

POLITECNICO DI TORINO

Laurea Magistrale in Ingegneria Informatica

Tesi di Laurea Magistrale



**Politecnico  
di Torino**

StepByStep: un'applicazione educativa per il  
Digital Wellbeing

**Relatori:**  
Luigi De Russis  
Alberto Monge Roffarello

**Candidato:**  
Giulio Piacentini

Ottobre 2021



# Ringraziamenti

Questa tesi di laurea è il passo conclusivo del mio lungo percorso universitario, che è stato possibile solo grazie alla collaborazione di molte persone, che voglio ringraziare.

Un ringraziamento particolare va fatto a Chiara, non solo per l'avermi accompagnato per tutti questi anni, ma soprattutto per essere stata il motivo primario per cui molti errori ortografici sono stati eliminati da questo elaborato.

Alla mia famiglia, per tutto quello che hanno fatto per permettermi di essere qui. A mamma, per essere fonte di validi consigli. A papà, per essere sempre riuscito a sdrammatizzare qualsiasi cosa. A Gaia, perché è Gaia. Ultima ma non ultima, a Michela. In generale, un grazie a tutte le persone che fanno parte della mia famiglia allargata.

Al teamBarge, per l'innata capacità di eliminare la serietà da qualsiasi contesto. A Su Col Morale e PoliAL, per essere stati fonte di molte risate durante gli anni di università.

Ai miei compagni del rugby, per avermi sempre dato la valvola di sfogo di cui ho avuto bisogno in questi anni.



# Indice

Elenco delle tabelle	VI
Elenco delle figure	VII
<b>1 Introduzione</b>	<b>2</b>
<b>2 Background e stato dell'arte</b>	<b>5</b>
2.1 Fondamenti teorici	5
2.1.1 Autoconsapevolezza e cambiamento comportamentale	5
2.1.2 Abitudini e principali teorie	7
2.1.3 <i>L'Habit Alteration Model</i>	9
2.2 Approcci tradizionali al Digital Wellbeing	11
2.2.1 Formazione di abitudini	11
2.2.2 Rottura di abitudini	12
2.2.3 Metodi comuni delle app di Digital Wellbeing	13
<b>3 Progettazione</b>	<b>16</b>
3.1 Casi d'uso	18
3.2 Design	20
3.2.1 <i>Path ed intervention</i>	20
3.2.2 Suggerimento proattivo	24
3.2.3 Prima Settimana	26
3.2.4 Architettura	27
3.3 Scheletro delle schermate	27
3.3.1 Monitoraggio	28
3.3.2 Notifiche	29
3.3.3 <i>Paths</i>	31
3.3.4 <i>Interventions</i>	32
3.3.5 Risultati e Riepilogo	33

<b>4</b>	<b>Implementazione</b>	<b>36</b>
4.1	Architettura . . . . .	36
4.2	MVVM e dati . . . . .	38
4.2.1	Firebase . . . . .	39
4.3	Navigazione e fragments . . . . .	40
4.3.1	Monitoraggio . . . . .	40
4.3.2	Notifiche . . . . .	40
4.3.3	Paths . . . . .	42
4.4	Interventions . . . . .	43
4.4.1	Intervention 1 . . . . .	45
4.4.2	Intervention 2 . . . . .	48
4.4.3	Passaggio di livello . . . . .	51
4.5	Prima apertura . . . . .	53
<b>5</b>	<b>Valutazioni</b>	<b>57</b>
5.1	Struttura del test . . . . .	57
5.2	Risultati . . . . .	58
5.2.1	Partecipanti . . . . .	58
5.2.2	Usabilità . . . . .	59
5.2.3	Efficacia . . . . .	61
5.3	Discussione . . . . .	64
<b>6</b>	<b>Conclusioni</b>	<b>67</b>
6.1	Limiti della ricerca . . . . .	67
6.2	Sviluppi futuri . . . . .	68
<b>A</b>	<b>Questionari</b>	<b>70</b>
A.1	Questionario 1 . . . . .	70
A.2	Questionario 2 . . . . .	71

# Elenco delle tabelle

4.1	Valutazioni date sul numero di sblocchi del telefono al termine dell' <i>intervention</i> 1.1 . . . . .	47
4.2	Valutazioni date sul numero di aperture delle applicazioni al termine dell' <i>intervention</i> 1.2 . . . . .	48
4.3	Penalità considerate dall' <i>intervention</i> 2 . . . . .	51
4.4	Valutazioni date in base alle penalità al termine dell' <i>intervention</i> 2	51
4.5	Regole per il passaggio di livello nel <i>Path</i> 1 . . . . .	53
4.6	Regole per il passaggio di livello nel <i>Path</i> 2 . . . . .	53

# Elenco delle figure

2.1	Fasi del <i>Transtheoretical Model</i> e interventi che possono aiutarne l'avanzamento. Immagine tratta da Djik et al [12] . . . . .	6
2.2	Rappresentazione del modello dei due sistemi. Immagine tratta da Lyngs et al. [4] . . . . .	9
2.3	Rappresentazione dell' <i>Habit Alteration Model</i> . Immagine tratta da Pinder et al. [8] . . . . .	10
3.1	Schema riassuntivo delle funzionalità principali di StepByStep . . . . .	18
3.2	Configurazioni dell' <i>intervention</i> 1.1 divise per livello . . . . .	22
3.3	Configurazioni dell' <i>intervention</i> 1.2 divise per livello . . . . .	23
3.4	Configurazioni dell' <i>intervention</i> 2 divise per livello . . . . .	24
3.5	Scheletro delle schermate dell'applicazione . . . . .	28
3.6	Mockup della schermata di notifica di uso eccessivo di un applicazione	30
3.7	Mockup della schermata di notifica di inattività del <i>Path</i> 1 . . . . .	30
3.8	Mockup della schermata <i>Home</i> . . . . .	31
3.9	Mockup delle schermate di avvio di un <i>Path</i> . . . . .	32
3.10	Mockup delle schermate di un <i>intervention</i> 1 Versione 1 . . . . .	33
3.11	Mockup della schermata di esito di un <i>intervention</i> . . . . .	34
3.12	Mockup della schermata di passaggio di livello . . . . .	34
4.1	Principali componenti di StepByStep . . . . .	37
4.2	Schermate di monitoraggio di StepByStep . . . . .	41
4.3	Schermata di riepilogo alla fine del periodo di monitoraggio . . . . .	44
4.4	Schermate per l'avvio dell' <i>intervention</i> 1 (versione 1) . . . . .	45
4.5	Schermate per l'avvio dell' <i>intervention</i> 2 . . . . .	49
4.6	Schermate di riepilogo per il passaggio di livello di entrambi i <i>Path</i>	52
4.7	Alcune delle schermate di tutorial di StepByStep . . . . .	54
5.1	Risposte alla domanda sulla percezione di utilizzo del cellulare, con una scala Likert da pessimo (1) a ottimo (5) . . . . .	59



5.2	Risposte alla domanda sull'intenzione di migliorare il proprio utilizzo del cellulare, con una scala Likert da poco (1) a molto (5) . . . . .	59
5.3	Grafico del numero di aperture totali delle applicazioni su cui si è attivato il <i>Path 1</i> . . . . .	63
5.4	Grafico del numero di aperture totali delle applicazioni su cui è stato attivato il <i>Path 1</i> di un singolo partecipante, con particolare delle giornate di utilizzo di un <i>intervention</i> . . . . .	63
5.5	Grafico del numero di ore totali spese su applicazioni su cui si è attivato il <i>Path 2</i> . . . . .	64



# Capitolo 1

## Introduzione

Lo smartphone è ormai diventato una presenza costante nella nostra vita quotidiana e, sebbene porti con sé numerosi vantaggi, il suo utilizzo può essere a volte legato a problematiche di varia natura. Diverse ricerche hanno dimostrato che lo smartphone è spesso fonte di distrazione, il suo uso eccessivo può essere fonte di stress e portare a problemi di salute mentale [1, 2], fino ad arrivare a casi estremi di vere e proprie dipendenze da Internet o dai social [3].

In risposta a ciò negli ultimi anni si è manifestato un crescente interesse per la tematica del Digital Wellbeing, ovvero il raggiungimento e mantenimento di una situazione di benessere per la persona all'interno di un contesto digitale e tramite mezzi appartenenti allo stesso contesto. Sono nate molte applicazioni, anche di colossi del settore quali Google ed Apple, che si propongono di fare ciò, aiutando le persone a controllare l'uso che fanno dei propri device e combattendo la "dipendenza da cellulare".

La maggior parte di queste applicazioni fa uso di tecniche ormai comuni e consolidate, la più diffusa delle quali è l'implementazione di alcuni interventi, come timer e blocchi, che limitino l'utente dall'usare eccessivamente il telefono. Molte applicazioni, inoltre, implementano una qualche forma di monitoraggio dell'uso del cellulare e/o delle applicazioni. In percentuale minore sono invece presenti tecniche relative all'avanzamento di obiettivi o una qualche forma di reward, con caratteristiche di gamification o condivisione sociale [4, 5].

Strumenti quali timer e blocchi, per quanto siano molto diffusi, hanno delle limitazioni, poiché sono dei vincoli che l'utente deve autoimporsi per impedire a sé stesso di usare il cellulare in maniera "dannosa". Quindi si fornisce all'utente un mezzo per controllarsi ma allo stesso tempo lo si rende dipendente da questo mezzo, la cui rimozione comporterebbe un ritorno dell'utente al suo abituale comportamento "errato" [6]. Non si cerca in alcun modo di insegnare all'utente come usare correttamente il proprio cellulare o cambiare il proprio comportamento abituale.

Quello che si è quindi fatto in questo percorso di tesi è l'ideazione, creazione e verifica della funzionalità di un'applicazione mobile, StepByStep, che si inserisce nel contesto del Digital Wellbeing senza appoggiarsi a soluzioni tradizionali ma proponendo un approccio più educativo, insegnando all'utente qual è il comportamento per lui più corretto e aiutandolo a mantenerlo. Questo avviene tramite la creazione di un percorso che vada a favorire, nel suo avanzamento, l'indipendenza dell'utente nei confronti dell'applicazione, in modo che, idealmente, il comportamento acquisito si possa mantenere autonomamente.

Le basi teoriche di questa applicazione si fondano sulla teoria psicologica dei due sistemi di Kahneman [7], già da molti indicata come un ottimo punto di partenza per l'implementazione di strumenti di autoregolazione e autocontrollo in ambito Digital Wellbeing [4, 8]. Secondo questa teoria, ogni nostra decisione è risultante dall'azione di due sistemi di pensiero: il Sistema 1, veloce, emozionale ed inconsapevole, e il Sistema 2, più lento e razionale. Come verrà spiegato in seguito, molti dei comportamenti abitudinari scorretti sono dipendenti dal primo sistema. Il principio fondamentale di StepByStep sarà quindi quello di favorire un'attivazione del Sistema 2, consapevole, e quindi inibire il Sistema 1.

Un maggiore approfondimento delle teorie psicologiche che fanno da base all'applicazione, in particolare dal punto di vista applicativo nell'ambito del Digital Wellbeing, è trattato nel Capitolo 2.

Nel Capitolo 3, si espone il processo di ideazione e progettazione dell'applicazione, discutendo in maniera più approfondita gli elementi che coprono il ruolo di attivatori del Sistema 2.

L'implementazione vera e propria dell'applicazione è discussa nel Capitolo 4.

Nel Capitolo 5, viene descritto il test attuato sull'applicazione per verificarne usabilità ed efficacia, e ne sono discussi i risultati.

Infine, nel Capitolo 6, vengono presentate alcune riflessioni conclusive, con un focus sui limiti di questa ricerca e sui possibili sviluppi futuri.



# Capitolo 2

## Background e stato dell'arte

### 2.1 Fondamenti teorici

#### 2.1.1 Autoconsapevolezza e cambiamento comportamentale

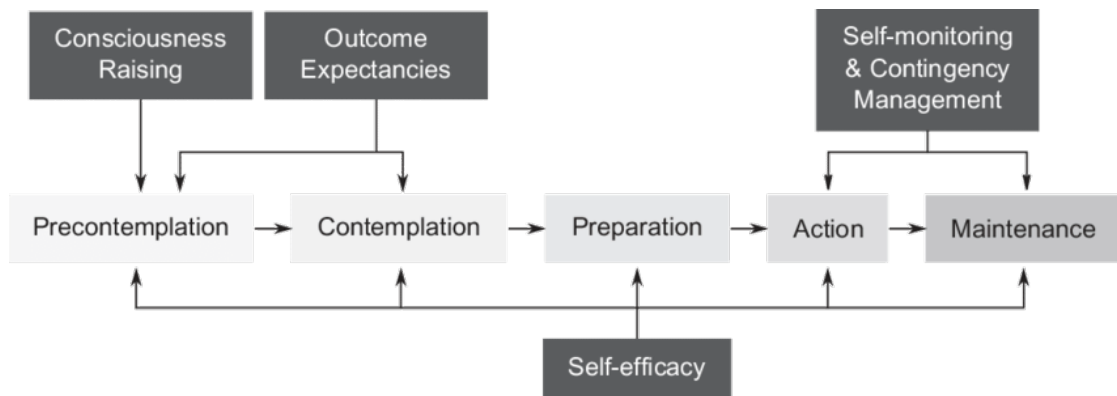
Come introdotto nel capitolo precedente, gran parte delle applicazioni di Digital Wellbeing esistenti implementa una qualche forma di monitoraggio. Il fine di ciò è ovviamente permettere all'utente di controllare efficacemente l'uso che fa del proprio smartphone e, in ultimo, innescare un cambiamento di comportamento quando non si trovi soddisfatto di quello che fa. Tuttavia, sebbene l'idea che il monitoraggio possa favorire una spinta verso l'autocontrollo e la modifica del proprio comportamento sia intuitiva, la realtà è ovviamente più complessa.

La tecnica del monitoraggio e autocontrollo è già ampiamente usata, con un'efficacia dimostrata, in altri domini, sempre collegati al cambiamento comportamentale, come il monitoraggio della dieta o del consumo energetico [9, 10]. Secondo la teoria dell'autoregolazione di Bandura [11] gli effetti del monitoraggio sono principalmente dovuti al confronto tra il comportamento monitorato e alcuni standard e obiettivi; in seguito a ciò, i comportamenti diversi dallo standard vengono aggiustati. Da un punto di vista comportamentista gli effetti possono ricollegarsi a una forma di condizionamento operante, in cui la deviazione dallo standard è percepita come una punizione. Un altro fattore da considerare è anche la presenza di un "observer effect": il sentirsi osservato, anche se solo da un dispositivo elettronico, può incentivare l'utente a mantenere un comportamento che considera corretto. [12]

Tipicamente, però, iniziare a modificare un comportamento richiede più del semplice "sapere cosa si dovrebbe fare" [12]. In particolare, il *Transtheoretical Model* [13] viene indicato come un buon fondamento teorico per l'implementazione di applicazioni che stimolino cambiamenti comportamentali [12].

Secondo il *Transtheoretical Model*, il processo che conduce al cambiamento comportamentale è suddiviso in cinque stadi: precontemplazione, contemplazione, preparazione, azione e mantenimento.

Il processo comincia ben prima dell'effettivo inizio del nuovo comportamento. Nella fase di precontemplazione, la persona, ancora non consapevole del problema dovuto al proprio comportamento, non ha alcuna motivazione per modificarlo. Nella fase di contemplazione, la persona diventa consapevole del problema ma non è ancora intenzionata ad avviare un cambiamento, cosa che invece avverrà nella fase di preparazione, in cui si pianificherà come modificare il proprio comportamento. Nella fase di azione si attuerà il cambiamento del comportamento, mentre la fase di mantenimento è incentrata sul consolidare il nuovo comportamento, per evitare ricadute [12].



**Figura 2.1:** Fasi del *Transtheoretical Model* e interventi che possono aiutarne l'avanzamento. Immagine tratta da Djik et al [12]

Facendo riferimento alla Figura 2.1, in questo modello vi sono una serie di possibili interventi, i quali facilitano il progresso di una persona da una fase a quella successiva, che esporremo di seguito nel contesto del Digital Wellbeing:

- *consciousness raising*: un input fornito all'utente che aumenti la sua consapevolezza sulla presenza di un problema dovuto a un suo modello comportamentale;
- *outcome expectancies*: processo con il quale vengono resi noti all'utente i vantaggi del cambiamento comportamentale, che non sempre sono facilmente percepibili dalla persona, soprattutto nelle fasi iniziali, in cui gli svantaggi riguardanti l'avvio del cambiamento appaiono maggiori
- *self-efficacy*: spesso indicata come una componente; fondamentale per il cambiamento del comportamento, è la percezione che una persona ha della propria capacità di raggiungere finalità e obiettivi (in questo caso, adottare

un nuovo comportamento) [11]. Può venire influenzata da diverse fonti di informazione, tra cui, per esempio, esiti positivi di azioni compiute nel passato dalla persona [11];

- *contingency management and self-monitoring*: la persona deve monitorare il proprio avanzamento, tenendo conto degli ostacoli che si presentano; ciò si può implementare fornendo feedback immediati all'utente sul proprio progresso [11].

In ambito di Digital Wellbeing, si è dimostrato che l'utilizzo del monitoraggio per aumentare la comprensione di un utente del proprio comportamento e innescare un cambiamento comportamentale abbia un'innegabile efficacia, ma non universalmente. In alcuni casi, far diventare l'utente consapevole di alcuni aspetti del suo comportamento non causa un avvio del processo di cambiamento, soprattutto quando i nuovi dati non sono di reale interesse per l'utente [12]. Il tipo di monitoraggio che si è rivelato più promettente è quello riguardante il progresso di un utente verso un obiettivo o una norma [11].

Il fornire supporto all'utente durante tutte le fasi che caratterizzano il cambiamento comportamentale è quindi fondamentale nello sviluppo di un applicazione di Digital Wellbeing, così come avere una particolare attenzione alla *self-efficacy* [12].

## 2.1.2 Abitudini e principali teorie

Per poter parlare correttamente del nostro uso quotidiano del cellulare, però, è necessario definire il concetto di abitudine. Per abitudine si intende una risposta comportamentale automatica ad uno stimolo, originatasi a partire dalla ripetizione del comportamento in determinati contesti [14]. Nello specifico, in seguito all'esecuzione di una nuova azione, si crea un'associazione mentale fra la stessa e l'antecedente interno o contestuale che l'ha innescata. La ripetizione di questa azione rafforza e stabilisce questo collegamento nella memoria, rendendo questo comportamento più probabile e accessibile nella medesima situazione, fino ad assumere la caratteristica di automaticità. La forza di questa connessione nella traccia mnemonica rende molto difficile modificare un'abitudine, anche quando ve ne sia l'intenzione [15].

Quei comportamenti che gli strumenti di Digital Wellbeing si propongono di combattere rientrano in questo contesto abitudinario: si tratta di azioni ripetute nel tempo e riconducibili alla struttura *context-response* definita prima (e.g. ho bisogno una pausa dallo studio → apro Facebook). L'uso del cellulare è contraddistinto da sezioni di breve durata ma frequenti, definite *checking habits*, in cui l'utente interrompe momentaneamente altre attività per accedere a del contenuto dinamico, spesso associato ai social [16]. Questo continuo interrompersi, sia esso dovuto alla ricezione di notifiche o autoindotto, ha effetti molto negativi sulla produttività di



una persona [17, 18, 19]. La grande sfida delle applicazioni di Digital Wellbeing è trovare il modo di rompere questi comportamenti abitudinari, che sono però molto resistenti al cambiamento [8].

Per i Comportamentisti le abitudini sono collegamenti stimolo-risposta eseguiti fuori dal sistema decisionale cosciente, tramite due meccanismi: condizionamento classico ed operante, che, rispettivamente, indicano l'apprendimento di una relazione fra eventi (stimolo  $\rightarrow$  risposta) e l'apprendimento di una relazione fra le risposte comportamentali e le loro conseguenze (stimolo-risposta  $\rightarrow$  risultato), positive o negative [20]. Quindi, premiare un determinato comportamento rinforza questi collegamenti e ne favorisce la ripetizione. Un'abitudine è definita tale quando la rimozione del premio non provoca una diminuzione del comportamento. Una delle critiche mosse al Comportamentismo riguarda il fatto che non prende in considerazione i processi mentali sottesi ad un dato comportamento, come intenzioni, aspettative o la definizione degli obiettivi, nella formazione delle abitudini [8].

Al contrario, invece, le teorie Cognitiviste, in particolare la Teoria del Comportamento Pianificato [21], specificano che il comportamento è guidato dalle intenzioni di una persona e dalla sua abilità nell'eseguirlo. La teoria ha tuttavia limitate applicazioni alle abitudini, poiché non considera la ripetizione automatica di un comportamento [8].

Queste due scuole di pensiero classiche hanno delle notevoli limitazioni nel contesto delle abitudini, per questo, le teorie principali su cui si basa StepByStep appartengono a modelli integrati (di cui fa parte il *Transtheoretical Model*), che considerano entrambe le correnti.

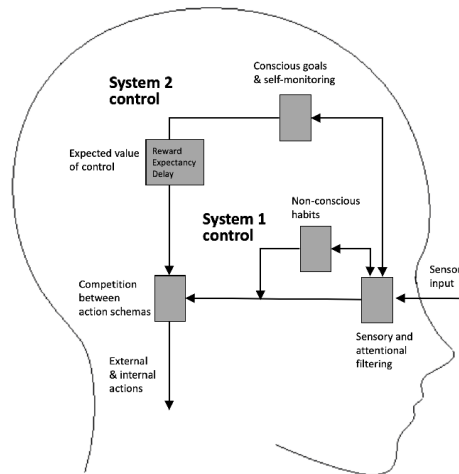
### ***Dual-Process Theory***

La *Dual-Process Theory* [7] definisce il comportamento come un risultato tra due diversi sistemi:

- sistema di Tipo 1 - automatico, inconsapevole, impulsivo, associativo, contestuale e rapido;
- sistema di Tipo 2 - più lento, razionale, conscio, guidato dagli obiettivi.

Il sistema di Tipo 1 riceve gli input dall'ambiente circostante e mappa l'attuale situazione, assieme allo stato interno, a risposte istintive o abitudini; è spesso definito automatico, poiché permette di iniziare a compiere delle azioni senza inizialmente esserne del tutto consapevoli. Il sistema di Tipo 2 è invece guidato da obiettivi, intenzioni e regole presenti nella memoria esplicita. L'effettivo comportamento sarà il risultato della competizione dei due sistemi. [4]

Sebbene questa teoria indichi il sistema di Tipo 1 come principale contributore alle abitudini, non nega la possibilità di influenzarle con il sistema di Tipo 2. Il comportamento è una risultante dell'interfacciarsi dei due sistemi, e il sistema di



**Figura 2.2:** Rappresentazione del modello dei due sistemi. Immagine tratta da Lyngs et al. [4]

Tipo 1 prevale soprattutto quando le “risorse” del sistema di Tipo 2 sono esaurite. Vi è una limitata capacità di informazioni che il sistema di Tipo 2 può tenere nella memoria di lavoro, tipicamente  $7 \pm 2$  *chunk* [22]. Questo è influenzato anche dallo stato emotivo della persona e soffre degli effetti della fatica, non sono perciò rari i casi in cui il sistema, soggetto a particolari pressioni (e.g durante distrazioni, carico cognitivo elevato, umore avverso, stress o fatica), non funzioni correttamente e non riesca a mantenere le informazioni necessarie in memoria, venendo quindi dominato dal sistema di Tipo 1 [8, 4].

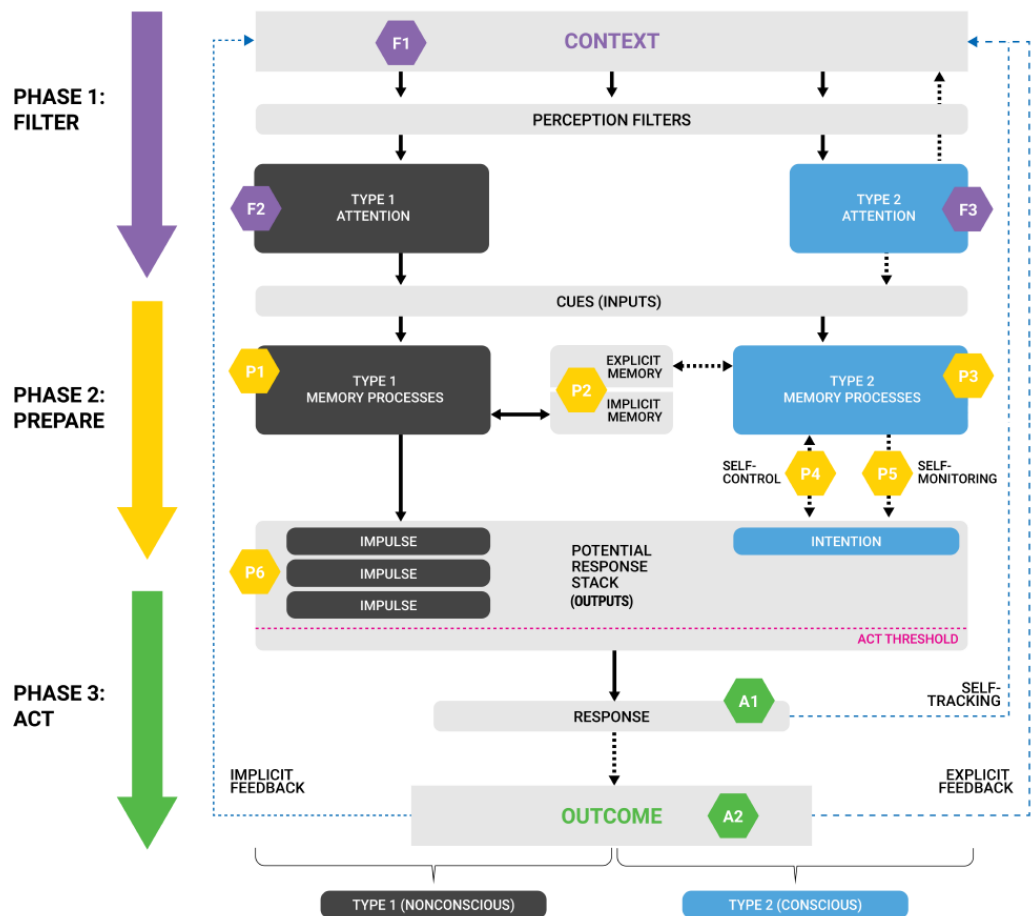
### ***Goal Setting Theory***

La *Goal Setting Theory* [23, 24] esplora in che modo formare gli obiettivi per guidare il comportamento quando predominano i processi di Tipo 2. In particolare, si pone accento sul fatto che sia necessario un continuo feedback sul progresso del percorso verso l’obiettivo e che l’efficacia sia direttamente proporzionale a due fattori: la difficoltà e la specificità. È meglio quindi definire un obiettivo specifico e difficile rispetto ad uno vago e semplice [8].

### **2.1.3 L’*Habit Alteration Model***

L’*Habit Alteration Model* è un modello concettuale virtuale che combina *Dual Process Theory*, *Goal-Setting Theory* e le moderne teorie delle abitudini [8].

Come mostrato in Figura 2.3, il processo che porta all’attuazione di un comportamento dati alcuni segnali (*cues*) è diviso in tre stadi:



**Figura 2.3:** Rappresentazione dell'*Habit Alteration Model*. Immagine tratta da Pinder et al. [8]

- *Filter*  
In questa fase i segnali, che possono provenire sia dall'ambiente esterno che interno, vengono captati e filtrati prima dal filtro di percezione e poi dai filtri di attenzione di Tipo 1 e Tipo 2.
- *Prepare*  
I risultati della fase precedente vengono trattati dai processi mnestici di Tipo 1 e Tipo 2, che li correlano con le potenziali risposte, impulsi (Tipo 1) e intenzioni (Tipo 2), le quali popoleranno il *Potential Response Stack*.
- *Act*  
Una competizione tra le possibili risposte nello *Stack* determina il comportamento che verrà attuato. Per poter risultare come vincitori le risposte devono

superare una certa soglia (*act threshold*). Dall'esito del comportamento risultante vengono poi raccolte varie informazioni, che fanno da feedback al modello e ai due processi (similmente al condizionamento operante).

Il comportamento è quindi funzione di contesto, segnali, differenze individuali e la mappatura di impulsi e intenzioni da parte dei due sistemi [8].

Con una sufficiente ripetizione di comportamenti semplici in un contesto stabile vi è un'automatizzazione del processo *Filter-Prepare-Act*. Si va a rafforzare e velocizzare il collegamento stimolo → risposta e ad aumentare la sua posizione nel *Potential Response Stack*. Si struttura così una nuova abitudine.

Quando più comportamenti superano la *act threshold* nell'ultima fase, i processi di Tipo 2 (consapevoli) vengono allertati per decidere quale scegliere [25].

## 2.2 Approcci tradizionali al Digital Wellbeing

### 2.2.1 Formazione di abitudini

Ora che abbiamo visto alcune delle teorie su comportamento e abitudini, è utile presentare come queste vengano applicate all'interno del contesto del Digital Wellbeing.

Le funzionalità di monitoraggio usate da gran parte delle applicazioni di Digital Wellbeing [4, 6], nonostante supportino il cambio comportamentale, come illustrato a Sezione 2.1.1, hanno un limitato collegamento alla formazione delle abitudini. L'automonitoraggio rimane efficace solo se continuo nel tempo; nel caso in cui cessasse, il vecchio comportamento riemergerebbe [6, 14].

Poiché la chiave nel consolidamento di un'abitudine risiede nella ripetizione del tempo di un comportamento, le applicazioni di questo tipo dovrebbero aiutare a mantenere questa ripetizione e fornire un supporto continuo, per tutto il tempo necessario. Questa durata è altamente dipendente dalla complessità del comportamento che si vuole ottenere e può variare dai 18 giorni per comportamenti particolarmente semplici fino a circa 254 giorni per quelli più complessi [26].

Alcuni dei possibili interventi implementabili dalle applicazioni per favorire la formazione delle abitudini sono:

- *Cues*  
Sono definiti tali degli stimoli esterni che vengono innescati da eventi. Il loro basarsi su eventi li rende molto efficaci, poichè è molto più semplice per una persona compiere un'azione all'interno di una determinata routine (e.g. prendere un medicinale dopo cena) piuttosto che in un determinato momento (e.g. prendere un medicinale alle 10:00) [6].
- *Reminders*  
Anche in questo caso si tratta di stimoli esterni ma bastati sul tempo, quindi

innescati in determinati orari. La loro utilità è limitata, poichè, per quanto il rispondere a un *reminder* possa essere visto come automatico, in realtà non si automatizza in alcun modo il comportamento in sè. Anche se il loro utilizzo può avere effetti iniziali benefici sulla ripetizione, la loro efficacia decade velocemente nel tempo [6].

- *Rewards*  
Consistono nell'implementazione del rinforzo positivo nel contesto del condizionamento operante: dare un premio per lo svolgersi un una determinata azione. Questi premi possono anche essere intrinseci (e.g. sensazioni di piacere o soddisfazione) [6]. Sono efficaci solo nel caso in cui il premio sia diverso da quello che è l'obiettivo in sè [26].

I *cues*, rispetto ai *reminders*, sono molto più lenti nello sviluppo di automatismi, ma anche di gran lunga più efficaci di questi ultimi, il cui effetto, per quanto ottimo inizialmente, finisce con l'essere controproducente proseguendo nel tempo. Il rinforzo positivo è efficace, ma in minor modo, e l'automonitoraggio è utile nei primi stadi del processo di formazione di un'abitudine, soprattutto per aiutare la persona a definire i propri obiettivi e mantenere la motivazione [6].

## 2.2.2 Rottura di abitudini

Per quanto invece riguarda la rottura delle abitudini, esistono vari approcci. Nel contesto dell'*Habit Alteration Model*, per rompere un abitudine è necessario agire in qualche punto del sistema in modo che non si arrivi al risultante comportamento abitudinario non voluto.

Una soluzione comune è il popolamento del *Potential Response Stack* con varie intenzioni per competere con impulsi non voluti, in modo che più risultati superino la soglia e venga attivato il sistema di Tipo 2. Tuttavia, come detto nella sezione 2.1.2, non sempre le "risorse" del sistema di Tipo 2 sono disponibili e, nel caso in cui questo non possa venire attivato, la risposta più probabile sarà quella con il maggior valore, la più consolidata nel sistema mnestico [8].

Altri metodi si basano sull'addestramento dell'autocontrollo, che si ritiene essere parte del processo di Tipo 2 e quindi soffre della stessa dipendenza dal carico cognitivo della persona [8].

In ugual modo si può intervenire nella fase di *Filter*, rimuovendo gli stimoli che portano al comportamento indesiderato, per quanto eliminarli completamente non sia banale, o riducendone la rilevanza tramite altre "distrazioni" in modo che non passino dai filtri iniziali [8].

### 2.2.3 Metodi comuni delle app di Digital Wellbeing

Per quanto agire sul processo di formazione/rimozione delle abitudini sia un metodo molto promettente nel campo del Digital Wellbeing [4, 5, 6], non tutti gli interventi proposti da queste applicazioni, automonitoraggio in primis, hanno una chiara collocazione in questo contesto [6].

Facendo riferimento a uno studio di Lyngs del 2019 [4], da un'analisi delle applicazioni di Digital Wellbeing presenti su vari Store risulta l'utilizzo di tecniche comuni in queste app, come anticipato nel capitolo 1. Le più diffuse, in ordine di frequenza, sono:

- una qualche forma di **blocco o rimozione** delle distrazioni, che impedisce all'utente di utilizzare determinate applicazioni o il telefono intero. Sono presenti vari metodi per fare ciò: imposizione di un blocco vero e proprio o di un timer, limitazione del numero di possibili aperture di un'applicazione, aggiunta di un delay durante il lancio di una funzionalità;
- **automonitoraggio** e visualizzazione di informazioni sull'uso del cellulare, distinti tra dati relativi all'intero device o alle specifiche applicazioni [5];
- interventi collegati all'**avanzamento di obiettivi**, principalmente costituiti da *reminders*, nella forma di pop-up, che ricordino un obiettivo sul tempo massimo da spendere su una applicazione o anche qualcosa di più generico, impostato dall'utente;
- **premi e punizioni**, elargiti all'utente come conseguenza dell'uso che è stato fatto del device, utilizzando caratteristiche della *gamification*, classifiche e condivisione sociale o usando premi e punizioni reali (e.g. perdere dei soldi).

Analizzandole in chiave dell'*Habit Alteration Model*, automonitoraggio e *reminders* vanno principalmente a collocarsi nel Sistema di Tipo 2, mentre blocchi e rimozioni hanno come obiettivo l'arginare le abitudini, facenti parte del Sistema di Tipo 1. Infine, i premi influiscono sul feedback dell'intero sistema [4].

L'implementazione di blocchi e rimozioni è di gran lunga la tecnica più diffusa (presente nel 74% delle applicazioni [4]) ma ha numerose limitazioni all'interno di questo modello. Per quanto i blocchi impediscano effettivamente il compiersi di comportamenti "errati", interrompendo il ripetersi di abitudini dannose, lo fanno impedendo materialmente l'utente con un ostacolo esterno. Un blocco non ha nessun influsso all'interno del sistema dell'*Habit Alteration Model*: l'utente, ricevendo lo stimolo esterno, è comunque spinto a compiere il comportamento "incriminato" e solo a quel punto il blocco gli impedisce di farlo. Non si va in alcun modo a intervenire nel sistema per sostituire l'impulso come output e quindi provocare una modifica del comportamento e il cambiamento dell'abitudine.

Nel momento in cui il blocco o l'applicazione di Digital Wellbeing venissero rimossi, l'utente, in assenza dell'ostacolo, tornerebbe al suo comportamento precedente. Pertanto, la grossa critica a blocchi e rimozioni è che il mantenimento di un comportamento dipende dal blocco di per sé, l'utente non fa alcuno sforzo cognitivo per migliorarsi (se non avere l'autodisciplina di non eliminare il blocco) e diventa esso stesso dipendente dall'applicazione di Digital Wellbeing.





## Capitolo 3

# Progettazione

Dall'analisi delle varie teorie psicologiche esposte nel Capitolo precedente e dalle limitazioni degli approcci tradizionali al Digital Wellbeing illustrati nella Sezione 2.2 si è ricavata l'idea per un'applicazione: StepByStep.

I principi fondamentali dell'applicazione si basano sulla teoria dell'*Habit Alteration Model*, esposta nella sezione 2.1.3. In particolare, l'applicazione cerca di frapporti tra i comportamenti automatici risultanti dal Sistema di Tipo 1, che portano all'uso scorretto dello smartphone, e stimolare, invece, l'attivazione del Sistema di Tipo 2. L'obiettivo è aiutare l'utente a sviluppare una maggiore capacità di autocontrollo nell'uso dello smartphone e, nel mentre, combattere le abitudini "errate". Ciò viene fatto tramite un vero e proprio programma di apprendimento, che punta ad insegnare un comportamento corretto all'utente e a fornire supporto nel mantenerlo, con aiuti che, tuttavia, diventano progressivamente più rari, in modo che il comportamento si consolidi indipendentemente da StepByStep e non sia dipendente da essa.

L'*intervention* è l'entità fondamentale che implementa la parte di sviluppo del comportamento corretto. L'*intervention* viene avviato manualmente prima di utilizzare il telefono in determinati momenti e aiuta l'utente nell'attivazione del Sistema di Tipo 2. L'effettivo ruolo di questi *intervention* dipende dal tipo di comportamento che si vuole contrastare. Si distinguono due categorie:

- il costante uso di un'applicazione per sessioni anche brevi ma troppo frequenti, che interrompono lo svolgersi di altre attività, definite *checking habits*;
- l'uso eccessivamente prolungato di un'applicazione.

Due tipi di *interventions* sono dedicati a queste categorie.

Nell'*intervention* 1, si vogliono contrastare le *checking habits*. L'*intervention* andrà attivato prima dell'inizio di un'attività che impegni le risorse attentive (e.g.

studio, lavoro) e richiederà all'utente di bloccare il cellulare e posizionarci sopra un oggetto "inusuale". Ogni volta che l'utente avrà l'impulso di utilizzare il telefono, questo oggetto dovrebbe aiutarlo a fermarsi, attivando il Sistema di Tipo 2.

Nell'*intervention 2*, si vuole contrastare l'uso prolungato di un'applicazione. L'*intervention* dovrà essere attivato ogni volta che si vuole utilizzare l'applicazione e cercherà di insegnare all'utente due pratiche: decidere a priori quanto tempo passare su questa applicazione e controllare frequentemente l'ora, in modo da rispettare questo orario.

Ognuno di questi *intervention* farà parte di un *Path*, l'entità che costituisce il percorso progressivo con la graduale rimozione degli aiuti. Esattamente come gli *interventions*, anche i *Paths* sono divisi in tipo 1 e tipo 2, che coincidono con il tipo del proprio *intervention*. Ogni *Path*, quindi, contiene un *intervention* di cui richiede l'attivazione in determinati momenti.

Ogni *Path* partirà da un livello base e, dopo il passaggio di alcuni giorni e l'utilizzo degli *interventions*, avvierà un *Recap*, in cui i risultati di questi ultimi saranno considerati per un eventuale passaggio di livello. L'aumentare del livello comporterà un minor numero di aiuti forniti dall'applicazione durante l'esecuzione degli *interventions*. Superare il livello massimo porta a una conclusione del *Path*. Il livello potrà aumentare, diminuire e mantenersi uguale, secondo l'andamento dell'utente negli ultimi *interventions* eseguiti. Questo percorso, oltre a favorire lo sviluppo autonomo e indipendente del comportamento, tramite la rimozione degli aiuti, supporta l'utente nel mantenere la propria motivazione tramite feedback frequenti e obiettivi ben definiti (riferimento a Sezione 2.1.1).

Oltre a questo, un altro ruolo di StepByStep è quello di suggerire proattivamente essa stessa le applicazioni sulle quali iniziare dei *Path*. Analizzando le statistiche di utilizzo dello smartphone verrà comunicato all'utente se un'applicazione sia stata usata eccessivamente. Tuttavia, non ci si limiterà ad una semplice esposizione dei dati, ma verrà prima chiesto all'utente di stimare quale sia questa applicazione e quanto sia stata usata. Ciò ha il fine di aumentare la consapevolezza che l'utente ha sull'uso del proprio smartphone.

Una volta determinata l'applicazione su cui si vuole lavorare, durante il primo periodo l'utente dovrà solamente usare l'app in modo normale e stimare quotidianamente, similmente a prima, l'uso che ne ha fatto. Oltre a lavorare sulla consapevolezza, questo periodo di monitoraggio è necessario per registrare l'uso abituale di questa applicazione e, al suo termine, esporre un riepilogo all'utente in cui consigliare quale *intervention* possa essere più utile.

A Figura 3.1 è rappresentato una schema esemplificativo delle funzionalità spiegate in questa sezione.

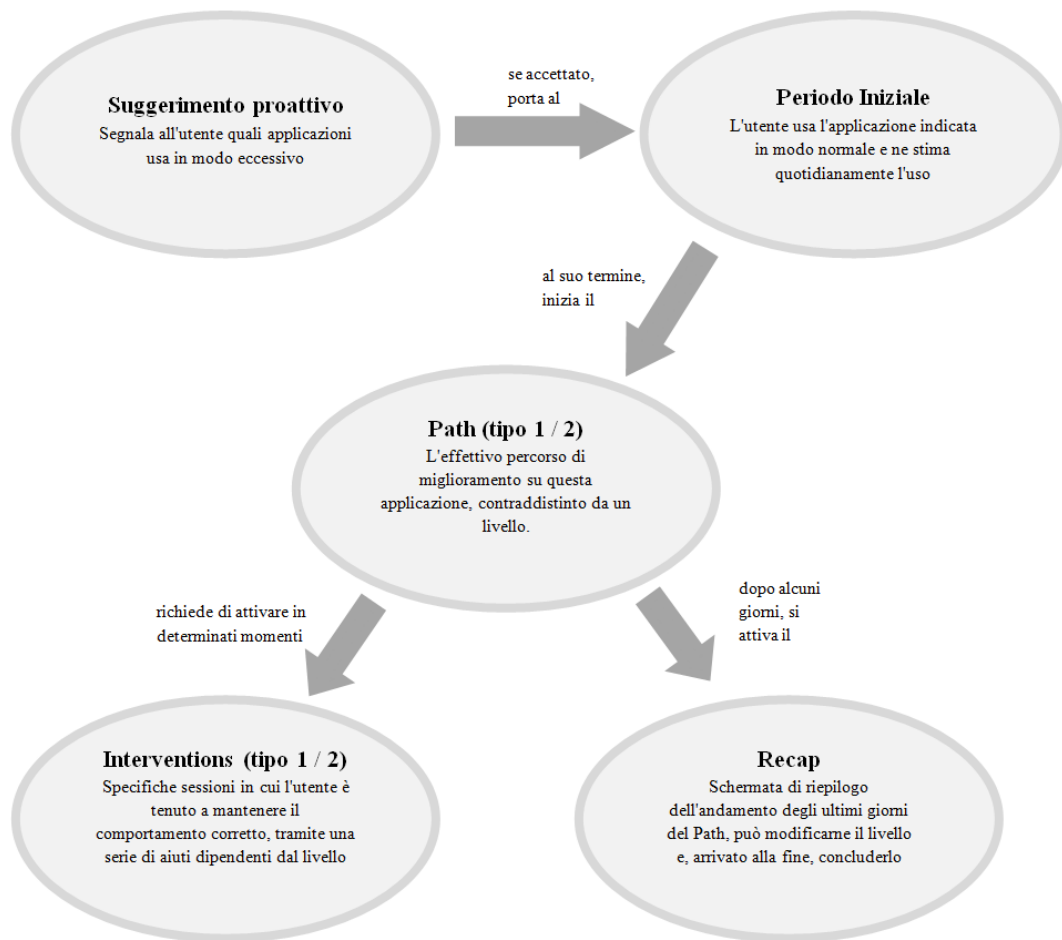


Figura 3.1: Schema riassuntivo delle funzionalità principali di StepByStep

### 3.1 Casi d'uso

Di seguito esponiamo alcuni casi d'uso dell'applicazione StepByStep.

#### Uso globale

Marco, dopo aver installato StepByStep, la apre e, osservando le statistiche, conferma il suo sospetto di usare eccessivamente Reddit e Youtube. Decide quindi di avviare un percorso su queste due applicazioni.

Durante la settimana seguente, alla fine di ogni giorno inserisce una stima dell'uso di queste applicazioni, accorgendosi che spesso sottovaluta il tempo trascorso su queste app.

Alla fine della prima settimana viene di nuovo colpito da quanto abbia sbagliato a

stimare il tempo passato su queste app. Decide di impegnarsi nel limitare l'uso giornaliero di entrambe le applicazioni. Quindi attiva tutti i possibili *Path*: un *Path* 1, che comprende entrambe le app ed è dedicato a limitare il numero di aperture, e due *Path* 2, uno per Reddit e uno per YouTube, dedicati a limitare l'utilizzo programmato della singola applicazione.

Durante la settimana seguente usa tutti gli *interventions*, seguendo le istruzioni dell'applicazione. Nel fine settimana, il *Recap* gli fa notare che è andato molto bene nel *Path* 1 e in quello di uso programmato Youtube, che quindi passano al livello 2, mentre la media di punteggio del *Path* 2 di Reddit è relativamente bassa e quindi rimane a livello 1.

Alla fine della quarta settimana, conclude il *Path* 2 su Youtube, essendo sempre riuscito a mantenere un punteggio alto e a salire di livello ogni settimana, mentre il *Path* 1 passa a livello 4 e il *Path* 2 su Reddit a livello 3.

Alla fine della quinta settimana, anche il *Path* 1 viene concluso, mentre, essendo andato molto male nell'uso di Reddit, il relativo *Path* torna a livello 2.

Finirà il *Path* 2 su Reddit solo alla fine dell'ottava settimana. Nel frattempo, si è reso conto di aver iniziato a usare Instagram in maniera eccessiva, quindi inizia a considerare se tenere installata StepByStep e avviare un altro percorso su questa.

### **Uso *intervention* 1**

Chiara ha davanti a sé un intenso pomeriggio di studio e non vuole venire interrotta in alcun modo dal suo cellulare. Apre StepByStep, accede al *Path* 1 creato in precedenza e avvia l'*intervention* 1 in modo da impedire a sé stessa di controllare continuamente il cellulare.

Seguendo le istruzioni, e trovandosi al livello 2, prende il pupazzetto che si è scelta come oggetto "inusuale" e lo posiziona di fianco al cellulare. Durante il pomeriggio, nella maggior parte dei casi, riesce ad accorgersi del pupazzo e a fermarsi prima di sbloccare il cellulare ma un paio di volte se ne accorge troppo tardi, solo dopo averlo sbloccato.

Alle 19, Chiara decide di smettere di studiare. Apre quindi l'applicazione e conclude l'*intervention*. Avendo sbloccato il cellulare solo due volte per tutta la sua durata, le viene dato un voto positivo.

### **Uso *intervention* 2**

Sono le 19:30 e Luca decide di poter dedicare un po' di tempo a sfogliare le storie di Instagram. Apre Instagram e subito un pop-up di StepByStep gli ricorda che dovrebbe avviare l'*intervention* 2 di uso programmato, avendo creato un *Path* 2 sull'applicazione Instagram. Infastidito dall'essersene dimenticato, Luca lo fa e decide di stare su Instagram 10 minuti.

Riprendendo a guardare le storie, viene "interrotto" ogni 2 minuti circa da una

leggera vibrazione, che, come sa, lo invita a guardare l'ora. Grazie a questo, dopo circa 10 minuti si accorge di quanto tempo è passato e chiude Instagram. Aprendo StepByStep per controllare com'è andato, Luca riceverà un buon punteggio, avendo come unica penalità l'essersi dimenticato di avviare l'*intervention* da solo.

## 3.2 Design

### 3.2.1 *Path* ed *intervention*

Come introdotto a inizio Capitolo, il fulcro di StepByStep sono *Path* e *intervention*, due entità fondamentali dell'applicazione e distinte fra di loro, per quanto interconnesse. L'*intervention* è una funzionalità specifica che richiede di essere attivata manualmente dall'utente in determinati momenti. Ogni *intervention* è legato a un *Path*, il percorso graduale che idealmente porta l'utente dal comportamento errato a quello corretto.

Ogni *Path* avrà un solo *intervention* di cui richiede l'attivazione in specifici momenti. Una volta avviato, l'*intervention* esporrà all'utente il comportamento giusto da tenere e, per la propria durata, lo aiuterà ad aderire allo stesso, tramite una serie di input. Allo stesso tempo, monitorerà il comportamento dell'utente e mostrerà, alla fine, l'aderenza che ha avuto rispetto al comportamento ideale presentato. Il risultato di ogni attivazione dell'*intervention* sarà salvato e mantenuto all'interno del *Path*.

Una volta passato un numero determinato di giorni ed eseguito un certo numero di *interventions*, il *Path* entrerà in una fase di riepilogo e modifica del livello. In base ai risultati avuti negli *interventions* svolti, l'utente potrà salire o scendere di livello, o anche mantenere lo stesso. Questa struttura dinamica permette all'app di adattarsi meglio alle necessità dell'utente. Con una quantità di aiuti inversamente proporzionale al livello, un utente ad un livello alto riceverà un numero limitato di suggerimenti durante l'uso degli *interventions*, permettendogli di sviluppare un certo grado di indipendenza da StepByStep, mentre un utente di livello basso riceverà molti aiuti, così da facilitare il processo iniziale di formazione dell'abitudine.

L'applicazione fornisce frequentemente feedback all'utente, sotto forma di "risultati" ottenuti alla fine dell'esecuzione di un *intervention* o di riepilogo durante il passaggio di livello. Questo, assieme alla definizione di obiettivi molto specifici durante gli *intervention* (e.g. non stare più di 5 minuti su Facebook) fa parte dei principi della *Goal Setting Theory*, definiti nella Sezione 2.1.2.

Come visto precedentemente, l'utente ha la possibilità di attivare due *Path* (e, di conseguenza, poter utilizzare due *interventions*) diversi fra loro: uno dedicato a limitare il tempo che viene speso su un'applicazione, l'altro a ridurre il numero di aperture e controlli che l'utente fa su un'applicazione o sull'intero cellulare in determinati periodi.

### ***Path ed intervention di Tipo 1***

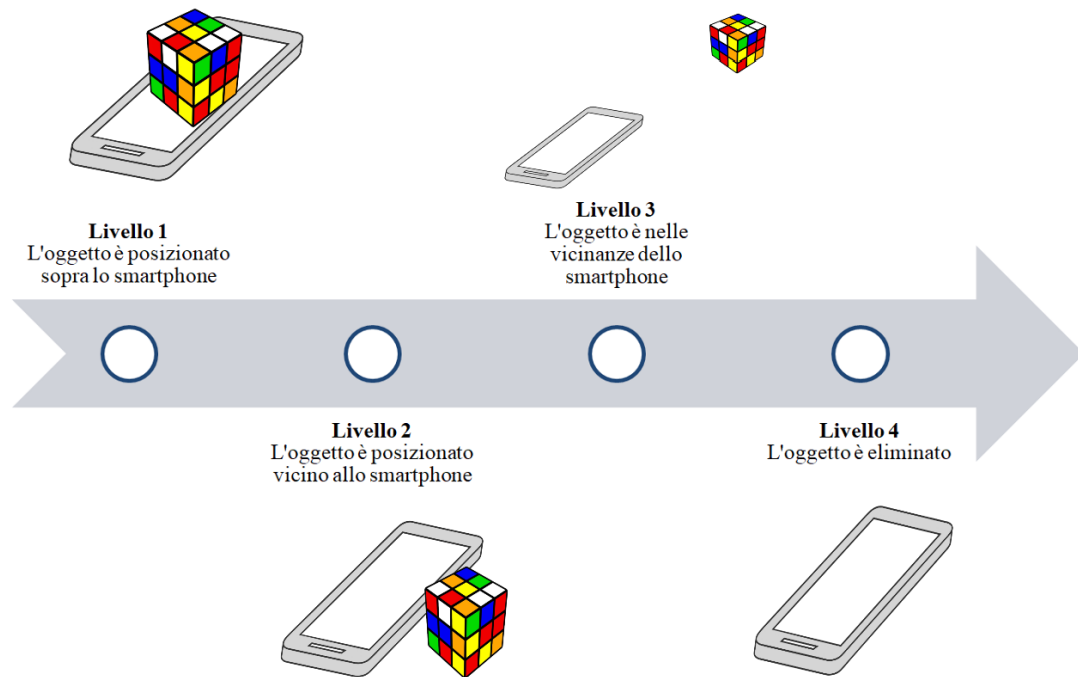
Il *Path 1* è dedicato a limitare il numero di aperture di applicazioni e, appunto, a contrastare le *checking habits* durante i momenti della giornata in cui l'utente si dedica ad altre attività (e.g. lavoro, studio) che richiedono concentrazione. Il relativo *intervention* dovrà essere attivato manualmente prima dell'inizio delle attività richiedenti concentrazione citate sopra e l'utente dovrà specificarne la durata, che quindi sarà variabile. L'utente, che sarà notificato allo scadere della durata inserita, avrà però la possibilità di estenderla in qualsiasi momento prima della sua estinzione. Esistono due versioni di questo *intervention*, distinte dalla portata delle funzionalità del cellulare che si vogliono bloccare.

Nella prima versione, l'obiettivo è impedire l'uso dell'intero cellulare. All'attivazione dell'*intervention* verrà chiesto all'utente di bloccare il cellulare e metterlo da parte, posizionandoci sopra (o nelle immediate vicinanze, dipendentemente dal livello) un oggetto "insolito" a sua scelta. Lo scopo di questo oggetto non è banale: il suo essere posizionato vicino al cellulare, alterando l'ambiente "normale", può prevenire l'insorgere dei *cues* che portano alla *checking habit* e, in ogni caso, la sua presenza all'interno del campo visivo può attivare il Sistema 2 nel caso in cui l'utente abbia la tentazione di prendere il telefono [8]. Viene quindi ostacolato il normale svolgimento dell'abitudine automatizzata nel Sistema 1. Come si può vedere dalla Figura 3.2, con l'aumentare di livello, si chiederà all'utente di posizionare l'oggetto progressivamente più lontano dal cellulare, fino ad arrivare alla sua rimozione nel livello 4.

Avendo come obiettivo quello di impedire l'uso dell'intero telefono, l'*intervention* considererà una penalità ogni volta che l'utente arriverà a sbloccare il cellulare durante la sua esecuzione. Avvenuto ciò, un altro avviso, nella forma di pop-up, ricorderà all'utente che un *intervention* è in corso ma, oltre a ciò, non si bloccherà il cellulare in alcun modo. StepByStep non applica restrizioni dure che obblighino l'utente a tenere un determinato comportamento e nel caso in cui questo voglia procedere nell'utilizzare lo smartphone nonostante l'oggetto "insolito" e il popup potrà farlo.

Per quanto la versione 1 abbia la sua utilità, è limitata nella sua considerazione dell'uso che un utente fa del cellulare, avendo come obiettivo il blocco completo. Poiché esistono casi in cui il cellulare viene utilizzato in maniera utile per coadiuvare lo svolgersi di altre attività importanti, come controllare la mail, bloccarlo completamente può essere controproducente. Per questo motivo esiste anche una versione 2 dell'*intervention*, in cui l'obiettivo sarà impedire l'accesso ad alcune app, mentre il cellulare in sé potrà essere usato liberamente.

All'attivazione di questo *intervention* l'utente sceglierà una serie di applicazioni da segnare per cui, nel caso in cui una di queste venga aperta durante l'esecuzione



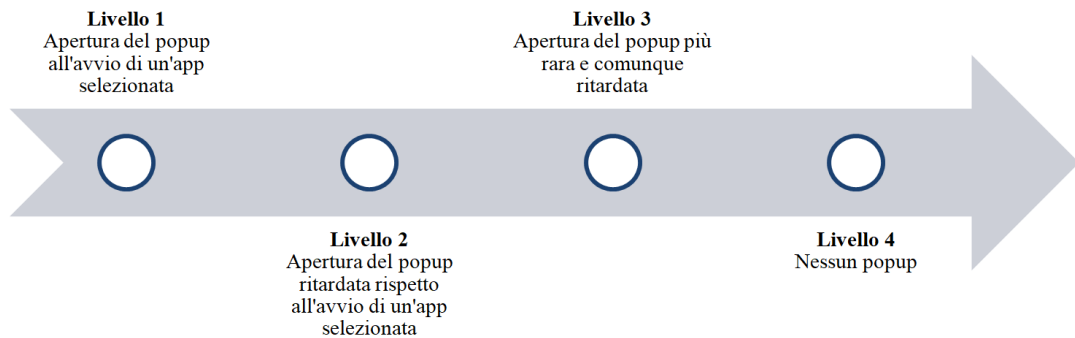
**Figura 3.2:** Configurazioni dell'*intervention* 1.1 divise per livello

dell'*intervention*, un pop-up ricorderà all'utente che non dovrebbe farlo. Le applicazioni preselezionate sono quelle su cui il *Path* si focalizza, ma potranno venire aggiunte e rimosse, a discrezione dell'utente, ad ogni avvio dell'*intervention*, per avere una maggiore flessibilità. Durante l'esecuzione, una penalità sarà considerata ad ogni apertura di un'applicazione segnata e, con l'avanzare dei livelli, i pop-up all'apertura di una di queste si faranno più rari e ritardati, fino alla loro completa assenza al livello 4, come mostrato in Figura 3.3.

Questa versione propone una soluzione più simile al blocco classico, con cui quindi condivide le limitazioni. Tuttavia il suo inserimento all'interno di un *Path* con vari obiettivi specifici, l'avanzamento dei livelli e il feedback continuo sono elementi che, per la *Goal-Setting Theory* (Sezione 2.1.2) e il *Transtheoretical Model* (Sezione 2.1.1) aiutano la modifica del comportamento, che di per sé contrasta lo svolgersi delle *checking habits*.

### ***Path* ed *intervention* di Tipo 2**

Il *Path 2* è dedicato ai casi in cui l'utente decide di passare del tempo su una determinata applicazione. L'obiettivo del *Path* è tenere sotto controllo la durata di



**Figura 3.3:** Configurazioni dell'*intervention* 1.2 divise per livello

queste sessioni e fare in modo che l'utente sia in grado di iniziarle e interromperle consapevolmente.

Il relativo *intervention* richiede quindi di essere attivato ogniqualvolta l'utente voglia usare l'applicazione indicata. Nel caso in cui l'utente se ne dimentichi, un pop-up gli ricorderà di farlo e una penalità minore sarà considerata. All'attivazione, l'*intervention* richiederà all'utente di decidere per quanto tempo intende stare sull'applicazione. Dopo aver confermato ciò, l'utente potrà procedere nell'utilizzare normalmente l'applicazione indicata ma gli verranno forniti una serie di *cues*, nella forma di leggere vibrazioni del cellulare, che hanno lo scopo di fargli controllare l'ora. Oltre a questi, una serie di pop-up verranno aperti in determinati momenti (e.g. tempo quasi scaduto, tempo limite superato di un tot... ). Con l'avanzare dei livelli, le vibrazioni, così come i pop-up, saranno progressivamente più rari.

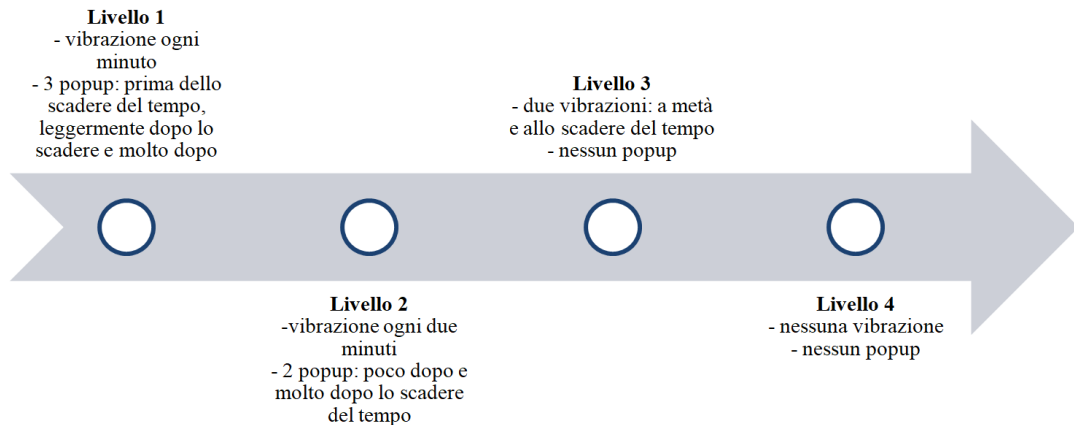
Dal punto di vista dell'*Habit Alteration Model*, si lavora in ambito di formazione di un'abitudine, con due obiettivi principali:

- all'apertura di un'applicazione, l'utente deve decidere consapevolmente quanto tempo passarci;
- l'utente deve essere in grado di rispettare il limite datosi.

Nel primo caso, porre all'utente la semplice domanda di scegliere quanto tempo passare sull'applicazione ha come effetto l'attivazione del Sistema di Tipo 2, mentre, di norma, l'apertura di un'applicazione può essere dovuta a un'azione automatizzata dell'utente. Per quanto riguarda il rispetto del vincolo temporale imposto, StepByStep si limita a segnalare il tempo trascorso, senza ovviamente forzare l'utente a chiudere l'applicazione. I *cues* che StepByStep fornisce, quindi, invitano semplicemente l'utente a controllare l'ora, sarà quest'ultimo a dover utilizzare questa conoscenza con l'obiettivo di chiudere l'applicazione.



La graduale rimozione dei *cues* con i livelli (mostrata a Figura 3.4) dovrebbe favorire il rafforzamento della nuova abitudine in modo indipendente da questi e, quindi, da StepByStep, fino a portare, idealmente, ad una totale autonomia dell'utente, che sarà in grado di mantenere questo comportamento nel tempo senza ulteriori aiuti esterni.



**Figura 3.4:** Configurazioni dell'*intervention* 2 divise per livello

Alla fine di ogni esecuzione dell'*intervention*, all'utente verranno mostrati i risultati, in cui si considererà se l'*intervention* è stato avviato in maniera autonoma e se le tempistiche scelte sono state rispettate e di quanto. Le stesse misure verranno considerate per il passaggio di livello.

I due *Path* hanno in comune fra di loro la struttura a livelli. Entrambi, indipendentemente dalle differenze nel loro obiettivo, condividono come fine il raggiungimento e mantenimento dell'indipendenza da parte dell'utente. Quindi l'ultimo livello di ciascun *Path* è contraddistinto da un'assenza totale di aiuti durante le relative esecuzioni degli *interventions*. In questo modo, l'utente, riuscendo a completare il livello finale senza alcun aiuto, avrà acquisito una nuova abitudine tramite le varie ripetizioni del comportamento corretto. Perciò il *Path* non sarà più necessario e il comportamento dovrebbe essere mantenuto anche in assenza di questo o dopo la disinstallazione di StepByStep.

### 3.2.2 Suggerimento proattivo

Un altro ruolo fondamentale di StepByStep è aiutare l'utente ad iniziare il proprio percorso di miglioramento, proponendogli attivamente su quali applicazioni agire e come. Come esposto nel *Transtheoretical Model* (Sezione 2.1.1), il processo di

cambiamento comportamentale inizia ben prima che si compia la decisione di modificare un comportamento. Facendo riferimento alle cinque fasi del modello, e in particolare a quella della Precontemplazione, si è deciso di implementare quell'intervento che viene indicato come *consciousness raising*. StepByStep, analizzando i dati di uso giornaliero dello smartphone, nell'eventualità che riscontri casi di utilizzo troppo elevato di un'applicazione, proporrà all'utente, tramite una notifica, di iniziare a lavorare su questa applicazione.

In questo modo si facilita, come indicato dal *Transtheoretical Model*, il passaggio alla fase di Contemplazione, in cui l'utente diventa effettivamente consapevole del suo comportamento scorretto. Tuttavia, questa notifica di "uso eccessivo" non si limita solo ad esporre i dati sull'uso delle applicazioni e proporre l'inizio di un percorso di miglioramento. L'utente viene infatti invitato a dialogare in maniera più interattiva con StepByStep, poiché gli viene chiesto di provare a stimare quale app abbia usato eccessivamente e per quanto, e solo dopo gli vengono esposti i dati reali a riguardo. Ciò ha numerosi effetti: dall'invitare l'utente a riflettere sull'uso abituale del cellulare, promuovendone un utilizzo più controllato e consapevole, all'evidenziargli esplicitamente la differenza fra il tempo percepito e quello reale. Le misure scelte per analizzare l'uso di un'applicazione ed eventualmente creare questa notifica sono due:

- numero di aperture, usata come metro di rappresentazione delle *checking habits*, sessioni di controllo brevi e frequenti che caratterizzano l'uso quotidiano dei nostri cellulari [16];
- tempo passato con un'applicazione in primo piano, che, naturalmente, è un indicatore di quanto tempo sia speso sulla data applicazione, senza considerare la lunghezza e frequenza delle sessioni.

Se i valori di queste misure registrati da StepByStep per una determinata applicazione superano, uno o entrambi, una certa soglia, l'utilizzo che è stato fatto di questa app viene considerato "scorretto" e si crea la notifica esposta sopra.

Idealmente, il superamento del limite per una di queste misure corrisponde ad un modello di comportamento preciso: nel caso del numero di aperture, le *checking habits*, nel caso del tempo speso in primo piano, sessioni eccessivamente lunghe che l'utente ha sull'applicazione. Nel primo caso, si dovrebbe limitare l'impulso che l'utente ha, in vari momenti della giornata, di aprire l'applicazione mentre, nel secondo, si cercherà di aiutare l'utente ad autocontrollarsi e ridurre il tempo speso sull'app.

Poiché questa coppia di misure è relativamente elementare e potrebbe non considerare alcuni casi particolari a cui l'utente è soggetto, si lascerà allo stesso la possibilità di scelta. Il consiglio sarà, ovviamente, quello di iniziare un percorso di miglioramento sull'applicazione ma l'utente sarà libero di ignorarlo o di bloccare

direttamente le future analisi dell'uso di tale applicazione (nel caso sia un'app che ha necessità di usare per motivi non futuri).

L'inizio di un percorso non è comunque subordinato all'arrivo di una notifica e l'utente potrà avviarlo manualmente in qualsiasi momento scegliendo liberamente l'applicazione su cui lavorare

### 3.2.3 Prima Settimana

Una volta avviato il percorso, la prima settimana sarà costituita da un semplice periodo di monitoraggio. In riferimento al *Transtheoretical Model*, nella fase di Contemplazione la persona è consapevole del suo comportamento scorretto ma non ha ancora attuato un cambiamento comportamentale. Durante la prima settimana, quindi, l'utente è tenuto a mantenere lo stesso uso abituale che ha dell'applicazione. Tuttavia gli sarà richiesto di accedere una volta al giorno a StepByStep per interagire in maniera simile a quanto visto nella notifica iniziale: provare a stimare l'uso giornaliero dell'applicazione in esame. In particolare, gli si chiederà di inserire delle stime riguardanti le due misure viste sopra, il numero di accessi e il tempo speso, e in seguito gli verranno mostrati i dati reali.

La presenza di questo periodo iniziale di monitoraggio consente a StepByStep di raccogliere e studiare più approfonditamente i dati giornalieri sull'uso dell'applicazione per tutta la prima settimana. In tal modo si può costruire un modello dell'utilizzo abituale dello smartphone e quindi consigliare correttamente all'utente quale *Path* iniziare per correggere il proprio comportamento. Dal punto di vista dell'utente invece, si è già discusso nella Sezione 3.2.2 l'utilità dell'eseguire una stima del proprio uso di un'app. In particolare, ripetendo questo schema per tutto il periodo iniziale, si evidenzia all'utente la differenza tra il comportamento percepito e quello reale. Proprio tramite questo confronto e l'esposizione quotidiana dei dati reali, l'utente sarà portato a formare autonomamente degli obiettivi [6] per l'uso dell'applicazione e verrà introdotto alla fase di Preparazione.

Al termine del periodo di monitoraggio, eseguito il numero di stime necessario, verrà mostrato all'utente un riepilogo dell'ultima settimana. Verranno mostrate le misure totali e, in base ai valori medi giornalieri, StepByStep proporrà determinate soluzioni. Si rientra in piena fase di Preparazione del *Transtheoretical Model*. Come introdotto nella Sezione 3.2.2, il comportamento dell'utente viene mappato, basandosi sulle misure raccolte, a due modelli: *checking habits* o uso prolungato eccessivo. In base a ciò, si proporranno i due diversi *Path*, che hanno come obiettivo la limitazione del numero di accessi che l'utente fa in momenti critici della giornata (e.g. studio, lavoro) l'uno e la riduzione della durata delle sessioni di uso di un'applicazione l'altro.

Le misure prese in considerazione saranno la media giornaliera delle aperture di un'applicazione e del tempo passato su di essa. Nel caso queste superino una

determinata soglia, verrà consigliato di attivare sull'applicazione il *Path* dedicato. Ovviamente, nel caso in cui entrambe le soglie siano superate, entrambi i *Path* verranno proposti. In ugual modo, nel caso in cui il comportamento della settimana sia rientrato nei limiti e quindi l'eventuale notifica iniziale possa essere stata dovuta ad una situazione particolare, lo si comunicherà all'utente.

L'esposizione di questi dati e i consigli su come proseguire avvengono tramite un dialogo interattivo con l'utente, in cui esso potrà richiedere spiegazioni aggiuntive e, infine, scegliere autonomamente come proseguire. In ogni caso, infatti, l'utente sarà libero di interrompere il percorso o di scegliere quale soluzione attuare, indipendentemente dai consigli dati da StepByStep.

### 3.2.4 Architettura

Il sistema ha una struttura relativamente semplice, con due componenti principali:

- un'applicazione mobile, dove si concentrano le funzionalità descritte fino ad ora
- un database cloud, dove vengono salvati i dati degli utenti

Per quelle che sono le funzionalità principali, l'unico componente realmente necessario è l'applicazione mobile. Tutte le operazioni descritte possono infatti essere svolte completamente in locale, senza necessità di comunicare con alcun server. La presenza della componente cloud è necessaria solo per i fini di questo studio, per avere un database in cui i dati degli utenti possano venire salvati e acceduti liberamente durante la ricerca.

## 3.3 Scheletro delle schermate

Per quanto riguarda la struttura dell'applicazione mobile, la Figura 3.5 mostra l'organizzazione delle sue schermate.

StepByStep è costituita da tre schermate iniziali, tutte accessibili direttamente all'apertura:

- elenco dei *Path* attivati
- elenco delle notifiche dell'applicazione
- monitoraggio del cellulare

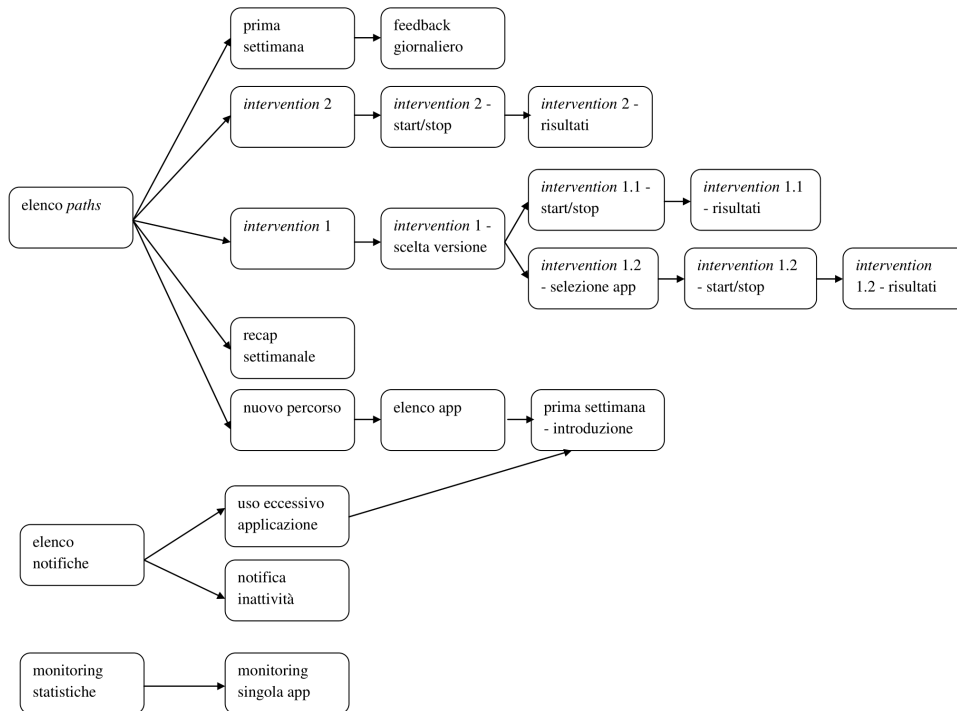


Figura 3.5: Scheletro delle schermate dell'applicazione

### 3.3.1 Monitoraggio

Nella schermata riguardante il monitoraggio vengono rappresentate le informazioni riguardanti l'uso del telefono.

Nonostante il monitoraggio non sia strettamente correlato al fulcro dell'applicazione, costituito da *Path* e *intervention*, si è ampiamente discusso nel Capitolo 2 dei benefici che questo può avere sui cambiamenti comportamentali, nonché formazione e rottura di abitudini. Inoltre, l'esclusione di questa funzionalità avrebbe eliminato uno strumento già ampiamente usato e verificato da altre applicazioni di Digital Wellbeing.

Nel caso di StepByStep, in particolare, la presenza di una schermata che mostrasse esplicitamente informazioni sull'uso del telefono costituisce un ottimo strumento per permettere all'utente di controllare come utilizza il cellulare e quindi, nel caso eventualmente notasse un'applicazione usata in modo scorretto, di iniziare un percorso su quest'ultima, senza dover attendere l'arrivo della notifica di uso eccessivo. Inoltre va anche considerata la possibilità che l'utente utilizzi il monitoraggio per avere informazioni aggiuntive sulle applicazioni su cui sta lavorando durante il

periodo iniziale di ogni percorso, o, più in generale, in qualsiasi momento possa ritenerlo opportuno.

Si è quindi deciso di implementare due schermate che mostrassero informazioni di monitoraggio del cellulare. Nella schermata principale sarà possibile vedere statistiche riguardanti l'uso dell'intero device con una suddivisione generica per applicazioni. Selezionando poi la singola applicazione, si aprirà un'altra schermata, con informazioni più specifiche riguardo l'uso della singola app selezionata.

### 3.3.2 Notifiche

Nella schermata delle notifiche sono elencate tutte le notifiche ricevute interne all'applicazione, da distinguere dalle notifiche di sistema del device dove StepByStep è installata. Queste notifiche si dividono in due tipi:

- **Notifica di uso eccessivo di un applicazione**

Come definito nella Sezione 3.2.2, con questa notifica viene implementato il suggerimento proattivo, dato autonomamente da StepByStep, di iniziare a lavorare su un'applicazione nel caso si sia registrato un uso scorretto della stessa nelle ultime 24 ore. Come viene mostrato nel mockup a Figura 3.6, la schermata richiede interattivamente all'utente di stimare quale applicazione ha usato in modo eccessivo e per quanto, in modo da incrementare la sua consapevolezza. A ogni risposta corrisponderà l'eventuale apertura di un pop-up per permettere all'utente di selezionare l'applicazione e il tempo.

Dopo l'esposizione dei dati reali, sarà permesso all'utente di iniziare a lavorare sull'applicazione presa in esame o anche di ignorare la notifica, nel caso in cui questo uso eccessivo sia dovuto a necessità della persona.

- **Notifica di inattività**

Come visto nella Sezione 3.2.1, l'*intervention* di tipo 1 richiede sempre un'attivazione manuale da parte dell'utente prima dell'inizio di un'altra attività. Poiché lo svolgersi di queste attività durante una giornata può essere variabile nel tempo e l'*intervention* non può percepire automaticamente il momento in cui dovrebbe venire attivato, a differenza di quello di tipo 2, il suo utilizzo è totalmente dipendente dalla memoria dell'utente. Per questo, si è deciso di considerare il caso in cui l'utente possa essersi dimenticato di utilizzare l'*intervention* e creare così la notifica di inattività. La notifica viene creata nel momento in cui non si sia attivato l'*intervention* di un *Path* di tipo 1 per un determinato periodo di tempo (tipicamente, nell'ordine di grandezza dei giorni).

Facendo fede sulla sincerità dell'utente, la notifica, come si vede dalla schermata rappresentata dal mockup in Figura 3.7, richiede di ammettere se ci sia effettivamente stata una dimenticanza da parte dell'utente (e in tal caso,

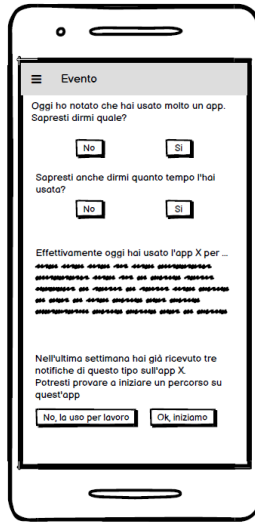


Figura 3.6: Mockup della schermata di notifica di uso eccessivo di un applicazione

una penalità sarà considerata durante il passaggio di livello) o se non sia mai stato necessario attivare l'*intervention*. Come da Figura 3.7, in seguito all'inserimento di una risposta apparirà un pop-up di conferma.

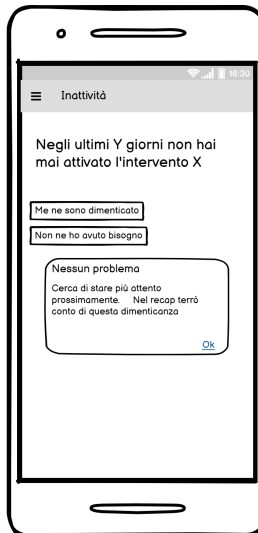


Figura 3.7: Mockup della schermata di notifica di inattività del *Path 1*

### 3.3.3 Paths

Infine, l'ultima delle schermate iniziali di StepByStep, anche *Home* dell'applicazione, è costituita dall'elenco di tutti i percorsi attivi iniziati dall'utente, come si può vedere a Figura 3.8. Verranno visualizzati qui tutti i *Path* e i periodi di monitoraggio, con relative informazioni. Selezionando uno di questi si verrà portati alla schermata di avvio del relativo *intervention*, nel caso del *Path*, alla schermata per inserire le stime, nel caso del monitoraggio iniziale, o alle schermate di riepilogo, nel caso queste siano disponibili.

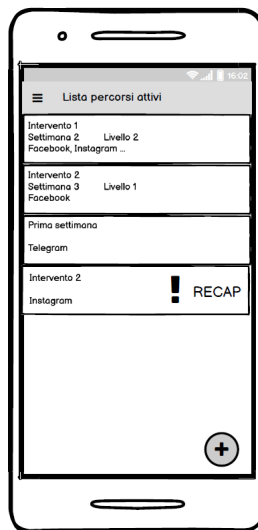
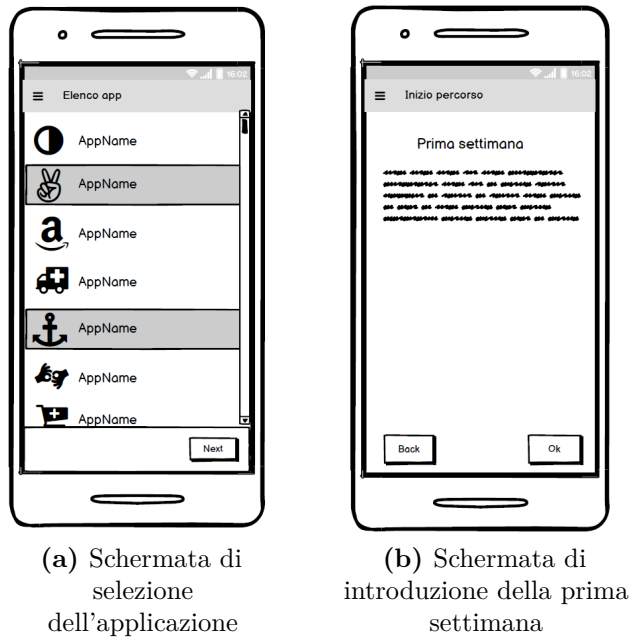


Figura 3.8: Mockup della schermata *Home*

Per quanto riguarda le stime del periodo di monitoraggio iniziale, verrà richiesto di eseguirle in un determinato momento della giornata, in modo da avere dati significativi e uniformi tra le varie misure. All'utente sarà ricordato di eseguire la stima tramite una notifica. La schermata dovrà ovviamente adattarsi in modo da mostrare informazioni coerenti nel caso l'utente vi acceda prima dell'ora indicata o dopo aver già eseguito la stima.

Dalla schermata *Home* sarà anche possibile selezionare manualmente un'applicazione sulla quale far partire un percorso di miglioramento. L'utente sceglierà su quale applicazione lavorare e, una volta selezionata, gli verrà presentata l'organizzazione del periodo iniziale di monitoraggio, in modo che sia informato su cosa debba fare. I mockup di entrambe le schermate sono presentati a Figura 3.9. Nel caso ci siano già dei *Path* attivi sull'applicazione, l'utente verrà informato prima della creazione del nuovo percorso. Inoltre, l'utente potrà cancellare un *Path* in qualunque momento, indipendentemente dallo stato di questo.



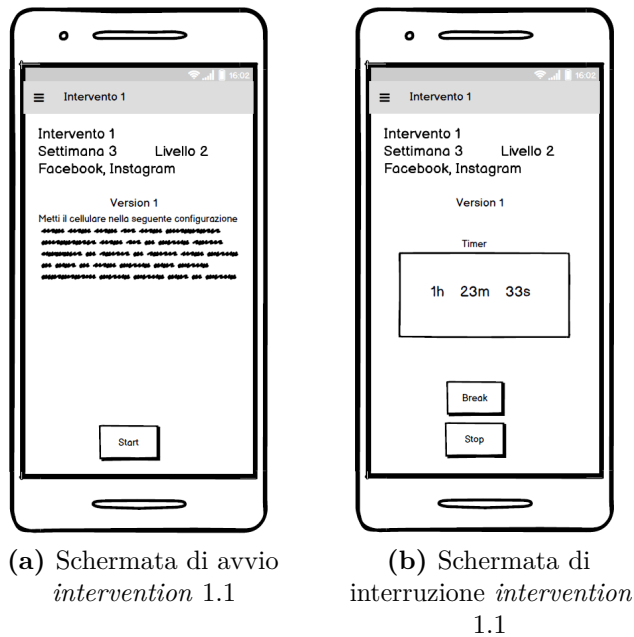


**Figura 3.9:** Mockup delle schermate di avvio di un *Path*

Poiché, come introdotto nella Sezione 3.2.1, il *Path* di tipo 1 funziona a livello globale, limitando l'uso dell'intero telefono o di una serie di app, e non è incentrato su una singola applicazione, a differenza del *Path* di tipo 2, ha poco senso avere più *Path* 1 attivi contemporaneamente. Ci potrà quindi essere solo *Path* di questo tipo e, quindi, nel caso in cui alla fine del periodo di monitoraggio di un applicazione ci sia un *Path* di tipo 1 già esistente, verrà proposto di incorporare l'applicazione in questo, aggiungendola all'elenco delle app. Per il *Path* di tipo 2, come spiegato sopra, questo problema non si pone.

### 3.3.4 *Interventions*

Nel caso in cui, nella schermata *Home*, venga selezionato un *Path*, l'utente verrà portato alla schermata di avvio del relativo *intervention*. Queste schermate saranno leggermente diverse fra di loro, in base al tipo di *intervention* e versione, ma condividono una struttura base costituita da: richiesta inserimento dati necessari (durata, app selezionate...) e pulsante di inizio. Ad *intervention* iniziato, il pulsante sarà sostituito da uno di fine e verranno mostrate informazioni riguardanti gli input inseriti e la durata. Nella Figura 3.10 sono rappresentati due mockup di esempio per queste due schermate, nel caso di *intervention* 1 Versione 1.



**Figura 3.10:** Mockup delle schermate di un *intervention* 1 Versione 1

Inoltre, per quanto riguarda l'*intervention* 1, si è deciso di implementare, come è osservabile sempre in Figura 3.10, la possibilità di fare una pausa. Viste le caratteristiche dell'*intervention*, da attivare durante un periodo di studio/lavoro, e l'elevata durata nel tempo che può avere, l'implementazione di una funzionalità simile era necessaria. L'utente selezionerà una durata per la pausa e, nel corso di questa, saranno disabilitati i controlli dell'*intervention*. Sarà possibile, durante queste pause, attivare anche degli *interventions* di tipo 2 per utilizzare determinate app.

### 3.3.5 Risultati e Riepilogo

Alla fine di ogni utilizzo degli *interventions* verranno mostrati all'utente i risultati ottenuti (Figura 3.11). La presenza di questo feedback immediato e preciso dovrebbe rafforzare la motivazione verso l'obiettivo (e.g. il passaggio di livello) dell'utente e, in ogni caso, fornire una comprensione immediata degli esiti dell'*intervention*.

Infine, dalla stessa schermata *Home* sarà possibile accedere, nel caso in cui le condizioni su durata e numero di *interventions* siano rispettate, alla schermata di riepilogo di livello, sempre cliccando sul relativo *Path*. In questa schermata, come mostrato nel mockup a Figura 3.12, saranno presentati, in modo conciso, i risultati dei vari *interventions* eseguiti e l'esito riguardante il passaggio di livello.



**Figura 3.11:** Mockup della schermata di esito di un *intervention*

In base alla media dei risultati, infatti, sarà possibile salire, scendere o rimanere nello stesso livello.



**Figura 3.12:** Mockup della schermata di passaggio di livello

Sia nel caso della schermata dei risultati a fine esecuzione dell'*intervention* che in quella di riepilogo verrà applicato un semplice rinforzo positivo, per cui i dati verranno presentati anche con una serie di “frasi di incoraggiamento” piuttosto che da un’esposizione dicotomica “dati → bene/male”.



# Capitolo 4

## Implementazione

L'applicazione StepByStep è stata sviluppata in ambiente Android, utilizzando quasi esclusivamente il linguaggio nativo Kotlin. Il codice dell'applicazione può essere liberamente visionato su <https://git.elite.polito.it/giuliopiacentini/thesis-app>.

### 4.1 Architettura

La struttura dell'applicazione, con le sue principali componenti, è mostrata a Figura 4.1.

Le schermate dell'applicazione in sé sono principalmente costituite da Fragments, che vengono raggruppati nelle sei Activities che vediamo in Figura 4.1 in base alle operazioni che compiono:

- **MainActivity** viene lanciata all'avvio dell'applicazione e si occupa di eseguire determinati controlli e conseguentemente ridirigere verso altre Activities;
- **TutorialActivity** contiene le schermate del tutorial dell'applicazione;
- **PermissionActivity**, insieme al **PermissionManager** e al proprio Fragment (**PermissionFragment**), gestisce la richiesta dei permessi necessari per il funzionamento di StepByStep;
- **LoginActivity** si occupa di gestire il login dell'utente;
- **HomeActivity** contiene un alto numero di Fragment e costituisce il vero e proprio nucleo dell'applicazione;
- **PopupWindow** costituisce tutte le schermate di avviso (popup) che vengono autonomamente aperte dall'applicazione in determinati momenti.

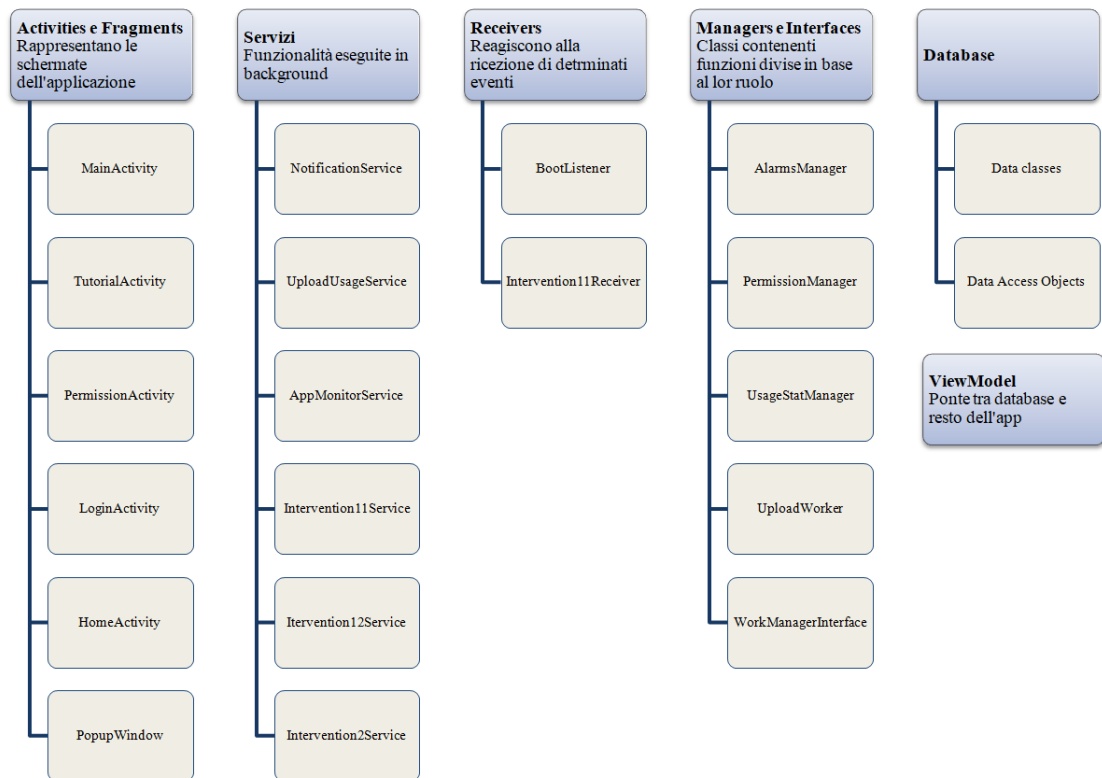


Figura 4.1: Principali componenti di StepByStep

Volendo esporre brevemente le funzionalità di alcuni degli altri componenti, abbiamo:

- **NotificationService**, Servizio che si occupa della creazione, in determinate situazioni, delle notifiche di sistema;
- **UploadUsageService**, che, assieme ad **UploadWorker** e **WorkManagerInterface**, gestisce il caricamento dei dati dell'applicazione sul cloud database;
- **AppMonitorService**, Servizio continuamente attivo che controlla l'avvio di altre applicazioni;
- **Intervention11Service**, **Intervention12Service**, **Intervention2Service** e **Intervention11Receiver** costituiscono, assieme ai relativi Fragments, i veri e propri *interventions*;
- **BootListener** viene lanciato all'avvio dello smartphone e avvia le funzionalità che devono sempre essere attive;

- **AlarmsManager** si occupa di schedulare l'avvio di determinate operazioni in certi momenti della giornata;
- **UsageStatManager** gestisce richiesta, ricezione ed elaborazione delle statistiche di uso dello smartphone.

Per quanto riguarda il database cloud, introdotto a Sezione 3.2.4, si è deciso di implementarlo utilizzando un database Firestore, facente parte dei servizi gratuiti Firebase<sup>1</sup>, di Google. Il database manterrà dati appartenenti a tre categorie: operazioni fatte sui *Paths*, risultati delle esecuzioni degli *interventions* e statistiche giornaliere dell'uso dello smartphone.

## 4.2 MVVM e dati

La struttura generale dell'applicazione è basata sul modello architetturale Model-View-ViewModel (MVVM). I tre elementi di questo modello sono:

- **Model** indica le componenti in cui vengono mantenuti i dati e che rendono possibile accedere ad essi. Nel caso di StepByStep, è rappresentato dal database Room e dai vari *Data Access Object* (DAO), i quali saranno discussi più approfonditamente in seguito.
- **View** indica la struttura, il layout e in generale tutto quello che l'utente vede sullo schermo, costituisce la *User Interface* dell'applicazione. In StepByStep, questo ruolo viene ricoperto dalle varie Activity e Fragment.
- **View-Model** è l'astrazione tramite la quale si popola la View con dati sensibili, costituisce un ponte tra la View e i dati, permettendo a questa di venire automaticamente aggiornata. In StepByStep, è l'HomeViewModel.

Questa suddivisione permette una netta separazione dei ruoli, in cui la gestione dei dati è lasciata unicamente al Model, la View si occupa soltanto della creazione della *User Interface* e la manipolazione dei dati, con il fine di popolare la UI, avviene tramite il View-Model.

In StepByStep, la persistenza dei dati viene garantita tramite l'uso della libreria Room<sup>2</sup>, che fornisce un livello di astrazione su SQLite, permettendo un facile accesso al database tramite la definizione di query SQL. In particolare, il database Room è composto da tre elementi:

---

<sup>1</sup><https://firebase.google.com/>

<sup>2</sup><https://developer.android.com/training/data-storage/room>

- il dato in sé, costituito da classe in cui le proprietà sono accompagnate da varie annotazioni che definiscono le caratteristiche, tra cui eventuali *foreign keys*, della corrispondente tabella all'interno del database;
- il DAO, interfaccia con cui si definiscono i metodi tramite i quali si può interagire con i dati (vengono definite qui le query);
- la classe database in sé, che fornisce le specifiche di configurazione del database vero e proprio ed è il punto di accesso primario a questo per il resto dell'applicazione.

### 4.2.1 Firebase

A differenza di Room, sul quale è utile avere solo una rappresentazione dei dati attuali, che quindi verranno via via sovrascritti ad ogni modifica, sul database Firestore viene mantenuto uno storico completo delle operazioni compiute dall'utente. Proprio per questa ragione i dati su Firestore saranno leggermente diversi da quelli sul database del cellulare: in particolare, avranno, come campi aggiuntivi, la data dell'operazione e una descrizione della stessa (e.g. `PATH_CREATION`, `INTERVENTION_DONE`, `LEVEL_UP`....)

La modifica della struttura dati dalla forma locale a una forma adatta al database Firestore viene eseguita dalla **WorkManagerInterface**. In seguito a ciò, i dati sono mandati all'**UploadWorker**, classe che invece si occupa del vero e proprio caricamento su Firestore. L'**UploadWorker** è un'estensione della classe `Worker`<sup>3</sup> di Android. L'utilizzo di questa classe permette di schedulare lavori in background (nel nostro caso, l'upload dei dati su un database esterno) in modo che vengano eseguiti in certe condizioni (e.g. quando il telefono è connesso ad una rete Wi-Fi) e ripetuti dopo un certo intervallo di tempo nel caso si presenti un errore di qualche tipo.

Nel momento in cui avviene un'operazione sul database Room, quindi, i relativi dati vengono mandati alla **WorkManagerInterface**, in modo che siano opportunamente modificati, e poi all'**UploadWorker**, per salvarli sul server. L'utilizzo di questa classe permette ai dati di venire salvati in modo sicuro sul server e al cellulare di eseguire queste operazioni pianificandole liberamente nel tempo.

---

<sup>3</sup><https://developer.android.com/reference/kotlin/androidx/work/Worker>



## 4.3 Navigazione e fragments

Lo spostamento tra i vari Fragments che costituiscono le schermate centrali dell'applicazione è stato gestito tramite il framework Navigation<sup>4</sup> di AndroidX. Un drawer menu, che viene inizializzato nella **HomeActivity**, permetterà all'utente di accedere alle tre destinazioni *root* del grafico di navigazione: *Paths*, Monitoraggio e Notifiche

### 4.3.1 Monitoraggio

La pagina di monitoraggio è suddivisa in: un grafico nella parte superiore, che indica l'uso totale che si è fatto del telefono e alcune delle app più utilizzate, e un elenco di tutte le app usate con informazioni più specifiche, ovvero tempo di utilizzo e numero di aperture.

Il grafico è stato implementato utilizzando il PieChart della libreria MpAndroidCharts<sup>5</sup> mentre l'elenco è costituito da una RecyclerView<sup>6</sup>. Premendo su un elemento dell'elenco, si apre un'altra schermata con informazioni più dettagliate sull'uso dell'app selezionata, tra cui un grafico che suddivide le statistiche nelle varie ore della giornata. Entrambe le schermate sono mostrate a Figura 4.2

Questi dati sono ottenuti dall'**UsageStatManager**, che a sua volta utilizza la classe UsageStatsManager<sup>7</sup> di Android, la quale fornisce accesso alle statistiche di utilizzo del cellulare, aggregate in intervalli temporali, che il sistema Android salva internamente di default.

### 4.3.2 Notifiche

La schermata dell'applicazione dedicata alle notifiche è anch'essa costituita da una RecyclerView con l'elenco delle “notifiche” interne create da StepByStep. Queste in realtà sono uno specifico tipo di dato presente sul database Room; nel momento in cui vengono aperte, il dato è aggiornato a “letto” e la notifica non viene più mostrata nell'elenco. I due tipi di notifiche, inattività del *Path* 1 e uso anomalo di un'app, sono rappresentate dalle proprie classi, mentre una classe Notification copre il ruolo di Wrapper per queste e costituisce il dato mostrato effettivamente dalla RecyclerView.

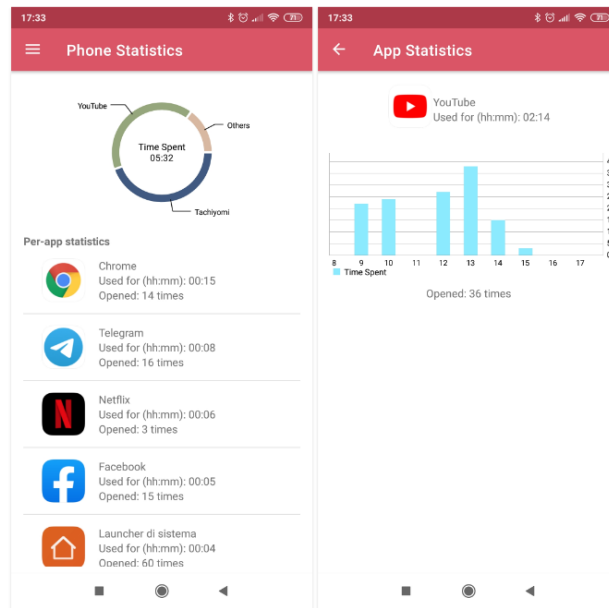
---

<sup>4</sup><https://developer.android.com/jetpack/androidx/releases/navigation>

<sup>5</sup><https://github.com/PhilJay/MPAndroidChart>

<sup>6</sup><https://developer.android.com/jetpack/androidx/releases/recyclerview>

<sup>7</sup><https://developer.android.com/reference/android/app/usage/UsageStatsManager>



**Figura 4.2:** Schermate di monitoraggio di StepByStep

Alla selezione di una notifica di uso scorretto, la schermata che si apre invita l'utente a stimare, se ne è in grado, quale app sia stata usata in modo eccessivo e per quanto tempo. Dopodiché sono esposte le statistiche reali e vengono date all'utente tre scelte: iniziare un percorso su questa app, escludere l'app da quelle considerate nella creazione di notifiche (nel caso sia un'app che deve usare molto per motivi di lavoro, per esempio) oppure non fare nulla.

Le notifiche vengono create in un determinato momento della giornata dal **NotificationService**. Per quanto riguarda la creazione della notifica di uso eccessivo, si fa nuovamente riferimento ai dati che si ricevono dall'**UsageStatManager**, guardando in particolare l'uso del cellulare nelle ultime 24 ore. Nel caso in cui un'applicazione superi l'ora e mezza di utilizzo o le 50 aperture viene creata una notifica su di essa. Invece, la notifica di inattività è costruita se risulta, da database, la presenza di un *Path* di tipo 1 di cui non si sia attivato l'*intervention* da più di 24 ore.

Questo Servizio, oltre a creare i dati che StepByStep usa per rappresentare le notifiche, crea, nel caso in cui si siano costruiti questi dati, delle vere e proprie notifiche del sistema Android, per avvisare l'utente di questi avvenimenti, tramite l'uso della classe `NotificationCompat`<sup>8</sup>. Un altro ruolo ricoperto dal Servizio è la

<sup>8</sup><https://developer.android.com/reference/androidx/core/app/NotificationCompat>

creazione, nel caso sia in corso un periodo iniziale di monitoraggio, di un'altra notifica di sistema che avvisi l'utente di effettuare le stime giornaliere.

Il lancio del **NotificationService** e, quindi, la creazione dei dati di notifica di StepByStep e delle notifiche vere e proprie di Android avviene ogni giorno alle ore 21. Similmente, l'**UploadService** è un'altra classe che viene avviata in uno specifico istante, le 23:55, e ha la funzione di caricare sul database Firestore, tramite **UploadWorker** e **WorkManagerInterface**, le informazioni sull'uso del cellulare nelle ultime 24 ore, che ovviamente ottiene tramite **UsageStatManager**. L'esecuzione di questi Servizi in specifici momenti è possibile tramite l'uso di **AlarmManager**<sup>9</sup>, classe di Android che dà accesso ai servizi di allarme del sistema e permette di programmare l'esecuzione di determinate task in determinati momenti futuri, le funzionalità della quale sono gestite dall'**AlarmsManager**.

Il codice di **AlarmsManager** viene eseguito da **HomeActivity** all'apertura dell'applicazione, previo controllo che non si sia già impostato quell'Alarm (creato in una precedente apertura), in modo da garantire la corretta programmazione di **NotificationService** e **UploadService** agli orari indicati. Inoltre, i due Servizi, una volta eseguita la loro funzione primaria, reimpostano la propria esecuzione il giorno seguente alla stessa ora, sempre tramite **AlarmsManager**. Poiché questo codice deve, per garantire il corretto funzionamento dell'applicazione, venire sempre eseguito almeno una volta al giorno, è necessario che gli allarmi vengano creati anche all'accensione del telefono, in modo che i relativi Servizi siano eseguiti anche se StepByStep non viene mai aperta.

Ciò viene fatto dal **BootListener**, estensione della classe **BroadcastReceiver**<sup>10</sup>, la quale è una componente del sistema Android che permette di implementare un paradigma *publish-subscribe*. All'avvenire di determinati eventi, vengono creati dal sistema dei *broadcast messages*; impostando adeguatamente un Receiver, si potrà rilevare questi messaggi ed eseguire del codice in maniera asincrona. Nel nostro caso, il **BootListener** è in ascolto degli eventi **ACTION\_BOOT\_COMPLETED** o **android.intent.action.QUICKBOOT\_POWERON**, rispettivamente accensione e riavvio, e in seguito alla loro ricezione chiamerà le funzioni di schedulazione dell'**AlarmsManager**.

### 4.3.3 Paths

Anche la schermata *Home* è costituita da una **RecyclerView**, la quale mostra un elenco di *Paths*. Nel caso in cui per uno di questi sia disponibile un riepilogo (e.g. fine del periodo di monitoraggio, passaggio di livello...), ciò verrà rappresentato

---

<sup>9</sup><https://developer.android.com/reference/android/app/AlarmManager>

<sup>10</sup><https://developer.android.com/reference/kotlin/android/content/BroadcastReceiver>

all'interno dell'elenco. Selezionando uno degli elementi dell'elenco, si verrà mandati alle schermate di avvio dei relativi *interventions* o a quelle di riepilogo, che saranno discusse in maniera più approfondita nella Sezione 4.4.

La cancellazione di un *Path* è implementata tramite il classico swipe da destra a sinistra. Questo viene fatto tramite l'uso di una callback adeguata usando la classe `ItemTouchHelper`<sup>11</sup>. Nel caso si tenga premuto su di un *Path*, un `Toast` indicherà di utilizzare lo swipe nel caso si volesse cancellarlo.

In questa pagina è anche presente la possibilità di avviare manualmente un percorso. Premendo sul *Floating Action Button*, si viene mandati su una schermata in cui l'utente dovrà selezionare l'applicazione su cui ha intenzione di lavorare. In questa pagina una `RecyclerView` mostra tutte le applicazioni installate sul cellulare, con relativo nome e icona. Questi dati, e quello più fondamentale del package, vengono ottenuti usando il `PackageManager`<sup>12</sup> di Android. La funzione che abbiamo implementato per fare ciò esclude le app di sistema da questo elenco.

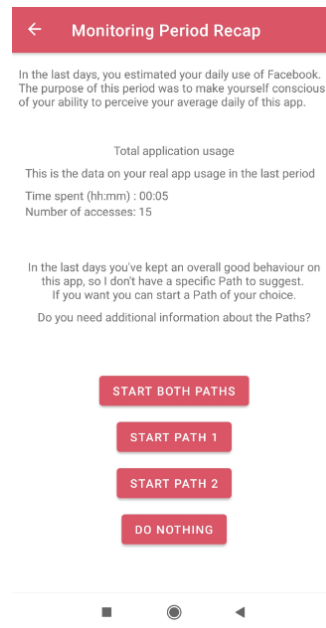
## 4.4 Interventions

Il periodo iniziale di monitoraggio avrà una durata di 7 giorni. Ogni giorno, alle 21, l'app richiederà all'utente, ricordandoglielo tramite una notifica di sistema creata dal `NotificationService`, di inserire una stima dell'uso quotidiano che si è fatto dell'app su cui il periodo è incentrato. Le misurazioni che `StepByStep` richiede di inserire sono due: il tempo speso sull'applicazione, approssimato in ore, e il numero di aperture. Al termine di ciò, verrà mostrato all'utente l'uso effettivo dell'applicazione, paragonandolo alla stima fatta. I dati reali saranno poi raggruppati e salvati sul database in modo da poter venire presentati all'utente una volta che abbia fatto 7 stime (ovvero, salvo dimenticanze, dopo 7 giorni). Nella pagina di riepilogo (Figura 4.3), a cui si potrà accedere al termine del periodo, verrà spiegato all'utente quale *Path* si ritiene più adatto, in base ai dati raccolti. Se, mediamente, il tempo passato quotidianamente sull'applicazione supera l'ora e mezza, sarà consigliato il *Path 2*, se l'app è stata aperta in media più di 50 volte al giorno, il *Path 1*. Sarà anche possibile che vengano consigliati entrambi o nessuno dei due. Verrà poi chiesto all'utente se necessita di avere informazioni aggiuntive relativamente a *Path* e *intervention*, e, nel caso di risposta affermativa, un `Dialog` esporrà brevemente le loro caratteristiche. La scelta su cosa iniziare e se iniziare sarà, in ultimo, lasciata all'utente.

---

<sup>11</sup><https://developer.android.com/reference/androidx/recyclerview/widget/ItemTouchHelper>

<sup>12</sup><https://developer.android.com/reference/android/content/pm/PackageManager>



**Figura 4.3:** Schermata di riepilogo alla fine del periodo di monitoraggio

Ogni *intervention* viene implementato tramite due componenti: un Fragment, tramite il quale si interagisce con l'utente, permettendogli di iniziarne e terminarne l'esecuzione, e un Servizio, che, una volta avviato, da background, controllerà il comportamento dell'utente e lo memorizzerà. Al termine dell'esecuzione dell'*intervention*, questi dati saranno mandati al Fragment che mostrerà i risultati all'utente.

Per via di alcune limitazioni che l'ambiente Android applica alle task eseguite in background è stato necessario, visto che le operazioni del Servizio sono osservabili dall'utente, utilizzare dei `ForegroundServices`<sup>13</sup>, ovvero dei Servizi che creano una notifica permanente, in modo che l'utente sappia che stanno venendo eseguiti. Per cui, all'avvio di tutti gli *interventions*, il Servizio dedicato, come prima cosa, creerà questa notifica. La notifica mostrerà un semplice "Intervention X running", rimarrà fino a che non si eliminerà il Servizio e, nel caso in cui nel caso venga cliccata, porterà alla schermata dell'*intervention*. La presenza di questa notifica rende meno probabile che l'utente si dimentichi di terminare un *intervention*.

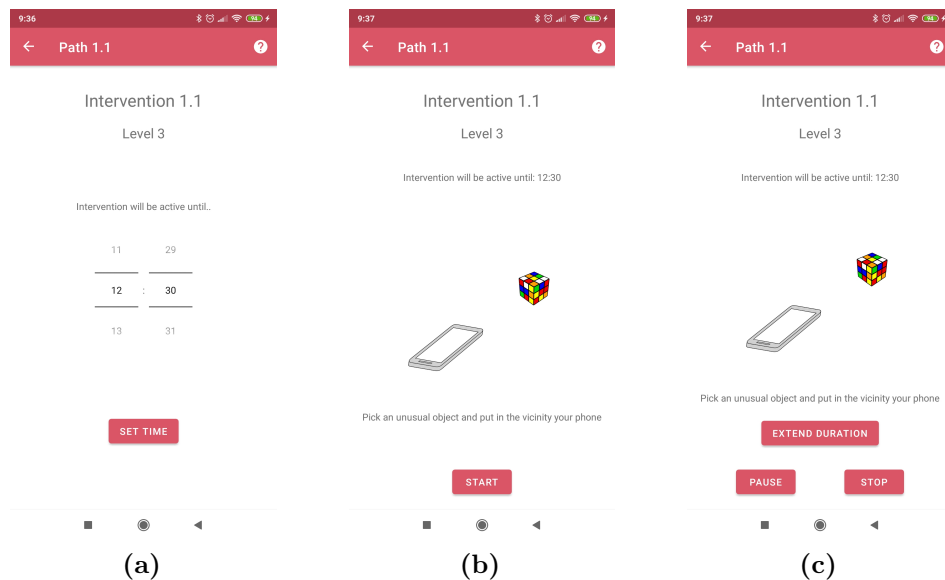
Inoltre, i Servizi degli *interventions* rientrano, per definizione, anche nel gruppo dei `BoundServices`<sup>14</sup>. Un `BoundService` è un Servizio che permette ad altre componenti di collegarsi e interagire con sé, implementando un modello client-server.

<sup>13</sup><https://developer.android.com/guide/components/foreground-services>

<sup>14</sup><https://developer.android.com/guide/components/bound-services>

Nel caso di StepByStep, l'utente ha effettivamente necessità di interagire con il Servizio (e.g. per interromperlo, metterlo in pausa...). Il necessario utilizzo di questi BoundService comporta un livello di complessità ulteriore all'interno del Fragment. Infatti, laddove normalmente il Servizio viene creato e collegato alla pressione del tasto "Start" (riferimento a Figure 4.4 e 4.5), nel momento in cui l'utente riapre il Fragment a *intervention* attivo, è necessario che, prima della creazione del layout della pagina, ci si colleghi al Servizio corretto. Quindi, ad ogni apertura del Fragment, sarà necessario controllare lo stato del Servizio (un dato riguardante ciò sarà presente sul database), collegarsi a questo nel caso sia già stato avviato e aggiornare la View di conseguenza.

#### 4.4.1 Intervention 1



**Figura 4.4:** Schermate per l'avvio dell'*intervention* 1 (versione 1)

Poiché l'*intervention* di tipo 1 è dotato di due versioni, la prima cosa che l'utente dovrà fare dopo averlo selezionato dalla pagina *Home* sarà scegliere la versione. Una semplice schermata con due pulsanti e una rapida descrizione della differenza tra le due gli consentirà di farlo.

La parte che le due versioni condividono è quella riguardante la durata. Entrambi i Fragment presentano un TimePicker<sup>15</sup> con cui l'utente potrà selezionare il momento

<sup>15</sup><https://developer.android.com/reference/kotlin/android/widget/TimePicker>

(ora e minuto) fino a cui far durare l'esecuzione dell'*intervention*. L'intervallo minimo è di 30 minuti, la scelta di un orario che non rispetti questa restrizione non è permessa. Nel caso in cui si selezioni un'ora precedente all'attuale, questa sarà considerata come appartenente al giorno successivo. Una volta avviato l'*intervention* tramite il pulsante "Start" (Figura 4.4), la schermata si aggiornerà, mostrando, indipendentemente dalla versione, i pulsanti per fermarlo, metterlo in pausa o estenderne la durata.

Ad ogni pressione di un pulsante corrisponde la chiamata di una funzione del Servizio collegato.

All'attivazione dell'*intervention*, il Servizio verrà creato, riceverà i dati riguardanti la durata e avvierà conseguentemente un timer tramite la classe `CountDownTimer`<sup>16</sup>, che permette di eseguire del codice dopo un tempo specificato. Una volta terminato il timer tutte le funzionalità di controllo del Servizio, spiegate nelle sezioni seguenti, saranno interrotte. Altri timer invece sono creati in modo da modificare la notifica del Servizio così che mostri "Intervention 1 about to expire!" a 10 minuti dalla fine del tempo e "Intervention 1 expired" a tempo scaduto.

La pressione del pulsante "Stop" (Figura 4.4) comporta la cancellazione dei timer, nel caso non fossero già terminati, e l'interruzione delle funzionalità di controllo, le quali possono anche venire interrotte momentaneamente tramite i pulsanti "Pause" e "Restart" (Figura 4.4). Alla pressione del pulsante "Pause", l'utente selezionerà tramite un Dialog quanto far durare la pausa e il Servizio inizierà un timer con adeguato valore, allo scadere del quale una notifica avviserà l'utente.

La pressione del pulsante di "Extend Duration" (Figura 4.4) comporta semplicemente una reinizializzazione dei timers in modo che il termine dell'intervento sia posticipato del numero di minuti che l'utente avrà scelto utilizzando lo specifico Dialog (è possibile selezionare intervalli di 10 minuti che vanno da un minimo di 10 a un massimo di 50).

## Versione 1

Nella versione 1 dell'*intervention*, dopo la selezione dell'orario, la schermata sarà modificata in modo da mostrare un'immagine che rappresenti la configurazione in cui disporre il cellulare assieme all'oggetto "insolito" (rappresentato da un cubo di Rubik), dipendentemente dal livello. Assieme alle immagini, una breve scritta descriverà questa configurazione.

Alla pressione del tasto "Start" (Figura 4.4) verrà creato il Servizio del relativo *intervention* e lo schermo del cellulare verrà bloccato utilizzando la funzione `locknow` del `DevicePolicyManager`<sup>17</sup> di Android.

---

<sup>16</sup><https://developer.android.com/reference/kotlin/android/os/CountDownTimer>

<sup>17</sup><https://developer.android.com/reference/android/app/admin/DevicePolicyManager>

Il Servizio, in questa versione, è relativamente semplice. Oltre alle funzioni in comune descritte nella Sezione precedente, il suo unico compito sarà istanziare, eliminare e mettere in pausa un BroadcastReceiver dedicato alla ricezione di due tipi di messaggio:

- ACTION\_USER\_PRESENT, mandato quando l'utente sblocca il telefono
- ACTION\_SCREEN\_OFF, mandato quando lo schermo viene bloccato

Nel primo caso, il Receiver incrementerà di uno il conteggio del numero di sblocchi del cellulare e avvierà un timer, con durata di un minuto, allo scadere del quale si aprirà una schermata della nostra applicazione, ricordando all'utente di avere l'*intervention* attivo. Nel secondo caso, il Receiver interromperà il timer.

Al concludersi dell'*intervention* l'applicazione indirizzerà l'utente alla schermata dove gli verrà fornita una valutazione (fare riferimento alla Tabella 4.1).

Numero di sblocchi	Valutazione
$< 3$	Excellent!
$3 \leq x \leq 9$	Not bad, but you can do better
$> 9$	C'mon, I know you can do better

**Tabella 4.1:** Valutazioni date sul numero di sblocchi del telefono al termine dell'*intervention* 1.1

## Versione 2

Nella versione 2, la schermata di avvio dell'*intervention*, oltre ad avere il TimePicker per la scelta della durata, presenta all'utente le applicazioni sulle quali l'*intervention* si focalizza. L'utente può modificare queste applicazioni tramite un apposito pulsante che lo porterà ad un elenco di tutte le app installate, tramite cui potrà selezionare ed escludere quelle che desidera, per poi confermare la scelta fatta.

Una volta avviato con la pressione del pulsante "Start" (Figura 4.4), il relativo Servizio dovrebbe, come descritto nella Sezione 3.2.1, aprire una schermata davanti all'utente quando questo avvia una delle app selezionate. Purtroppo, il lancio manuale di applicazioni (tramite il click sulla relativa icona) in un sistema Android avviene con l'utilizzo di *Explicit Intents*. Questi Intent, per le specifiche di sicurezza Android, sono esclusivi dell'app con il dato package e non possono venire utilizzati da altre applicazioni. È quindi impossibile creare un BroadcastReceiver che capti questi Intent e possa mostrare la schermata di StepByStep di conseguenza, quindi in modo asincrono. Per arrivare a implementare le funzionalità di controllo dell'*intervention* 1.2 è stato necessario utilizzare un meccanismo di *polling*.



Tutto ciò viene gestito dalla classe `AppListener`, interna al Servizio. Tramite l'uso di `TimerTask`<sup>18</sup> è possibile programmare l'esecuzione di un codice ogni secondo. Questo codice utilizzerà lo **UsageStatManager** per accedere agli eventi del telefono nell'ultimo secondo e controllarli. In particolare gli eventi interessanti sono:

- `ACTIVITY_RESUMED` o `MOVE_TO_FOREGROUND` (in base alla versione di Android), indicano l'apertura dell'applicazione di cui forniscono il package.
- `ACTIVITY_PAUSED` o `MOVE_TO_BACKGROUND` (in base alla versione di Android), indicano la chiusura dell'applicazione di cui forniscono il package.

Dopo aver controllato in ordine cronologico tutti i dati restituiti, nel caso si sia registrata la permanenza su un'app segnata, verrà aperta la schermata con cui si ricorda all'utente che l'*intervention* è attivo, dipendentemente dal livello:

1. apertura immediata della schermata;
2. apertura ritardata di 20 secondi;
3. apertura solo la metà delle volte e ritardata di 20 secondi;
4. nessuna apertura.

Al termine dell'*intervention* viene mostrata all'utente una valutazione, esposta a Tabella 4.2

Numero di aperture applicazioni	Valutazione
0	Excellent!
$1 \leq x \leq 3$	Nice . Now aim for the top
$4 \leq x \leq 7$	Not bad, but you can do better
$>7$	You still have some work to do

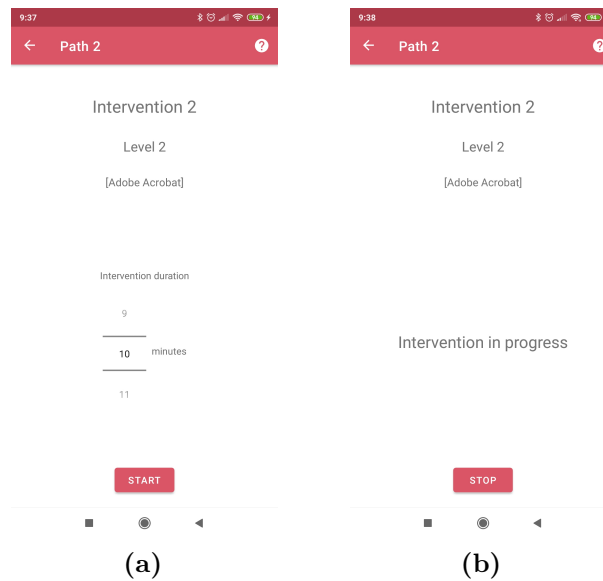
**Tabella 4.2:** Valutazioni date sul numero di aperture delle applicazioni al termine dell'*intervention* 1.2

#### 4.4.2 Intervention 2

L'*intervention* di tipo 2 è gestito da un Fragment, il cui layout presenta un `NumberPicker`<sup>19</sup> che permette all'utente di selezionare il numero di minuti da

<sup>18</sup><https://developer.android.com/reference/kotlin/java/util/TimerTask>

<sup>19</sup><https://developer.android.com/reference/kotlin/android/widget/NumberPicker>



**Figura 4.5:** Schermate per l'avvio dell'*intervention 2*

passare sull'applicazione a cui è dedicato l'*intervention*. Si è scelto di impostare una durata minima di 2 minuti e massima di 15.

Come da Sezione 3.2.1, l'*intervention* dovrebbe ricordare all'utente di avviarlo nel caso questo se ne dimentichi e apra l'applicazione. Questo si traduce in un'apertura automatica della schermata quando l'utente apre l'applicazione che l'*intervention* copre senza aver prima avviato quest'ultimo. Come visto per l'*intervention 1.2*, l'unico modo per rilevare l'apertura di applicazioni esterne a StepByStep è utilizzare una meccanica di *polling*.

Questa operazione viene gestita da **AppMonitorService**, che, in modo analogo a quanto esposto nella Sezione 4.4.1, utilizzerà una `TimerTask` per controllare gli eventi del cellulare ogni 5 secondi, aprendo la pagina del relativo *intervention 2* nel caso si registrasse l'apertura di una determinata applicazione (notare che, al più, la pagina sarà aperta 5 secondi dopo). Questa operazione deve essere fatta in modo continuo ed ininterrotto sul cellulare, con ovvie conseguenze sulla batteria del telefono. Poiché si vuole evitare che il Servizio sia sommariamente terminato dal sistema, dovrà essere un `ForegroundService` e quindi sul cellulare dell'utente dovrà essere costantemente presente una notifica fissa (semplicemente costituita da "StepByStep running").

L'applicazione memorizzerà sempre se l'apertura del `Fragment` sia avvenuta in maniera manuale o automatica, in modo che una penalità possa venire considerata nel secondo caso.

Una volta selezionata la durata e premuto il tasto “Start” (Figura 4.5), questo viene sostituito da uno “Stop” e viene creato il Servizio dell’*intervention*. L’implementazione dell’*intervention* tramite questo Servizio avviene basandosi completamente sulla classe `CountDownTimer`, introdotta nella Sezione 4.4.1. In particolare, verranno utilizzati due timer:

- *timer*, che sarà impostato per coprire la durata inserita dall’utente;
- *secondTimer*, avviato allo scadere del primo timer e con durata di 3 minuti.

Tramite questi timer è possibile programmare sia le vibrazioni periodiche del cellulare che l’apertura di una schermata popup al loro scadere. La frequenza della vibrazione e le operazioni che verranno effettuate allo scadere dei timer dipendono dal livello dell’*intervention*:

- **Livello 1**

- Primo timer: vibrazioni ad ogni minuto, ad un minuto dalla fine verrà aperta una schermata che lo ricordi all’utente.
- Secondo timer: vibrazioni ogni minuto, ad un minuto (quindi un minuto dopo lo scadere del tempo) una schermata con questa informazione verrà mostrata all’utente, allo scadere (quindi dopo tre minuti) si aprirà un’altra schermata.

- **Livello 2**

- Primo timer: vibrazioni ogni due minuti, nessuna schermata.
- Secondo timer: vibrazioni ogni minuto, schermata un minuto dopo l’avvio, schermata allo scadere.

- **Livello 3**

- Primo timer: due vibrazioni totali, una a metà della durata e una alla fine, nessuna schermata.
- Secondo timer: una sola vibrazione allo scadere del tempo, nessuna schermata.

- **Livello 4**

Nessuna vibrazione o schermata in entrambi i timer.

Allo “Stop” (Figura 4.5), il Servizio confronta lo stato dei timer, da cui determina quanto tempo è passato, con l’utilizzo effettivo dell’applicazione dall’inizio dell’*intervention*, ottenuto usando `UsageStatManager`. In base a questi e all’apertura della schermata, automatica o meno, vengono impostate tre penalità, indicate a Tabella 4.3.

Le due misure sono confrontate e vengono assegnate delle penalità all'utente se: la durata selezionata è stata superata (scadere del primo timer) e se è stata superata di più di 3 minuti (scadere del secondo timer), rispettivamente indicate come P2 e P3. Viene anche salvato il numero di secondi di eccesso rispetto al tempo inserito e considerato se l'utente abbia avviato l'*intervention* spontaneamente o se la schermata si sia dovuta aprire autonomamente davanti all'applicazione (indicata come P1).

P1	Schermata aperta in maniera automatica
P2	Durata selezionata superata di poco (< 3 minuti)
P3	Durata selezionata superata di molto (> 3 minuti)

**Tabella 4.3:** Penalità considerate dall'*intervention 2*

I risultati sono mandati alla schermata che espone all'utente lo stato relativo alle tre penalità, il numero di secondi per cui il limite temporale è stato superato, nel caso sia successo, e un esito generale dell'*intervention* basato sui risultati (riferimento a Tabella 4.4).

P1	P2	P3	Valutazione
NO	NO	NO	Excellent!
NO	< 60 sec	NO	Nice. Now aim for the top
SI	NO	NO	Nice. Now aim for the top
NO	> 60 sec	NO	Not bad, but you can do better
SI	< 60 sec	NO	Not bad, but you can do better
NO	SI	SI	You still have some work to do
SI	> 60 sec	NO	You still have some work to do
SI	SI	SI	C'mon, I know you can do better

**Tabella 4.4:** Valutazioni date in base alle penalità al termine dell'*intervention 2*

### 4.4.3 Passaggio di livello

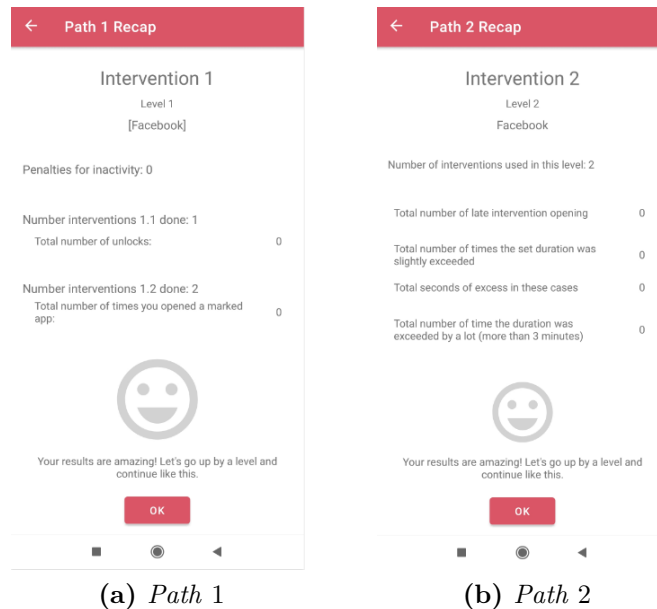
In entrambi i *Path*, dopo un determinato numero di giorni ed *interventions* eseguiti, viene mostrato all'utente un riepilogo dell'ultimo periodo, che conduce ad una modifica del livello. In StepByStep, il limite temporale è fissato a 4 giorni, mentre è necessario un minimo di 2 esecuzioni di *interventions* nel caso del *Path 1* e 4 per il *Path 2*. La possibilità di eseguire questa operazione su di un *Path* sarà resa nota tramite una modifica della relativa casella nella schermata di *Home*, la selezione della quale porterà ad una delle schermate visibili a Figura 4.6 e non alla schermata di avvio dell'*intervention*. Dopo l'apertura di un riepilogo, la data

del *Path* viene aggiornata e quindi sarà necessario far passare altri 4 giorni per innescare nuovamente il passaggio di livello.

Come si può vedere in Figura 4.6, entrambe le schermate mostrano all'utente i risultati raggruppati dei vari *intervention* eseguiti, risultati che poi saranno utilizzati per determinare la modifica del livello.

Per il *Path 1* si considerano: le penalità di inattività (introdotte nella sezione 4.3.2) e il numero di *interventions* attivati per Versione 1 e Versione 2, con, rispettivamente, il numero totale di sblocchi del cellulare e il numero totale di aperture di applicazioni “segnate”.

Per il *Path 2* vengono sommate le occorrenze delle varie penalità per tutti gli *interventions* eseguiti. Sarà considerato, per il passaggio di livello, la frequenza di queste penalità.



**Figura 4.6:** Schermate di riepilogo per il passaggio di livello di entrambi i *Path*

Le regole utilizzate per determinare la modifica di livello sono visibili nelle tabelle seguenti, in particolare, a Tabella 4.5 le regole del *Path 1* e a Tabella 4.6 quelle del *Path 2*. Come si può vedere, si è scelto di dare alla penalità di inizio automatico P1 (Tabella 4.3) un peso alquanto ridotto per quanto riguarda passaggio di livello. La penalità, infatti, non sarà presa in considerazione per il *level down* e sarà ammessa una presenza nel 90% dei casi per il *level up*.

Una volta arrivati al riepilogo del livello 4, nel caso lo si sia svolto in modo da innescare un avanzamento di livello, il *Path* sarà concluso. Tramite un Dialog

l'applicazione si complimenterà con l'utente e il *Path* verrà automaticamente rimosso.

Inattività (tot)	Sblocchi Int1.1 (avg)	Aperture Int 1.2 (avg)	Condizione	Risultato
> 4	> 8	> 8	OR	level down
≤ 1	≤ 3	≤ 3	AND	level up
else				same level

**Tabella 4.5:** Regole per il passaggio di livello nel *Path 1*

P1 (%)	P2 (%)	P3 (%)	Condizione	Risultato
none	> 50% (more than 1 second)	> 40%	OR	level down
≤ 90%	≤ 30% (less than 1 second)	≤ 20%	AND	level up
else				same level

**Tabella 4.6:** Regole per il passaggio di livello nel *Path 2*

## 4.5 Prima apertura

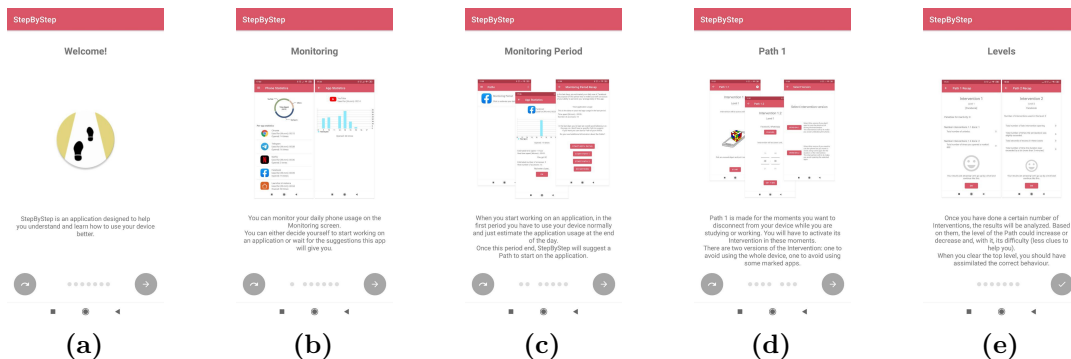
Alcune delle funzionalità dell'applicazione presentate nelle Sezioni precedenti richiedono la concessione di alcuni permessi specifici del sistema Android. Per questo motivo, alla prima apertura di StepByStep, dopo l'installazione, l'app indirizzerà ad un'Activity dedicata alla richiesta di questi permessi. I permessi devono essere forniti manualmente dall'utente per mezzo di alcune schermate dell'applicazione Impostazioni del sistema Android, schermate che possono venire aperte direttamente da StepByStep tramite l'utilizzo di alcuni Intent. In particolare, questi Intent e, di conseguenza, i permessi richiesti, sono:

- `DevicePolicyManager.ACTION_ADD_DEVICE_ADMIN`  
abilita l'app come applicazione di amministrazione del sistema, necessario per permettere a StepByStep di bloccare il telefono
- `Settings.ACTION_USAGE_ACCESS_SETTINGS`  
consente l'accesso alle statistiche di utilizzo del cellulare tramite l'uso di UsageStatsManager
- `Settings.ACTION_MANAGE_OVERLAY_PERMISSION`  
consente l'apertura di alcune schermate di StepByStep sopra ad altre app

- `Settings.ACTION_REQUEST_IGNORE_BATTERY_OPTIMIZATIONS` consente di bypassare le impostazioni di ottimizzazione della batteria del telefono, garantendo così che l'applicazione si attivi quando necessario e non venga terminata sommariamente dal sistema operativo
- permesso di Autostart, il cui Intent è dipendente dal produttore del cellulare (si sono inseriti quelli delle principali aziende), necessario per permettere ad alcune componenti di StepByStep di avviarsi autonomamente quando necessario

La richiesta di questi permessi viene gestita da tre componenti dell'applicazione:

- **PermissionFragment**, con cui si gestisce semplicemente la schermata che viene mostrata all'utente, la quale presenta una breve spiegazione del permesso e il Button con cui fornirlo
- **PermissionManager**, in cui sono raccolte le funzioni con cui si controlla lo stato di un determinato permesso
- **PermissionActivity**, che in base al valore di ritorno delle funzioni di PermissionManager aprirà il relativo PermissionFragment, passando ad esso l'Intent corretto e i dati per l'UI



**Figura 4.7:** Alcune delle schermate di tutorial di StepByStep

Data la relativa complessità di StepByStep, la cui modalità d'uso potrebbe non essere immediatamente intuibile, è necessaria la presenza di un tutorial, in cui si spieghi all'utente il funzionamento dell'applicazione. Per implementare il tutorial si è utilizzata la libreria AppIntro<sup>20</sup>, con la quale si è creata una semplice serie

<sup>20</sup><https://github.com/AppIntro/AppIntro>

di schermate in cui un'immagine e un breve testo introducono le funzionalità di StepByStep all'utente. Si possono vedere alcuni esempi in Figura 4.7. Oltre a venire automaticamente aperto al primo avvio dell'applicazione, si può accedere al tutorial in qualsiasi momento, tramite un apposito pulsante nel menù della schermata *Home* dell'applicazione.

Infine, l'ultimo dei compiti richiesti all'utente alla prima apertura, prima dell'accesso all'effettiva applicazione, è effettuare il login su Firebase. Questa operazione è necessaria solamente per permettere la profilazione dell'utente e il salvataggio dei suoi dati sul server Firestore. Poiché l'autenticazione non è fondamentale per le effettive funzionalità di StepByStep, ma solo per la sua fase di test, si è scelto di limitarla, permettendo solamente l'uso di un account Google<sup>21</sup>.

---

<sup>21</sup><https://firebase.google.com/docs/auth/android/google-signin>





# Capitolo 5

## Valutazioni

L'applicazione è stata testata *in-the-wild* tramite un modello di test *within-subject*. Tutti i tester, quindi, hanno utilizzato StepByStep durante la loro giornata in condizioni reali e avendo a disposizione tutte le funzionalità dell'applicazione. I partecipanti sono persone appartenenti alla mia sfera sociale, che sono state reclutate tramite messaggi privati, usati per introdurre le varie fasi dell'esperimento.

### 5.1 Struttura del test

Il test ha avuto una durata di 24 giorni e si divideva in:

- questionario iniziale, visibile nell'Appendice A.1
- utilizzo dell'app per un periodo di 24 giorni
- questionario finale, visibile nell'Appendice A.2

La durata del test e i relativi parametri dell'applicazione, discussi nel capitolo precedente (riferimento a Sezione 4.4.3), sono stati definiti in modo tale che gli utenti avessero la possibilità completare un *Path* dall'inizio alla fine durante la fase di test.

Il questionario iniziale, oltre a richiedere l'approvazione del modulo di consenso sul trattamento dei dati e a fornire alcune informazioni generiche sull'applicazione, era principalmente incentrato sulla raccolta di informazioni demografiche sull'utente. Tramite alcune domande, inoltre, si esplorava la percezione dell'utente in merito all'uso del proprio smartphone e la sua eventuale intenzione di migliorarlo. Al termine di questo, veniva chiesto al partecipante di installare l'applicazione tramite un link che consentiva l'accesso all'apk.

Durante il periodo di 24 giorni, il tester era invitato ad utilizzare le funzionalità dell'app il più possibile mentre StepByStep collezionava i dati del tester sul database

Firestore secondo le modalità descritte a Sezione 4.2.1. Sul database sono stati memorizzati dati riguardanti l'app in sé, quindi informazioni su *Path* (e.g. creazione, cancellazione, passaggio di livello...) e risultati degli *interventions* eseguiti, ma anche le statistiche dell'uso giornaliero del cellulare. In tal modo è stato possibile studiare eventuali cambiamenti nell'uso di certe applicazioni e verificare in via preliminare l'efficacia dell'applicazione.

Il questionario finale era costituito da:

- il questionario System Usability Scale (SUS) [27], per lo studio dell'usabilità dell'applicazione;
- alcune domande aperte con cui è stato chiesto all'utente di condividere la propria opinione sull'applicazione.

Nelle domande aperte è stata indagata l'utilità delle notifiche usate per segnalare l'uso scorretto di un'applicazione dal punto di vista dell'utente. Inoltre, si chiedeva ai partecipanti di comunicare le problematiche incontrate ed esprimere eventuali suggerimenti per migliorare l'applicazione.

Al termine del questionario finale, è stato richiesto di disinstallare StepByStep. Nei giorni seguenti la fine del test, sono state fatte indagini aggiuntive tramite l'utilizzo di messaggi privati, in modo da approfondire, nei casi in cui si rivelava necessario, eventuali risposte fornite nel questionario.

## 5.2 Risultati

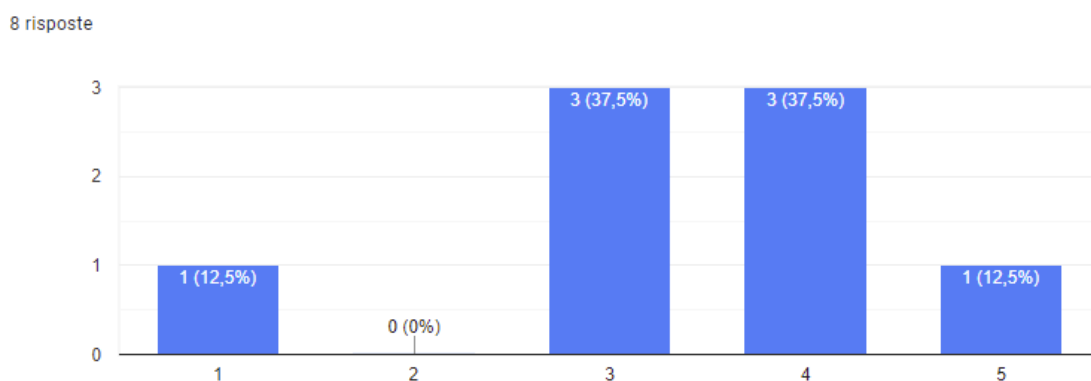
### 5.2.1 Partecipanti

Dei 10 partecipanti reclutati inizialmente, 8 hanno completato l'intero percorso di test. I loro dati saranno quelli considerati per le analisi.

Degli 8 partecipanti 6 sono uomini e 2 donne, l'età media è di 25 anni e, per quanto riguarda l'occupazione, 2 sono lavoratori, 5 studenti e 1 fa parte di entrambe le categorie. Il 50% dei partecipanti, inoltre, ha affermato di non aver lavorato e/o studiato durante il periodo di test. Questa domanda è stata fatta in modo da poter controllare l'eventuale presenza di forti deviazioni nell'uso dell'applicazione che potessero essere provocate dalle vacanze.

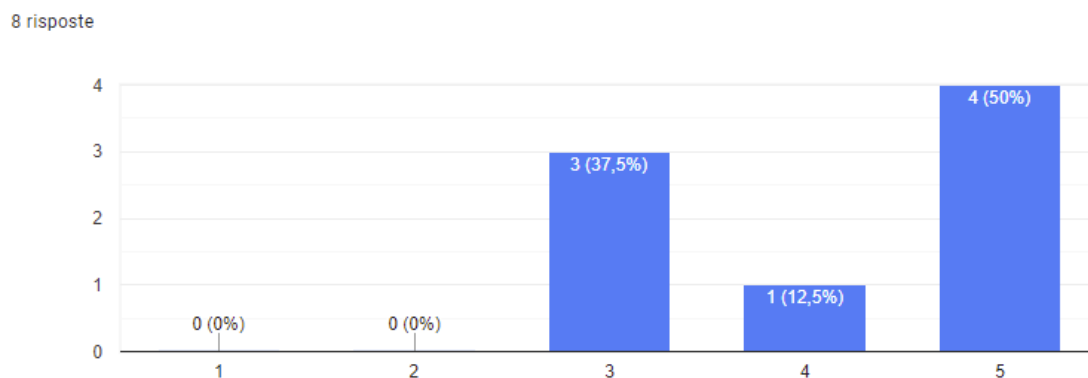
Per quanto riguarda la percezione sull'utilizzo dello smartphone, le risposte sono state mediamente positive. Solo un partecipante ha valutato negativamente il proprio uso del telefono, come si può vedere dal grafico a Figura 5.1

Eppure, nonostante le valutazioni per lo più positive date per questa domanda, il 75% (6) dei partecipanti ha affermato di utilizzare alcune app in maniera eccessiva. Tra queste, la più indicata è Instagram, segnata da 6 partecipanti, seguita da YouTube, 3, e da TikTok e Whatsapp, 1 per entrambe.



**Figura 5.1:** Risposte alla domanda sulla percezione di utilizzo del cellulare, con una scala Likert da pessimo (1) a ottimo (5)

L'ultima domanda, riguardante l'intenzione di migliorare l'utilizzo del proprio smartphone, se ce ne fosse possibilità, ha avuto risposte generalmente positive, come è possibile vedere da Figura 5.2.



**Figura 5.2:** Risposte alla domanda sull'intenzione di migliorare il proprio utilizzo del cellulare, con una scala Likert da poco (1) a molto (5)

## 5.2.2 Usabilità

L'usabilità totale dell'applicazione è stata verificata nel questionario finale, mediante l'uso della System Usability Scale [27]. Il questionario SUS è un metodo comunemente utilizzato per calcolare un punteggio in centesimi che indichi l'usabilità di un sistema. StepByStep è stata valutata con un punteggio medio di 87,19 (con un minimo a 72,5 e un massimo a 97,5). Ciò la pone ben sopra il valore medio di 68, che la scala utilizza come limite, ed è indice di un'assenza di grossi

difetti che compromettano l'usabilità dell'applicazione e un'esperienza dell'utente complessivamente positiva su di essa. Ciò è stato rafforzato anche dalle domande aperte finali, in cui non sono stati segnalati problemi significativi nell'utilizzo di StepByStep.

La totalità dei partecipanti ha trovato utile la notifica di uso eccessivo di un'applicazione e suggerimento di inizio *Path*. Nonostante ciò, in molte domande aperte (3), è stata indicata come problematica la creazione di notifiche per alcune applicazioni che non si erano usate molto e, inoltre, alcuni dei miglioramenti suggeriti (2) includevano una maggiore complessità delle notifiche. Quindi, per quanto gli utenti abbiano trovato un'utilità nella presenza di queste segnalazioni di uso eccessivo, le misure determinanti per la creazione delle notifiche sono state considerate troppo semplici.

I partecipanti hanno avuto modo di segnalare nel questionario finale la presenza di eventuali problematiche avute nell'uso dell'applicazione. Le risposte raccolte posso essere suddivise in due insiemi: problemi dovuti ad errori nel codice, quindi risolvibili, e problemi dovuti a mancanze nelle funzionalità dell'applicazione. Rientrano nella prima categoria:

- "-1" come numero di sblocchi cellulare nell'*intervention* 1 versione 1 nel caso non si sbloccasse mai il cellulare e si interrompesse l'intervento dopo lo scadere del tempo;
- l'errato conteggio delle statistiche di alcune applicazioni che portava alla visualizzazione di dati errati e quindi alla creazione di notifiche non necessarie;
- un'incostanza generale nella creazione delle notifiche, segnalata da tutti gli utenti, sulla quale è necessario fare test più approfonditi per determinarne la causa;
- un'inconsistenza percepita tra le statistiche di un'applicazione mostrate nella notifica di uso eccessivo e quelle visibili dalla pagina di monitoraggio, dovuta in realtà a una non chiara esposizione dei differenti intervalli temporali delle due statistiche.

Fanno invece parte del secondo gruppo:

- una durata poco chiara del periodo iniziale di monitoraggio;
- l'insufficienza dei 15 minuti massimi per la durata dell'*Intervention* 2, segnalata da un utente facendo particolare riferimento al suo uso di YouTube, che spesso consisteva nel guardare video di maggiore durata mentre svolgeva altri compiti (e.g. cucinare, attività fisica...).

I partecipanti sono inoltre stati molto prolifici per quanto riguarda i suggerimenti per eventuali miglioramenti dell'applicazione. Degli 8 partecipanti, 6 hanno condiviso le proprie idee, di seguito elencate:

- mandare una maggiore quantità di notifiche durante la giornata per ricordare di attivare gli *interventions*;
- prevedere la definizione di periodi della giornata dedicati allo studio e allo svago, nonché l'implementazione di una "modalità vacanza", per poter dare una maggiore flessibilità;
- rendere gli *intervention* di tipo 2 attivabili su più applicazioni, in modo da poter passare senza limitazioni quel momento di svago programmato sulle 2/3 applicazioni che l'utente è solito utilizzare in questo modo;
- modificare le notifiche di uso eccessivo in modo che non considerino solo l'uso di un'app nelle ultime 24 ore ma anche quello nei giorni passati, in modo da segnalare solo le applicazioni che abitualmente si usano in modo scorretto (indicato da due partecipanti);
- creare un tutorial più approfondito, che non si limiti a mostrare una serie di slide ma faccia effettivamente provare all'utente una versione semplificata dell'app;
- unificare i due *interventions*, in modo che non siano utilizzati separatamente l'uno dall'altro ma in maniera combinata, e renderli programmabili per riuscire a definire periodi tempo anche lunghi in cui creare un modello per regolare lavoro e svago.

Per quanto alcuni suggerimenti siano interessanti, altri vanno in conflitto con quelli che sono gli obiettivi di StepByStep, che vuole essere un'applicazione per aiutare l'utente a utilizzare meglio il cellulare con un particolare accento sulla promozione della sua indipendenza.

### 5.2.3 Efficacia

Durante il periodo di valutazione di 24 giorni gli utenti hanno complessivamente avviato il monitoraggio di 23 applicazioni. Di queste, tutte hanno portato alla creazione di un *Path* alla fine del periodo di monitoraggio iniziale e questi *Path* sono stati utilizzati fino al termine della fase di test. Le cancellazioni, 3 in totale, possono essere catalogate come facenti parte di una fase esplorativa dell'applicazione. In tutti i tre casi, infatti, i periodi di monitoraggio sono stati ricreati poco dopo.

Le app prese in considerazione sono:

- Instagram (6 volte);
- YouTube, Telegram e Whatsapp (3 volte);
- Facebook e Twitter (2 volte);
- TikTok, Chrome, GuardianTales e Scacchi (1 volta).

In totale, sono stati creati 8 *Path* di tipo 1 (si consideri che in molteplici casi più applicazioni sono state integrate in un singolo *Path* di questo tipo) e 8 *Path* di tipo 2. Complessivamente, 17 delle 24 applicazioni iniziali hanno portato alla creazione di un *Path* 1, contro le 8 del *Path* 2, rivelando una leggera preferenza dei partecipanti per i servizi offerti dal *Path* 1.

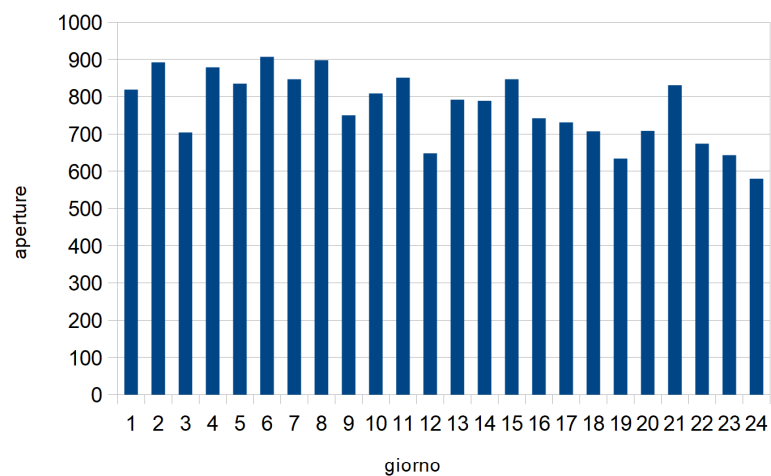
Sono stati complessivamente eseguiti 72 *interventions* di tipo 1 (25 della versione 1, 47 della versione 2) e 712 *interventions* di tipo 2. Nessun partecipante è riuscito durante i 24 giorni a portare a completamento un *Path*. Dei 16 *Path* iniziati, 3 sono rimasti a livello 1, 7 a livello 2, 4 a livello 3 e solamente 2 sono stati portati a livello 4.

Durante l'esecuzione degli *intervention* 1 versione 1, i partecipanti hanno complessivamente sbloccato lo schermo del proprio smartphone 47 volte mentre, per la versione 2, si sono aperte per 242 volte le applicazioni segnate. Per quanto riguarda l'*intervention* 2, complessivamente si sono registrate 336 penalità P1 (*intervention* non avviato manualmente, riferimento a Tabella 4.3), 62 P2 (tempo limite superato di poco) e 22 P3 (tempo limite superato di molto). Per entrambi gli *intervention*, la distribuzione di queste penalità è uniforme sui livelli.

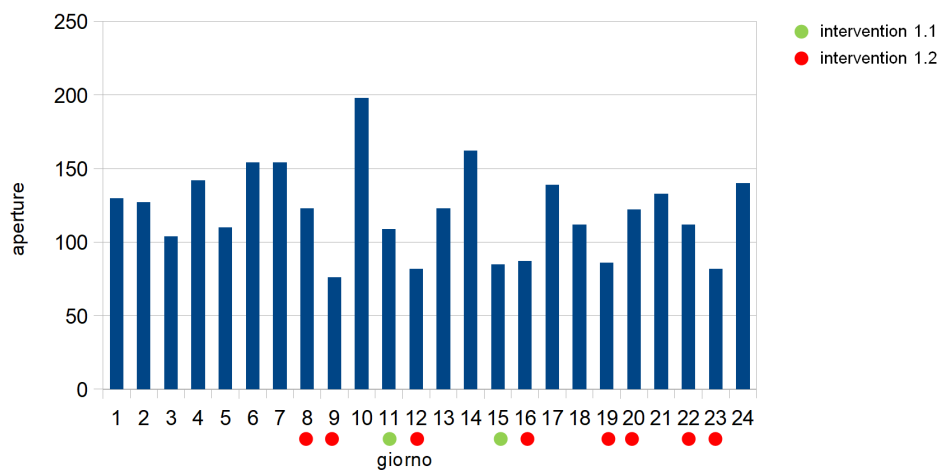
Per verificare l'efficacia dell'applicazione si sono presi in considerazione i dati riguardanti il numero di aperture delle applicazioni su cui si è attivato il *Path* 1 e il tempo speso sulle applicazioni su cui si è attivato il *Path* 2

La Figura 5.3 mostra il dato giornaliero circa il numero di aperture delle applicazioni dei *Path* 1. Confrontando i dati facenti parte dell'intervallo di giorni 1-7 (periodo di monitoraggio) con quelli dell'intervallo 8-24 (attivazione ed utilizzo dei *Path*), si è passati da una media di circa 840 aperture giornaliere a circa 743, con una diminuzione del 12,5%.

É necessario però precisare che la maggior parte dei partecipanti ha utilizzato il *Path* 1 in maniera sporadica, dimenticandosi spesso di attivarlo, come dimostra il numero relativamente alto di notifiche di inattività create (15). Nella Figura 5.4 si prende come esempio le aperture giornaliere delle applicazioni di un singolo partecipante, con evidenziati i giorni di esecuzione degli *intervention*. Prendendo in considerazione solamente questi giorni rispettivamente a tutti i rimanenti, si ottiene una riduzione del numero medio di aperture del 25%.



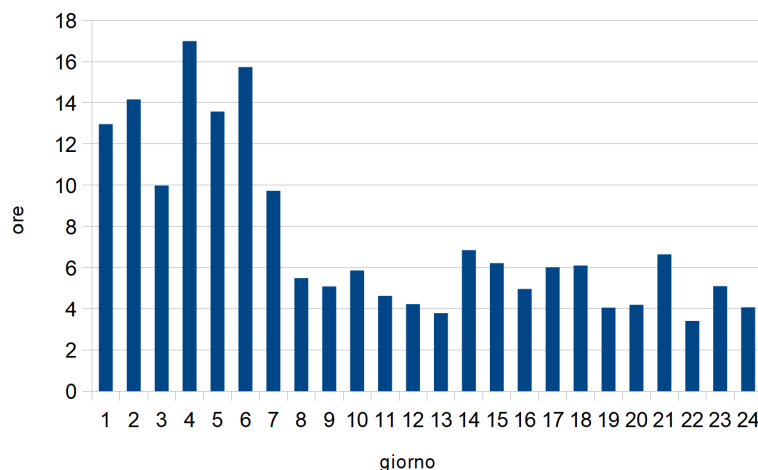
**Figura 5.3:** Grafico del numero di aperture totali delle applicazioni su cui si è attivato il *Path 1*



**Figura 5.4:** Grafico del numero di aperture totali delle applicazioni su cui è stato attivato il *Path 1* di un singolo partecipante, con particolare delle giornate di utilizzo di un *intervention*



A Figura 5.5 è invece rappresentato il numero di ore spese sulle applicazioni dei *Path 2*. Come si può vedere, dal momento della creazione dei *Path* (giorno 8) vi è una riduzione giornaliera media del 62%. Risultati rilevanti, soprattutto considerando che non sono esito della presenza di blocchi restrittivi, ma delle decisioni consapevoli degli utenti, supportate da StepByStep.



**Figura 5.5:** Grafico del numero di ore totali spese su applicazioni su cui si è attivato il *Path 2*

### 5.3 Discussione

Il principale obiettivo di questo lavoro di tesi ha riguardato lo sviluppo e il testing di un applicazione con un approccio educativo al Digital Wellbeing. Per sviluppare questa app, si è cercato di implementare degli interventi basati sui principi di alcune delle teorie che si collocano nel contesto del cambio comportamentale, nominalmente il *Transtheoretical Model* [13] e la *Dual-Process Theory* [7]. In accordo con la letteratura precedente, già ampiamente discussa nel Capitolo 2, i risultati ottenuti sono promettenti per quanto riguarda l'applicazione di queste teorie nel contesto del Digital Wellbeing.

Durante la fase di test, tutti i partecipanti (8) arrivati alla fine della prima settimana hanno utilizzato StepByStep, più o meno frequentemente, per l'intero periodo. Ciò, in aggiunta alla valutazione positiva che è stata data riguardo al suggerimento proattivo, va a confermare l'utilità dei due interventi indicati dal *Transtheoretical Model* come *consciousness raising* e *contingency management* (Figura 2.1). In linea con il lavoro di Dijk et al [13] e Bandura [11], si è dimostrato che fornire all'utente feedback continuo e immediato, in questo caso nella forma di

risultati dell'esecuzione di *intervention* e passaggi di livello, favorisca l'avanzamento del percorso di cambio comportamentale.

Considerando i risultati incoraggianti ottenuti dall'*intervention 2* e il numero ridotto di penalità di superamento del tempo P2 e P3 rilevate (rispettivamente, 62 e 22 su 712 esecuzioni totali dell'*intervention*, l'8% e 3% circa dei casi) si può cautamente definire come efficace l'uso della vibrazione come *cues* per attivare il Sistema 2. Come indicato dalla *Dual-Process Theory* [7, 4] e dall'*Habit Alteration Model* [8], l'utilizzo di *cues* esterni come attivatori del Sistema di tipo 2 si è confermato essere un intervento efficace per quanto riguarda il cambiamento comportamentale.

Inoltre, nonostante le limitazioni della presente ricerca nell'ambito dell'acquisizione di un nuovo comportamento, esposti approfonditamente a Sezione 6.1, possono essere fatte le seguenti considerazioni. Data la distribuzione uniforme nei livelli delle varie penalità nelle esecuzioni degli *intervention*, i partecipanti sono stati in grado in media di mantenere il comportamento suggerito anche con la diminuzione degli aiuti forniti da StepByStep. Ciò suggerisce un'effettiva interiorizzazione dei comportamenti consigliati dall'app ma il mantenimento nel tempo di questa autonomia acquisita andrebbe ulteriormente verificato.

Infine, l'assenza di effettivi blocchi sull'uso del cellulare all'interno di StepByStep permette di considerare come molto interessanti i risultati ottenuti, in particolare la riduzione del 62% dell'uso delle applicazioni dei *Path 2*. Come già spiegato, StepByStep si limita a informare e consigliare l'utente, non ad obbligarlo; l'utente è completamente libero di ignorare i consigli e continuare ad agire come preferisce. Nonostante quindi la prevalenza nello stato dell'arte di blocchi particolarmente restrittivi [4] e la generica preferenza dimostrata dagli utenti per questi [5, 28], in questo caso si sono ottenuti considerevoli miglioramenti con dei limiti molto *soft* e senza blocchi veri e propri. Questa discrepanza potrebbe in parte essere attribuita all'inserimento degli *intervention* (e quindi dei "blocchi") all'interno del percorso del *Path*, che, fornendo all'utente obiettivi a breve termine (e.g. passaggio di livello) e feedback continui, ne aumenta la motivazione e rafforza la capacità di autoregolazione.



# Capitolo 6

## Conclusioni

Per quanto questo studio abbia dato risultati molto promettenti, soprattutto per quanto riguarda l'*intervention 2*, è necessario fare delle doverose precisazioni a riguardo.

### 6.1 Limiti della ricerca

Lo studio è soggetto a due considerevoli limitazioni, entrambe legate alle restrizioni della fase di test:

- il ridotto numero di partecipanti e il campionamento di convenienza, che non permettono la generalizzazione dei risultati;
- la breve durata della fase di test.

Per quanto riguarda il tempo, poiché la durata della fase di test (24 giorni) e le durate delle varie fasi dell'applicazione (7 giorni di monitoraggio, 4 giorni per livello) sono interdipendenti, queste sono state impostate in modo da consentire ai partecipanti di poter potenzialmente portare a termine un *Path*. L'intervallo di 24 giorni è infatti suddivisibile in 7 giorni di monitoraggio, 16 di utilizzo dei *Path* (4 per livello) e 1 extra per permettere una maggiore flessibilità.

Tuttavia, questa durata si rivela insufficiente, dal punto di vista del test, per poter garantire una completa verifica di un'applicazione di Digital Wellbeing [29]. Vengono infatti considerati solo i risultati a breve termine e ignorati quelli a medio e lungo termine.

Inoltre, i 4 giorni per livello di un *Path* si rivelano eccessivamente corti, dal punto di vista delle abitudini, in considerazione dell'intervallo di tempo indicato per la formazione di queste (18-254 giorni) [26]. Andrebbe quindi considerato un aumento della durata di ogni livello.

Quindi, considerando entrambi i punti, per quanto si possa affermare l'efficacia dell'applicazione, in particolare dell'*intervention 2*, non si può dire la stessa cosa per quanto riguarda l'acquisizione dell'abitudine insegnata da StepByStep, a causa delle notevoli limitazioni temporali esposte prima.

## 6.2 Sviluppi futuri

Il primo passo da considerare per i possibili lavori futuri è, quindi, lo sviluppo di una fase di test più complessa, con un maggior numero di partecipanti e una durata che permetta di utilizzare adeguati parametri temporali dell'applicazione, in modo da poter ottenere risultati rilevanti per quanto riguarda l'acquisizione dell'abitudine. Gli stessi parametri temporali di StepByStep potrebbero venire a loro volta rifiniti tramite un'altro test, utilizzandoli come variabile indipendente.

Oltre a ciò, si potrebbero considerare alcune delle segnalazioni fatte dai partecipanti. Delle modifiche altamente consigliate potrebbero essere l'aumento della complessità dei dati utilizzati per la creazione del suggerimento proattivo e una maggiore presenza di notifiche di supporto per guidare l'attivazione dell'*intervention 1*. Un'altra considerazione si potrebbe fare sull'implementazione di una maggiore interfunzionalità tra i due *intervention*.

Infine, tenendo in considerazione i risultati minori ottenuti dal *Path 1*, per quanto possano essere legati all'incostanza di utilizzo degli *intervention*, si potrebbe valutarne un redesign, in particolare dell'*intervention 1.2*, per una migliore applicazione del principio di "attivazione del Sistema 2", introducendo un elemento simile all'oggetto "insolito" della versione 1 (e.g. cambiando o spostando le icone delle applicazioni segnate).



# Appendice A

## Questionari

### A.1 Questionario 1

<b>Domanda</b>	<b>Tipo</b>
Email	Testuale
Età	Scelta multipla
Genere	Scelta multipla
Occupazione	Scelta multipla
Andrai in ferie durante il periodo di test?	Scelta multipla
In generale, come valuti l'uso che fai del cellulare?	Scala Likert
Ci sono applicazioni che ritieni di usare in modo eccessivo?	Scelta multipla
Se hai risposto Sì alla domanda precedente, quali sono queste applicazioni?	Scelta multipla (multiopzione)
Se ne avessi la possibilità, quanto saresti intenzionato a migliorare l'uso che fai del tuo cellulare?	Scala Likert

## A.2 Questionario 2

Domanda	Tipo
Penso che mi piacerebbe utilizzare StepByStep frequentemente	Scala Likert
Ho trovato StepByStep complessa senza che ce ne fosse troppo bisogno	Scala Likert
Ho trovato StepByStep molto semplice da usare	Scala Likert
Penso che avrei avuto bisogno del supporto di una persona già in grado di utilizzare StepByStep	Scala Likert
Ho trovato le varie funzionalità di StepByStep ben integrate	Scala Likert
Ho trovato incoerenze tra le varie funzionalità di StepByStep	Scala Likert
Penso che la maggior parte delle persone potrebbe imparare a usare StepByStep facilmente	Scala Likert
Ho trovato StepByStep molto macchinoso da utilizzare	Scala Likert
Ho avuto molta confidenza con StepByStep durante l'uso	Scala Likert
Ho avuto bisogno di imparare molti processi prima di riuscire ad utilizzare al meglio StepByStep	Scala Likert

Domanda	Tipo
Email	Testuale
Hai trovato utili i suggerimenti automatici per iniziare a lavorare su delle applicazioni scelte da StepByStep?	Scelta multipla
Motiva brevemente il perché della tua risposta alla domanda precedente	Aperta
Indica brevemente se ci sono state parti dell'applicazioni poco chiare o che ritieni abbiano funzionato in modo errato	Aperta
Se hai suggerimenti su come modificare o ampliare l'applicazione, scrivili di seguito	Aperta



# Bibliografia

- [1] Michelle H van Velthoven, John Powell e Georgina Powell. «Problematic smartphone use: Digital approaches to an emerging public health problem». In: *DIGITAL HEALTH* 4 (2018), p. 2055207618759167. DOI: 10.1177/2055207618759167 (cit. a p. 2).
- [2] «The dark side of smartphone usage: Psychological traits, compulsive behavior and technostress». In: *Computers in Human Behavior* 31 (2014), pp. 373–383. ISSN: 0747-5632. DOI: 10.1016/j.chb.2013.10.047 (cit. a p. 2).
- [3] Chien Chou, Linda Condron e John Belland. «A Review of the Research on Internet Addiction». In: *Educational Psychology Review* 17 (dic. 2005), pp. 363–388. DOI: 10.1007/s10648-005-8138-1 (cit. a p. 2).
- [4] Ulrik Lyngs, Kai Lukoff, Petr Slovak, Reuben Binns, Adam Slack, Michael Inzlicht, Max Van Kleek e Nigel Shadbolt. «Self-Control in Cyberspace: Applying Dual Systems Theory to a Review of Digital Self-Control Tools». In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2019, pp. 1–18. ISBN: 9781450359702. DOI: 10.1145/3290605.3300361 (cit. alle pp. 2, 3, 8, 9, 11, 13, 65).
- [5] Alberto Monge Roffarello e Luigi De Russis. «The Race Towards Digital Wellbeing: Issues and Opportunities». In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2019, pp. 1–14. ISBN: 9781450359702. DOI: 10.1145/3290605.3300616 (cit. alle pp. 2, 13, 65).
- [6] Katarzyna Stawarz, Anna Cox e Ann Blandford. «Beyond Self-Tracking and Reminders: Designing Smartphone Apps That Support Habit Formation». In: apr. 2014. DOI: 10.1145/2702123.2702230 (cit. alle pp. 2, 11–13, 26).
- [7] Daniel Kahneman. *Thinking, fast and slow*. New York: Farrar, Straus e Giroux, 2011. ISBN: 978-0-374-27563-1 (cit. alle pp. 3, 8, 64, 65).

- 
- [8] Charlie Pinder, Jo Vermeulen, Benjamin R. Cowan e Russell Beale. «Digital Behaviour Change Interventions to Break and Form Habits». In: *ACM Trans. Comput.-Hum. Interact.* 25.3 (giu. 2018). ISSN: 1073-0516. DOI: 10.1145/3196830 (cit. alle pp. 3, 8–12, 21, 65).
- [9] Lora Burke e Jing Wang. «Self-Monitoring in Weight Loss: A Systematic Review of the Literature». In: *Journal of the American Dietetic Association* 111 (gen. 2011), pp. 92–102. DOI: 10.1016/j.jada.2010.10.008 (cit. a p. 5).
- [10] Kathryn Buchanan, Riccardo Russo e Ben Anderson. «The question of energy reduction: The problem(s) with feedback». In: *Energy Policy* 77 (2015), pp. 89–96. ISSN: 0301-4215. DOI: 10.1016/j.enpol.2014.12.008 (cit. a p. 5).
- [11] Albert Bandura. «Social cognitive theory of self-regulation». In: *Organizational Behavior and Human Decision Processes* 50.2 (1991). Theories of Cognitive Self-Regulation, pp. 248–287. ISSN: 0749-5978. DOI: 10.1016/0749-5978(91)90022-L (cit. alle pp. 5, 7, 64).
- [12] Elisabeth T. Kersten-van Dijk, Joyce H.D.M. Westerink, Femke Beute e Wijnand A. IJsselsteijn. «Personal Informatics, Self-Insight, and Behavior Change: A Critical Review of Current Literature». In: *Human-Computer Interaction* 32.5-6 (2017), pp. 268–296. DOI: 10.1080/07370024.2016.1276456 (cit. alle pp. 5–7).
- [13] James O. Prochaska e Wayne F. Velicer. «The Transtheoretical Model of Health Behavior Change». In: *American Journal of Health Promotion* 12.1 (1997), pp. 38–48. DOI: 10.4278/0890-1171-12.1.38 (cit. alle pp. 5, 64).
- [14] Phillippa Lally e Benjamin Gardner. «Promoting habit formation». In: *Health Psychology Review* 7 (gen. 2011). DOI: 10.1080/17437199.2011.603640 (cit. alle pp. 7, 11).
- [15] Benjamin Gardner, De de Bruijn e Phillippa Lally. «A Systematic Review and Meta-analysis of Applications of the Self-Report Habit Index to Nutrition and Physical Activity Behaviours». In: *Annals of behavioral medicine : a publication of the Society of Behavioral Medicine* 42 (mag. 2011), pp. 174–87. DOI: 10.1007/s12160-011-9282-0 (cit. a p. 7).
- [16] Antti Oulasvirta, Tye Rattenbury, Lingyi Ma e Eeva Raita. «Habits make smartphone use more pervasive». In: *Personal and Ubiquitous Computing* 16 (giu. 2012), pp. 105–114. DOI: 10.1007/s00779-011-0412-2 (cit. alle pp. 7, 25).
- [17] Laura Dabbish, Gloria Mark e Victor Gonzalez. «Why Do I Keep Interrupting Myself?: Environment, Habit and Self-Interruption». In: mag. 2011, pp. 3127–3130. DOI: 10.1145/1978942.1979405 (cit. a p. 8).

- [18] Cary Stothart, Ainsley Mitchum e Courtney Yehnert. «The Attentional Cost of Receiving a Cell Phone Notification». In: *Journal of experimental psychology. Human perception and performance* (giu. 2015). DOI: 10.1037/xhp0000100 (cit. a p. 8).
- [19] Martin Pielot, Karen Church e Rodrigo de Oliveira. «An in-situ study of mobile phone notifications». In: set. 2014, pp. 233–242. DOI: 10.1145/2628363.2628364 (cit. a p. 8).
- [20] Robert West e Ainsley Hardy. *Theory of addiction / Robert West, with Ainsley Hardy*. Malden, MA: Blackwell Pub., 2005. ISBN: 1405113596 (cit. a p. 8).
- [21] Icek Ajzen. «The theory of planned behavior». In: *Organizational Behavior and Human Decision Processes* 50.2 (1991), pp. 179–211. ISSN: 0749-5978. DOI: 10.1016/0749-5978(91)90020-T (cit. a p. 8).
- [22] Nelson Cowan. «The Magical Mystery Four: How Is Working Memory Capacity Limited, and Why?» In: *Current Directions in Psychological Science* 19.1 (2010), pp. 51–57. DOI: 10.1177/0963721409359277 (cit. a p. 9).
- [23] Edwin Locke e Gary Latham. «A Theory of Goal Setting and Task Performance». In: *The Academy of Management Review* 16 (apr. 1991). DOI: 10.2307/258875 (cit. a p. 9).
- [24] Edwin Locke e Gary Latham. «New Directions in Goal-Setting Theory». In: *Current Directions in Psychological Science* 15 (ott. 2006). DOI: 10.1111/j.1467-8721.2006.00449.x (cit. a p. 9).
- [25] Wendy Wood e David Neal. «A New Look at Habits and the Habit-Goal Interface». In: *Psychological review* 114 (nov. 2007), pp. 843–63. DOI: 10.1037/0033-295X.114.4.843 (cit. a p. 11).
- [26] Phillippa Lally, Cornelia Jaarsveld, Henry Potts e Jane Wardle. «How are habits formed: Modeling habit formation in the real world». In: *European Journal of Social Psychology* 40 (ott. 2010). DOI: 10.1002/ejsp.674 (cit. alle pp. 11, 12, 67).
- [27] John Brooke. «SUS: A quick and dirty usability scale». In: *Usability Eval. Ind.* 189 (nov. 1995) (cit. alle pp. 58, 59).
- [28] Markus Löchtefeld, Matthias Böhmer e Lyubomir Ganev. «AppDetox: Helping users with mobile app addiction». In: dic. 2013. DOI: 10.1145/2541831.2541870 (cit. a p. 65).
- [29] Predrag Klasnja, Sunny Consolvo e Wanda Pratt. «How to evaluate technologies for health behavior change in HCI research». In: mag. 2011, pp. 3063–3072. DOI: 10.1145/1978942.1979396 (cit. a p. 67).