apposito bottone riportato a fianco alla figura dell'attività. Tale bottone ha una duplice funzionalità: indicare in primis la tipologia di attività, e in secondo luogo permettere di aprire una scheda informativa all'utente in cui gli viene spiegata e ricordata la differenza tra attività settimanali e giornaliere, nonché il loro funzionamento all'interno di tutta l'applicazione. In questo modo se l'utente dovesse per un periodo non utilizzare l'applicazione e rientrare in un secondo momento non ricordandosi più la differenza tra attività settimanali e giornaliere, potrà riconsultare una breve ma efficace spiegazione, come riportato in Figura 9.

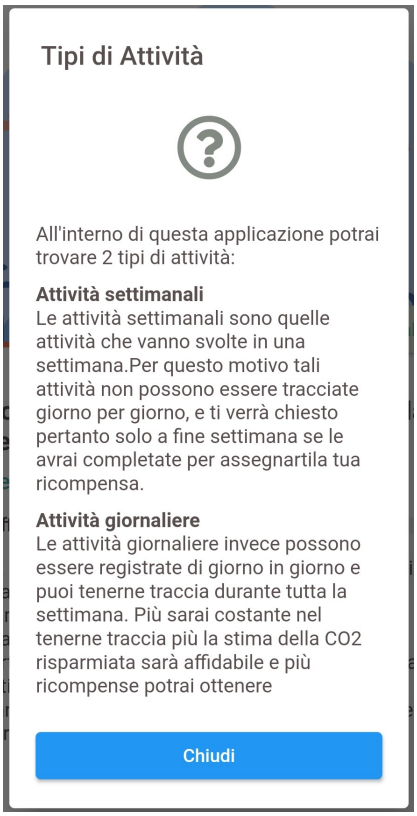



Figura 9: Box informativo per ricordare all'utente la differenza tra attività settimanali e giornaliere

Una volta che l'utente ha deciso di seguire l'attività, attraverso l'apposito bottone a forma di cuore in alto a destra, all'utente verrà mostrato uno slider in fondo alla pagina per poter indicare che ha portato a termine quell'attività, come riportato in Figura 10(a). L'utente potrà decidere anche di annullare il completamento dell'attività, allo stesso modo di come ha agito per segnalare all'applicazione il completamento dell'attività: infatti lo slider cambierà aspetto per poter annullare tale operazione, come visto in Figura 10(b). Questo comportamento è presente per le sole attività giornaliere, in quanto quelle settimanali avranno un'apposito form di compilazione a fine settima-

na. Lo slider è stato implementato tramite un apposito plugin per Flutter chiamato slide_to_act. In questo caso abbiamo voluto utilizzare volontariamente uno slider per incentivare l'uso delle **gesture**, particolarmente adatte in quelle operazioni per cui è richiesta l'attenzione e la volontà dell'utente nel compiere l'azione. Infatti la registrazione di un'attività è un operazione critica per tale applicazione (tutto il suo funzionamento ruota attorno a questa operazione), pertanto abbiamo voluto utilizzare una gesture per evitare che l'utente possa fare tap su un semplice bottone involontariamente, contrassegnando così l'attività come completata anche se effettivamente tale attività non era stata portata a termine. Inoltre va ricordato che le gesture migliorano l'accessibilità delle interfacce perché tollerano una minore precisione e vengono usate dove viene richiesta un'iterazione veloce e priva di errori.


(a) Slider per completare un'attività

(b) Slider per annullare il completamento di un'attività

Figura 10: Funzionamento dello slider per contrassegnare o meno un'attività come completata

## 3.7 Profilo

Il profilo utente è stato progettato al fine di mostrare all'utente i propri progressi in termini di esperienza ottenuta e premi. Oltre a ciò abbiamo inserito anche una parte per la selezione dell'avatar e il cambio del nome, che potrà essere poi maggiormente sviluppata in futuro per creare un profilo social. Nella prima parte di questa pagina, abbiamo inserito un avatar e un nome predefinito per l'utente, la cui modifica può essere eseguita premendo l'apposito bottone presente in alto a destra dell'avatar. Nella parte di modifica è possibile anche selezionare un nuovo avatar tra quelli disponibili, al momento solo 3 a titolo esemplificativo. In questo modo l'utente può personalizzare la sua esperienza nell'app rendendola più coinvolgente, così che i messaggi a lui rivolti contengano il suo nome. Inoltre questa parte potrà poi essere molto estesa in vista di funzionalità "social" integrabili in futuro nell'applicazione.

In seguito sotto il nome utente abbiamo deciso di inserire un bottone per la conclusione della settimana, attraverso cui l'utente può decidere durante il fine settimana (in particolare alla domenica) quali attività settimanali segnare come svolte. In questo modo si creerà uno *snapshot* della settimana corrente in cui verrà esposto il risultato ottenuto nello storico delle città accessibile dalla homepage. L'utente avrà tempo da domenica fino al sabato (compreso) della settimana successiva per segnalare le attività settimanali svolte, altrimenti verranno perse e potranno essere segnate solamente le attività della nuova settimana. Le attività giornaliere verranno comunque salvate automaticamente in real-time.

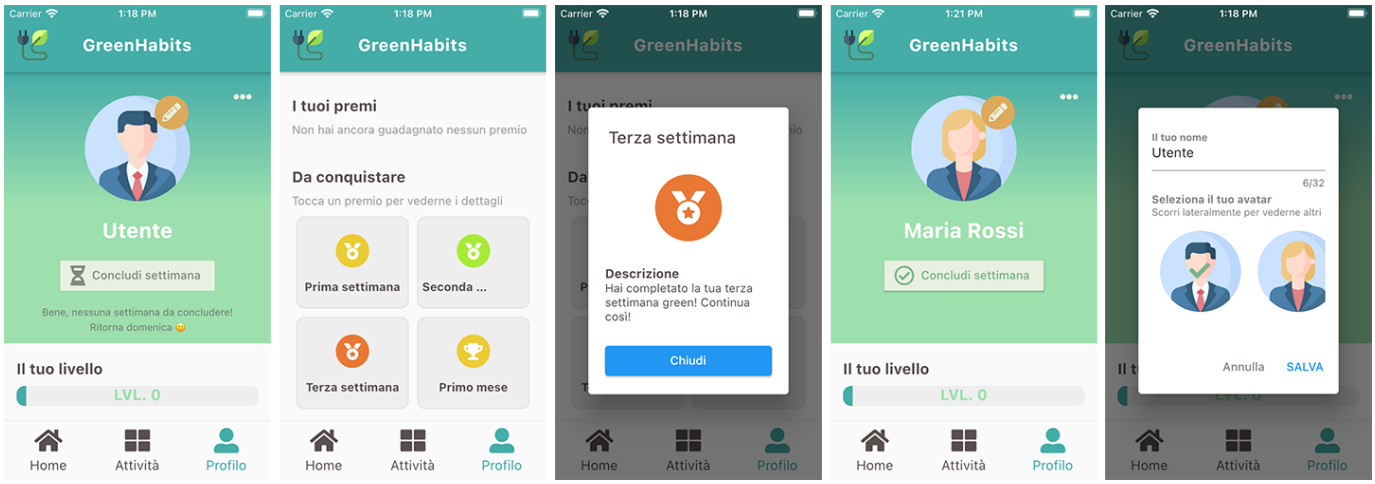



Figura 11: Da sinistra verso destra: Profilo utente (1 di 2), Profilo utente (2 di 2), Dettaglio premio, Profilo utente personalizzato, Form di modifica del profilo.

Proseguendo nel profilo utente, è possibile trovare una parte per il livello e per i premi raggiunti o da raggiungere. Il livello sale in base all'esperienza ottenuta dal completamento delle attività della settimanali o giornaliere. I premi invece, li abbiamo inseriti sempre a titolo esemplificativo per premiare gli utenti con dei punti bonus che fanno progredire il livello, favorendo ancora la gamification. Al momento un premio può essere ottenuto tramite la costanza di un utente nel concludere le attività settimanali, ma non escludiamo in una futura versione di aumentare le tipologie di premi caratterizzandoli ad esempio per quantità di attività svolte in un giorno o per quantità di attività seguite. Con un semplice tap su un premio inoltre è possibile vederne i dettagli con la descrizione per essere conseguito. Infine, i premi ottenuti riportano anche la data di acquisizione nella descrizione del premio stesso.

### 3.7.1 Crediti e funzioni sperimentali

All'interno della pagina profilo abbiamo voluto inserire anche un bottone in alto a destra formato da tre puntini per illustrare i crediti dell'applicazione, nonché la versione e la build corrente. I crediti riportati riguardano anche le immagini di terze parti utilizzate per sviluppare l'applicazione, in ottemperanza alle direttive richieste dalle singole parti in materia di copyright e diritti d'autore.

In aggiunta ai soli fini del progetto accademico, abbiamo deciso di inserire delle opzioni di debugging che saranno visibili direttamente al di sopra dei crediti. L'attivazione di queste funzionalità non dovrebbe alterare troppo il funzionamento normale dell'applicazione, sebbene non possiamo garantirne la piena stabilità. Ad esempio con l'attivazione delle funzioni di machine learning, la batteria potrebbe consumarsi più velocemente, essendo appunto ancora in via sperimentale e non pensata per l'utente finale.

## 3.8 Tema e animazioni

La tavolozza colori dell'applicazione è stata studiata in modo che rispecchiasse toni chiari sul colore dominante del verde, tale che per un utente fosse facilmente ricollegabile al contesto green dell'applicazione. In molte sezioni, abbiamo cercato di rendere gli elementi molto colorati, così da suscitare all'utente una sensazione amichevole mantenendo comunque una certa uniformità, come fatto ad esempio per le tile presenti nella sezione "Attività" (Fig. 8). Inoltre gran parte degli elementi nell'applicazione si basano sulla stessa saturazione del verde pur con tonalità differente, in modo da mantenere una certa coerenza nelle varie sezioni dell'app.

Per rendere l'app ancora più carina e che suscitasse emozioni positive all'utente durante il suo utilizzo, abbiamo deciso di utilizzare anche qualche animazione messa a disposizione da Flutter. Ad esempio una volta che l'utente fa tap su di un'attività questa si apre con un animazione che porta l'immagine dalla posizione in cui si trova nella lista fino a riempire completamente la parte superiore dello schermo. Tale animazione e fornita da una classe di Flutter chiamata Hero.

Inoltre anche gli slider per completare o meno un'attività, riportati in Figura 10, sono animati, in modo da rendere l'esperienza utente appagante al completamento di un'attività. Trascinando la freccia da destra verso sinistra, essa ruota, dopodiché una volta terminata la gesture, lo slider si condensa al centro con un'icona a forma di check-mark, per indicare l'effettiva registrazione dell'attività come completata. La stessa animazione viene ripetuta anche nel caso in cui si voglia annullare l'attività.

Nella homepage abbiamo inserito due diversi tipi di sfondo della città, uno più chiaro e uno più scuro, che cambiano sulla base dell'orario corrente. Insieme allo sfondo abbiamo aggiunto anche un icona di un sole e di una luna, che compaiono in base all'orario attuale in cui si trova l'utente a usare l'app, così da aumentare l'esperienza di coinvolgimento dell'utente nell'applicazione.

Nella pagina dello storico delle città abbiamo cercato di suscitare una serie di emozioni all'utente nel momento in cui questo decida di scorrere tra le varie settimane, mostrando il cambiamento della città con delle animazioni sulle sfumature, che alterassero lo sfondo in base alla CO2 risparmiata in quella specifica settimana. Questo ovviamente è stato fatto insieme ai colori dei messaggi di supporto e con lo "smile" che cambia in base a quanta CO2 è stata risparmiata, come è possibile notare nella Figura 7.

# 4 Sviluppo e testing

## 4.1 Dispositivi, simulatori e workspace

Nel corso dello sviluppo l'app è stata provata su un diverso numero di dispositivi con dimensioni e specifiche diverse. Principalmente sono stati usati dispositivi personali e simulatori, messi a disposizione da Android SDK e XCode, rispettivamente per Android e iOS.

I dispositivi fisici utilizzati sono stati i seguenti:

- iPhone 8 (iOS 14), 2gb di ram, 4.7 pollici, 64GB
- OnePlus 8T (Android 11), 8gb di ram, 6.55 pollici, 128GB
- Google Pixel 2 (Android 11), 4GB di ram, 5 pollici, 64GB

Per quanto riguarda i simulatori, sono stati utilizzati i seguenti:

- Simulatore x86 Pixel 3a, Android 11, 1.5 GB di ram
- Simulatore x86 Pixel 3a, Android 11, 0.5 GB di ram, risoluzione ridotta 480x800
- Simulatore x86 iPod touch (7th gen.), iOS 14, 4 pollici
- Simulatore x86 iPhone 11 (con Notch), iOS 14, 6.06 pollici

Per quanto concerne l'organizzazione del workspace per il progetto dell'applicazione abbiamo utilizzato Git per il versionamento del nostro progetto, mantenuto con una repository presente su Github. Vista l'entità del progetto, abbiamo inserito una parte di *continuous integration* (usando le Github Actions) che ci permettesse quindi di sviluppare per integrazioni di componenti attraverso controlli automatici effettuati a ogni incremento del software. In questo modo qualunque modifica è stata validata automaticamente con un controllo di compilazione automatico che determinasse il funzionamento dell'app sulla piattaforma Android. Questo ci ha permesso di avere sempre un'applicazione funzionante in ogni momento lavorando in parallelo e seguendo il principio di fail-fast, tale per cui l'incremento venisse invalidato se il software conteneva degli errori, permettendoci quindi di accorgerci più rapidamente delle eventuali correzioni da effettuare.

## 4.2 Dimensioni dello schermo

Al fine di determinare una più possibile ampia compatibilità con tutti i dispositivi attualmente in commercio, abbiamo cercato di restringere la fase di testing durante lo sviluppo ai dispositivi con caratteristiche più critiche. Siamo partiti testando l'app ad ogni rilascio sui dispositivi iOS e Android di più piccola dimensione, pari a 4 pollici (es. *iPhone SE 1st gen.*) e in parallelo abbiamo provato anche dimensioni maggiori fino a 6.5 - 7 pollici (es. *phablet*).

Per quanto riguarda la parte di debugging su iOS, nel corso dello sviluppo sono state fatte alcune accortezze a livello grafico affinché l'applicazione funzionasse correttamente su dispositivi di almeno 4 pollici. In diversi punti dell'applicazione il testo è stato ridotto e, all'occorrenza, tagliato con tre punti di sospensione, senza sacrificare troppo l'informazione. Ad esempio nelle *tile* delle attività, il testo viene automaticamente formattato con tre punti di sospensione se si superano le due righe di testo.

Adottando un approccio di questo tipo nello sviluppo, abbiamo costantemente controllato che la nostra applicazione fosse pienamente compatibile con i principali schermi presenti nel mercato dai 4 ai 7 pollici, per entrambi i sistemi operativi. Per quanto riguarda l'interfaccia inoltre abbiamo optato per una visualizzazione in modalità *Portrait* (verticale) dal momento che non ci sarebbero stati benefici nell'uso della modalità *Landscape* (orizzontale).

## 4.3 Play Store

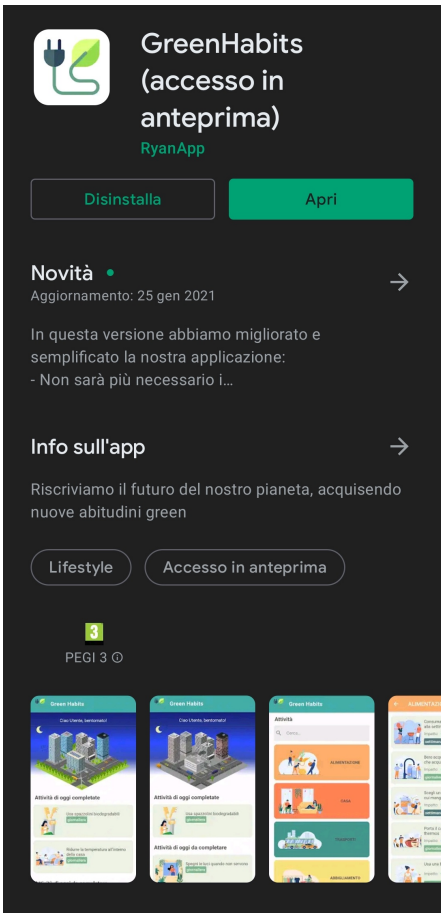



Figura 12: Pagina del Play Store di GreenHabits con accesso in anteprima

Per poter testare maggiormente la nostra applicazione anche con i nostri amici e parenti, così da ricevere preziosi feedback in merito alla compatibilità tra diversi device Android e in merito all'usabilità ed efficacia dell'app, abbiamo deciso di pubblicare la nostra applicazione sul Play Store. E' sufficiente infatti navigare a questo link per accedere direttamente alla pagina dello store e scaricare l'applicazione per poterla provare sul proprio dispositivo.

Al momento sono più di 10 i dispositivi su cui l'applicazione è stata scaricata e testata, che comprendono diverse versioni di Android e diversi tipi di schermo. Nessuno dei nostri amici ci ha riferito problemi di compatibilità, quindi possiamo essere abbastanza confidenti nell'affermare che questa applicazione risulta essere funzionante nella maggior parte degli smartphone ad oggi in commercio.

Rispetto a quanto visto in classe nel corso delle lezioni dedicate agli store, abbiamo notato alcune differenze che riportiamo in maniera sintetica qui di seguito:

- Per poter creare un account Google Developer e quindi accedere alla Google Play Console, sono richiesti ad oggi 30$ per sempre. Inoltre non è richiesta una carta di credito per poter usufruire del servizio, ma è sufficiente anche una carta prepagata per poter effettuare il pagamento unico.

- L'URL con le informazioni relative alle norme sulla privacy non è più necessario, se non per quelle applicazioni dedicate ad un pubblico al di sotto dei 13 anni.
- Sono richiesti 2 tipi diversi di descrizione: una molto breve di massimo 80 caratteri e una completa di massimo 4000 caratteri.
- I tempi attualmente richiesti per la pubblicazione dell'app sono di circa una settimana per la prima pubblicazione. Dopodiché eventuali aggiornamenti sono pubblicati nel giro di un'ora.

## 4.4 Considerazioni su Android vs iOS

Nel corso dello sviluppo abbiamo riscontrato una spiccata semplicità nell'eseguire debugging di schermi di diverse dimensioni dal momento che in caso di overflow degli elementi dell'applicazione, il framework segnala l'errore a video e nella console. In entrambi i sistemi operativi questa funzionalità di Flutter funziona correttamente e ci ha permesso di mantenere un buon ritmo di sviluppo. Inoltre riguardo alle altre funzionalità offerte da Flutter, non abbiamo riscontrato particolari problemi nell'implementazione e nell'uso dei plugin presenti per quanto concerne il design grafico.
Un problema di maggior rilevanza lo abbiamo riscontrato nell'esecuzione di pezzi di codice in background, reso semplificato per Android, ma altamente incompatibile con iOS. Questo non ci ha permesso di mettere pienamente a punto la parte di *machine learning*, dal momento che l'uso di processi in background richiede lo sviluppo di una parte di app con linguaggio nativo su iOS e, per motivi di tempo, non abbiamo voluto dedicarci a ulteriori sviluppi.

## 4.5 Problemi riscontrati e difficoltà

Di seguito elenchiamo tutti i problemi riscontrati durante lo sviluppo di questa applicazione:

- Non siamo riusciti ad implementare lo *swipe* nelle pagine principali dell'applicazione, dal momento che questo entrava in contrasto con le gesture di sistema di iOS e di Android.
- Come riportato spiegato in appendice, l'algoritmo predittivo di deep-learning non risulta ad oggi affidabile per il tracciamento automatico delle attività e dunque la visualizzazione per l'utente delle attività svolte durante la giornata per migliorare le stime di CO2 risparmiata dall'utente. Per tale feature erano state sviluppate appositamente delle "Fitness Tile" con il quale l'utente sarebbe stato informato sulle attività tracciate, della loro durata e della CO2 risparmiata.
- Non siamo riusciti ad aggiungere le notifiche push in background per spronare l'utente a usare più spesso l'app, dato che questo richiedeva la registrazione e l'implementazione attraverso un servizio esterno di backend, non attuabile considerando il tempo dedicato al progetto.
- Il tutorial è stato leggermente riadattato per fare spazio ai bottoni "Avanti" e "Salta" visto che nativamente non erano presenti nel plugin utilizzato. Per questo motivo i tutorial a singola

scheda presentano un solo bottone ("Concludi"), mentre i tutorial con più schede li hanno entrambi.

- Sebbene la documentazione di Flutter sia molto sostanziosa, non è stata di facilissima lettura vista l'organizzazione dei contenuti e visti gli esempi non del tutto esplicativi.

# 5 Sviluppi futuri

Per la nostra applicazione sono state numerose le funzionalità che non abbiamo potuto implementare e approfondire maggiormente, per ragioni di tempo. Una tra queste consiste nel raffinamento dell'algoritmo per il calcolo delle abitudini dell'utente, coinvolgendo una parte di machine learning per comprendere quanto un utente è propenso a portare a termine una data attività, analizzando anche possibili correlazioni quali difficoltà dell'attività, stile di vita dell'utente e attività di fitness.
Anche lo sviluppo di un città dinamica, realizzata eventualmente con Unity, potrebbe rendere più piacevole e appagante l'uso dell'applicazione. Il corredo con qualche mini gioco, potrebbe inoltre intrattenere maggiormente l'utente.

Una tra le funzionalità che richiederà sicuramente maggior tempo per essere approfondita e sviluppata riguarderà l'algoritmo predittivo di deep-learning per introdurre anche il tracciamento delle attività automatico e fitness tile già previste in questa prima versione.

Inoltre l'aggiunta di notifiche con relativi messaggi di incoraggiamento sfruttando firebase, coinvolgerebbe maggiormente gli utilizzatori dell'applicazione, ricordandogli alcuni dei compiti fondamentali richiesti dall'app. Questo permetterebbe inoltre un maggior utilizzo da parte degli utenti, che non potranno scordarsi di compilare le attività portate a termine giorno per giorno. Infine, effettuando un collegamento remoto con un server e database online, si introdurrebbe nell'applicazione il concetto di "community". In questo modo altri gli utenti potrebbero cooperare, competere su nuove ed emozionanti sfide e vedere le attività portate a termine dai propri amici e parenti. Inoltre per incentivare maggiormente l'apprendimento di abitudini green, si potrebbe far vedere all'utente la CO2 risparmiata da tutta la "community" grazie all'impegno di tutte le persone che si impegnano ad usare l'applicazione ogni giorno.

# Appendice

## A Logica e funzionamento

### A.1 Struttura del database

In questa sezione tratteremo in maniera veloce e sintetica la parte relativa alla progettazione e logica del database. Abbiamo deciso dunque di riportare qui di seguito il diagramma ER del nostro database. Il database è stato progettato con in mente l'idea che quest'ultimo in un futuro possa essere messo online per supportare più utenti che usano contemporaneamente l'app. In questa prima versione dell'app abbiamo deciso di lavorare con un database locale all'interno dell'applicazione stessa in modo da essere più veloci nello sviluppo e nella configurazione di esso. Per tali motivi abbiamo scelto dunque di utilizzare SQLite, un database SQL molto facile da installare, veloce, semplice e self-contained, implementabile in Flutter grazie al plugin sqflite.

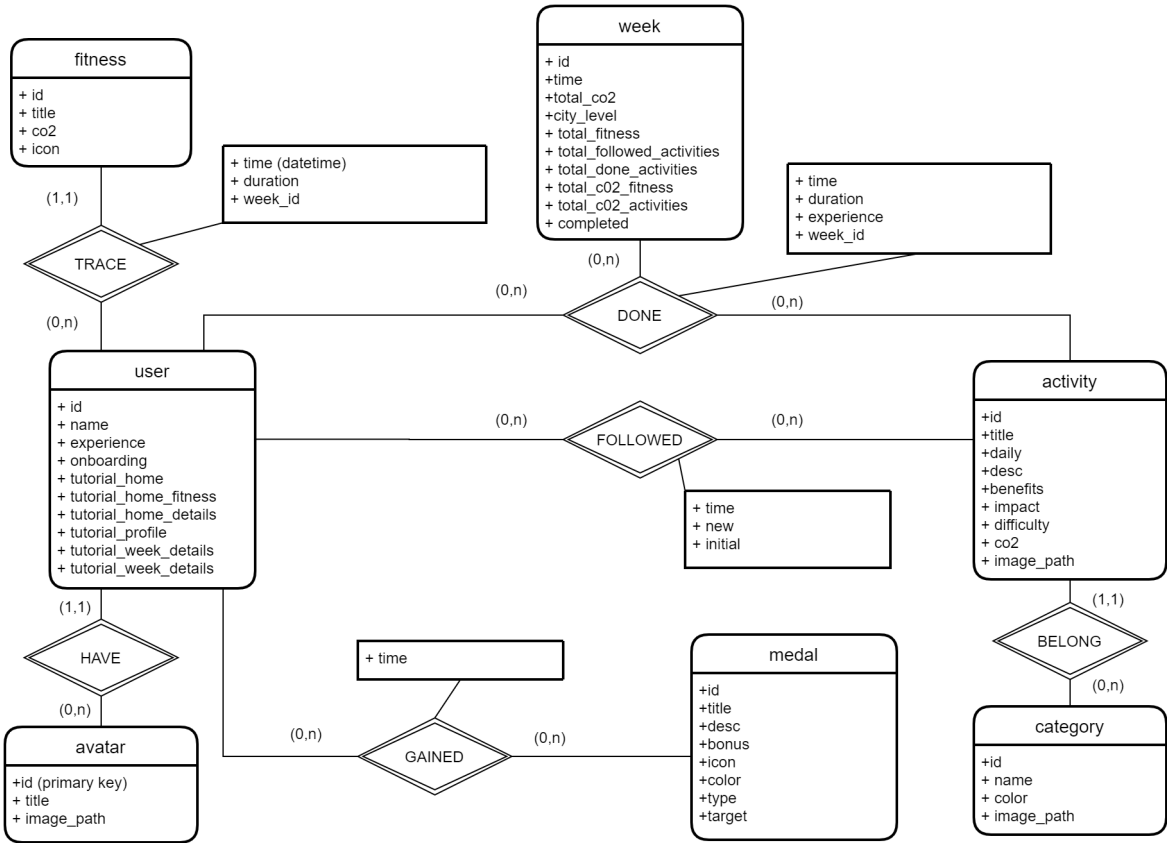



Figura 13: Digramma Entità-Relazione del database dell'applicazione

Vediamo ora velocemente le entità più importanti presenti al suo interno e il loro scopo:

- **user**: L'entità utente è incaricata di mantenere in memoria tutti i dati forniti dall'utente e per tracciare i suoi avanzamenti all'interno dell'applicazione. Infatti tale entità è incaricata di salvare il nome dell'utente, l'esperienza che esso ha guadagnato nel tempo, e tenere traccia di tutti i relativi tutorial che ha completato via via nell'app.

- **activity**: L'entità attività invece si occupa di tenere salvate tutte le informazioni che riguardano ogni singola attività, come il suo titolo, la sua descrizione testuale e come questa impatta nell'ambiente, nonché i livelli di difficoltà. Inoltre sono presenti tutte le informazioni logiche necessarie all'applicazione ai fini di riconoscere la categoria di appartenenza dell'attività, la CO2 risparmiata al suo completamento, l'immagine associata e il suo tipo (settimanale o giornaliera).

- **week**: Per ogni settimana trascorsa all'interno dell'applicazione verrà creata una nuova entità per salvare i progressi dell'utente settimana per settimana, in modo da poter poi ripercorrere la storia dell'utente come spiegato nella sezione "Storico delle città". Per questo motivo tale entità è incaricata di memorizzare per ogni settimana conclusa il totale della CO2 risparmiata, il livello della città raggiunto (scarso, normale, ottimo) e altre informazioni statistiche utili all'utente come il totale di attività seguite e quelle poi portate a termine.

- **medal**: L'entità medal rappresenta letteralmente le medaglie e i premi che l'utente può conseguire svolgendo un certo tipo di attività per un certo periodo di tempo. Attraverso questa entità vengono salvate le relazioni con i premi conseguiti dall'utente che verranno poi mostrati nella sezione profilo dell'applicazione.

## A.2 Calcolo della CO2 giornaliera e settimanale

Una delle parti più significative dell'applicazione è il calcolo di come deve essere aggiornato il livello della città. Come spiegato nella sezione "Homepage" la città dispone di tre livelli: scarsa, normale, ottima. Per scegliere in tempo reale il livello di città da far visualizzare all'utente è stato scelto di implementare una formula semplice ma al contempo efficace. Innanzitutto è stata stimata la CO2 che in media un utente dovrebbe risparmiare in una giornata-tipo; questo valore si attesta su 6 Kg. A questo punto abbiamo deciso che la giornata dell'utente inizi ipoteticamente alle 8 del mattino e, a partire da tale orario, viene considerato che l'utente inizi a impegnarsi nel completamento delle attività. Ora a partire da questa informazione sappiamo che dalle 8 di mattina fino alla mezzanotte l'utente dovrà aver risparmiato 6 Kg di CO2. A questo punto per conoscere quanta CO2 l'utente dovrà risparmiare in media ogni ora è sufficiente distribuire i 6 Kg di CO2 in 16 ore, ossia 24 ore totali in un giorno meno le 8 ore del mattino non considerate. Questo ci dice che in media ogni ora l'utente dovrebbe risparmiare 400 g di CO2.

A questo punto a qualsiasi ora del giorno possiamo stimare quanta CO2 l'utente dovrebbe aver risparmiato. Ora possiamo definire tre intervalli in cui l'utente vedrà comparire i tre diversi tipi di

città a seconda della CO2 risparmiata e del calcolo che abbiamo fatto per stimare la CO2 che l'utente avrebbe dovuto risparmiare in quell'istante. Gli intervalli scelti sono:

- $[2000 : +\infty]$ : Nel caso in cui l'utente abbia risparmiato più di 2 Kg di CO2 rispetto alla CO2 stimata, allora in questo caso l'utente vedrà comparire nella home la città più bella, e ciò significa che l'utente si è impegnato molto.
- $[-1000 : 2000]$: Se la CO2 risparmiata dall'utente si discosta di qualche Kg rispetto a quella richiesta, allora l'utente vedrà la città normale nella home.
- $[-\infty : -1000]$: Se l'utente si è impegnato poco e risulta essere indietro di più di un 1 Kg rispetto alla CO2 prestabilita in quel momento, allora l'utente vedrà la città peggiore in quanto non si è ancora impegnato abbastanza, così da invogliarlo a fare meglio nel resto della giornata.

Infine per decretare il livello della città al completamento della settimana, che poi sarà salvato anche nello storico utente, il calcolo risulta essere più semplice e intuivo. Stimando che in media ogni giorno un utente possa risparmiare 6 Kg di CO2 è sufficiente moltiplicare tale numero per 7 per ottenere la quantità di CO2 da risparmiare a settimana, in questo caso appunto 42 Kg. Considerato che un utente possa un giorno (o qualche giorno) dimenticarsi di segnare qualche attività come completata, abbiamo optato per un piccolo bonus e fissato la soglia di 35 Kg di CO2 totali a settimana per potersi guadagnare la città migliore. Infine abbiamo scelto quale soglia al di sotto della quale la città sarebbe stata la peggiore possibile, e tale soglia è stata fissata a 15 Kg. Pertanto nell'intervallo tra 15 Kg e 35 Kg di CO2 risparmiata in una settimana l'utente vedrà la città normale alla fine della settimana.

Gli intervalli e le soglie per decretare la città dell'utente non sono frutto di ricerche scientifiche e potrebbero non essere adatte ad ogni utente. Questo poiché potrebbero esserci utenti più interessati al tema e quindi invogliati nel completare più attività possibili e altri invece, ancora poco sensibili al tema che si dimostrerebbero più svogliati e che pertanto potrebbero essere scontenti e amareggiati di poter raggiungere solo la città normale o la più scarsa, abbandonando così l'applicazione. Per tali motivi sarebbe utile innanzitutto svolgere uno studio più approfondito sul tema ambientale per capire gli intervalli corretti da inserire nella nostra applicazione, affinché la terra possa prosperare nel prossimo futuro. Inoltre tramite machine learning, questi stessi intervalli potrebbero essere modificati in base alle abitudini dell'utente, all'attitudine e all'impegno di quest'ultimo. Sarebbe inoltre anche possibile, attraverso l'implementazione di *recommender-systems*, proporre all'utente attività per le quali è più probabile che quest'ultimo possa portare a termine, analizzando i suoi interessi e le relazioni di similarità tra altri utenti che utilizzano l'app.

## A.3 Algoritmo per le abitudini

Ciascuna attività che l'utente segue rappresenta un obiettivo che questo si pone. Dopo un po' che viene completato ricorrentemente comincia a diventare un obiettivo abitudinario. In questo modo

vogliamo instaurare nell'utente un comportamento responsabile e attento al tema nel momento in cui esso si ritrovi a compiere delle scelte eco-sostenibili o meno. Chiaramente prima che un'attività possa realmente diventare un'abitudine di vita, c'è bisogno di molto tempo e ciò può cambiare da persona a persona, sia per propria spinta personale che per proprio bagaglio di vecchie abitudini dure a mutare. Tale problema è già stato ampliamento studiato negli anni, e uno degli strumenti più rappresentativi viene proposto attraverso il Modello di Fogg[1].

Nel nostro caso, abbiamo voluto mantenere un approccio che potesse instaurare delle abitudini nel più breve tempo possibile, ragionando a livello di algoritmo non tanto in termini di propensione personale, ma quanto più in termini di attività svolte in piccole unità di tempo. Considerando il tempo di utilizzo dell'app, abbiamo voluto gratificare più velocemente l'utente rispetto agli effettivi tempi tecnici del nostro cervello richiesti per acquisire un'abitudine, verificando e controllando quotidianamente se un certo tipo di attività venisse effettivamente portata avanti come abitudine o meno.

L'algoritmo controlla nell'uso quotidiano dell'applicazione l'andamento delle attività svolte dall'utente, sulla base delle attività seguite. Nello specifico a seconda della tipologia di attività, verrà contrassegnata come abitudine nel seguente modo:

- **Attività giornaliera**: l'utente deve svolgere per almeno 7 giorni consecutivi tale attività seguita.
- **Attività settimanale**: l'utente deve svolgere per almeno 2 settimane consecutive tale attività seguita.

Quando un'attività diventa abitudine compare un'etichetta direttamente nella tile dell'attività in homepage, e questo permette all'utente di vedere a colpo d'occhio quali cambiamenti sta perseguendo con l'uso dell'app. Per mantenere un'abitudine viene fatto un controllo quotidiano che verifica se l'utente sta comunque svolgendo l'attività. In particolare, un'abitudine smette di essere tale se:

- **Attività giornaliera**: l'utente smette di registrare un'attività giornaliera per più di 7 giorni dall'ultima registrazione.
- **Attività settimanale**: l'utente non porta a termine un'attività settimanale per più di 1 settimana dall'ultima registrazione.

Chiaramente questo tipo di controllo verrà perfezionato in futuro nel momento in cui si riuscisse ad avere maggiori informazioni sullo stile di vita dell'utente, diventando così più mirato e più preciso per il singolo utente. Sfruttando un controllo di breve periodo per il momento, è più facile verificare la costanza e l'impegno che l'utente sta mettendo per perseguire i suoi obiettivi.

[1] FOGG, B. A behavior model for persuasive design

## A.4 Integrazione con il machine learning

Una caratteristica peculiare di questa app è l'integrazione di un algoritmo di deep-learning, per la predizione dell'attività svolta attualmente dall'utente (HAR - Human Activity Recognition). Grazie al progetto svolto per un altro corso magistrale, abbiamo deciso di provare ad integrare l'algoritmo di predizione realizzato anche all'interno di questa applicazione. Attraverso l'uso di un plugin per Flutter chiamato tflite_flutter, è stato possibile in maniera agevole e veloce eseguire il modello predittivo trasformato in un modello Tensorflow Lite all'interno dell'applicazione.

L'algoritmo implementato ad oggi è in grado di riconoscere, attraverso il solo utilizzo dei dati provenienti dall'accelerometro e dal giroscopio del telefono, alcune delle attività rilevanti per la nostra applicazione. Le attività tracciabili sono: camminata, bicicletta, scale in salita e scale in discesa ed infine la sedentarietà, in cui l'utente non sta svolgendo nessuna delle attività rilevanti. Purtroppo a causa della scarsa generalizzazione di questi algoritmi e per come essi vengono addestrati, ad oggi l'algoritmo integrato non è stato in grado di ottenere risultati ottimali in ambienti realistici e non controllati, come quelli in cui ci si trova normalmente nell'utilizzo dello smartphone. Al minimo movimento del telefono, infatti ci siamo accorti che spesso l'algoritmo tende a classificare questo comportamento come una delle attività su cui l'algoritmo è stato addestrato, anche se tale comportamento non ha nulla a che vedere con le attività elencate prima. Inoltre in situazioni particolari e diverse, come la camminata in spazi chiusi, dove sono presenti anche ostacoli da evitare, l'algoritmo non è in grado di predire correttamente l'attività e questo problema è imputabile sempre alla scarsa generalizzazione e alla mancanza di un addestramento adeguato della rete neurale. Per tali motivi, nonostante siamo riusciti a far eseguire un modello neurale all'interno dell'applicazione, abbiamo deciso di disabilitare la funzionalità, in attesa di un miglioramento in grado di predire correttamente l'attività svolta anche in quelle situazioni più comuni dove l'app viene utilizzata. In ogni caso sarà possibile visionare i nostri risultati scientifici ottenuti con tale modello predittivo in situazioni controllate in un paper dedicato che sarà a breve pubblicato.