

Progettazione di un'applicazione ubiqua per il Turismo 4.0

Sabrina Celia, Danilo Cistaro e Agostino Forestiero

RT-ICAR-CS-14-04

Dicembre 2014



Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)
– Sede di Cosenza, Via P. Bucci 41C, 87036 Rende, Italy, URL: www.icar.cnr.it
– Sezione di Napoli, Via P. Castellino 111, 80131 Napoli, URL: www.na.icar.cnr.it
– Sezione di Palermo, Viale delle Scienze, 90128 Palermo, URL: www.pa.icar.cnr.it

Sommario

Introduzione.....	3
Struttura dell'applicazione.....	5
Storytelling	5
Navigatore	5
Realtà aumentata	6
Progettazione	6
Descrizione delle interfacce.....	6
Caso d'uso "Consultazione delle informazioni":	7
Caso d'uso "Attiva realtà aumentata":	8
Caso d'uso "Attiva il navigatore":	8
Architettura.....	10
Interfaccia grafica (GUI)	10
Scheda di identità	12
Dettaglio POI.....	13
Controller.....	14
Controller scheda di identità.....	15
Servizio lettura GPS	15
Servizio download POI.....	16
Controller dettaglio POI	17
Text to speech.....	17
Servizio notifica.....	19
Alarm Manager.....	20
Ricerca e integrazione di nuove funzioni utente.....	20
Ottimizzazione dell'interfaccia grafica	23
Integrazione dei contenuti di Cicerone su piattaforma IOS.....	24
Analisi di una feature di Realtà Aumentata.....	24

Introduzione

Il presente rapporto tecnico è stato realizzato nell'ambito dell'attività A2.5.3 del progetto di ricerca INMOTO – Information MObility for TOurism, del Programma Operativo Nazionale Ricerca e Competitività 2007-2013 - Smart Cities and Communities and Social Innovation Asse II - Azioni Integrate per lo Sviluppo Sostenibile Ambito: Smart Culture e Turismo in collaborazione con: (i) ACI Informatica S.p.A., (ii) Università della Calabria, (iii) Università di Catanzaro e (iv) TSC Consulting s.r.l.. Il progetto Cultura e Turismo: DiCet-InMoto ha l'obiettivo di definire e realizzare modelli, processi e strumenti innovativi per lo sviluppo sostenibile di un territorio intelligente attraverso la valorizzazione dei suoi beni culturali e risorse ambientali e la promozione e commercializzazione della loro offerta turistica. Tale obiettivo risponde alle esigenze emergenti di promuovere l'evoluzione del territorio verso un modello più sostenibile e intelligente coerentemente con i recenti orientamenti comunitari sul tema di "Smart Cities" e le politiche d'innovazione contenute nella direttiva europea "Europe 2020". Lo scopo del progetto è quello di individuare un nuovo modello strategico di Destination Management Organization e implementare una declinazione Turismo 4.0 come una formulazione innovativa di strumenti e servizi per la promozione-commercializzazione dell'Offerta Turistica e Culturale con la finalità di:

- standardizzare, facilitare e razionalizzare la creazione e lo scambio di contenuti turistici tra enti che svolgono compiti di promozione o che operano nei servizi di mobilità, siano essi operatori del settore o cittadini del territorio;
- rendere efficienti i processi di distribuzione di servizi turistici, realizzando un ecosistema di piattaforme, smart objects, apps mobile e Web su logiche 4.0, mirato a costruire e distribuire contenuti strutturati su itinerari geografici ed esperienziali;
- agevolare il turismo in mobilità, con la definizione di servizi e la predisposizione di tecnologie in grado di garantire da un lato una larga ed accurata disponibilità di informazioni sui flussi in mobilità, orientata ad un'innovazione della programmazione logistica del territorio, e dall'altro la condivisione di mezzi di trasporto ad uso privato;
- promuovere percorsi e itinerari turistici esperienziali attraverso l'aggregazione di contenuti informativi su base geografica e tematica provenienti anche da piattaforme social;
- supportare la formazione diffusa e la conoscenza delle risorse turistiche e dei beni culturali presenti sul territorio per cittadini e turisti.

L'obiettivo del presente rapporto è quello di descrivere il concept e la progettazione della applicazione ubiqua Cicerone, pensata per la fruizione dei Point Of Interest turistici in stile navigatore, realizzata nell'ambito delle attività del progetto INMOTO.

Cicerone, fruibile da smartphone, è progettato puntando sul paradigma dello storytelling, con il supporto del navigatore integrato e di altre informazioni reperibili dai sensori di uno smarphone.

L'uso del termine Cicero come sinonimo di guida turistica nacque allorquando i cittadini romani, improvvisandosi guide locali, cominciarono ad accompagnare i facoltosi visitatori di Roma tra le sue

meraviglie archeologiche, dimostrando una abilità oratoria tale da ricordare l'antico avvocato romano Marco Tullio Cicerone. Usando come punto di riferimento i Ciceroni di Roma, Cicerone 2.0 mira a simulare tutte quelle funzioni necessarie a raccontare il territorio che un potenziale turista sta visitando. L'obiettivo è di invogliarlo alla visita, attraverso il racconto, segnalandogli nuovi punti di interesse proprio nel momento in cui si muove su un territorio.

Per garantire la realizzazione dell'obiettivo si è reso necessario l'estrazione e l'elaborazione delle seguenti informazioni:

- Rilevazione della posizione dell'utente
- Estrazione dalla piattaforma INMOTO dei contenuti in prossimità dell'utente, ordinati in base alla distanza
- Funzioni text to speech per la fruizione vocale dei contenuti (stile audio guida), utile anche per la fruizione da parte di utenti diversamente abili (non vedenti)
- Possibilità di fruizione standard da dispositivo mobile: selezione categorie, schede informative, fotogallery
- Integrazione con funzioni di navigazione verso il POI.

Gli obiettivi che l'applicativo si propone di realizzare sono:

- notificare al turista la presenza di un punto di interesse sul territorio oggetto di visita turistica
- offrire delle facilities per il raggiungimento del punto di interesse utilizzando applicazioni di terzi esistenti
- fornire ulteriori informazioni del POI attraverso la realtà aumentata.

Le informazioni provengono dalla piattaforma INMOTO e la tipologia di dati presentati al turista sono:

- Luoghi della cultura: sono comprese le aree e i parchi archeologici, i monumenti, i complessi monumentali e le altre strutture espositive permanenti destinate alla pubblica fruizione;
- Eventi culturali: informazioni sulle manifestazioni culturali (mostre, conferenze, convegni, seminari, presentazioni cataloghi, ecc.) organizzate dal Ministero e dagli Istituti periferici;
- Comunicati (News)*: i comunicati stampa del Ministero e tutte le comunicazioni provenienti dalle unità periferiche.

L'applicativo è progettato per essere utilizzato su smartphone quindi potendo contare sulle seguenti funzionalità:

- GPS: per la localizzazione della posizione sul territorio;
- Traffico dati: per il download delle informazioni;
- Fotocamera e bussola: per la realtà aumentata.

Struttura dell'applicazione

L'applicativo, denominato Cicerone 2.0, è stato progettato per smartphone ed è composto da tre sottosistemi logici fondamentali: Storytelling, Navigatore, Realtà aumentata.

Storytelling

La funzione di storytelling sta al centro di tutto il progetto. Ha il compito di fornire tutte le informazioni localizzate sul territorio utilizzando come primo approccio il sistema di notifiche.

Indipendentemente da quale sia la piattaforma mobile sul quale l'applicativo andrà ad essere eseguito, lo smartphone dovrà essere necessariamente dotato di un sistema GPS in modo da rendere più versatile possibile la notifica dei punti di interesse.

Uno dei requisiti principali con cui è stata progettata il sistema è la non invasività: il POI (punto di interesse) dovrà quindi essere notificato solo se l'utente si trova effettivamente in una zona a lui sconosciuta. Le possibili strategie adottabili sono:

- Permettere all'utente di inserire la posizione dell'abitazione e di un raggio (in km) entro il quale non vuole ricevere nessun tipo di notifiche
- Dare la possibilità all'utente di impostare una regione entro il quale non si intende ricevere alcuna notifica.

Definite le modalità di presentazione delle notifiche è necessario analizzare più da vicino la struttura dell'applicativo. La funzione di storytelling infatti rende disponibile l'accesso alle informazioni interagendo in modo diretto con la barra di stato e le icone di notifica.

Avviata l'interfaccia grafica di Cicerone 2.0, il quadro informativo sarà così composto:

- Testo informativo: contiene un'accurata descrizione del punto d'interesse;
- Immagini illustrative: consentono di migliorare l'esperienza utente usando una galleria fotografica degli elementi peculiari dell'attrattore turistico in modo da creare suggestioni visive con in grado di catturare l'attenzione del visitatore;
- Video documentari: brevi clip che completano il quadro informativo inserendo informazioni di vario genere relative al punto di interesse.

Il testo informativo, oltre ad essere letto potrà anche essere ascoltato utilizzando servizi di text to speech.

Navigatore

Requisito del sottosistema di navigazione è la facilità d'uso. La funzionalità di navigatore dovrà essere facilmente attivabile dall'interfaccia utente di Cicerone 2.0: l'utilizzatore dovrà essere accompagnato verso il punto di interesse in modo semplice e senza fornire alcuna informazione, sarà l'applicativo a comunicare i dati (posizione attuale e destinazione) al sistema di mappe già integrato nello smartphone.

Il software di navigazione che Cicerone 2.0 andrà ad utilizzare dipenderà dal sistema operativo in uso e dagli applicativi installati dall'utente.

Al raggiungimento del POI il software chiederà all'utente se è interessato a scansionare l'ambiente in realtà aumentata.

Realtà aumentata

L'obiettivo finale di questo sottosistema è quello di arricchire il quadro informativo facendo uso delle realtà aumentata. Un dispositivo mobile per usufruire di tale funzionalità dovrà essere necessariamente dotato di:

- Sistema di posizionamento globale (GPS);
- Magnetometro (bussola);
- Fotocamera.

Lo smartphone inquadra in tempo reale l'ambiente circostante, al mondo reale verranno successivamente sovrapposti dei livelli di contenuto forniti dai punti di interesse (POI) geolocalizzati, ed alcune informazioni verranno raccontate dal sintetizzatore.

Progettazione

Descrizione delle interfacce

Vediamo ora un tipico caso d'uso di Cicerone 2.0

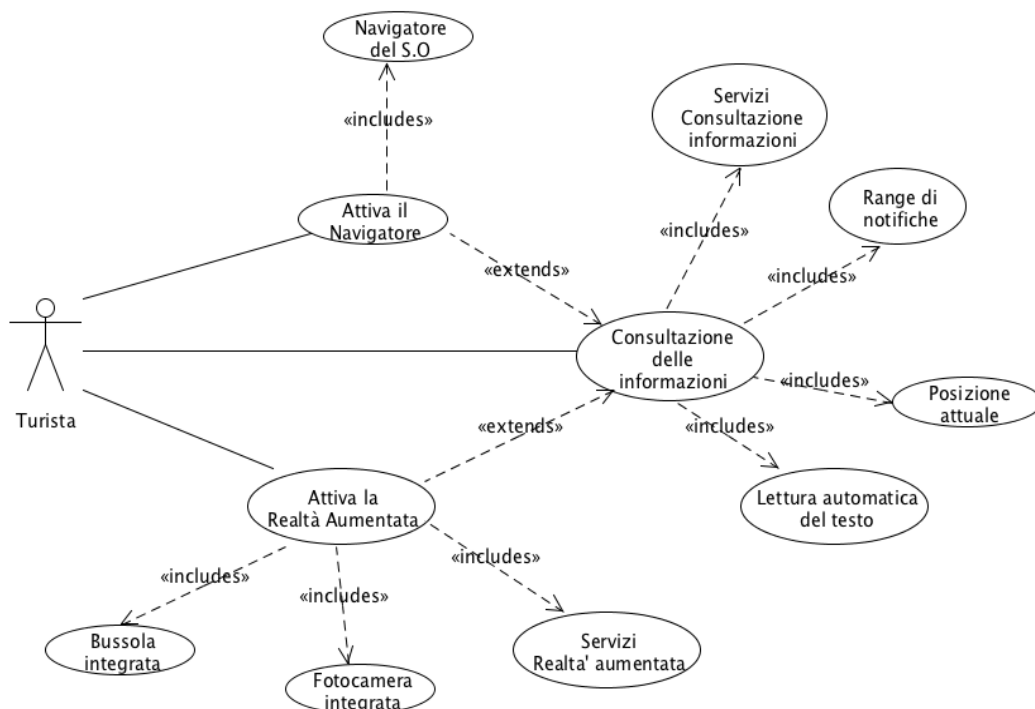


Figura 1 Diagramma casi d'uso

Caso d'uso "Consultazione delle informazioni":

Nome	Consultazione delle informazioni
Attori coinvolti	Turista
Scopo	Fornire all'utilizzatore tutte le informazioni disponibili per quel POI
Casi d'uso inclusi	Range di notifiche, Posizione attuale, Lettura automatica del testo
Casi d'uso estesi	Nessuno
Condizione d'ingresso	L'utente riceve una notifica che gli indica di essere in prossimità di un POI
Corso degli eventi	<ul style="list-style-type: none">▪ L'utente preme sulla notifica avviando Cicerone 2.0;▪ L'utente consulta il quadro completo delle informazioni (testo, audio, foto, video) in base alla sua posizione e in base al range di notifiche;▪ Può modificare il range di notifiche;▪ Può disabilitare la lettura automatica del testo.
Servizi	Servizio info_testo, Servizio info_video, Servizio info_foto, Servizio notifica, Servizio sincronizzazione
Condizione d'uscita	L'utente termina l'applicazione, oppure avvia un'altra funzione del software
Requisiti particolari	Sistema GPS

Caso d'uso "Attiva realtà aumentata":

Nome	Attiva realtà aumentata
Attori coinvolti	Turista
Scopo	Aumentare le informazioni disponibili nella realtà
Casi d'uso inclusi	Navigatore del S.O
Casi d'uso estesi	Consultazione delle informazioni
Condizione d'ingresso	L'utente deve raggiungere un POI
Corso degli eventi	<ul style="list-style-type: none">▪ L'utente inquadra un punto di interesse con la fotocamera dello smartphone;▪ L'utente può focalizzare il suo interesse su una particolare informazione (testo, immagini o video).
Condizione d'uscita	L'utente termina l'applicazione, oppure avvia un'altra funzione del software
Requisiti particolari	Fotocamera e bussola

Caso d'uso "Attiva il navigatore":

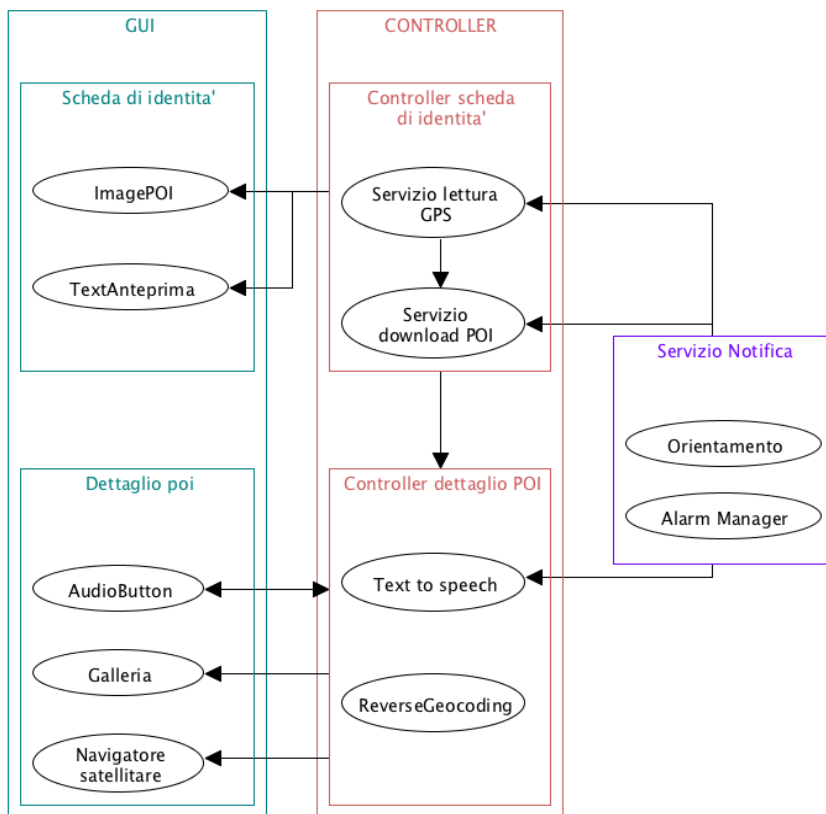
Nome	Attiva il navigatore
Attori coinvolti	Turista
Scopo	Guidare il turista al raggiungimento di un punto di interesse
Casi d'uso inclusi	Bussola integrata e fotocamera integrata
Casi d'uso estesi	Consultazione delle informazioni
Condizione d'ingresso	L'utente interagisce con l'ambiente circostante

Corso degli eventi	<ul style="list-style-type: none">▪ L'utente sarà guidato dal software di navigazione verso il punto di interesse;▪ Al raggiungimento del POI l'utente torna su Cicerone 2.0;▪ L'utente può scegliere se attivare la realtà aumentata.
Condizione d'uscita	L'utente termina l'applicazione, oppure avvia la modalità "Realtà aumentata"
Requisiti particolari	Software di navigazione, sistema GPS

Architettura

Prima di analizzare nel dettaglio l'architettura interna di Cicerone 2.0, è necessario avere una mappa concettuale rappresentante lo schema tecnico dell'applicativo suddiviso in sezioni:

Interfaccia grafica (GUI)



L'intera sezione grafica è stata interamente realizzata nel linguaggio XML sia per dispositivi Android che IOS.

Di seguito saranno mostrati gli elementi grafici utilizzati per lo splashscreen e l'icona di cicerone:



Figura 3 Splashscreen



Figura 4 Icona

Scheda di identità

Il concetto di scheda d'identità è basato sul fatto che un singolo POI deve essere presentato all'utente in modo da riassumere tutte le informazioni che lo rappresentano in un'unica scheda. L'home di Cicerone è composta da diverse schede, ognuna per ogni POI e con la seguente struttura:



Figura 5 Scheda di identità



Figura 6 Home page Cicerone2.0

Dettaglio POI

La schermata successiva alla scheda di identità è quella riguardante il dettaglio del POI che l'utente ha scelto di visionare. Qui sarà possibile accedere a un quadro completo d'informazioni, con la particolarità della narrazione del testo che compone la descrizione del POI.

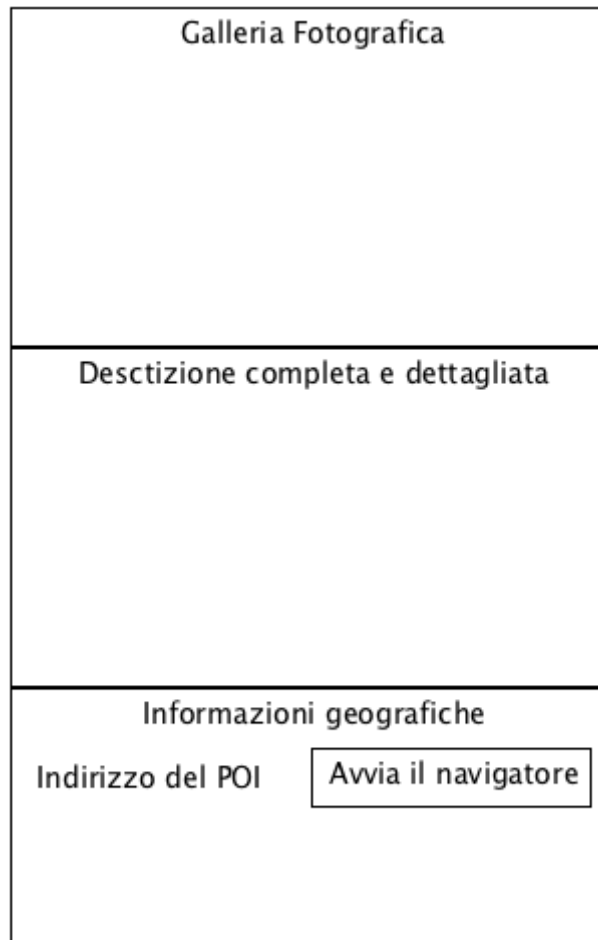


Figura 7 Progettazione Dettaglio POI

Il seguente screenshot mostra come sia possibile avere accesso ad un quadro completo di informazioni multimediali che caratterizzano il POI:



Figura 8

Ai fini di migliorare l'esperienza utente, se il POI che l'utente ha scelto di visionare non è nelle immediate vicinanze può sempre avviare il suo navigatore satellitare preferito, direttamente dall'applicazione in modo da poter essere guidato fino al suo raggiungimento.

Controller

I controller che stanno dietro alla GUI di Cicerone2.0 sono implementati seguendo le linee guida del MVC (model view controller).

MVC In informatica è un pattern architetturale molto diffuso nello sviluppo di sistemi software, in particolare nell'ambito della programmazione orientata agli oggetti, in grado di separare la logica di presentazione dei dati dalla logica di business.

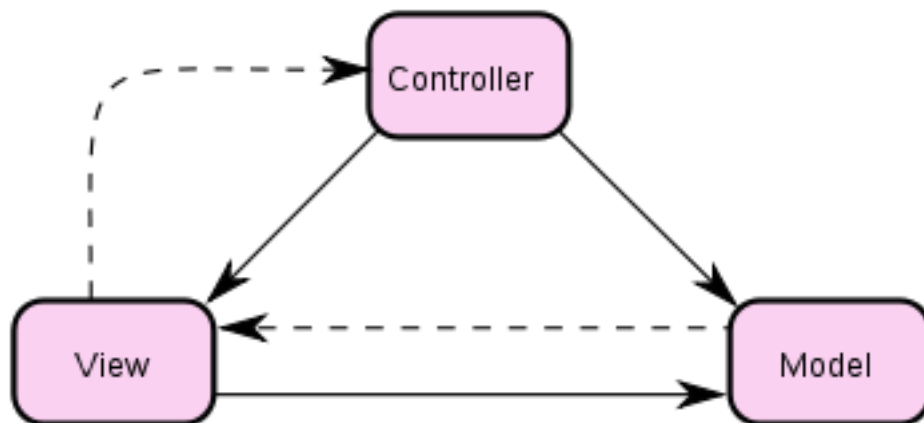


Figura 9 MVC (model view controller)

Controller scheda di identità

Il controller si occupa di gestire la vista “scheda d’identità” utilizzando principalmente 2 servizi fondamentali:

- Servizio download POI;
- Servizio lettura GPS.

Nella fase di precaricamento il controller della vista principale ha il compito di interrogare il sensore GPS, attraverso il corretto servizio, e una volta ricevute le coordinate saranno inoltrate al servizio di download POI.

Le informazioni ricevute da questo servizio saranno in formato JSON e saranno utilizzate per popolare l’home di Cicerone.

Servizio lettura GPS

Il servizio di lettura GPS consente di ottenere la posizione corrente dell’utente, nel momento in cui necessita all’applicazione. L’accuratezza della posizione dipende molto dal sensore che utilizza il device. Cicerone al fine di eseguire al meglio le sue funzioni, utilizza il sensore GPS in maniera ponderata, poiché ciò avrebbe un significativo impatto sui consumi energetici.

Il servizio nel momento in cui è attivato inizia a monitorare la posizione dell’utente con livello di accuratezza massimo, in altre parole, quel livello che consente di ottenere le coordinate combinando quelle di rete con quelle del GPS. Oltre a queste, il servizio ritorna anche, in un unico oggetto, la velocità con cui l’utente si sta spostando per eventuali controlli.

Durante la fase di monitoraggio del GPS, il servizio si arresta quando avviene una delle seguenti condizioni:

1. Viene richiamata la funzione di arresto del servizio,

2. Trascorrono più di 10 secondi dall'ultima coordinata: per evitare che attese prolungate in luoghi chiusi possano causare eccessivi consumi energetici.

```
var posizione = {  
    longitudine,  
    latitudine,  
    velocita  
}
```

Dato che il servizio è implementato in JavaScript, l'oggetto di ritorno è fornito attraverso una callback. L'oggetto ritornato è così strutturato:

Servizio download POI

In soldoni, gestisce tutte le comunicazioni di rete all'interno del client seguendo un protocollo ad oc per l'invio e la ricezione dei dati. La comunicazione avviene utilizzando il protocollo Http 1.1 eseguendo una richiesta di tipo GET.

- Specifiche Richiesta

Nome Campo	Descrizione	Esempio
latitudine	Latitudine utente	39,296174
longitudine	Longitudine utente	16,254686
lingua	Lingua da usare secondo ISO 639-1	it

- Specifiche Risposta: La risposta positiva conterrà nel body un oggetto JSON con le seguenti informazioni:

Nome Campo	Descrizione	Esempio
poi	Array dei poi	
nome	Nome del poi	Le tre colonne, di sacha sono
descrizione	Descrizione del poi	Tre colonne doriche ritagliate in lastre di marmo bianco di Sacha Sosno
tipologia	Tipologia del poi	Monumento
latitudine	Latitudine del poi	39,296174

longitudine	Longitudine del poi	16,254686
media	Array dei media	
mimeType	Mime type del documento multimediale	image/jpeg
uri	Url del documento multimediale	http://www.comune.cosenza.gov.it/moduli/output_immagin.php?id=4716

In caso di errore il servizio risponde con codice http 500.

Controller dettaglio POI

Il controller dettaglio POI si occupa della gestione della vista secondaria, ovvero quella che riguarda la descrizione dettagliata di uno dei POI scelti nella vista principale. Anche questo è implementato utilizzando il pattern MVC (model view controller).

Il controller riceve un oggetto in formato JSON dal controller principale e utilizza i dati contenuti all'interno per popolare la vista.

Nella fase di caricamento comunica alla vista i dati per :

- Costruzione della galleria fotografica, utilizzando i riferimenti in formato URL contenuti nell'oggetto JSON;
- Costruzione della view contenente il testo descrittivo del POI;
- Reverse geocoding delle coordinate per ottenere l'indirizzo del POI;
- Settaggio delle coordinate per il navigatore satellitare.

Finito il caricamento della vista il controller esegue il servizio di text to speech, raccontando il testo descrittivo del POI. In questo modo l'utente potrà comodamente ispezionare i contenuti del presenti all'interno della vista senza doversi soffermare nella lettura del testo.

Text to speech

Come descritto nel paragrafo precedente, Cicerone si avvale del servizio di Text To Speech per la riproduzione artificiale della voce umana e quindi per svolgere la funzione di storytelling.

Un sistema o motore di sintesi vocale è composto da due parti: una frontend e una backend.

La parte frontend si occupa della conversione del testo in simboli fonetici mentre la parte backend interpreta i simboli fonetici e li "legge", trasformandoli così in voce artificiale.

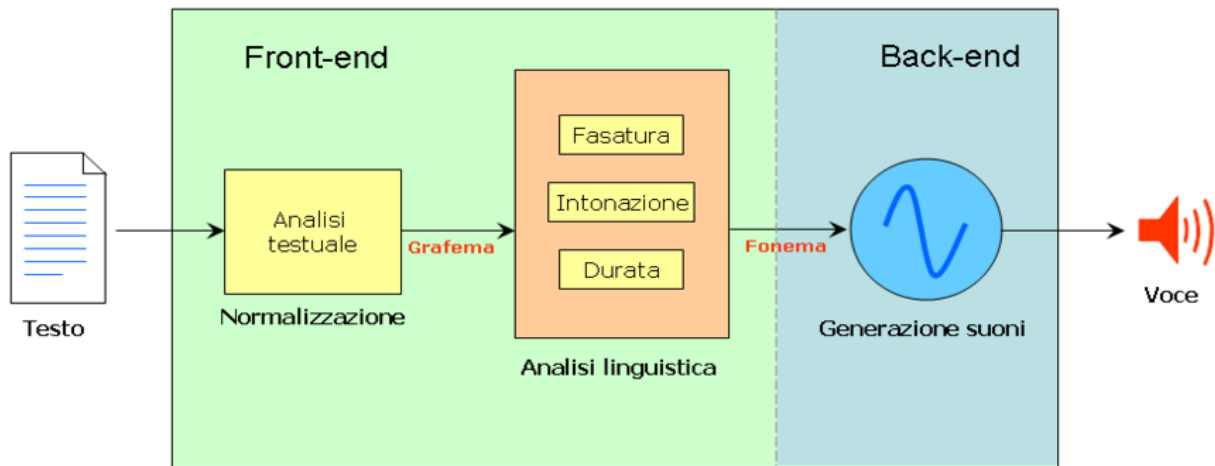


Figura 10 Schema di un sistema di sintesi vocale generico

Il frontend prevede due funzioni chiave: per prima cosa, viene eseguita un'analisi del testo scritto per convertire tutti i numeri, le sigle e le abbreviazioni in parole per esteso (es. il testo '2' viene convertito in 'due'). Questa fase di preelaborazione viene definita come normalizzazione o classificazione del testo (in inglese: tokenization). La seconda funzione consiste nel convertire ogni parola nei suoi corrispondenti simboli fonetici e nell'eseguire l'analisi linguistica del testo rielaborato, suddividendolo in unità prosodiche, ossia in proposizioni, frasi e periodi. Il processo di assegnazione della trascrizione fonetica alle parole è chiamato conversione da testo a fonema o da grafema a fonema (in inglese text to phoneme, TTP).

La trascrizione fonetica e le informazioni di prosodia combinate insieme costituiscono la rappresentazione linguistica simbolica che viene utilizzata dal backend per la conversione in suoni di tali informazioni ossia per il processo di sintesi vero e proprio.

Servizio notifica

Il core di Cicerone2.0 è rappresentato proprio dal servizio notifica. Esso infatti opera in background eseguendo le seguenti operazioni:

1. Ottiene la posizione dal sensore gps;
2. Comunica al server le coordinate del gps e la lingua;
3. Procedo al download delle informazioni dei POI;
4. Sceglie i POI presenti nel raggio di 40 metri;
5. Notifica i poi scelti al passaggio precedente.

Quando Cicerone notifica un POI, non si limita a farlo utilizzando solo il classico sistema di notifica, ma racconta letteralmente quello che in quel momento circonda l'utente.

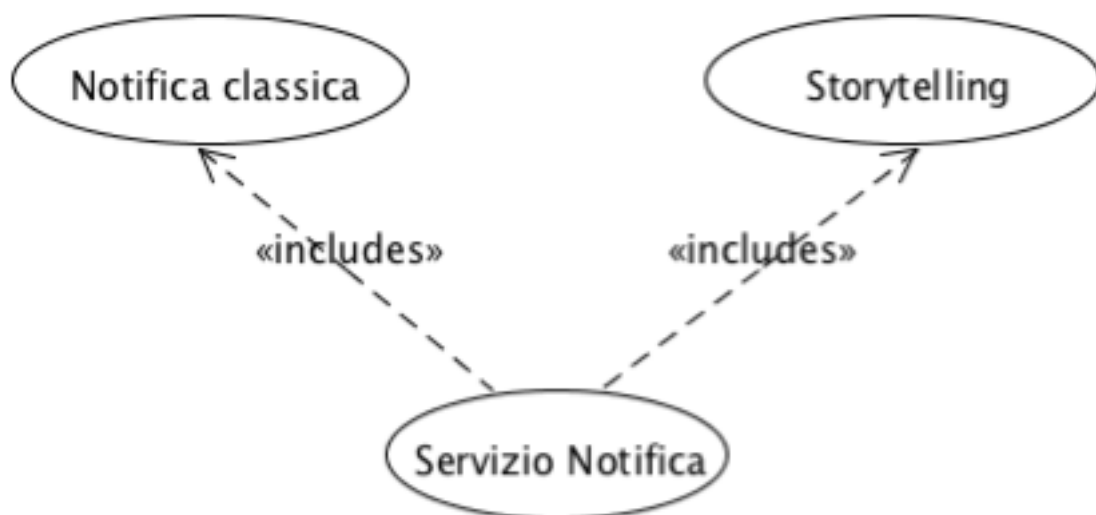


Figura 11 Servizio Notifica

Mentre viene notificato un POI, Cicerone inizia a raccontare all'utente quello che lo circonda in un raggio di 40 metri dalla sua posizione, indicando anche la direzione in cui bisogna guardare per individuare il punto di interesse.

Il servizio, come stato già anticipato sopra, lavora in background utilizzando 2 intervalli scelti in base alla distanza del POI più vicino all'utente:

- Intervallo di 1 minuto;
- Intervallo di 45 min.

Cicerone seleziona l'intervallo di 1 minuto quando nel raggio di 1 km sono presenti POI ancora da notificare, viceversa utilizza l'intervallo di 45 min. Quando un utente attende per troppo tempo in un luogo dove sono

presenti POI non ancora notificati, il controllo dell'area viene posticipato riducendo così il consumo energetico.

Alarm Manager

Il servizio notifica è interamente gestito dalla classe AlarmManager. Essa, da la possibilità di eseguire delle operazioni basate sul tempo e fuori dal ciclo di vita dell'applicazione.

Alarms presenta le seguenti caratteristiche:

- Permette di lanciare un operazione in un determinato momento / intervallo.
- Può essere usato in combinazione con un broadcast receivers per avviare servizi e completare altre operazioni.
- Opera fuori dal contesto dell'applicazione, così può essere usato per innescare eventi o azioni ogni volta che l'app non è avviata e ogni volta che il device è a riposo.
- Aiuta a minimizzare le risorse richieste dall'applicazione.

Una prima considerazione da fare nell'uso di un alarm si basa su quale tipologia deve essere usata. Ci sono due tipi di orologi per l'alarm: "elapsed real time" e "real time clock".

Elapsed real time, letteralmente tempo realmente trascorso, indica da quanto tempo il sistema è avviato. Utilizzato tipicamente quando è necessario eseguire l'alarm a particolari intervalli di tempo (per esempio, ogni mezz'ora).

Il real time clock invece dipende dall'ora locale corrente. Esso viene utilizzato nel momento in cui è necessario lanciare un evento in un particolare momento della giornata.

Nel caso del "servizio notifica" l'alarm manager è stato configurato utilizzando "elapsed real time", in modo da poter soddisfare le specifiche sopra citate.

È possibile configurare tale oggetto utilizzando i seguenti metodi messi a disposizione nel package android.app.AlarmManager:

- **cancel**(PendingIntent operation);
- **set**(int type, long triggerAtMillis, PendingIntent operation);
- **setExact**(int type, long triggerAtMillis, PendingIntent operation);
- **setRepeating**(int type, long triggerAtMillis, long intervalMillis, PendingIntent operation);
- **setTime**(long millis);
- **setTimeZone**(String timeZone);
- **setWindow**(int type, long windowStartMillis, long windowLengthMillis, PendingIntent operation).

Ricerca e integrazione di nuove funzioni utente

L'accesso ai contenuti di Cicerone inizialmente era possibile solo attraverso una lista, già generalizzata nei paragrafi precedenti, che forniva tutte le informazioni nelle vicinanze dell'utente.

Una fase di sperimentazione ha reso possibile l'individuazione di alcuni contesti in cui si avvertiva l'esigenza di fruire diversamente dei contenuti informativi. La ricerca subito dopo eseguita ha messo in luce come, in

un'applicazione del calibro di Cicerone, avvertisse la necessità di ampliare il ventaglio informativo rendendo possibile la visualizzazione dei POI su mappa.

L'integrazione della nuova funzione di fruizione dei contenuti, è stata progettata come una seconda interfaccia da affiancare a quella già presente ed equipaggiata con un modulo Google Maps Android API V2 per Titanium.

Esso è stato scelto poiché permette una gestione avanzata dei contenuti senza incidere sulle dimensioni dell'applicazione poiché il funzionamento del modulo stesso è strettamente legato alla rete, requisito essenziale per Cicerone.

L'accesso a tale funzione è stato reso possibile mediante l'inserimento di un'apposita voce nel menù.

Contestualmente sono state estese le funzioni dell'interfaccia principale dove, sempre grazie al nuovo modulo utilizzato per la visualizzazione della mappa, è stata progettata una nuova facility che rende possibile selezionare alcuni POI (oggetto d'interesse per l'utente) per l'individuazione e la visualizzazione su mappa.

Di seguito è esposto un piano di progettazione per nuova facility:

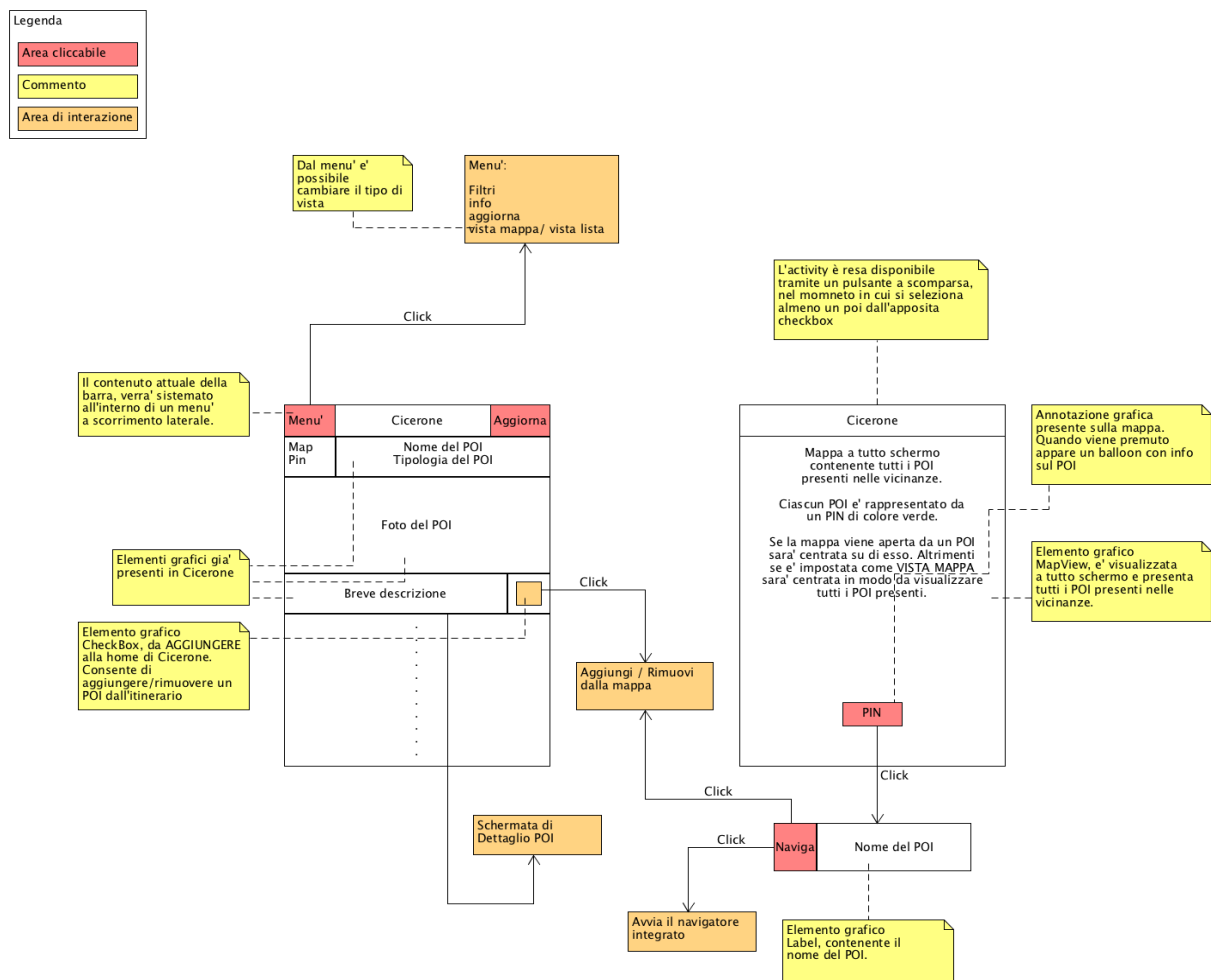


Figura 12 Schema progettuale

Dallo schema è possibile individuare quelli che saranno gli elementi che dovranno essere aggiunti all'interfaccia principale.

Nell'attuale interfaccia grafica è previsto l'inserimento di una CheckBox per ogni POI visualizzato, che consentirà di selezionarlo rendendolo disponibile in un'activity accessibile per mezzo di un popup a comparsa.

Il POI è identificato su mappa da un marker che se premuto darà la possibilità all'utente di abilitare il navigatore integrato, assegnandolo automaticamente come destinazione da raggiungere.

Ottimizzazione dell'interfaccia grafica

Giunti al punto di dover integrare nuove funzionalità all'interno di Cicerone, si è deciso di studiare un nuovo componente grafico che permettesse di offrire una gestione più elegante, intuitiva e meno invasiva, sia delle voci di menù che delle nuove componenti.

La ricerca ha portato a integrare il così detto **Navigation Drawer**, raffigurato nello schema sottostante:

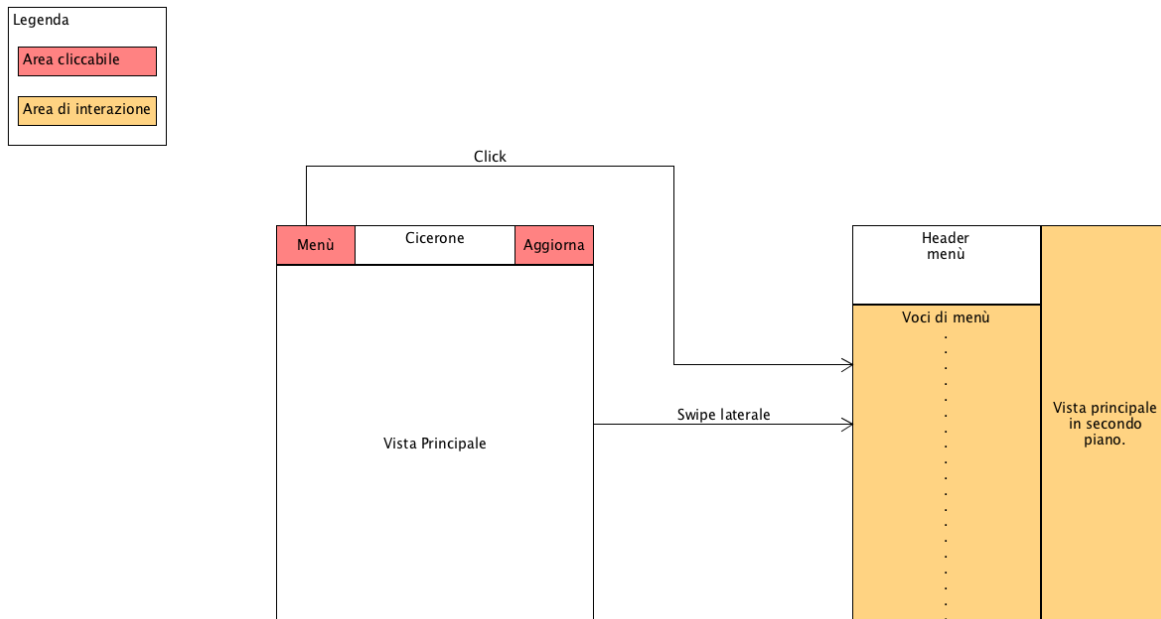


Figura 13 Schema Navigation Drawer

Da come si evince, è possibile accedere al menù laterale in due modi:

1. Click sul tasto menù,
2. Swipe laterale sulle viste.

A differenza del tasto menù, lo swipe laterale non sarà reso disponibile in tutte le viste bensì solo in quelle che non comportano la visualizzazione della mappa a tutto schermo. Questo perché per ovvie ragioni, non sarà possibile intercettare l'evento di swipe effettuato dall'utente.

Il Navigation Drawer permette inoltre una gestione ordinata e pulita delle voci già presenti con una notevole capacità di espansione senza dover apportare modifiche in futuro.

A titolo di abbellimento estetico, non mancano le animazioni grafiche che si attivano nel momento in cui il drawer viene aperto.

Integrazione dei contenuti di Cicerone su piattaforma IOS

Come anticipato nei paragrafi precedenti, l'applicativo di Cicerone è predisposto per l'integrazione anche su piattaforma IOS.

In prima battuta il framework di Titanium offre certamente un grosso aiuto nel mantenere la compatibilità tra le piattaforme, ma in ogni caso alcune accortezze sono decisive per il corretto funzionamento.

Come spiegato nei paragrafi precedenti per android, anche per IOS si è reso necessario affrontare uno studio adeguato per la creazione e la sperimentazione di moduli nativi che consentono l'integrazione di alcune funzionalità.

Altri studi invece, hanno portato a riorganizzare l'interfaccia grafica per consentire di mantenere la piena compatibilità tra le due piattaforme.

Analisi di una feature di Realtà Aumentata

Analizzando un po' le tendenze del momento si è visto come la Realtà Aumentata è diventata, sempre più, una tecnologia di massa in differenti settori dell'attività umana. Essa si basa essenzialmente sulla sovrapposizione di due livelli di presentazione: a un primo, reale, è sovrapposto un secondo che fornisce informazioni aggiuntive. Tutto ciò deve essere ovviamente elaborato in maniera ottimale, ovvero in modo tale che l'utente abbia la percezione di una singola scena nella quale il reale ed il virtuale sono due entità indistinguibili. Si viene così a creare una sorta di "mondo intermedio" tra la realtà e la virtualità, al cui interno si trova l'AR.

Per una possibile integrazione, diversi dispositivi sono stati oggetto di studio ponendo come obiettivo principale l'usabilità a costi contenuti.

Il primo caso di studio è stato affrontato sul **HMD** (head mounted display) visori composti da uno o due piccoli schermi, indossabili come un caschetto, che, hanno il vantaggio di fornire una sensazione di immersione nell'ambiente. In particolare, utilizzando un HMD a due schermi, è possibile ricreare la visione tridimensionale anche nell'AR creando così una percezione simile in tutto e per tutto a quella della realtà.



Figura 14 HMD

Questo tipo di tecnologia però, non soddisfa gli obiettivi di questa ricerca sia in termini di usabilità che di costi.

Anche i **Google glass** sono stati oggetto di studio. L'anima dei glass, è il pannello laterale touch, che attraverso sistemi di scorrimento e tap, permettono di controllare tutte le operazioni possibili.

- Risoluzione del display simile a uno smartphone medio,
- 16Gb di memoria interna,
- Conduzione ossea per l'audio,
- Pannello laterale sensibile al tocco,
- 24 ore di utilizzo in condizioni normali.



Figura 15 Google Glass

Essi però soddisfano l'obiettivo della ricerca solo in termini di comodità e usabilità, mentre continuano a rimanere alti i costi per l'utente finale ~2000\$.

In fine l'ultimo caso di studio riguarda l'AR su smartphone. La diffusione esponenziale di tablet e smartphone, che tramite il GPS integrato sono in grado di geolocalizzare l'ambiente circostante e tramite la telecamera di rappresentarlo sullo schermo, ha già dotato un alto numero di persone di strumenti perfetti per applicazioni AR su scala geografica: tutte le informazioni richieste possono essere recuperate e sovrainposte ai monumenti e ai luoghi d'interesse, limitando così al massimo i tempi e le difficoltà che comporterebbe ricavarle in altro modo.



Figura 16 AR su smartphone

In questo caso, per ovvi motivi, l'obiettivo della ricerca è soddisfatto sotto ogni aspetto, ragione per cui si è deciso di passare alla sperimentazione di una feature di AR, solo dopo un accurato studio riguardante lo stato dell'arte delle librerie di AR, che ha messo in luce l'SDK di Wikitude per via dell'esperienza nel settore e per i servizi offerti.

Wikitude SDK, infatti, è un prodotto tutto incluso che offre le seguenti feature già nel pacchetto base:

1. Image Recognition,
2. Geolocaliton AR,
3. 3D Augmentations.

Inoltre, oltre al fatto di essere multiplatforma, grazie alla predisposizione di un modulo l'SDK è compatibile anche con il framework di Titanium.