Energia
Territorio
Ambiente

Progetto Abitare sostenibile

Realizzato grazie al contributo
della Fondazione CRT -
Torino

Pensiero computazionale

Pensiero computazionale

Apprendimento cognitivo

L'apprendimento tradizionale impiega quattro importanti strategie per promuovere la competenza

Modelling: l'apprendista osserva ed imita il maestro che dimostra come fare;
Coaching: il maestro assiste continuamente secondo le necessità: dirige l'attenzione su un aspetto, dà feedback, agevola il lavoro
Scaffolding: è un aspetto particolare del coaching: il maestro fornisce un appoggio all'apprendista, uno stimolo, pre-imposta il lavoro, ecc.
Fading: il maestro elimina gradualmente il supporto, in modo da dare a chi apprende uno spazio progressivamente maggiore di responsabilità

L'apprendistato cognitivo pone maggiore attenzione alla dimensione metacognitiva, agli aspetti del controllo, ed alla variazione dei contesti di applicazione. Si introducono altre strategie:

Articolazione: si incoraggiano gli studenti a verbalizzare la loro esperienza
Riflessione: si favorisce il confronto con gli esperti
Esplorazione: si spinge a porre e risolvere problemi in forma nuova.

Pensiero computazionale

Che cos'è il pensiero computazionale?

Non esiste una definizione universalmente condivisa.

Quella che sembra mettere d'accordo il maggior numero di esperti è la definizione formulata dalla dottoressa Jeannette Wing, direttrice del Dipartimento di Informatica della Carnegie Mellon University, secondo cui "è il processo mentale che sta alla base della formulazione dei problemi e delle loro soluzioni così che le soluzioni siano rappresentate in una forma che può essere implementata in maniera efficace da un elaboratore di informazioni sia esso umano o artificiale".

Ovvero è lo sforzo che un individuo deve mettere in atto per fornire a un altro individuo o macchina tutte e sole le "istruzioni" necessarie affinché questi eseguendole siano in grado di portare a termine il compito dato.

Pensiero computazionale

Sono in molti oggi a credere che il pensiero computazionale costituisca la **quarta abilità di base** oltre a saper leggere, scrivere e fare di calcolo.

Papert, il padre della teoria dell'apprendimento nota come **costruzionismo,** sostiene che **la mente umana per poter imparare bene ha bisogno di creare artefatti**, ovvero rappresentazioni reali del mondo con cui interagisce.

E il computer, secondo Papert, è un ottimo strumento didattico poiché, grazie alla programmazione, può creare questi artefatti.

Pensiero computazionale

Perché è importante il pensiero computazionale?

Nonostante il termine “computazionale” possa indurre a pensare che il pensiero computazionale sia un’abilità utile solo a chi ha fatto dell’informatica la propria professione, si tratta di una **skill fondamentale che tutti dovrebbero possedere**, in particolare quei giovani che desiderano non farsi sfuggire le opportunità che il futuro porrà loro dinanzi nei prossimi anni. È ormai universalmente riconosciuto che per riuscire bene nel proprio futuro professionale i giovani dovranno “imparare a imparare” e non limitarsi a fornire risposte preconfezionate: in questa direzione si muovono le raccomandazioni dell’Unione Europea in materia di istruzione che sono state recepite dal MIUR anche con l’introduzione della programmazione nelle scuole a partire dalla primaria

Ciò che i nostri studenti imparano oggi, tra cinque anni sarà probabilmente obsoleto: questo vale in maniera particolare per le discipline tecniche, ma anche per le altre discipline. Inoltre, è davvero difficile pensare che gli studenti di oggi, che domani troveranno un lavoro, saranno in grado di mantenerlo fino alla fine della propria carriera lavorativa, come accadeva in passato ai loro padri o nonni: è molto più probabile che i giovani saranno chiamati a cambiare lavoro piuttosto frequentemente, imparando a destreggiarsi anche nell’ambito di differenti discipline.

Anche per quanto riguarda le nuove professioni: i lavori oggi più richiesti solo sei anni fa non esistevano neppure.

Proviamo a immaginare quante saranno le discipline che nasceranno nei prossimi anni senza che neppure esistano dei percorsi strutturati per poterle accostare.

La vera sfida per i nostri giovani sarà di porsi in un atteggiamento di lifelong learning e acquisire quelle abilità che consentiranno loro di sviluppare un’attitudine mentale utile ad affrontare problemi di ogni ordine e grado.

Pensiero computazionale

Il pensiero computazionale è la capacità di elaborare concetti e problemi in forma algoritmica e nello specifico:

Saper analizzare un problema

Saper individuare i dati del problema

Saper identificare, analizzare, implementare e verificare le possibili soluzioni con un'efficace ed efficiente combinazione di passi e risorse (avendo come obiettivo la ricerca della soluzione migliore secondo tali criteri)

Saper codificare, rappresentare i dati del problema tramite opportune astrazioni

Saper organizzare i dati in base a criteri logici

Saper automatizzare la risoluzione del problema definendo una soluzione algoritmica, consistente in una sequenza accuratamente descritta di passi, ognuno dei quali appartenente ad un catalogo ben definito di operazioni di base

Saper generalizzare un problema per poterlo trasferire ad un ampio spettro di altri problemi

Saper operare per artefatti e modelli che si possono utilizzare in diversi contesti.

Pensiero computazionale

Qual è il legame tra computer, informatica e pensiero computazionale?

Così come l'invenzione della stampa ha facilitato la diffusione dell'alfabetizzazione, così oggi la programmazione e i computer facilitano l'acquisizione e la diffusione del pensiero computazionale.

Il pensiero computazionale prende a prestito concetti e strumenti propri dell'informatica per trovare soluzioni innovative e creative ai problemi di ogni giorno.

Questo non significa che gli esseri umani devono imparare a pensare come i computer: il pensiero computazionale è il modo in cui gli esseri umani insegnano ai computer a risolvere i problemi e non viceversa. **I computer sono stupidi e noiosi** e solo grazie agli esseri umani possono diventare strumenti utili e interessanti: l'unico limite a quello che i computer sono in grado di fare è costituito dalla nostra **creatività e immaginazione**.

I processi mentali tipici del pensiero computazionale sono favoriti dall'approccio alla risoluzione dei problemi che viene messo in atto da coloro che sviluppano programmi per il computer: in altre parole il pensiero computazionale è quello che adotta un informatico quando affronta un problema. Esercitare il pensiero computazionale significa quindi molto di più che saper scrivere righe di codice.

Pensiero computazionale

Pratiche di pensiero computazionale:

Essere incrementali: la progettazione è un processo adattativo dove la pianificazione può cambiare man mano che ci si avvicina alla soluzione del problema.

Testare e debuggare: individuare problemi ed errori e correggerli.

Riusare (pattern recognition): riconoscere come alcune parti di soluzione possono essere riusate nella stesso modo o riapplicate a problemi simili.

Remixare (copiare per migliorare): grazie alla rete e all'ampia disponibilità di lavori di altri autori, è possibile prendere spunto da idee e codice per costruire cose più complesse di quelle che si sarebbero potute realizzare per conto proprio, dando un'ulteriore spinta alla propria creatività.

Astrarre: è il processo di riduzione della complessità, per far emergere l'idea principale mantenendo solo alcuni aspetti e tralasciandone altri.

Modularizzare (scomporre): è il processo che consente di scomporre un problema complesso in problemi più semplici, per cui risolvendo i problemi più semplici si risolve anche il problema complesso.

Attitudini di pensiero computazionale:

Esprimere se stessi: una persona dotata di pensiero computazionale vede nella tecnologia uno strumento per esprimere se stessi, la propria creatività e dire qualcosa di sé agli altri.

Essere connessi: saper comunicare e lavorare con gli altri per raggiungere un obiettivo o una soluzione condivisa.

Porre domande: saper sviluppare una mente vigile grazie alla quale è sempre viva la domanda di come un oggetto incontrato nel mondo reale possa funzionare.

Pensiero computazionale

Concetti di pensiero computazionale:

Sequenza: un’attività può essere espressa attraverso una serie consecutiva di singoli step o istruzioni.

Ciclo: è un meccanismo per eseguire più volte la medesima sequenza in maniera iterativa.

Evento: il verificarsi di un’azione causa lo scatenarsi di un’altra azione.

Parallelismo: significa eseguire sequenze di istruzioni differenti allo stesso tempo.

Condizione: è la possibilità di prendere decisioni sulla base del verificarsi di determinate situazioni.

Operatore: fornisce supporto per la manipolazione di numeri e stringhe di caratteri.

Dati: sono valori che possono essere salvati, recuperati e modificati durante l’esecuzione di un programma.

Pensiero computazionale

Il testimone di Papert è stato raccolto da Mitchel Resnick responsabile del Lifelong Kindergarten del MIT MediaLab che con i suoi collaboratori ha realizzato un frame work per l'insegnamento del pensiero computazionale e la valutazione dell'apprendimento che si fonda sulla convinzione che **i bambini possano acquisire il pensiero computazionale programmando storie interattive e videogiochi** (gli artefatti di cui parlava Papert).

Il lavoro di questi anni di Resnick e dei suoi collaboratori ha portato alla nascita di **Scratch**, un ambiente di programmazione visuale che consente ai ragazzi di creare in maniera semplice e intuitiva le proprie storie animate, giochi e simulazioni: oggi Scratch conta una community di giovani sviluppatori estesa in tutto il mondo ed è di fatto lo strumento di riferimento per insegnare ai bambini il pensiero computazionale attraverso la programmazione.

Pensiero computazionale

Il software **Scratch** è un «tool» di programmazione visuale (il codice del programma non deve essere digitato) ideato al Mit di Boston.

Blockley è un ambiente di programmazione grafico, l'utente sposta dei blocchi, simili a tasselli di un puzzle, per realizzare le proprie applicazioni.

Ricorda altri linguaggi di programmazione visuale, infatti storicamente la realizzazione di Blockley è stata influenzata da **App Inventor**, App Inventor è stato influenzato da **Scratch** il quale è stato influenzato da **StarLogo** e **Etoys** .

Il progenitore di tutti questi linguaggi di programmazione è stato il Logo, creato da Seymour Papert insieme a Jean Piaget, fautore della teoria dell'apprendimento costruttivista

Pensiero computazionale

- Dare i comandi alle macchine
- Preparare serie di azioni partendo dall'obiettivo
-
- Account gmail
- Ai.2.appinventor MIT
- Permetti
- Lingua italiano
- Progetti avvio nome
- Screen 1 titolo

Pensiero computazionale

Progettazione				Blocchi
Componenti	Visualizzazione	Componenti utilizzati	proprietà	
			colore	
			Immagini	
			testo	

Pensiero computazionale

Cosa si dovrà vedere alla fine

Screen 1	proprietà	
	Colore	
	Posizione schermo	
Immagine	multimediale	carica immagine
Immagine proprietà	altezza	
	larghezza	
immagine	carica file	
Casella di testo	lato 1 se rinominare da 3 parte	
Casella di testo	lato 2 se rinominare da 3 parte	
Pulsante	per schiacciare per calcolo area	
Etichetta	risultato	

Se vogliamo spazi inserire caselle etichette

Pensiero computazionale

Blocchi

Pulsante 1 per sempre quando pulsante 1 cliccato esegui

Etichetta 1 imposta etichetta 1 (testo) a

Operazioni matematiche moltiplicazione

Casella di testo 1 Casella testo 1 X

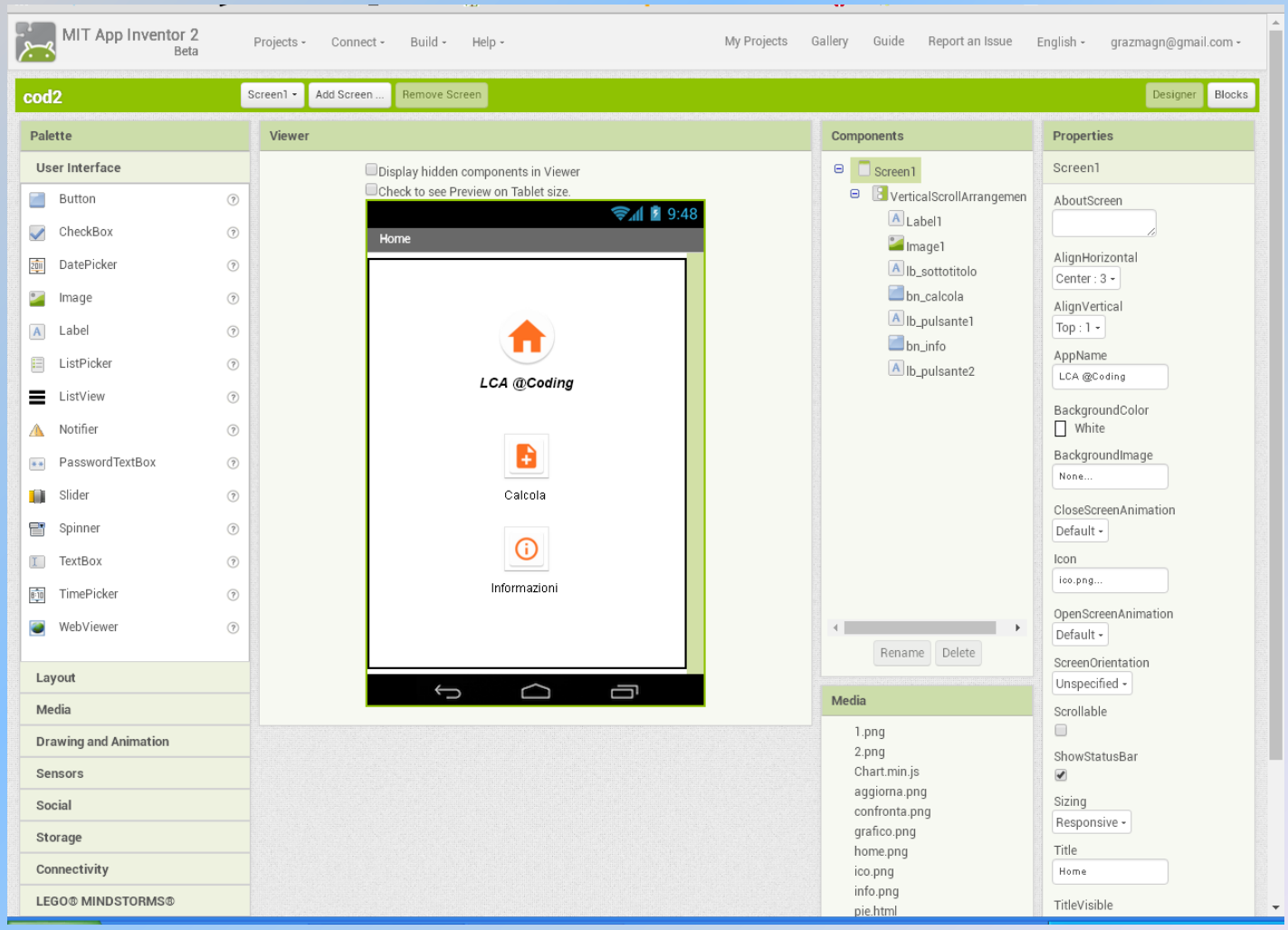
Casella di testo 2 Casella testo 2 X

Telefono schermata di blocco e sicurezza impostazioni sorgenti sconosciute

compila provvedi QR code

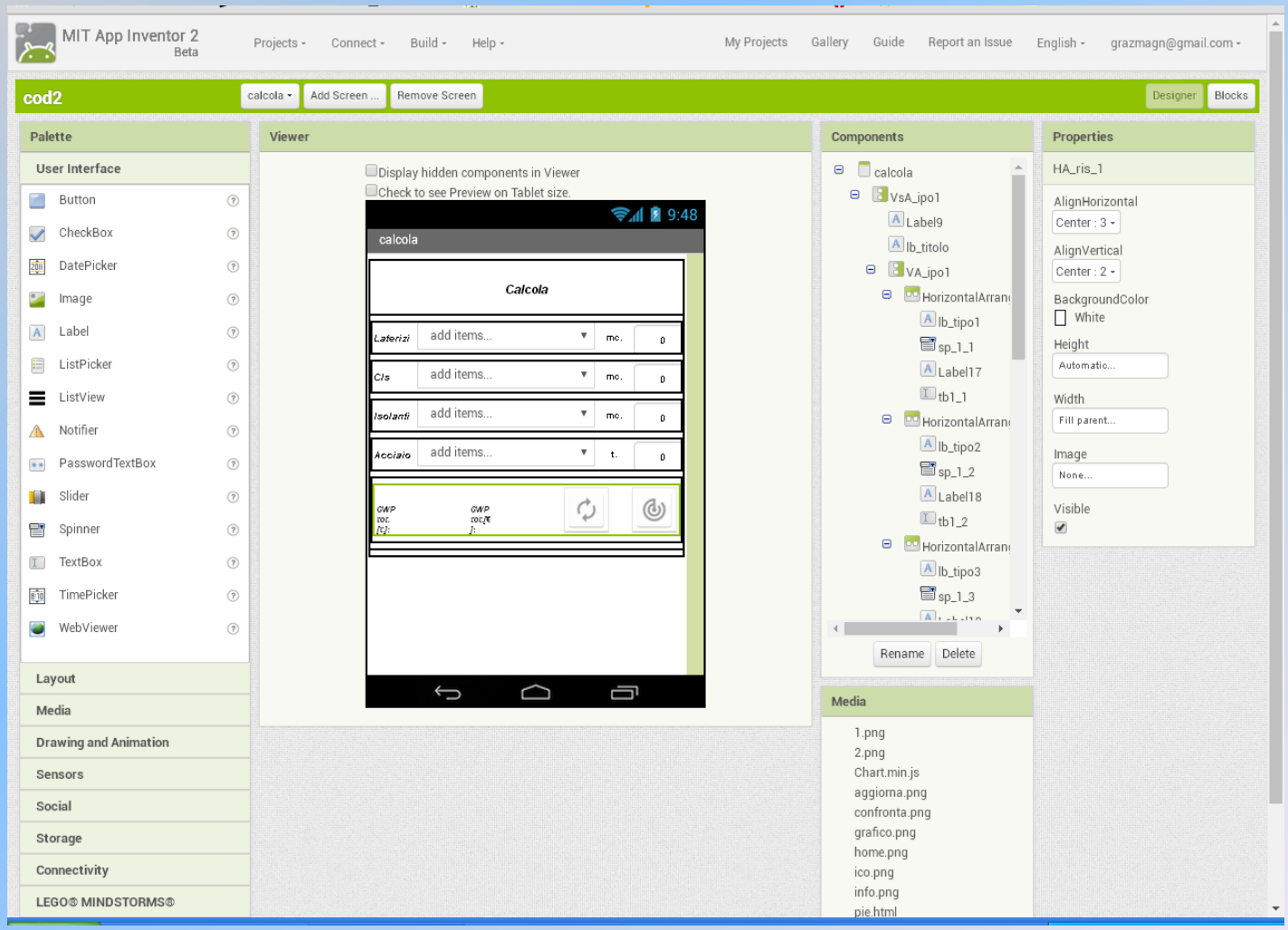
Pensiero computazionale

App inventor



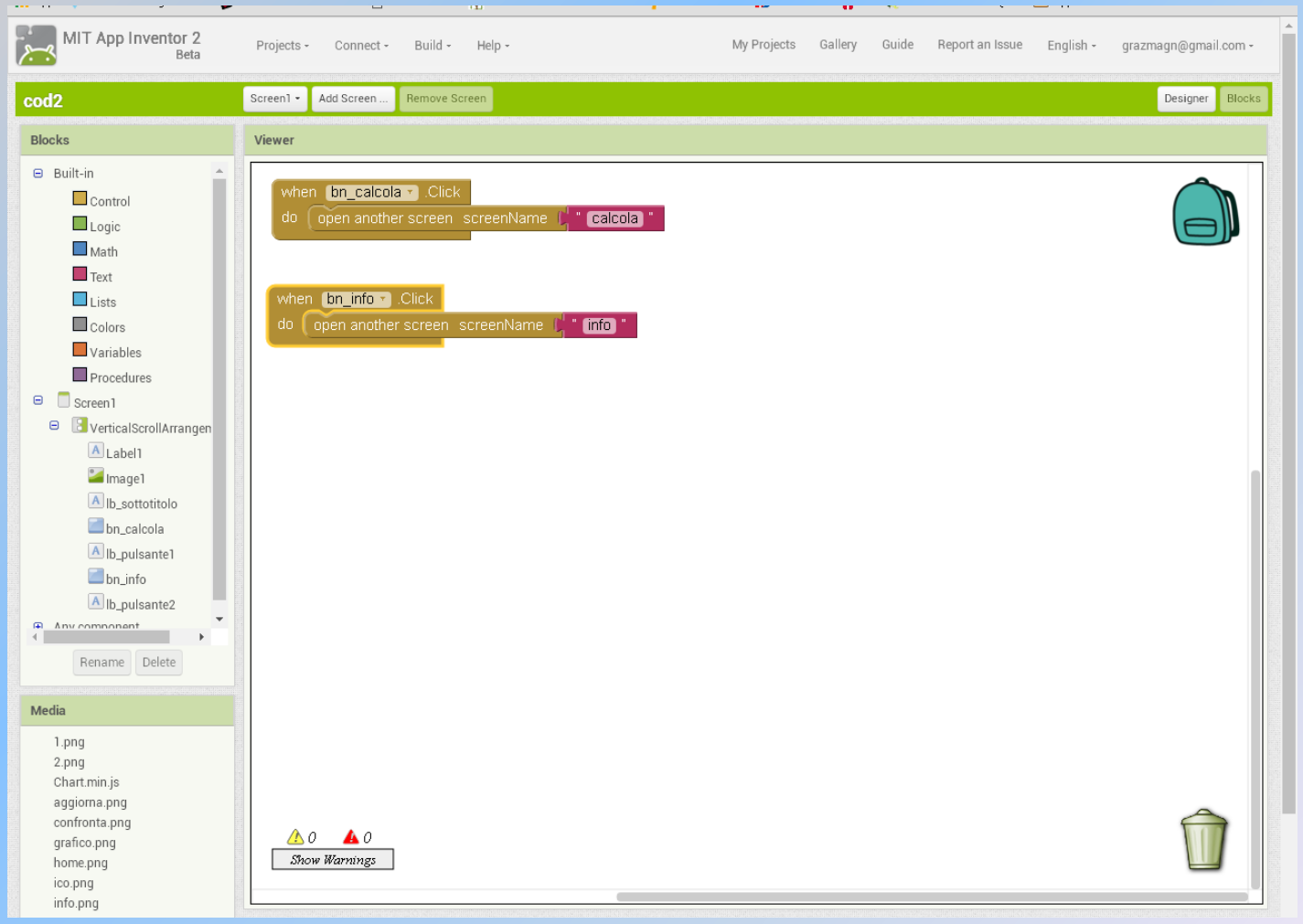
Pensiero computazionale

App inventor



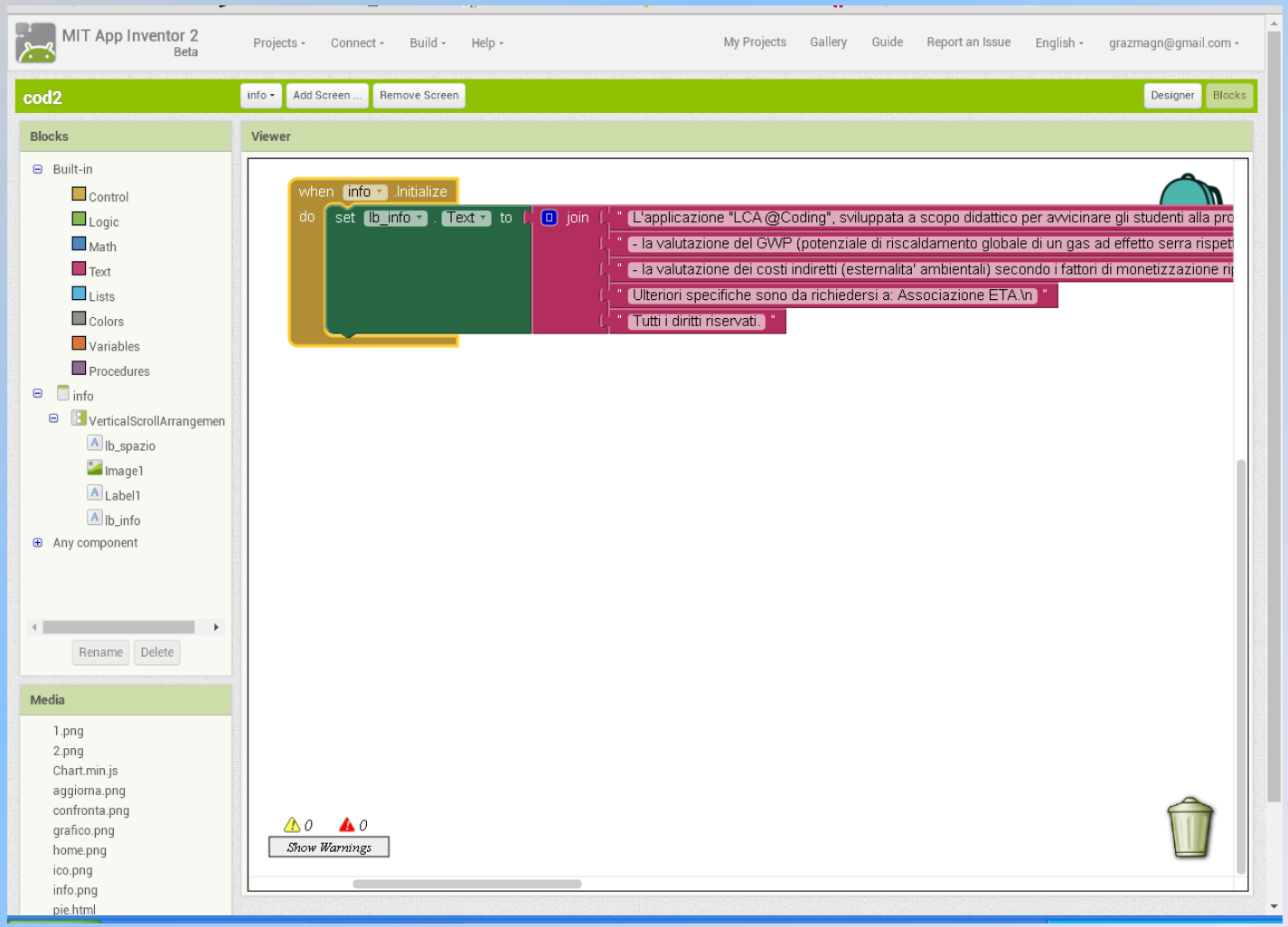
Pensiero computazionale

App inventor



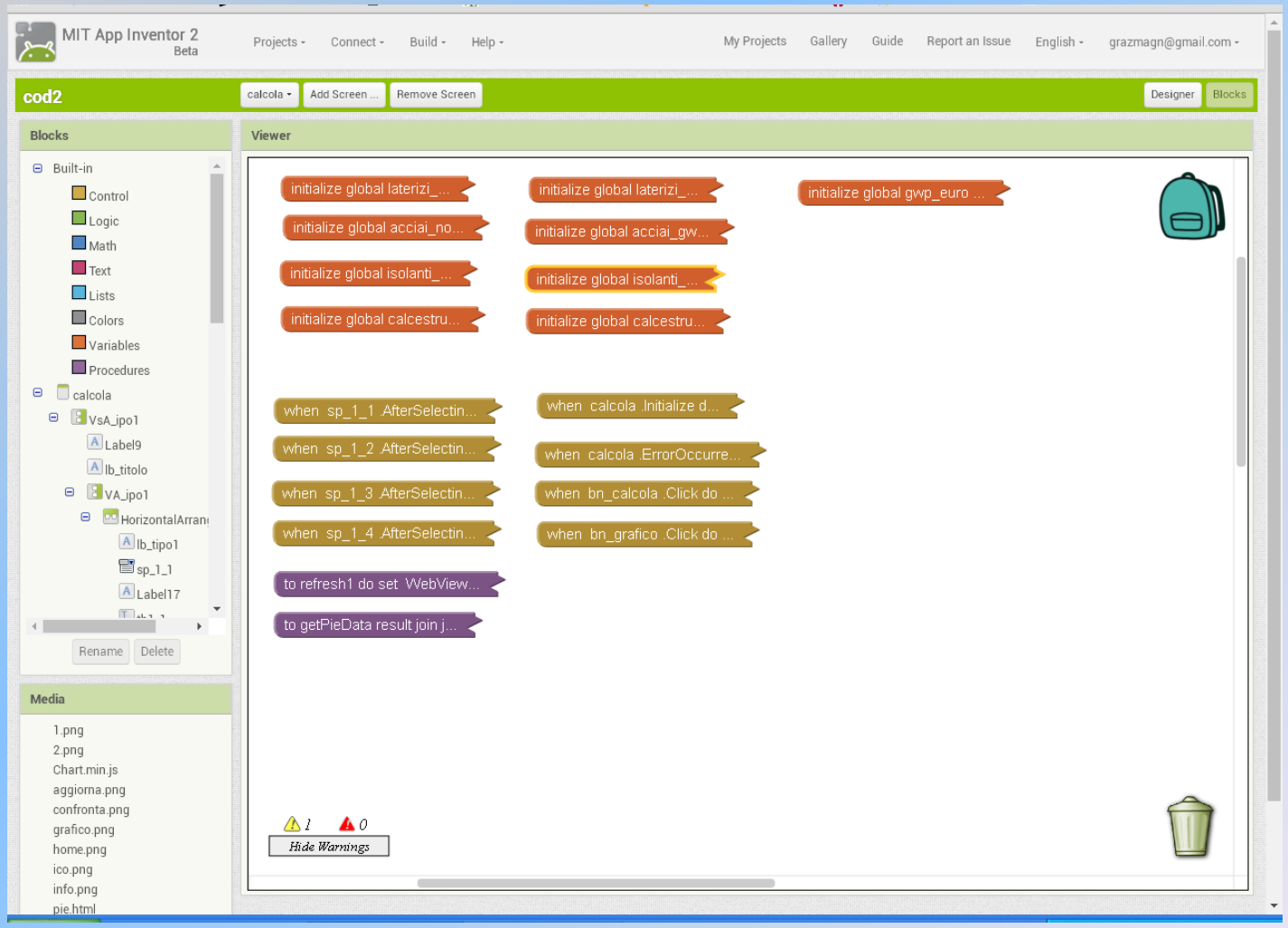
Pensiero computazionale

App inventor



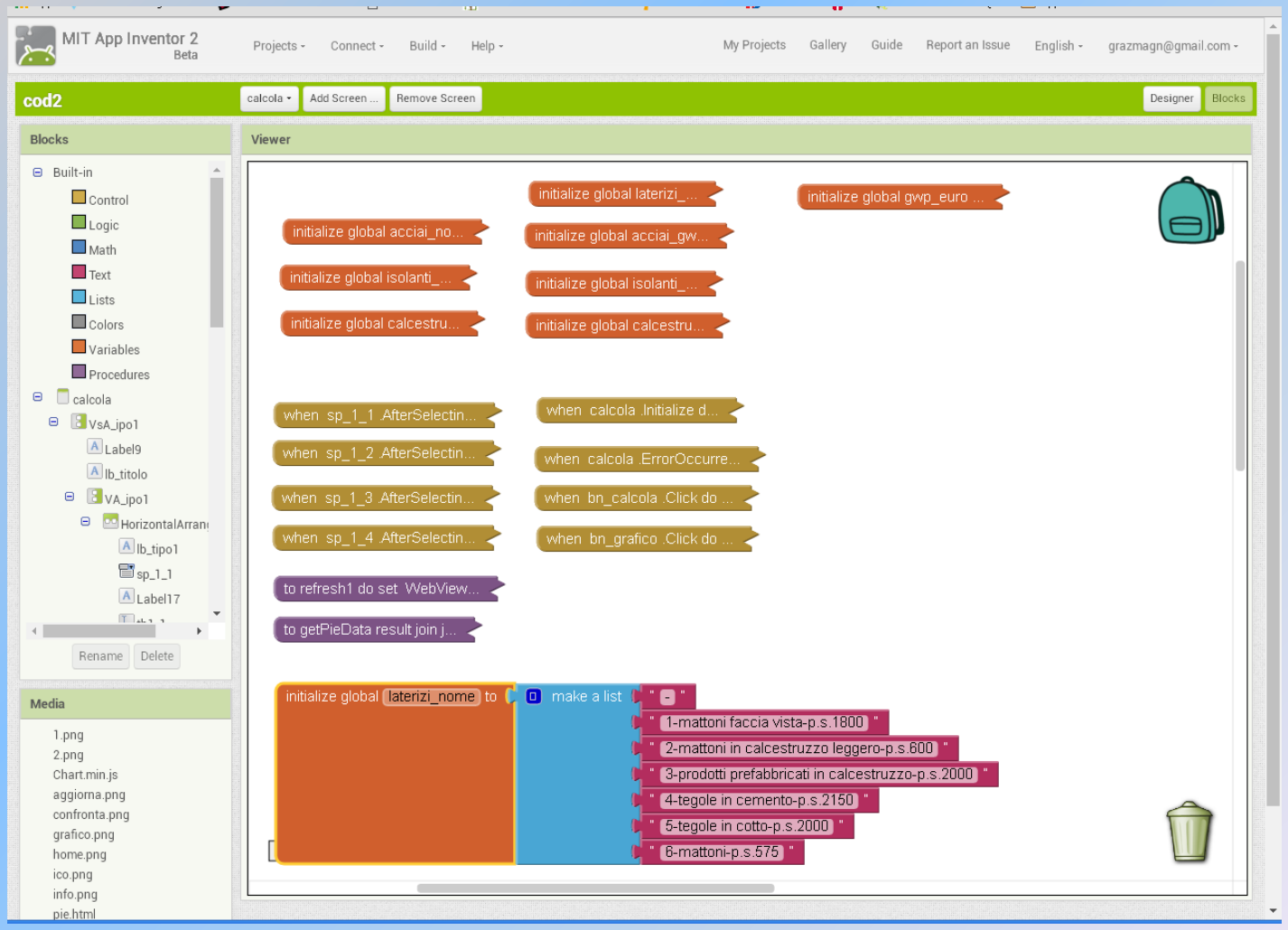
Pensiero computazionale

App inventor



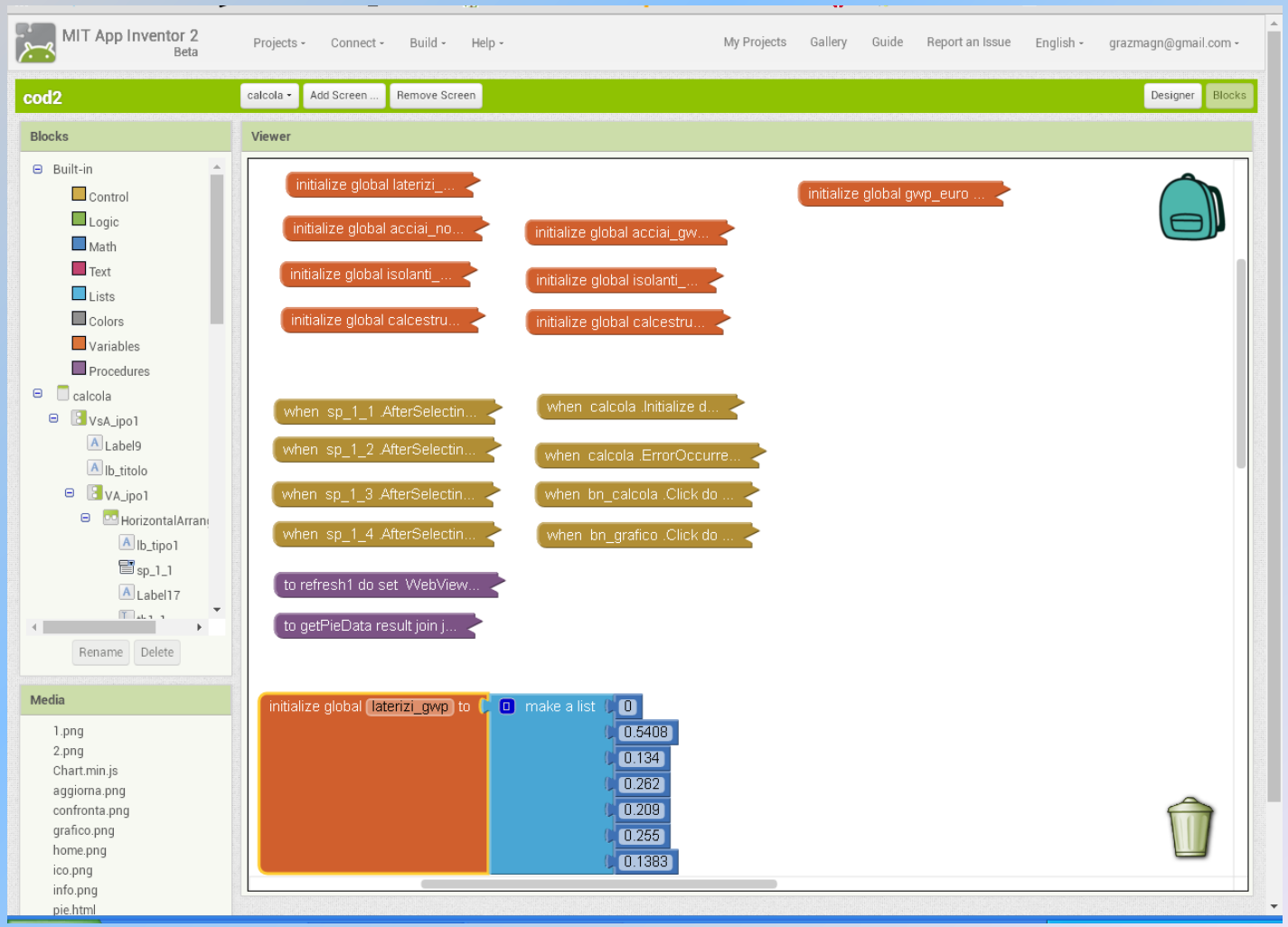
Pensiero computazionale

App inventor



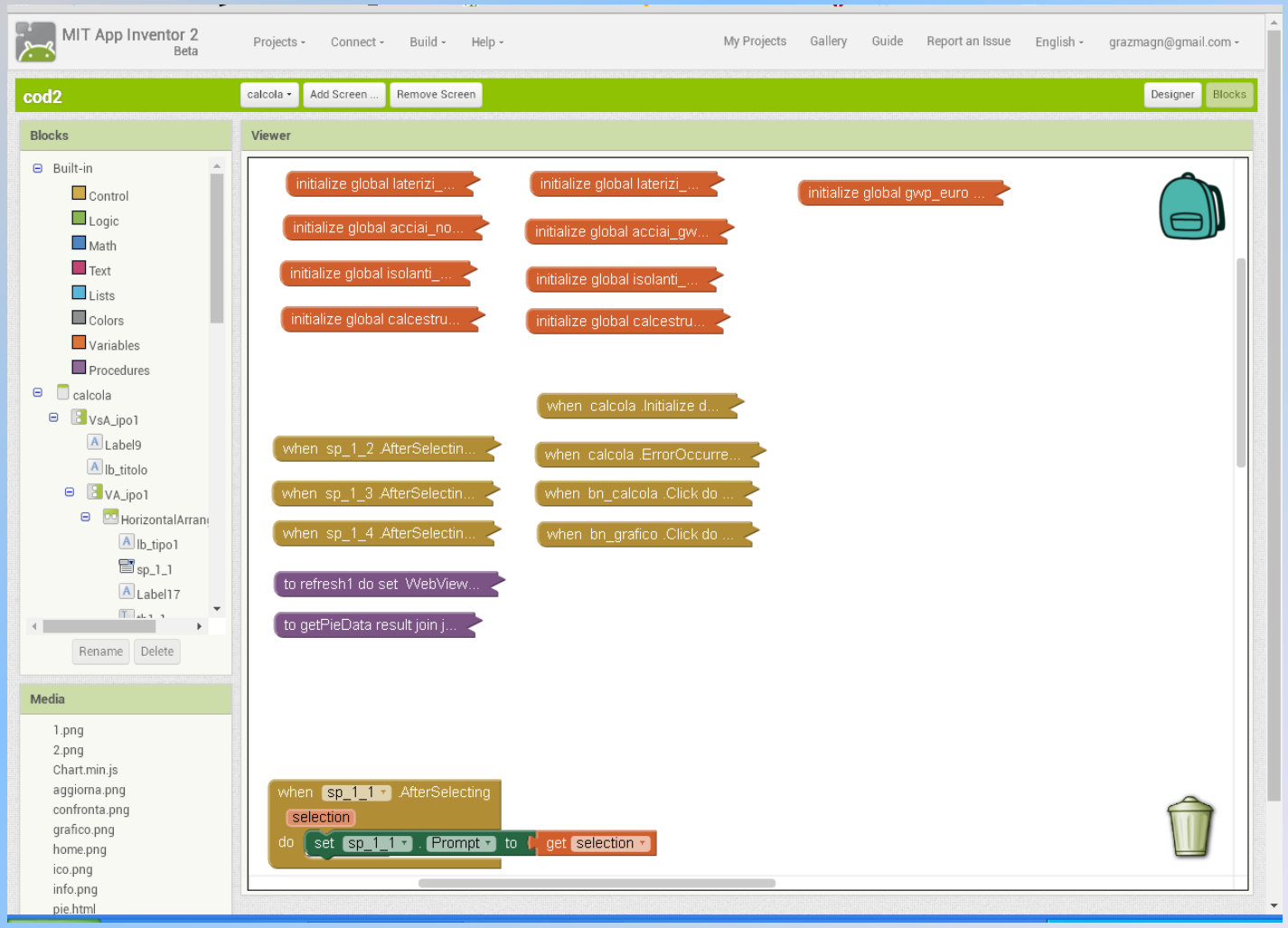
Pensiero computazionale

App inventor



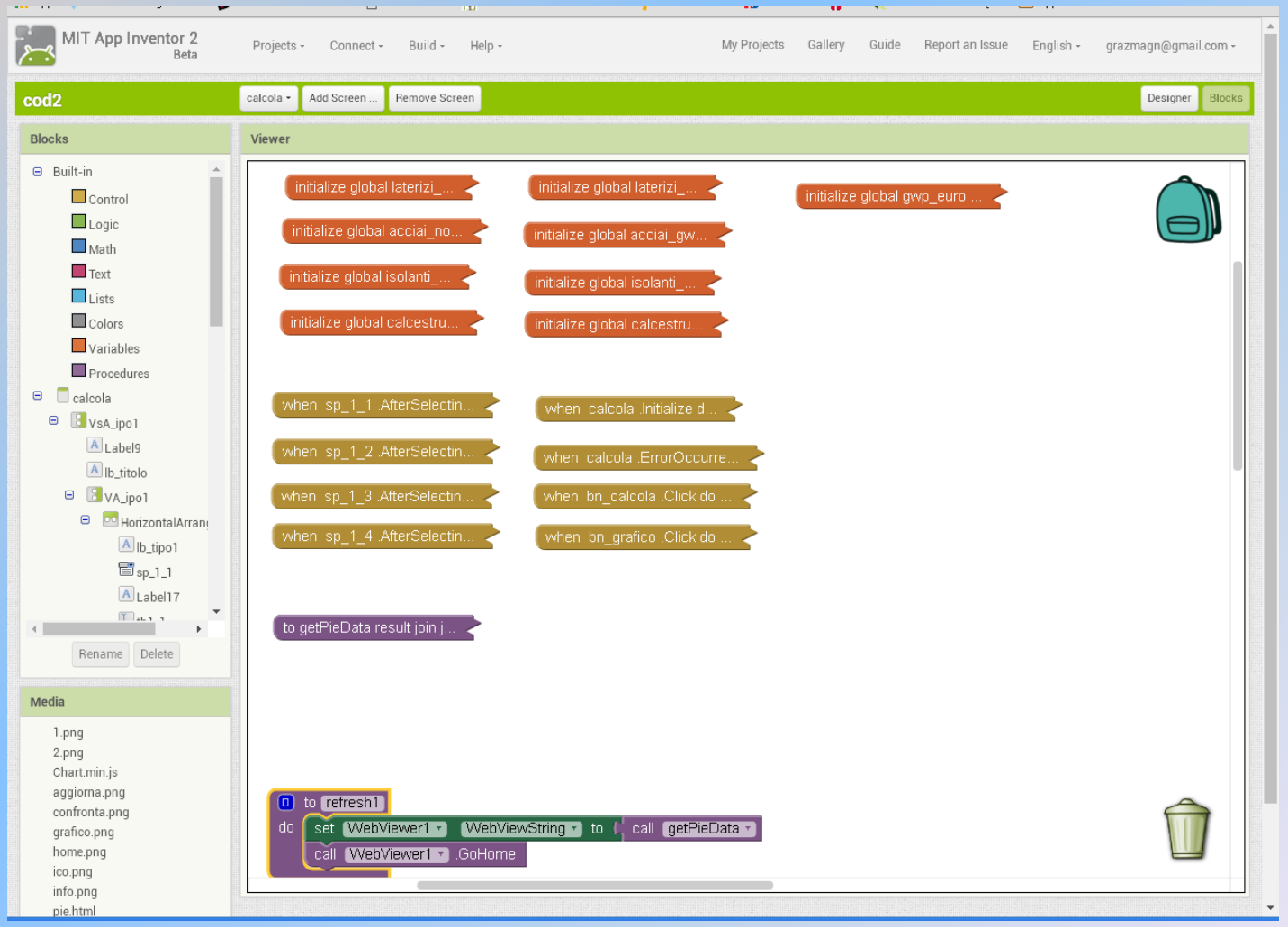
Pensiero computazionale

App inventor



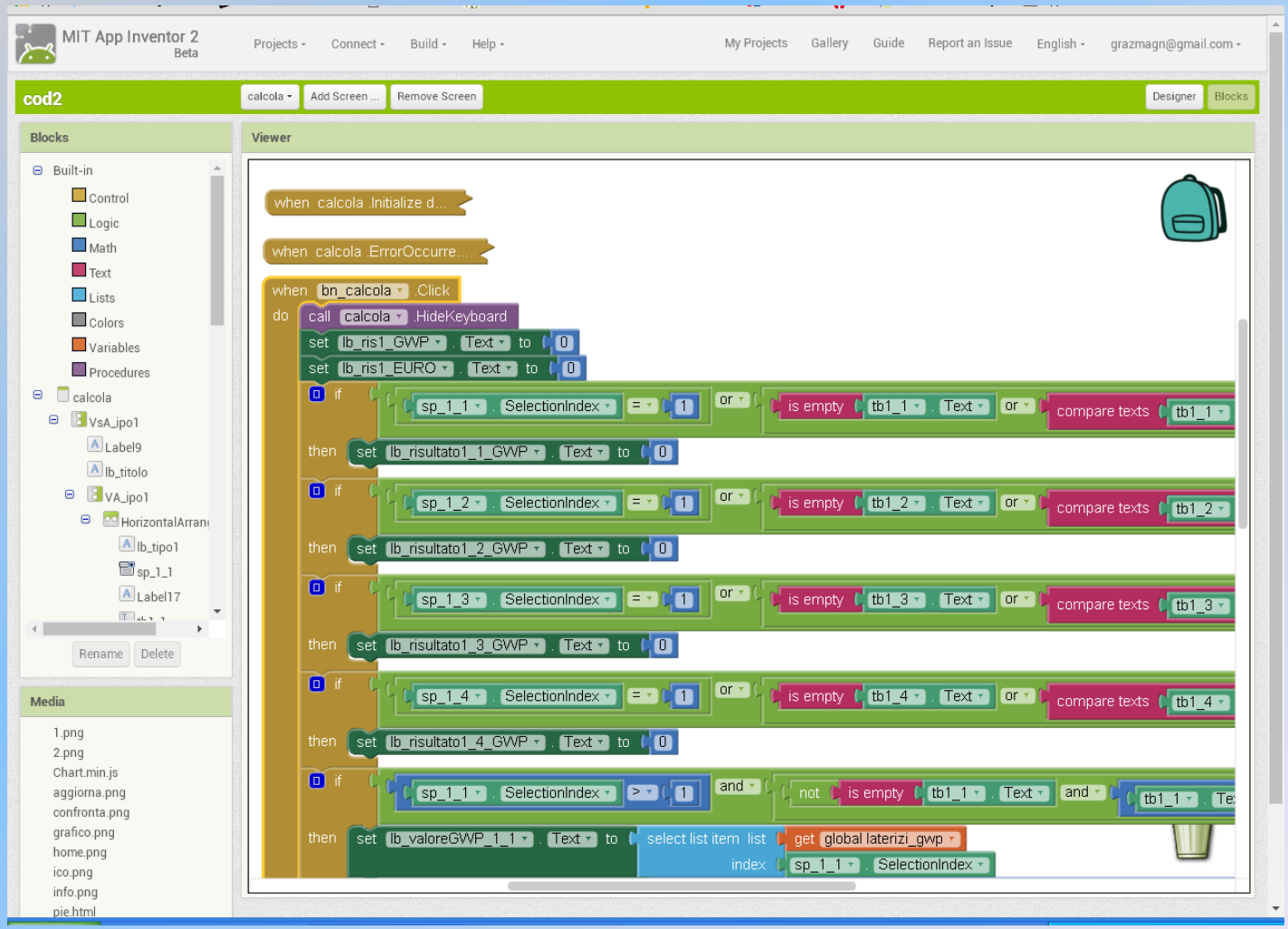
Pensiero computazionale

App inventor



Pensiero computazionale

App inventor



Pensiero computazionale

App inventor

Pensiero computazionale

App inventor