# La Sequenza

## Panoramica:

| | |
|---|---|
| Spiegazione del concetto | **Sequenza:** una serie ordinata di passaggi per l'esecuzione di un'attività. Per garantire che Codey Rocky possa portare a termine il compito come previsto, dovremmo dare istruzioni nel giusto ordine.<br>Esempio: Metti l'anguria in frigorifero.<br><br>**Bug**: errori che causano il fallimento dei programmi durante l'esecuzione<br><br>**Debug**: trova bug e risolvili. |
| Obiettivi formativi | 1. Comprendere il concetto di Sequenza;<br><br>2. Progetta una serie di passaggi per creare un'animazione;<br><br>3. Capire il concetto di Bug;<br><br>4. Trova bug e risolvili. |
| Preparazione dell'insegnamento | 1. Una lavagna e un pennarello per lavagna (oppure puoi usare una lavagna e gessetti);<br><br>2. Un Codey e un dongle Bluetooth (o il cavo USB) per ogni studente, ma va bene anche se 2 o 3 gli studenti condividono un set;<br><br>3. Un computer con mBlock 5 installato per ogni studente, ma va bene anche se 2 o 3 studenti condividono a computer;<br><br>4. Distribuisci un rapporto di auto-revisione e un rapporto sul progetto per ogni studente. |

| | |
|---|---|
| Durata | 90-120 min |

## Procedura di insegnamento:

### Fase 1: Ripasso -- Evento

Revisione:

- Che cos'è l'evento?
- Gli studenti possono pensare ad eventi della vita quotidiana?
- Quali eventi hai utilizzato nell'ultima lezione?

L'**evento** è un'azione che fa sì che le cose accadano. Ad esempio, quando si preme il pulsante, la luce è accesa. In questo caso, premere il pulsante è l'evento e il risultato è che la luce accesa. Gli eventi che gli studenti hanno utilizzato nella sessione dell'ultima lezione sono:
1) All'avvio del programma  2) Quando Pulsante A, B e C premuti.

### Fase 2: Specificare nuove conoscenze - La Sequenza (Sequencing)

Il nuovo concetto per questa lezione in classe è **la sequenza**. Ad esempio, se vogliamo mettere il cocomero nel frigorifero, dobbiamo seguire questi passaggi:

1. Aprire la porta del frigorifero.

2. Metti l'anguria nel frigorifero.

3. Chiudere la porta del frigorifero.

Se segui la sequenza sbagliata, non potresti mettere il cocomero nel frigorifero.

### Brainstorming

Chiedi agli studenti se riescono a pensare a casi in cui devono seguire una serie di passaggi per ottenere qualcosa. Gli insegnanti possono fornire degli esempi:
*Devi prima svitare il tappo della bottiglia, versare l'acqua nella bocca e poi avvitare il tappo della bottiglia.*
Se non segui i passaggi, non sarai in grado di bere l'acqua. (Suggerimento: ogni esempio dovrebbe offrire solo un ordine specifico. Significa che solo quando segui una serie specifica di passaggi, è possibile fare le cose nell'esempio.)
Per eseguire un'attività, è necessario seguire una serie di passaggi. L'ordine in cui vengono eseguiti i passaggi viene chiamato **Sequenza**.

## Fase 3: Gioco Iniziale - Sono un robot

L'insegnante si comporta come un robot: Si sposta per la classe va alla lavagna e disegnando una faccina sorridente. Gli studenti danno istruzioni al robot e li scrivono su carte.

## Istruzioni di gioco:

1. Gli insegnanti leggono le istruzioni degli studenti dall'inizio alla fine.
2. Chiedi agli studenti di lasciare istruzioni dettagliate nell'ordine corretto.

## Suggerimenti

1. Se gli studenti scrivono le istruzioni da sinistra a destra, gli insegnanti dovrebbero continuare a leggere le istruzioni da cima a fondo. In questo caso, esiste la possibilità che le istruzioni possano essere lette solo da sinistra.

2. Quando le istruzioni degli studenti non sono chiare, gli insegnanti devono comunque seguire le istruzioni per compiere le azioni. Ad esempio, se l'istruzione è: gira a sinistra, vai avanti di 4 metri, il robot dovrebbe esegui le istruzioni come: gira a sinistra e vai avanti. Questo è esattamente il modo in cui vengono eseguite le istruzioni del software. Quando non ci sono impostazioni specifiche per il tempo e l'angolo, il computer legge le semplici istruzioni per girare a sinistra e quindi legge le istruzioni per andare avanti;

3. Se è necessario rendere le istruzioni più specifiche, è possibile ricordare agli studenti che il robot appoggia su due gambe verticalmente. Pertanto, quando gli studenti stanno dando alcune istruzioni, devono assicurarsi che le istruzioni siano sufficientemente dettagliate. Ad esempio, se l'istruzione per il robot è quella di prendere una penna, l'istruzione deve includere i dettagli: con quale mano, il gesto della mano, dove disegnare la faccina esattamente sulla lavagna, ecc;

4. In considerazione del limite di tempo e dell'età degli studenti, gli insegnanti possono semplificare le istruzioni. Ad ogni modo, il punto chiave è chiaro: è necessario rendere le istruzioni specifiche e organizzarle nell'ordine corretto se si desidera che il robot faccia le cose mentre si programma.

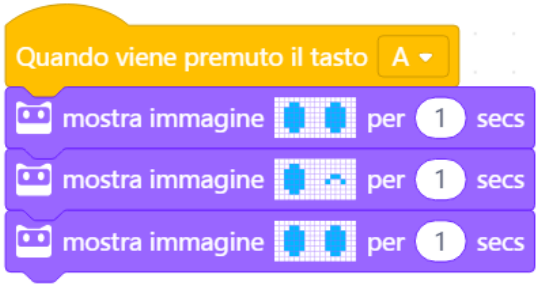
**Riepilogo:** quando programmiamo, organizziamo i blocchi nell'ordine dall'alto verso il basso per formare una serie di passaggi. Le macchine possono seguire i passaggi uno per uno per eseguire un'attività. Ci riferiamo alla serie di passaggi come la **Sequenza**.

## Fase 4: Attività dello studente - Progettazione dell'animazione

Crea animazioni con il blocco 【mostra l'immagine () per () secondi】. Tradizionalmente, i disegnatori di animazioni realizzano le animazioni nel modo seguente: Metti prima un foglio di disegno sul tavolo e apri un nuovo foglio di disegno sopra il primo foglio. I disegnatori tracceranno la cornice e modificheranno leggermente il disegno. Poi un altro nuovo foglio, delinea la cornice e cambia leggermente il disegno. I disegnatori ripetono i passaggi più e più volte finché non completano una serie di immagini leggermente diverse l'una dall'altra. Quindi, capovolgono velocemente i disegni per renderli più fluidi. In parole povere, i disegnatori animano i disegni.

## Attività 1: Seguire gli insegnanti per progettare animazioni

Dimostra: In questa sessione, gli insegnanti mostreranno come creare animazioni con il blocco 【mostra immagine () per () secondi】. Il metodo è semplice: usa l'immagine del blocco come immagine di partenza, duplica il blocco e modifica leggermente l'immagine. Ripeti i passaggi e quindi disponi questi blocchi in sequenza. Progetto di esempio: *Winking Eyes*




Programma Story:

Trascina il blocco 【mostra l'immagine () per () secondi)】 e modifica la prima immagine come gli occhi di Codey apri;
Duplica il blocco 【mostra l'immagine () per () secondi】 ma modifica un occhio per fare l'occhiolino;
Duplica il blocco 【mostra l'immagine () per () i secondi】 e questa volta fa aprire l'occhio occhiolino.

Aggiungi il blocco eventi - ""Quando il pulsante A è stato premuto"

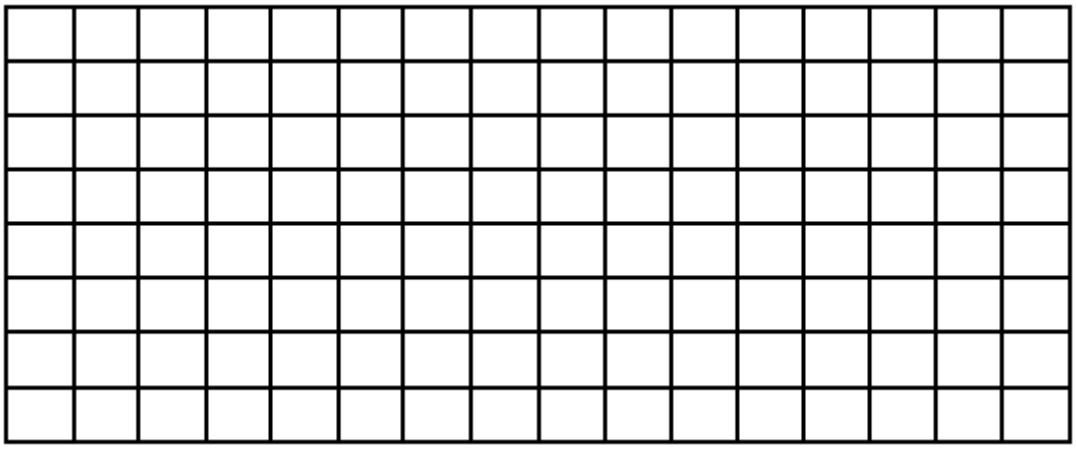
Carica i programmi sul dispositivo. Quando viene premuto il pulsante A, Codey ti farà l'occhiolino.

## Attività dello studente:

Gli studenti lavorano in coppia per fare le loro animazioni come facevano gli insegnanti.

## Attività 2: Progettare nuove animazioni

Alla fine di questa sessione, gli studenti devono mostrare i loro lavori e compilare i rapporti del progetto.
Gli studenti lavorano a coppie e riempiono le piccole scatole per formare prima un'immagine; disegna la stessa immagine con il blocco 【show image ( ) for ( ) secs (mostra l'immagine () per () i secondi)】; duplicare il blocco e cambiare leggermente l'immagine; ripeti il passaggio finché non avrai una serie di blocchi che stanno cambiando e disposti in sequenza.



Carica i codici per esaminare l'effetto dell'animazione. Gli studenti devono presentare le loro opere, ma devono prima completare il rapporto del progetto.
Dovrebbero mostrare le loro opere seguendo le domande del rapporto.

## Suggerimenti:

1. Se alcuni studenti finiscono il compito in anticipo, suggerisci di iniziare le loro animazioni con più eventi;

2. Oppure fagli cambiare per quanto tempo mostrano l'immagine;

3. Se ci sono studenti che non riescono a completare l'attività in tempo, invitali a spiegare cosa è successo quando stavano scrivendo programmi. Potrebbe essere successo qualcosa di divertente: una sfida da affrontare o un problema che hanno incontrato;

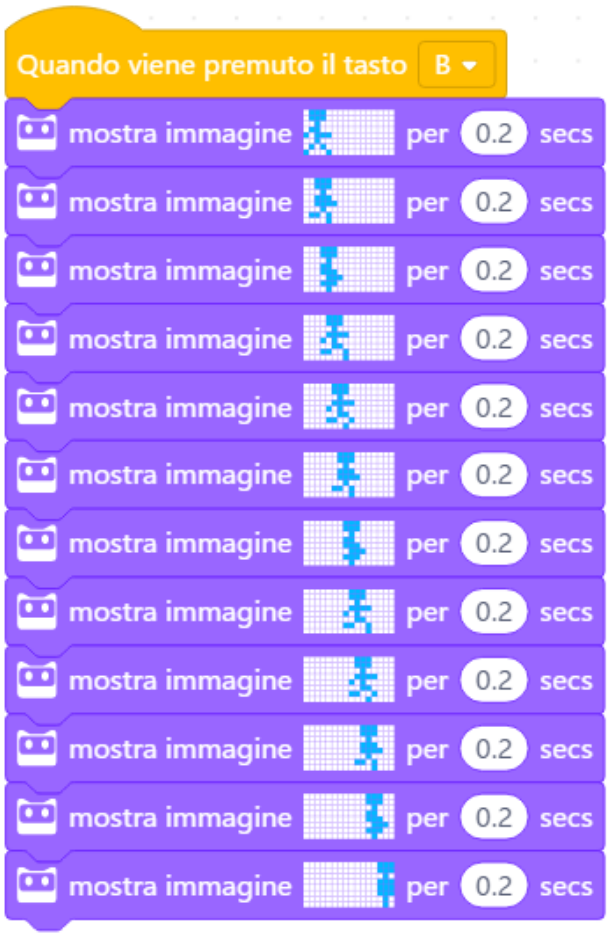
4. Gli studenti possono condividere le loro opere con l'intera classe, oppure possono a turno presentare le opere ad altri gruppi;

5. Gli insegnanti possono adattare il limite di tempo in base agli scopi didattici e al tipo di classe. Tempo consigliato: 10 min

Gli insegnanti possono mostrare progetti di esempio: *Growing Tree and Walking*


Program Story

Posizionare il display a matrice LED in posizione verticale. Quando viene premuto il pulsante B, vedrai  un alberello crescere




Program Story

Quando si preme il pulsante B, un bambino scorrerà da un lato del display a matrice LED all'altro lato.

## Suggerimenti per esperti: Bug e debug

Introduci i nuovi concetti Bug e Debug prima che gli studenti lavorino sui loro compiti. Quando scriviamo programmi, i bug sono inevitabili. *A Bug* significa “un insetto” in senso letterale ma si riferisce a un problema tecnico in un programma per computer. Nella fase iniziale in cui i computer sono stati inventati, a differenza dei computer portatili di oggi, erano di dimensioni estremamente grandi. (Immagine tratta da Wiki).

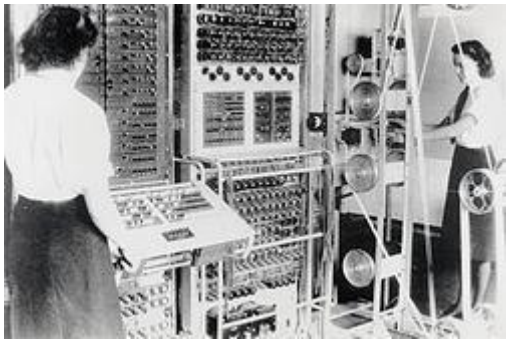


Figura 1 I computer colossus sono stati usati per decifrare i codici tedeschi durante la seconda guerra mondiale

Fu durante questo periodo che un computer colossus non funzionava correttamente. L'intero team di programmatori stava cercando faticosamente di scoprire il problema, ma senza successo. Alla fine, Grace Murray Hopper, una programmatrice donna, ha identificato il problema: una falena volava all'interno del computer e provocava il problema. Quando hanno rimosso la falena, tutto è tornato a posto. È stato il primo *bug* in un programma per computer che è stato trovato e i programmatori lo hanno apposto sul registro (vedi l'immagine sotto). Da allora, il termine bug è comune in uso quando le persone si riferiscono agli errori in un programma per computer. Naturalmente, Grace Murray Hopper è stata da allora considerata la madre di *Debug*. (Immagine tratta da Wiki)

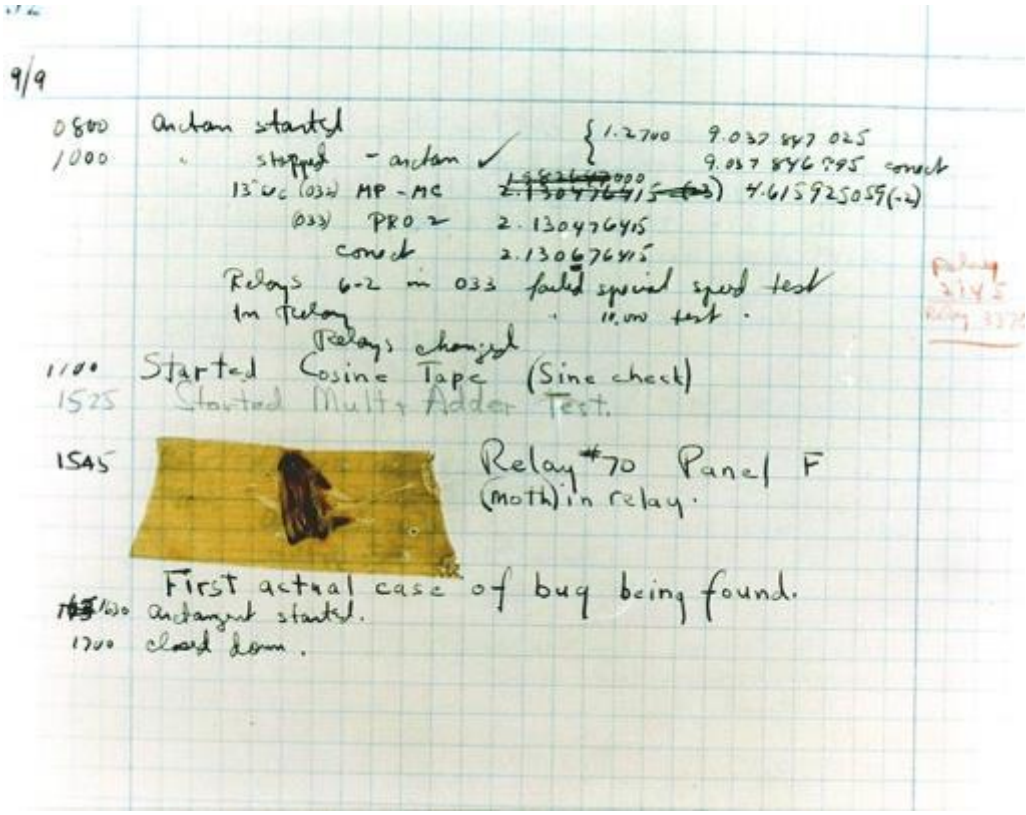


Figura 2: La falena ha causato l'errore del computer e questo è il primo errore in un programma per computer

Quando scriviamo programmi, dobbiamo scorrere attraverso ogni riga dei codici per trovare i bug e risolverli. Trova i bug nei seguenti programmi e prova a risolverli:

**1. Lombrico e insetto**


Program Story

Un piccolo lombrico incontra un grosso insetto mentre gattona per terra. Rimuovi l'insetto per far tornare di nuovo il lombrico (cambia l'immagine dell'insetto come immagine del lombrico che striscia).

**2. The Bomb Can not Count Down !!**


Program Story**:**

C'è una bomba che non c'è tempo. Aiutalo!

**3. La chiave dell'auto è stata rubata!**


Program Story:
non è possibile avviare l'auto perché la chiave dell'auto è stata rubata. Devi trovare la chiave adesso! (l'evento mancante).

## Fase 5: Conclusione

1. Riepiloga quale sia il concetto di Sequenza. La sequenza fa riferimento a una serie di passaggi ordinati per l'esecuzione di un'attività.

2. Nel software mBlock, la sequenza è dall'alto verso il basso. Se Codey Rocky non riesce a eseguire l'attività come programmato, puoi scorrere ogni riga di codici dall'alto verso il basso per trovare bug.
3. Quando si progettano animazioni, è possibile iniziare con un foglio di immagini di base, copiare l'immagine e modificarla leggermente ogni volta. Infine, riproduci la serie di immagini in sequenza per animarle.

4. Gli studenti completano il rapporto di auto-valutazione

## Rapporto di Autovalutazione

**Nome :** **Età :**

- Rispondi alle seguenti domande e registra il tuo risultato :

Descrivi ciò che hai imparato con una o due frasi.

| |
|---|
| |

Descrivi brevemente cosa ti piace di più e di meno in questa sessione di lavoro

| |
|---|
| Quello che mi piace di più |
| Quello che mi piace meno |

Disegna una **Sequenza** che ti è successa oggi :

| |
|---|
| |

Puoi disegnare come ti senti in questa sessione di classe nell'angolo in alto a destra del rapporto di autovalutazione.

## Rapporto di progetto

**Nome :** **Nome del gruppo :**

- Segui queste domande per presentare il tuo lavoro:

Quali compiti hai intrapreso? Descrivilo in una o due frasi.

| |
|---|
| |

Hai qualche idea su come adempiere ai compiti?

Annota o disegna ogni singolo pezzo che ti ispira, che sia buono o cattivo. Lo stai facendo per esplorare più possibilità. Se hai bisogno usa più fogli per registrare le tue idee.

| |
|---|
| |

Descrivi l'effetto finale del tuo progetto e perché scegli l'effetto in una o due frasi

| |
|---|
| Effetto<br>*Esempio: quando si preme il pulsante A, il lombrico inizia a strisciare in avanti.* |
| Perché scegli questo (questi) effetto?<br>*Esempio: perché è divertente.* |

Hai incontrato ostacoli? Hai qualche soluzione? Descrivilo con una o due frasi. (La soluzione può essere un livello approssimativo)

| | |
|---|---|
| Le difficoltà<br>*Esempio: impossibile connettere Codey Rocky a mBlock ; Impossibile caricare i codici;Impossibile raggiungere il consenso ...* | soluzioni<br>*Esempio: accendi Codey Rocky; Sasso carta forbici…* |

Gli studenti possono rispondere alle seguenti domande dopo la sessione di condivisione.

Ti piace il tuo programma? Descrivi cosa ti piace di più e meno del progetto con una o due frasi. E eventuali miglioramenti futuri?

| |
|---|
| ○Lo amo ○ Mi piace ○ Così così ○ Non mi piace ○ Odio |
| Quello che mi piace di più<br>*Esempio: l'animazione è vivida; il suono corrisponde all'animazione.* |
| Quello che mi piace meno<br>*Esempio: è difficile per le persone raccontare di cosa tratta l'animazione a prima vista.* |
| Migliorare<br>*Esempio: ridisegna l'animazione e rendila più semplice questa volta; aggiungi suoni e luci.* |

## Valutazione dell'insegnante:

1. Cooperazione (30%): valutare il rendimento del gruppo in termini di divisione del lavoro, collaborazione e coordinamento.

2. Completezza (20%): valutare se il progetto è sufficientemente completo. Naturalmente, il progetto deve attenersi prima all'argomento.

3. Innovazione (20%): valuta la creatività del progetto.

4. Funzionalità (20%): valutare se il lavoro è abbastanza funzionale?

5. Difficoltà (10%): valuta qual è il livello di difficoltà del lavoro?

## CSTA

| gradi | Identifier | Standard CSTA K-12 CS interinali | Concetto di lavoro | Concetto Pratico |
|---|---|---|---|---|
| **K-2** | 1A-A-5-2 | Costruisci programmi, per svolgere un'attività o come mezzo di espressione creativa, che include sequenze, eventi e loop semplici, usando un linguaggio di programmazione visiva basato su blocchi, sia in modo indipendente che collaborativo (ad es., Programmazione coppia). | Algoritmi e programmi | Creazione di artefatti computazionali |
| **K-2** | 1A-A-5-3 | Pianifica e crea un documento di progettazione per illustrare pensieri, idee e storie in un modo sequenziale (passo per passo) (ad esempio, una mappa della storia, uno storyboard, un organizzatore grafico sequenziale). | Algoritmi e programmi | Creazione di artefatti computazionali |
| **K-2** | 1A-A-3-7 | Costruire ed eseguire algoritmi (serie di istruzioni passo-passo) che includono sequenze e loop semplici per eseguire un compito, sia in modo indipendente che collaborativo, con o senza un dispositivo di calcolo. | Algoritmi e programmi | Riconoscendo e Definizione computazionale I problemi |
| **K-2** | 1A-A-6-8 | Analizzare e correggere (correggere) un algoritmo che include sequenze e loop semplici, con o senza un dispositivo di calcolo. | Algoritmi e programmi | Test e raffinazione |
| **3-5** | 1B-A-2-1 | Applicare strategie di collaborazione per supportare il problema risolvendo il ciclo di progettazione di unprogramma. | Algoritmi e programmi | collaborare |
| **3-5** | 1B-A-5-4 | Costruisci programmi, per risolvere un problema o per l'espressione creativa, che includono sequenze, eventi, loop, condizionali, parallelismo e variabili, usando un linguaggio di programmazione visuale basato su blocchi o un linguaggio basato sul testo, sia in modo indipendente che collaborativo (es. coppia di programmazione). | Algoritmi e programmi | Creazione di artefatti computazionali |

| | | | | |
|---|---|---|---|---|
| **3-5** | 1B-A-3-7 | Costruisci ed esegui un algoritmo (set di istruzioni passo passo) che include sequenze, loop e condizionali per eseguire un compito, sia in modo indipendente che collaborativo, con o senza un dispositivo di calcolo. | Algoritmi e programmi | Riconoscendo e Definizione computazionale I problemi |
| **3-5** | 1B-A-6-8 | Analizza ed esegui il debug (correzione) di un algoritmo che include sequenze, eventi, loop, condizionali, parallelismo e variabili. | Algoritmi e programmi | Test e raffinazione |
| **3-5** | 1B-I-1-17 | Cercare e confrontare diverse prospettive, in modo sincrono o asincrono, per migliorare un progetto. | Impatti di Informatica | Promuovere un Cultura informatica inclusiva |