thunkable

Giulio Galesi, ISTI-CNR

Cos’è Thunkable?

Thunkable è una piattaforma web per lo sviluppo di app senza codice.

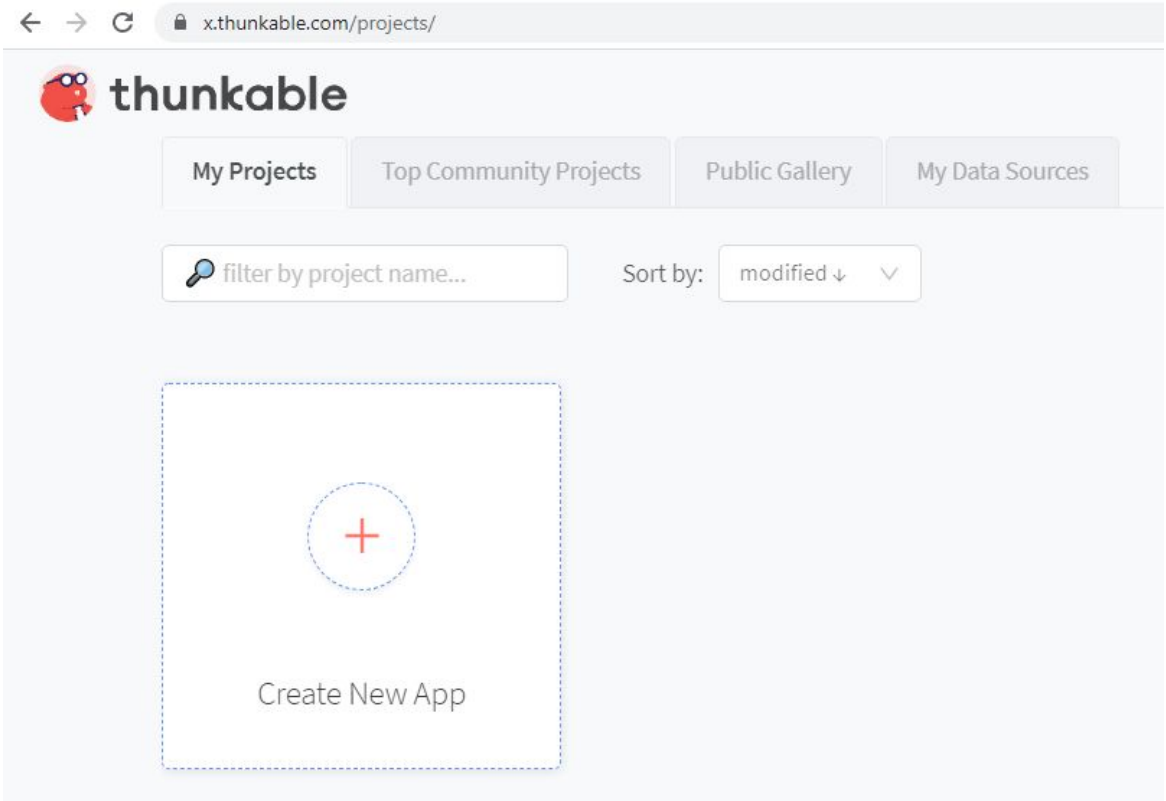
La versione gratuita consente a chiunque di progettare app, programmare potenti funzionalità con i blocchi e testare la propria app live sul proprio smartphone mentre con la versione Pro a pagamento (38$ mensili) è possibile caricare le app sul Google Play Store e sull'App Store di Apple.

E’ anche possibile creare web-app, ovvero app che non è necessario scaricare e a cui è possibile accedere direttamente online.

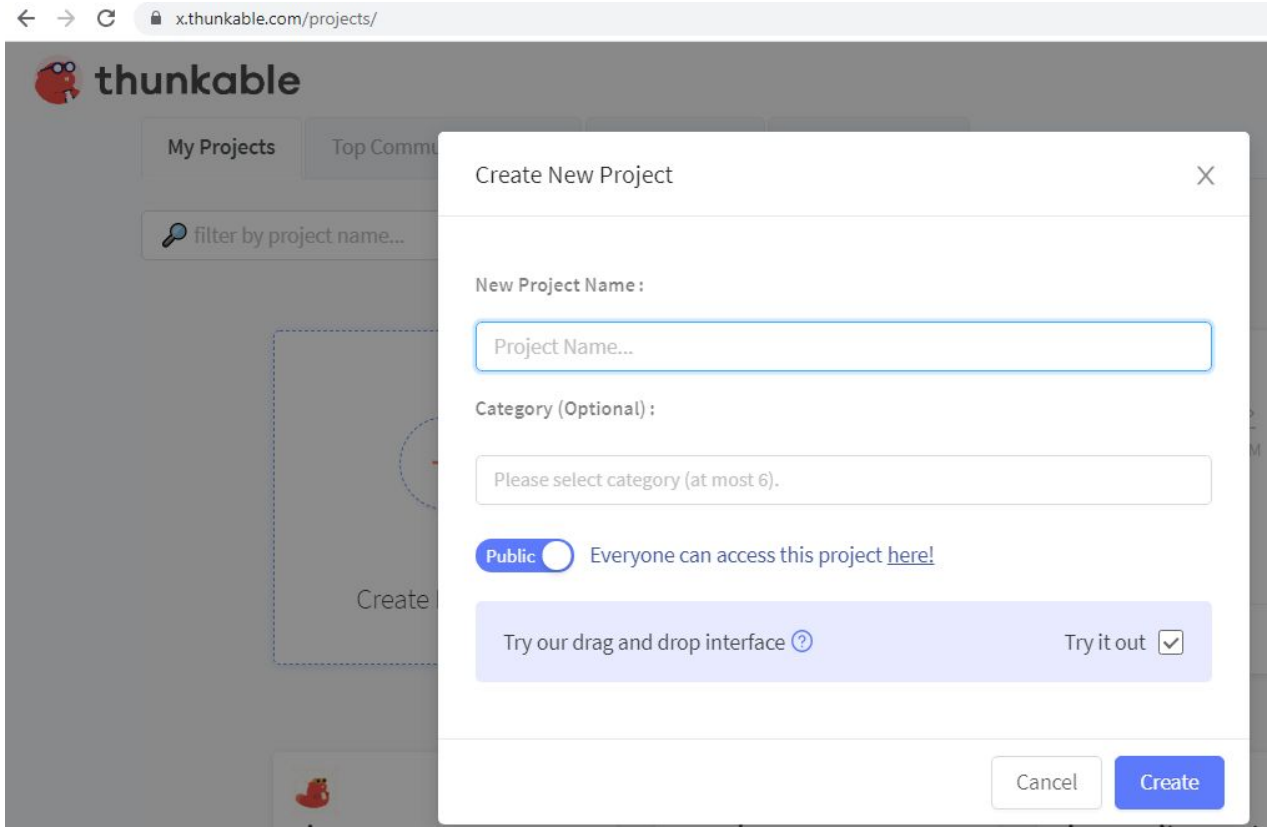
L’interfaccia è composta di una pagina di design in cui inserire le componenti dell’UI, e di una pagina di blocchi in stile puzzle per programmare il funzionamento della app.

Iniziamo ad usare Thunkable!

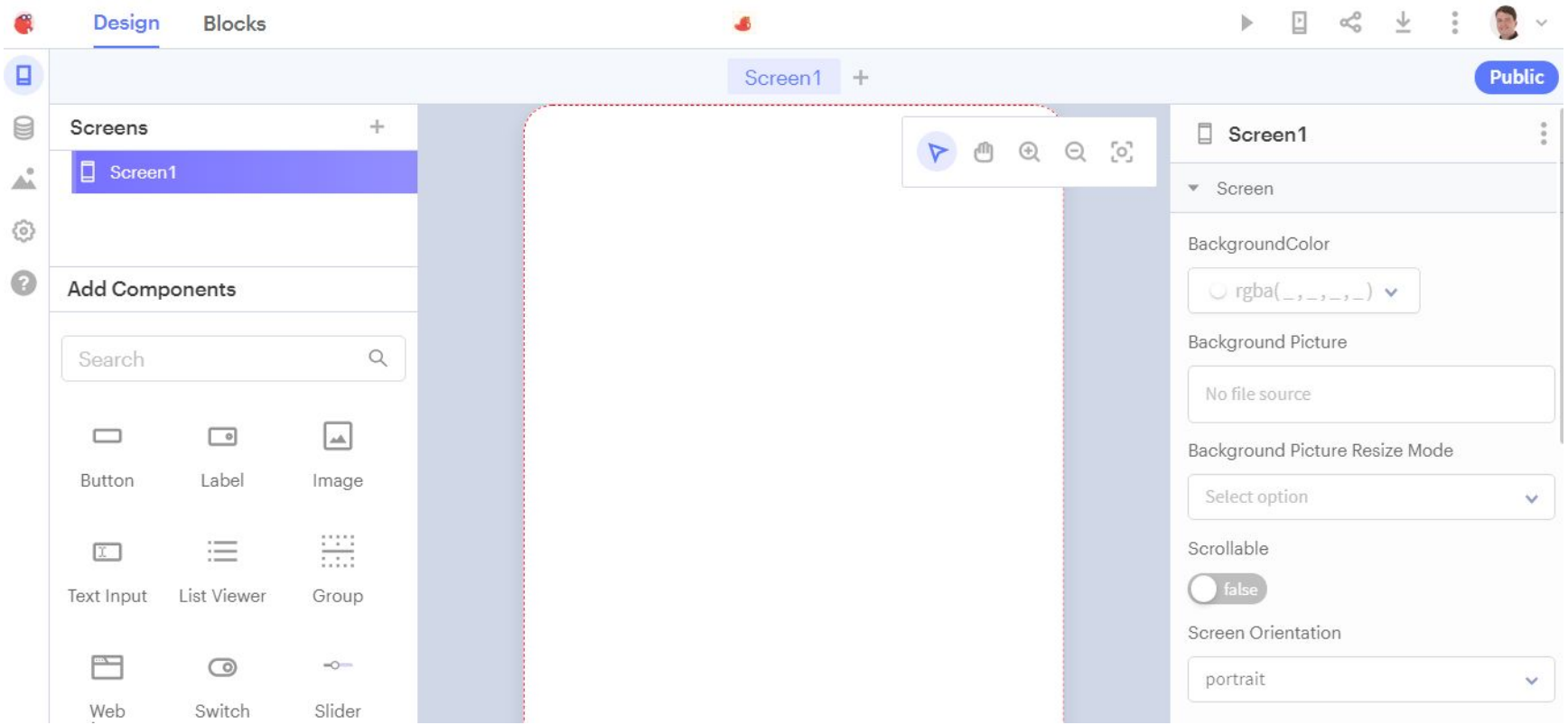
https://thunkable.com/



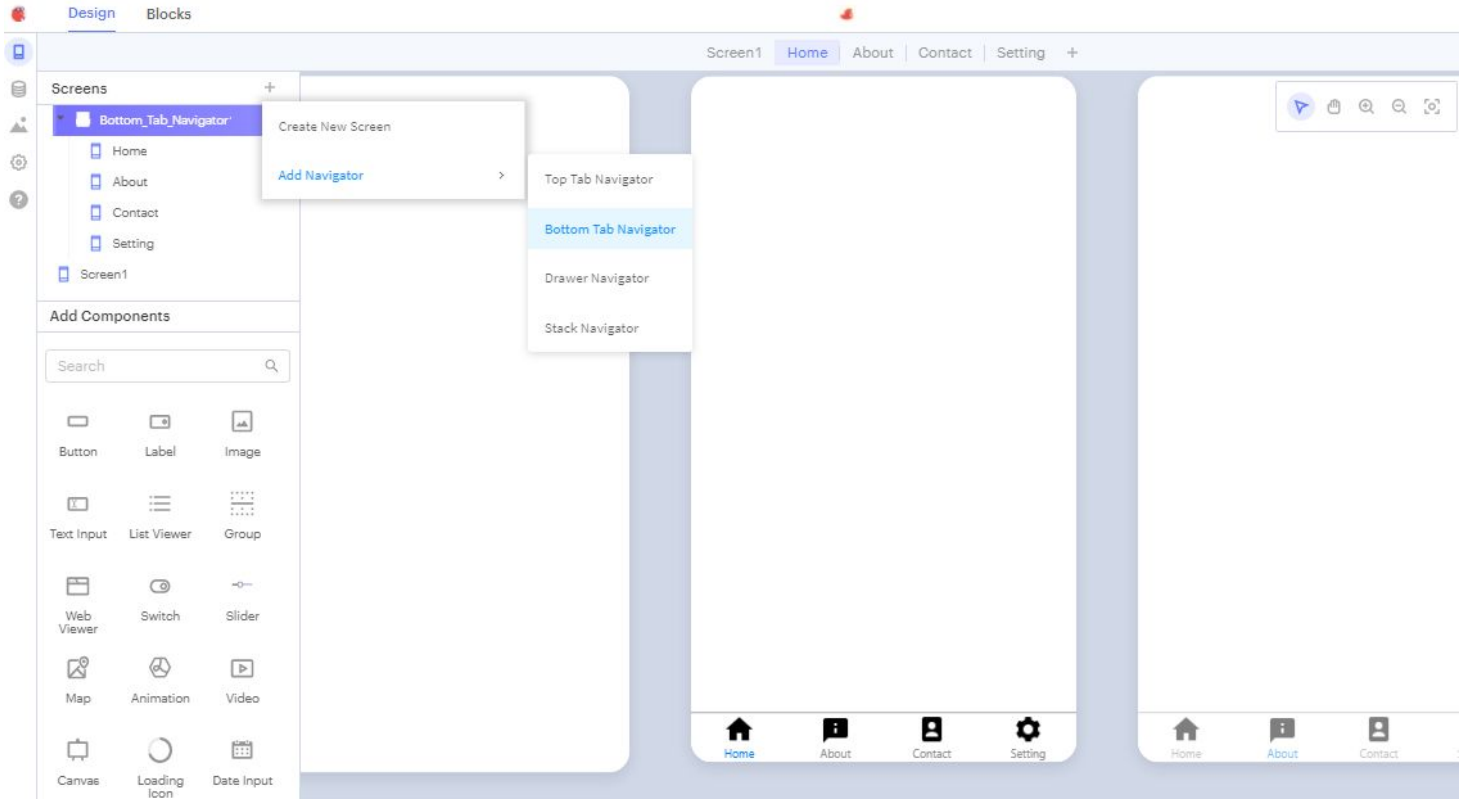
Creazione progetto usando l’interfaccia con drag and drop



Design Page - Screens

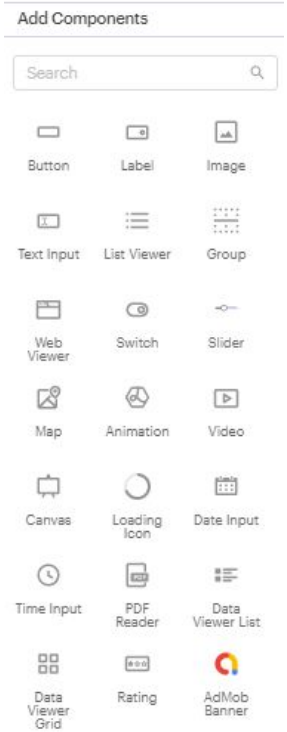


Design Page - Navigator



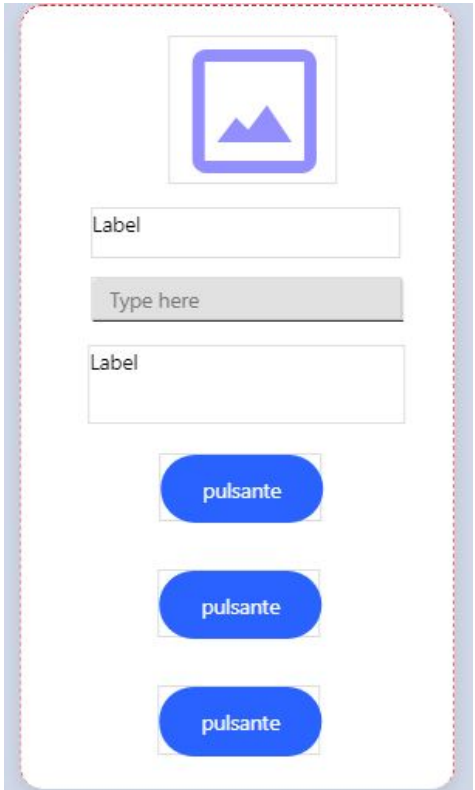
Design Page - Components

Si tratta delle componenti visibili dell'interfaccia utente:



- Button
- Label
- Image
- Text Input
- List Viewer
- Group
- Web Viewer
- Switch
- Slider
- Map
- Animation
- Video
- Canvas
- Loading Icon
- Date Input
- Time Input
- PDF Reader
- .. e molte altre

Proviamo a realizzare una interfaccia su Thunkable aggiungendo immagini, etichette e caselle di testo

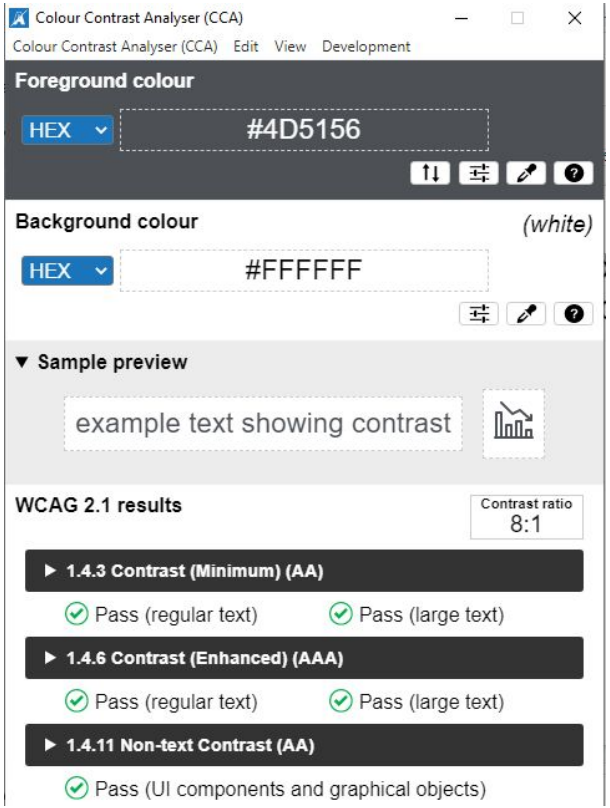


Su Thunkable manca la possibilità di inserire il testo alternativo per le immagini!

1) Se l'immagine che inseriamo è puramente decorativa andiamo a sovrapporre sopra l'immagine un label trasparente in cui inseriamo il testo alternativo che descrive l'immagine, utilizzando un font invisibile.
2) Se l'immagine deve poter essere cliccata, andiamo ad inserire un pulsante in cui impostiamo l'immagine come sfondo ed inseriamo come testo del pulsante la descrizione dell'immagine usando un font invisibile.

In questo modo la app creata sarà pienamente fruibile da tutti gli utenti.

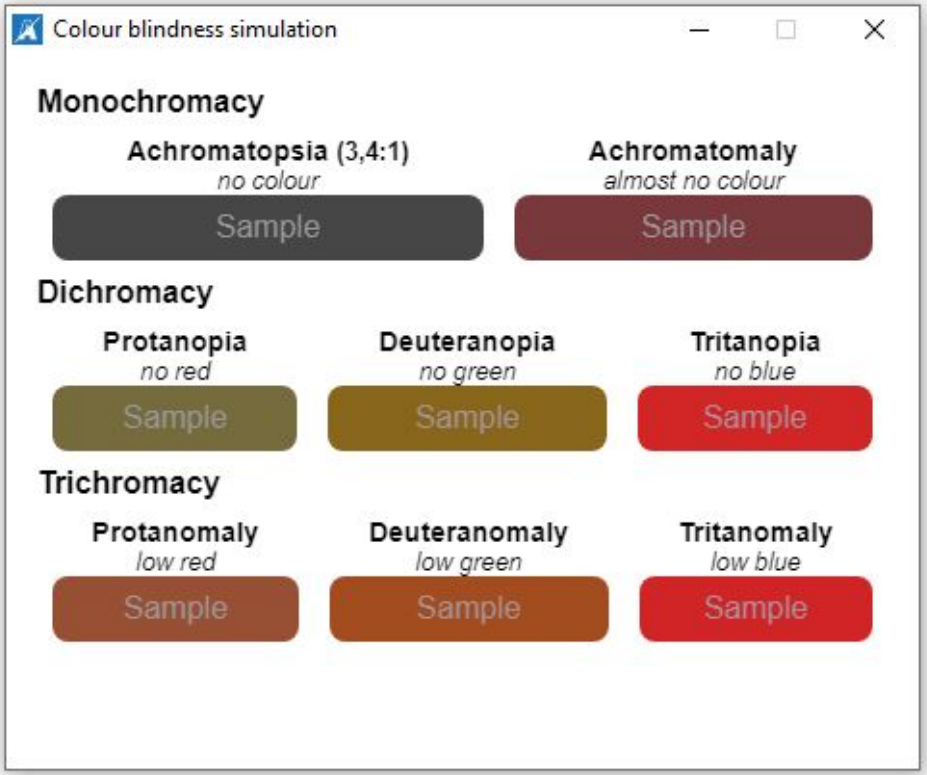
Verifica del contrasto con Colour Contrast Analyser

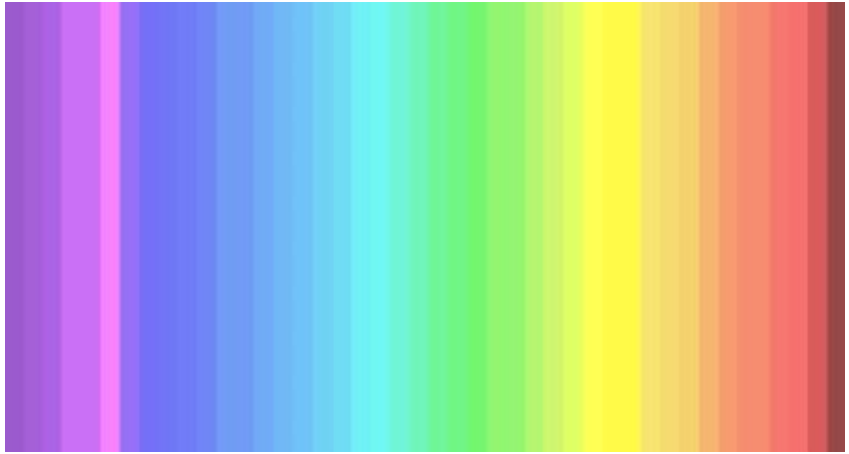


https://www.tpgi.com/color-contrast-checker/

- Uno strumento gratuito di controllo del contrasto cromatico che consente di determinare facilmente il rapporto di contrasto tra due colori permettendo di ottimizzare i contenuti, inclusi testo ed elementi visivi, per le persone con disabilità visive come daltonismo e ipovisione.
- Indicatori di conformità per le Linee guida per l'accessibilità dei contenuti Web 2.1 (WCAG 2.1)

Simulazione daltonismo con Colour Contrast Analyser





Se guardando l'immagine sopra (composta di 39 colonne con gradazioni di colore differente) una persona riesce a distinguere meno di 20 gradazioni di colore ha probabilmente un problema di daltonismo.

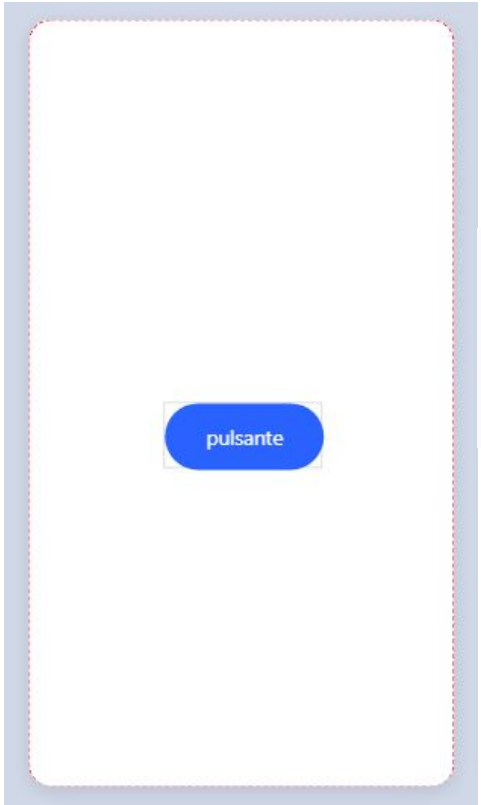
I blocchi associati alle componenti della UI

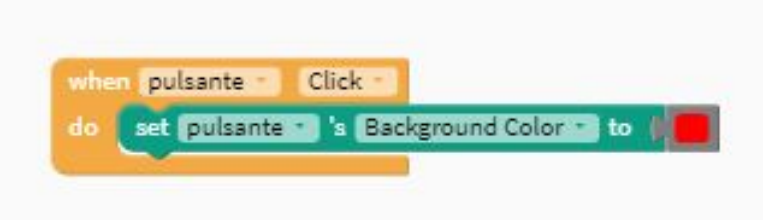
Dobbiamo considerare la app come una sequenza di eventi in cui sono coinvolte le componenti visibili dell'interfaccia utente nonchè le funzionalità non visibili ma che sono in esecuzione in background.
I blocchi relativi alle componenti inserite in una schermata non saranno usabili in altre schermate per cui bisogna utilizzare le variabili per salvare le informazioni tra le schermate (il funzionamento è semplice, lo vediamo tra breve).

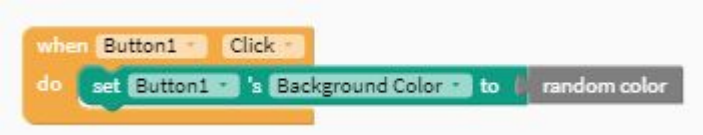
- Eventi

- Chiamate e Proprietà

Vediamo come funzionano i blocchi: creiamo un pulsante che quando viene cliccato cambia colore





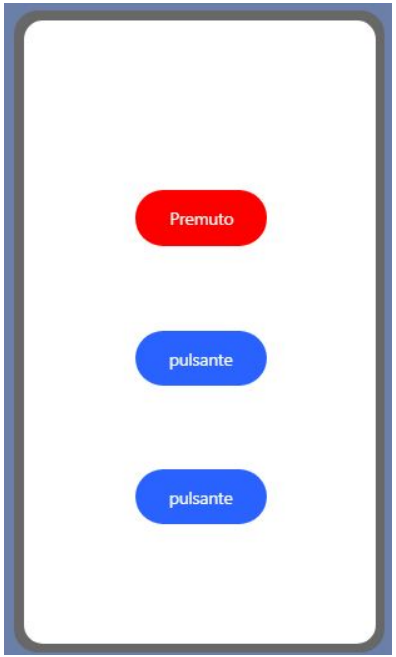


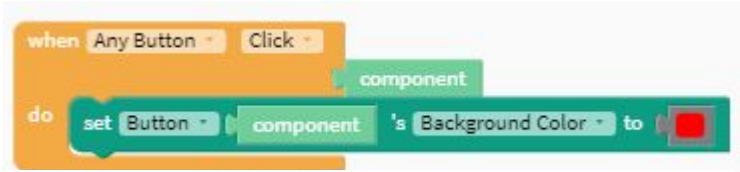
Aggiungiamo altri due pulsanti identici con la duplicazione

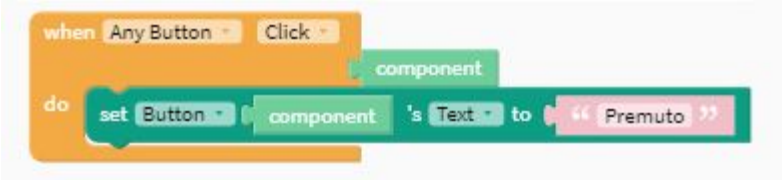
Usando la duplicazione delle componenti della UI, Thunkable ci avvisa che i relativi blocchi funzione non verranno duplicati quindi solo un pulsante cambierà colore una volta cliccato! Come procedere?

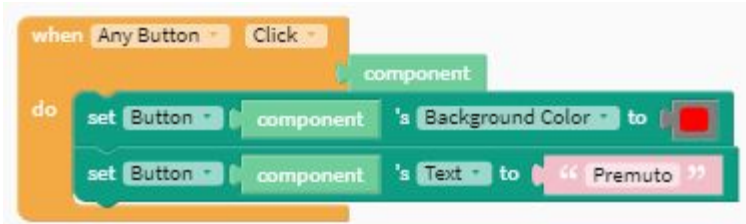
Usiamo il blocco avanzato “Any component”

Possiamo scegliere di modificare il testo del pulsante oltre che lo sfondo, andando a duplicare i blocchi e modificare la proprietà Text anzichè Background Color. Prestiamo attenzione a semplificare i blocchi quando possibile.



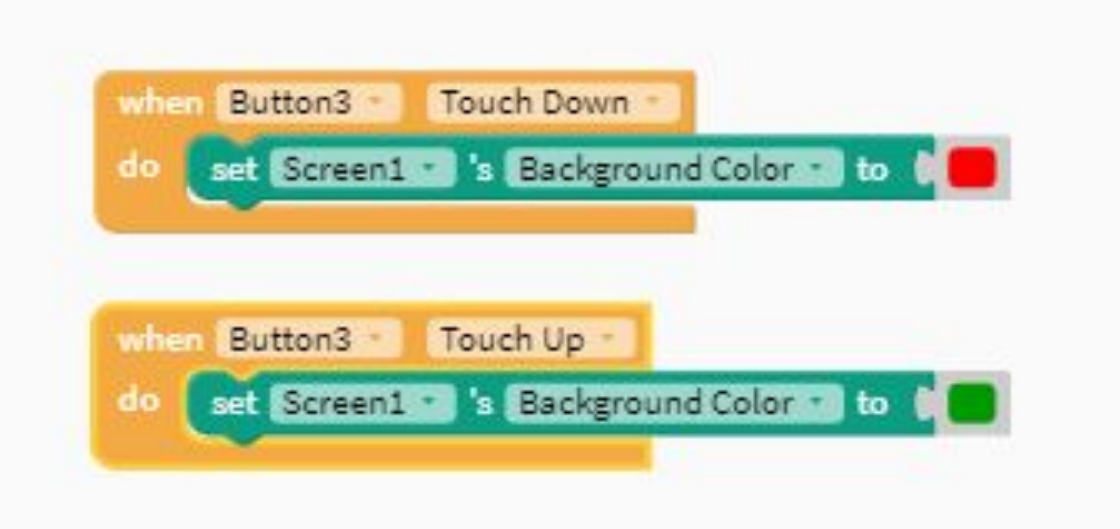






Usiamo le caratteristiche touch down e touch up del pulsante

Possiamo impostare un pulsante perchè effettui il cambio del colore di sfondo della schermata (o ad altro elemento della UI) solo mentre è premuto.



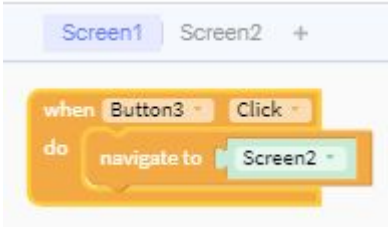
I blocchi principali (core) per programmare una app

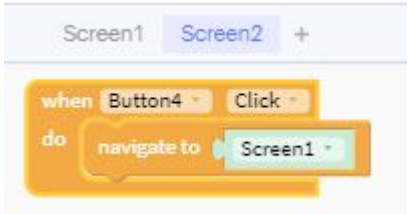
Abbiamo 10 categorie di blocchi principali che ci consentono di impostare regole e programmare la nostra app:

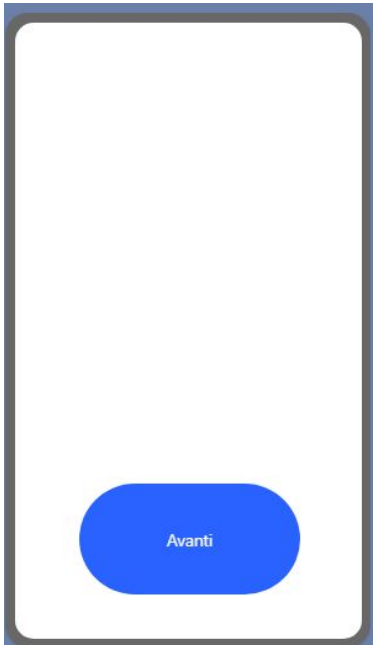
- Controllo (navigate, when->do, if->do, test, wait, loop,..)
- Logica (confronto, and-or, not, valori true,false,null)
- Matematica (operazioni con i numeri, valori random,..)
- Testo (impostare valori di testo e analisi testuali, join, contain, replace,..)
- Liste (creazione e modifica liste, insert, set,..)
- Colori (impostare colore da tavolozza o con valori rgb, hsv o esadecimali)
- Dispositivo (informazioni sul sistema operativo, dimensioni schermo,...)
- Oggetti (per gestire Web API che inviano dati in formato oggetto)
- Variabili (ove archiviare valori che poi verranno richiamati senza duplicarli)
- Funzioni

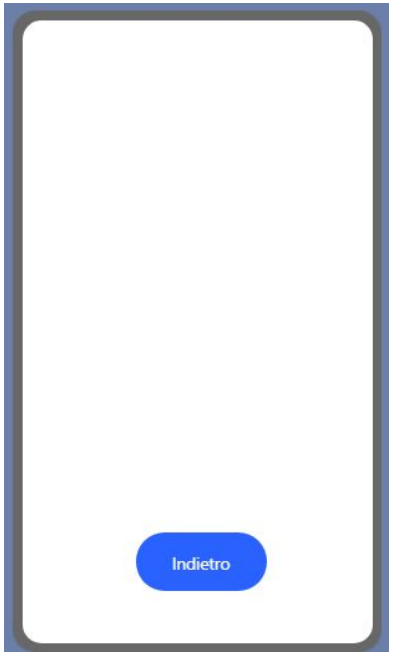
Blocco navigate

Consente di spostarci da uno screen all'altro, vediamo un esempio semplice creando due schermate ciascuna con un pulsante (schermata 1 con pulsante avanti e schermata 2 con pulsante indietro)



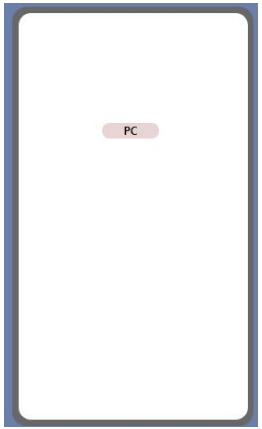




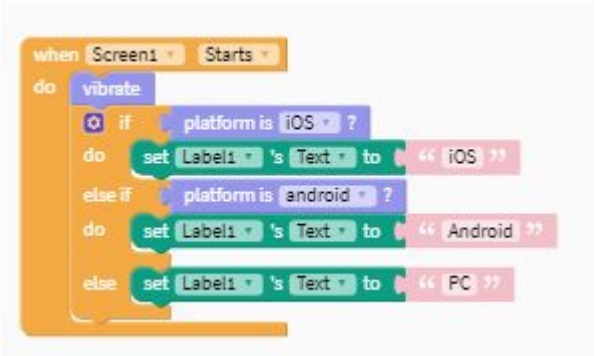


Blocchi if-do-else ed else-if

Il blocco if valuta la condizione e in base alla risposta (vero o falso) esegue l'istruzione associata a "do", altrimenti (else) esegue l'istruzione associata al blocco successivo. E' anche possibile annidare uno o più blocchi if all'interno di un blocco if o usare il blocco "else if". Creaiamo una app che emetta una vibrazione quando si avvia e che ci indichi semplicemente quale piattaforma stiamo utilizzando.

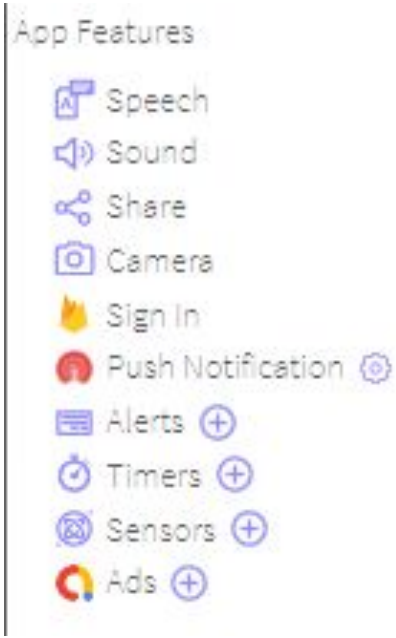


Android



Le funzionalità (App features)

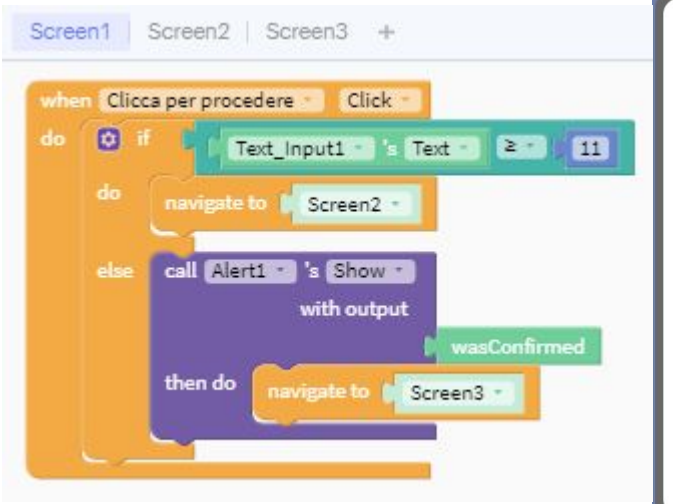
Abbiamo 10 categorie di funzionalità che ci consentono di usare blocchi con caratteristiche differenti:

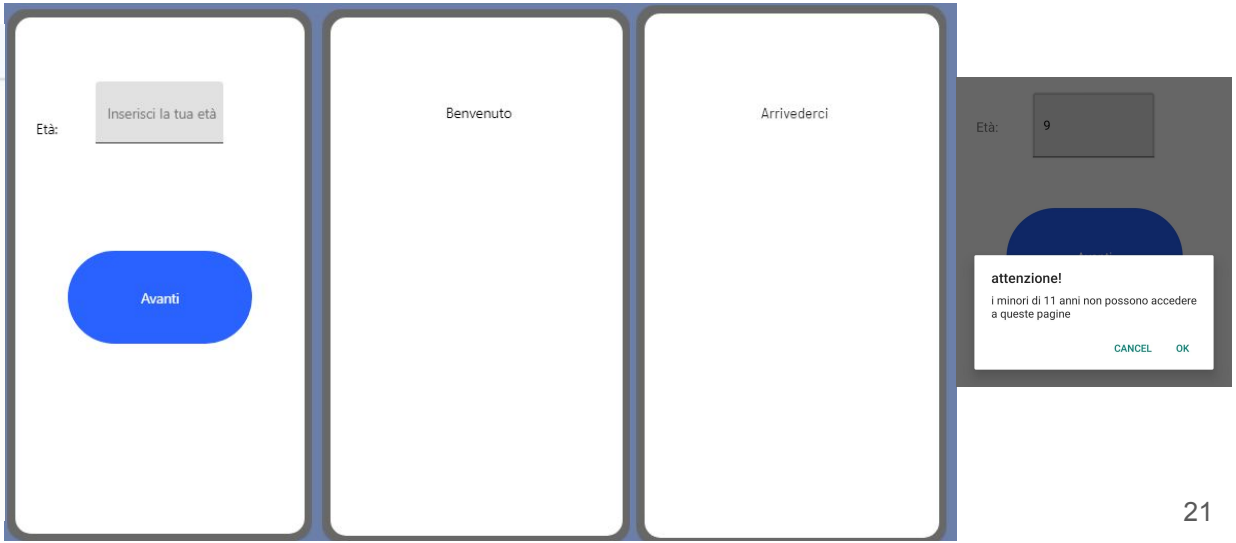


- Speech
- Sound
- Share
- Camera
- Sign In
- Push Notification
- Alerts
- Timers
- Sensors
- Ads

Esempio di app con alert

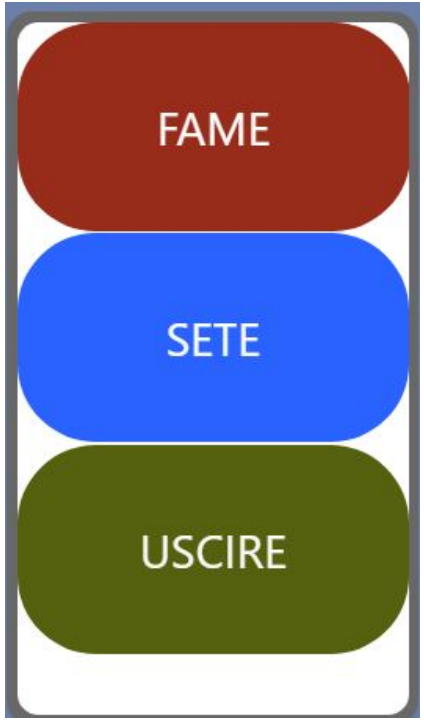
Proviamo a creare una app che non consenta a chi ha meno di 11 anni di usare la app. Possiamo chiedere l'età nella prima schermata e se questa è maggiore o uguale a 11 anni il sistema consentirà di passare alla schermata successiva, altrimenti aprirà un alert che avvisa della limitazione attiva.





Proviamo a realizzare una semplice app che riproduce suoni...

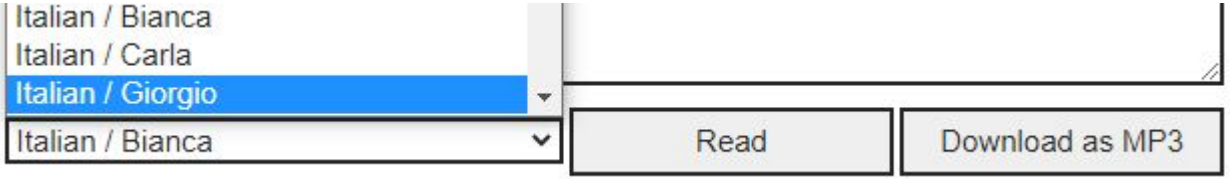
Possiamo pensare ad una app che verrà usata da una persona anziana che non riesce a comunicare con la propria voce.
All' avvio della app possiamo inserire un suono di benvenuto.
Dalla pagina di design inseriamo solo 3 pulsanti, ciascuno dei quali, una volta cliccato, riprodurrà un messaggio vocale differente.
Andiamo ad inserire 3 file audio in formato mp3 in Assets - Media Files e poi componiamo i blocchi per far funzionare la app in pochi secondi.



Free Text-To-Speech (TTS)

ttsmp3.com (https://ttsmp3.com/) consente di scaricare i file mp3 parlati dei testi che andiamo a digitare nella casella (fino a 3000 caratteri per volta).

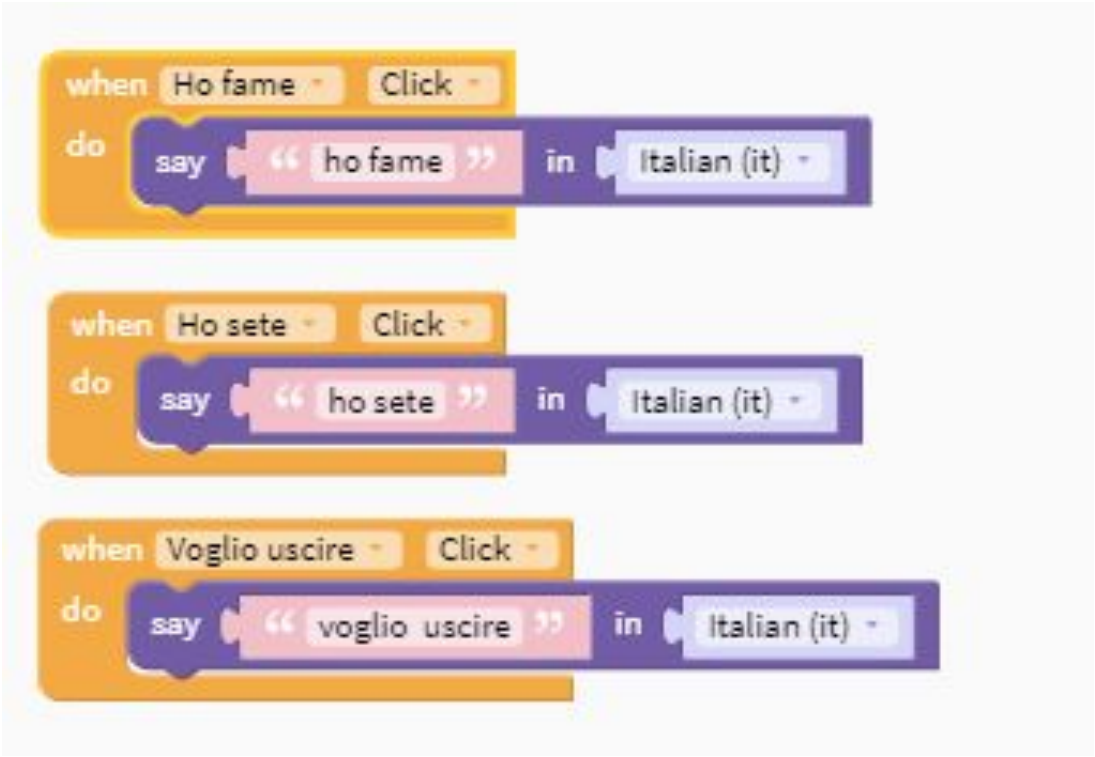
Una volta digitato il testo da trasformare in parlato e scelta la lingua/voce da utilizzare, basterà premere il tasto ”Download as MP3” per ottenere il file.



Anche Thunkable ha un suo modulo TTS

Modifichiamo la app realizzata usando i blocchi del text-to-speech di Thunkable anzichè i file audio mp3.

Funziona correttamente ma notiamo che la qualità della pronuncia è peggiore.

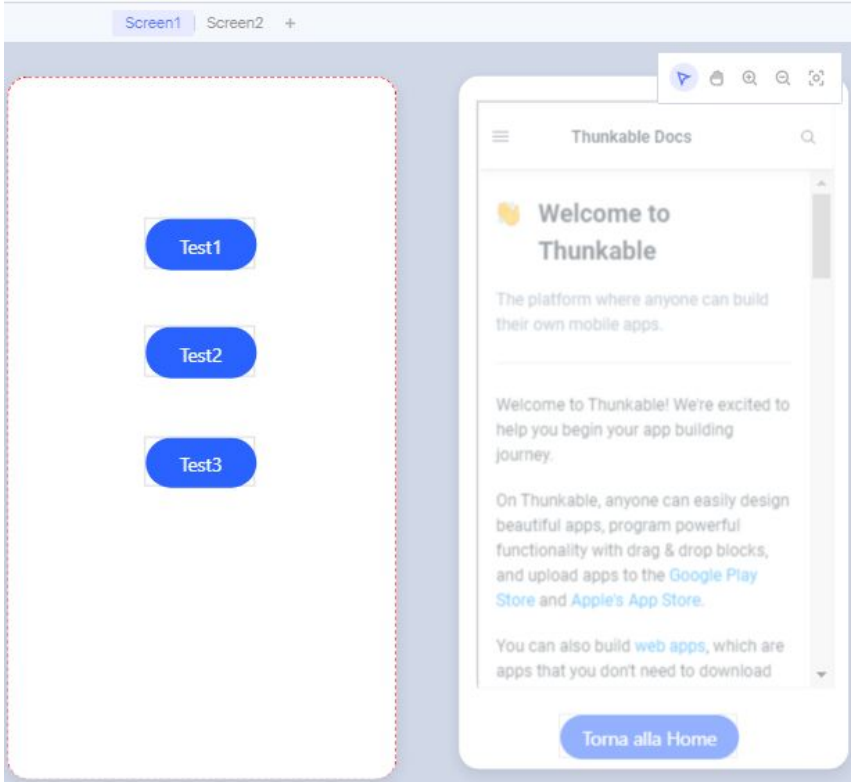


Utilizziamo le variabili in Thunkable

Esempio di una app che ci consente di visualizzare siti web con url predefiniti

Nella pagina di design:

1. creiamo due schermate
2. Nella prima inseriamo tre pulsanti
3. Nella seconda inseriamo un pulsante e un componente web viewer



Variabili in Thunkable

Nella pagina dei blocchi di Screen1 inizializziamo la variabile url e attiviamo i blocchi eventi dei tre pulsanti associando un url diverso a ciascun pulsante.

Nella pagina dei blocchi di Screen2 attiviamo il blocco eventi del visualizzatore web associando al suo attributo URL la variabile url.