

# Interazione uomo macchina

Jacopo De Angelis

24 marzo 2021



# Indice

<b>1</b>	<b>Modulo 1: Concetti di base</b>	<b>5</b>
1.1	Cos'è l'interazione uomo macchina? . . . . .	5
1.2	Perché è difficile progettare perché l'interazione sia "buona"? . . . . .	8
1.3	Temi dell'HCI . . . . .	8
1.4	Sistema socio-tecnico . . . . .	9
1.4.1	Proprietà emergenti . . . . .	10
1.4.2	Conseguenza inattesa . . . . .	11
1.5	Usabilità . . . . .	13
1.5.1	Bassa usabilità = danni e problemi . . . . .	13
1.5.2	La colpa è dell'utente o del progettista? . . . . .	13
1.5.3	Come capire a livello generico se l'interfaccia è chiara? . . . . .	14
1.5.4	Concetti estremi di usabilità . . . . .	14
1.6	Sei concetti centrali . . . . .	17
1.6.1	Definizione classica di interazione (ISO 9241) . . . . .	18
1.7	Affordance . . . . .	19

1.7.1	La legge di Fitt . . . . .	20
1.8	Scheumorfismo . . . . .	20
1.9	Vincoli . . . . .	21

# Capitolo 1

## Modulo 1: Concetti di base

### 1.1 Cos'è l'interazione uomo macchina?

Definizione standard

*"HCI (Human-Computer Interaction) è una disciplina che si occupa della progettazione, realizzazione e valutazione di sistemi interattivi con capacità computazionali destinati all'uso umano e dello studio dei principali fenomeni che li circondano" - Association for Computing Machinery*

L'usabilità di un sistema è spesso trascurata all'interno dell'ambito lavorativo italiano, non per faciloneria ma perchè le risorse da adibire a questo ambito sono una spesa che non viene vista come essenziale nella produzione del valore per andare avanti.

Il problema viene aggravato dall'outsourcing verso paesi dove il costo del lavoro sia inferiore e, ultimamente, anche da sistemi di ML che sono in grado, con grado di precisione sempre maggiore grazie al deep learning, di sviluppare codice funzionante. Ma cosa viene assegnato ai lavoratori esterni? Ciò che è altamente formalizzato, in modo da lasciare poche possibilità di variazione dalle richieste dei clienti.

Cosa rimane difficile da esternalizzare? Il contatto del cliente, sia durante la prima raccolta di requisiti funzionali, sia le successive interazioni con esso per cambiamenti incrementativi, variazioni di funzionalità o feedback.

L'usabilità è quella caratteristica che rende "facile la vita" al cliente.

---

*Piccola digressione*

**Perché la concorrenza moderna rende sempre più importante l'analisi dell'interazione uomo macchina?**

In un contesto monopolistico le aziende non sono invogliate a produrre la soluzione "migliore"<sup>1</sup> ma solo quella più efficiente a livello di costo marginale.

Cosa vuole dire questo? Che nella moderna concorrenza derivante da sistemi di sviluppo sempre più semplici e un'offerta più rapida tramite internet, per ottenere quote di mercato le aziende devono iniziare a pensare non solo al "funziona?" ma anche al "come lo faccio funzionare?"

---

**"Qui non si impara a fare delle interfacce usabili, si impara a riconoscere l'usabilità delle interfacce"** ovvero impariamo strumenti che ci possono portare a fare delle belle interfacce, certo, ma soprattutto ci permette di riconoscere cosa renda buona un'interfaccia.

La disciplina nasce negli anni '80 ma l'interazione con le macchine (intese come calcolatori) esiste dagli anni '40, semplicemente prima l'utente era ultra specializzato mentre ora quasi tutti possono accedere ad un PC e con questo interagire tramite un'interfaccia.

Dal 1983 si tiene la conferenza annuale [ACM CHI](#). In questi casi tre aree disciplinari si incontrano:

---

<sup>1</sup>Dove per migliore si intende quella che prende in considerazione più metriche come usabilità, efficienza, efficacia, design, ecc

- ergonomia
- informatica
- psicologia comportamentista

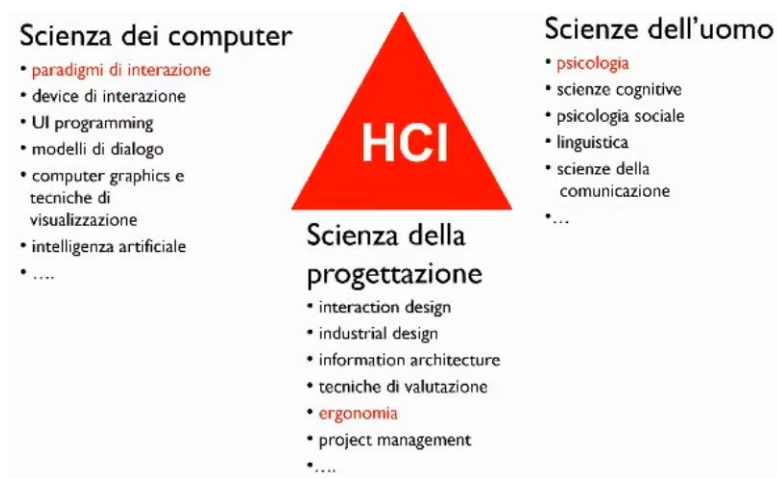


Figura 1.1: HCI e le sue componenti

Dobbiamo ricordare sempre una cosa: dobbiamo lasciarci alle spalle l'utente ideale, l'utente senza faccia e senza capacità, e iniziare a pensare all'utente reale, ovvero a chi, idealmente, è diretta la nostra interfaccia. L'utente non siamo noi. Proprio per questo le interviste, i test con esterni, i mockup sono utili, perchè ci permettono di vedere la nostra idea attraverso gli occhi di altri. Esempio banale: la nostra interfaccia basata sul colore potrebbe essere altamente confusionaria per un daltonico. Un'applicazione mobile dove tutti i comandi sono sulla destra potrebbe essere difficile da usare per un mancino.

**Ergonomia cognitiva:** studio dell'interazione tra l'uomo e gli strumenti per l'elaborazione di informazioni studiando i processi cognitivi coinvolti (percezione, attenzione, memoria, pensiero, linguaggio, emozioni) e suggerendo delle soluzioni per migliorare tali strumenti.

## 1.2 Perchè è difficile progettare perchè l'interazione sia "buona"?

Ci sono tre ragioni, idealmente:

1. **La varietà dei sistemi interattivi:** cellulari, computer, cloche, macchine da cucina, tablet...
2. **La varietà degli utenti:** fasce d'età, background culturale, condizioni mediche...
3. **La varietà degli scopi e degli usi:** contesti formativi, ludici, lavorativi e usi tramite dispositivi diversi, in luoghi differenti...

*Noi studieremo come progettare per la varietà e al volto delle procedure.*

## 1.3 Temi dell'HCI

- Criteri, metodi e strumenti per la **progettazione dell'interazione** fra esseri umani e sistemi interattivi. L'interazione è il contesto nel quale interagiamo;
- Criteri, metodi e strumenti per la **valutazione dell'usabilità** dei sistemi interattivi;
- Progettazione di nuove tecniche di interazione (dall'holo-lens ai sistemi che sfruttano i sistemi nervosi);
- Valutazione dell'**impatto dell'automazione** nei contesti umani
- **Sistema socio-tecnico**

Per valutazione dell'**impatto** ci si riferisce a due tipi di impatto:

- a breve termine: sui singoli e nel qui ed ora (usabilità)



- a medio-lungo termine: sono conseguenze inattese sulla collettività

---

### Errore comune

L'usabilità è un tipo di effetto/impatto sugli utenti (sui singoli utenti), NON è una caratteristica intrinseca di un sistema). Per questo non si può valutare l'usabilità senza il coinvolgimento degli utenti (di vario tipo) e sarebbe meglio valutarla in un contesto più simile possibile a una situazione reale.

Ricordiamo che l'usabilità non è solo un concetto di "correttezza funzionale" ma anche di "facilità d'uso".

---

Non si può parlare di impatto se non si parla di su che sistema socio-tecnico dovrà avere un impatto.

## 1.4 Sistema socio-tecnico

Banalmente possiamo dire che è un contesto di umani che lavorano assieme tramite delle tecnologie. Quindi:

- è un sistema, "un insieme di elementi interrelati ed eventualmente mutuamente dipendenti che, agli occhi di un osservatore esterno, appaiono come un'entità unitaria ma collettiva, con caratteristiche e comportamento proprio, solitamente autonomo ed intenzionale (cioè volto ad un obiettivo)";
- Un sistema in cui la componente umana (sociale) e quella tecnica (tecnologica) sono inestricabilmente legate tra loro e la loro interazione porta a fenomeni emergenti imprevedibili. Attenzione che tecnica è un termine generico proprio per intendere tutti gli strumenti, che siano fisici o dell'ingegno. In più si parla di interdipendenza perchè lo strumento

è fermo senza qualcuno che lo usa, l'umano è fermo se non ha uno strumento per agire;

- è un concetto di invarianza di scala, ovvero il concetto non cambia in base alla grandezza dell'ambiente sociale

Come progettisti dobbiamo essere pronti all'imprevisto, ovvero all'utente che usa lo strumento non nel modo prescritto.

STS<sup>2</sup> thinking: un approccio che è consapevole che le due componenti si integrano bene (fit) e si "ottimizzano" solo congiuntamente in configurazioni subottime (joint optimization). Bisogna quindi pensare al sistema nella sua interezza e come le parti si integreranno.

Non si deve pensare solo alla qualità degli strumenti ma anche al contesto nel quale vanno usati. Ad esempio la soddisfazione del lavoratore può migliorare l'interazione. Questo va a contribuire alla qualità della configurazione.

L'introduzione di una tecnologia in un contesto (sociale) è parte di un processo di cambiamento operante su più piani. Seguendo ciò che è detto prima si capisce che bisogna pensare non solo alla nostra tecnologia a livello funzionale (sistema tecnico) quando pensiamo all'interazione ma anche a dove andrà inserita e come migliorare quel contesto (sistema sociale).

#### 1.4.1 Proprietà emergenti

Sono di due tipi:

- **Sono proprietà funzionali**, riguardano il funzionamento dell'intero sistema una volta che tutte le sue parti, assemblate come devono, funzionano bene.
- **Sono proprietà non funzionali** che riguardano quanto bene opera il sistema in un determinato ambiente/contesto. Un elenco non esaustivo è affidabilità, sicurezza, performance, sicurezza, usabilità, comfort. (E.g. le informazioni

---

<sup>2</sup>Socio-technical system

sono giuste ma non aggiornate in un sistema medico, perchè? Perchè non era comodo farlo, quindi veniva visto più come un obolo rispetto ad uno strumento utile, vuole dire che è progettato male a livello di usabilità)

*L'emergenza è quando la somma delle parti vale più (a livello di usabilità) delle parti singole (e.g. una bici e le sue parti)*

### 1.4.2 Conseguenza inattesa

Iniziamo con un esempio, l'effetto Peltzman: l'introduzione obbligatoria del casco per i ciclisti rendeva questi più spericolati e anche i conducenti di macchina vedendone uno col casco. In più nei luoghi dove non c'era una distinzione netta tra marciapiede e carreggiata, gli automobilisti rallentavano automaticamente, riducendo il rischio di incidenti, questo perchè non potevano ideare una "zona sicura" da evitare ma per il resto vivere la strada come loro. Tutto ciò perchè gli umani tendono a bilanciare il rischio.

---

#### Conseguenza inattesa

Una conseguenza inattesa è quella cosa che può anche andare in maniera contro intuitiva rispetto alla progettazione, e.g. un social network fatto per conoscersi che diventa la base per le proteste di Washington.

---

Un tipo di conseguenza inattesa è il overreliance, ovvero il fatto di affidarsi maggiormente allo strumento a causa dei feedback positivi. Ci sono due tipi:

- **Overdependence:** mancanza di autonomia, abuso e uso al di là dei bisogni effettivi, mancanza o ignoranza di un piano di contingenza, ovvero come eseguire un compito senza l'uso della tecnologia in esame.

- **Overconfidence:** pensare che non ci saranno mai problemi, non ci saranno mai danni derivanti dall'uso e non si potrà mai sbagliare.

La **complacency** è la fiducia che il sistema tecnico funzionerà sempre come è stato progettato, riducendo così l'attenzione durante l'utilizzo (e.g. le macchine a guida autonoma che richiedono l'attenzione del conducente ma questo le ignora perché sicuro del funzionamento). è legata ai processi di monitoraggio.

L'**automation bias** è l'eccessiva fiducia nella risposta del supporto alle decisioni e quindi causa di errori di omissione o di azione quando i sistemi sono imperfetti (e.g. la calcolatrice che deve rispondere correttamente, non può essere altrimenti). è legato ai processi decisionali.

Le conseguenze inattese non sono intrinsecamente positive o negative, è solo la registrazione di un effetto non previsto.

Progettare sistemi usabili è progettare per l'uso, quindi per qualcosa che il progettista non controlla e che dipende dall'utente e da quello che fa. Per progettare sistemi usabili non esistono metodologie, è meglio:

- imparare per imitazione e per esperienza, diventando così capaci di basarsi sulla seconda per allontanarsi dalla prima
- essere creativi ma non troppo per non disorientare l'utente che non avrebbe basi conoscitive
- valutare il proprio sistema coinvolgendo utenti veri
- essere progettisti responsabili, ovvero ragionare sulle conseguenze impreviste, Essere progettisti responsabili significa sapere che il proprio sistema sarà il componente di un sistema socio-tecnico che può stravolgere o comunque modificare.
- aspettarsi che il lavoro possa avere conseguenze inattese e lavorare per minimizzarne l'impatto

## 1.5 Usabilità

### 1.5.1 Bassa usabilità = danni e problemi

Vari tipi di bassa usabilità:

- gli utenti non capiscono come svolgere i propri compiti con il sistema
- il sistema presenta un'eccessiva quantità di funzionalità e opzioni (low use)
- gli utenti non capiscono cosa il sistema stia facendo (poca trasparenza)

### 1.5.2 La colpa è dell'utente o del progettista?

La prima idea è che se il progettista ha creato il sistema a prova di stupido, ci siano tutte le guardie, gli avvisi e le informazioni necessarie allora la colpa sia dell'utente se qualcosa va storto. Nella storia di Dedalo e Icaro viene automatico pensare che Dedalo (il progettista) ha ragione, ha detto tutto a suo figlio e l'altro comunque è andato contro di ciò.

Certo, la colpa è qualcosa che può essere arginato tramite i termini d'uso, "qualsiasi utilizzo fuori da queste linee guida non è nostro problema", per la legge saremmo a posto ma ciò non cambia un dettaglio: se l'utente ha sbagliato c'è la possibilità che siamo stati noi a progettare male l'esperienza. "Mica gli ho detto io di mettere le dita così vicino alla lama dell'affettatrice" ma hai pensato "se il salume diventasse troppo piccolo le dita si avvicinerebbero troppo alla lama, meglio inserire un sistema comodo per spostarlo"?

Gli oggetti ben progettati sono facili da interpretare e comprendere: contengono indizi visibili (affordances) del loro funzionamento. Vedremo successivamente cosa voglia dire. Un esempio è l'indicatore in tempo reale di robustezza della password.

### **Affordance (invito alluso)**

La relazione tra le azioni possibili di un agente in un determinato ambiente.

---

Gli oggetti progettati male possono essere difficili e frustranti da usare: non offrono indizi o ne danno di sbaglianti, oppure sono stati progettati curando l'estetica più che la funzionalità.

#### **1.5.3 Come capire a livello generico se l'interfaccia è chiara?**

Uno dei metodi più semplici è tracciare una linea tra le interazioni per vedere quanto rispetti un ordine di lettura. Possiamo anche basarci su quanti ostacoli alla comprensione ci siano, ad esempio pulsanti che non fanno capire cosa facciano, spiegazioni verbose e non basate su simboli, rendendo difficile l'interpretazione per una persona che non sa leggere la lingua.

#### **1.5.4 Concetti estremi di usabilità**

##### **Porta di Norman**

<https://www.youtube.com/watch?v=yY96hTb8WgI>

Una porta di Norman è una porta il cui design suggerisce un'azione contraria a quella da fare o che richiede un segnale per spiegare come usarla.

Uno dei principi che possono andare in correzione qua è la "visibilità", ovvero la possibilità di scoprire quali azioni possano essere eseguite. Prendiamo un attimo come esempio i touchpad del computer: guardandoli non permettono di comprendere cosa

un singolo, doppio o triplo tocco possano fare, non c'è quindi scopribilità.

L'altro è il "feedback", ovvero un segnale che permetta la comprensione di cosa sia successo.

"La porta ideale è quella per la quale non devo pensare come aprirla"

---

### **Porta di Norman**

La tendenza del progettista a concepire oggetti o sistemi che non invitano all'uso (e all'uso corretto) sulla base di elementi visuali o indicazioni chiare (affordance), affidandosi quindi solo alla memoria, all'esperienza o all'intuizione e incontrando spesso il disorientamento o il fraintendimento dell'utente.

### **Discoverability (visibilità)**

Permette all'utente di individuare facilmente qual è la funzione dell'oggetto, e quindi riconoscere se è adatto allo scopo prefissato. È il risultato dell'applicazione di tutti gli altri principi elencati.

### **Feedback (reazione)**

È l'insieme di risposte che il sistema comunica all'utente, in base alle azioni effettuate fino a quel momento.

---

### **Toilette di Floyd**

È la serie di istruzioni della toilette in Odissea nello Spazio.

Una toilette di floyd è un qualcosa di progettato in maniera così poco intuitiva per il quale devo fornire una serie di istruzioni.

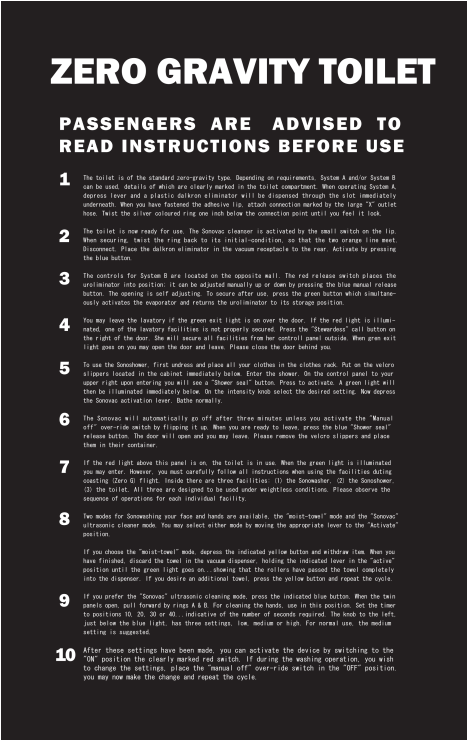


Figura 1.2: Lista di istruzioni della toilette

Toilette di Floyd

Tendenza del progettista a complicare affari semplici e a pretendere che basti descrivere una procedura per renderla ovvia a qualsiasi utente.



## 1.6 Sei concetti centrali

### Visibilità

Permette all'utente di individuare facilmente qual è la funzione dell'oggetto, e quindi riconoscere se è adatto allo scopo prefissato. È il risultato dell'applicazione di tutti gli altri principi elencati.

### Modello concettuale

È il modello che l'utente ha dell'oggetto in questione; il modello può essere superficiale e riferirsi alla sola conoscenza di relazione tra input e output, ma può anche essere più approfondito, arrivando a conoscere anche tutti i passaggi intermedi a livello macchina. Più il modello concettuale è fedele al funzionamento reale, maggiore è la probabilità di successo nell'interazione tra utente e macchina.

### Affordance (invito alluso) (vedere pagina 19)

Gibson: è la risorsa o supporto che l'ambiente offre all'utente

Norman: è una qualsiasi proprietà o qualità di un oggetto che definisce i suoi possibili utilizzi o rende chiaro come possa o debba essere usato.

Una affordance è qualsiasi proprietà di un oggetto che invita una persona competente all'azione mediata da tale oggetto.

### Significanti

Tutto ciò che aggiunge ulteriori indicazioni sull'esecuzione dell'azione, come direzione di movimento, senso di rotazione, ecc.

### Feedback (reazione)

È l'insieme di risposte che il sistema comunica all'utente, in base alle azioni effettuate fino a quel momento.

### Mapping

È il rapporto fra i comandi, il loro azionamento ed i risultati che ne derivano nel mondo esterno; permette all'utente di creare un collegamento diretto fra i comandi di controllo e le parti dell'oggetto di cui modificano rispettivamente lo stato.

#### 1.6.1 Definizione classica di interazione (ISO 9241)

**Un sistema interattivo** è una combinazione di componenti hardware e software che ricevono input da un utente umano e gli forniscono un output allo scopo di supportare l'effettuazione di un compito.



Figura 1.3: Mappa sullo scopo e sul metodo di completamento di un'azione

**Un'interfaccia** è l'insieme dei componenti di un sistema interattivo (software o hardware) che forniscono all'utente informa-

zioni e comandi per permettergli di effettuare specifici compiti attraverso il sistema.

In comune hanno il concetto di "effettuazione di un compito", da ciò si capisce che il sistema interattivo permette all'utente di conseguire un'obiettivo (compito).

## 1.7 Affordance

---

### Affordance (invito alluso)

Gibson: è la risorsa o supporto che l'ambiente offre all'utente

Norman: è una qualsiasi proprietà o qualità di un oggetto che definisce i suoi possibili utilizzi o rende chiaro come possa o debba essere usato.

Una affordance è qualsiasi proprietà di un oggetto che invita una persona competente all'azione mediata da tale oggetto.

---

Ci sono vari tipi di affordance:

- **Cognitive:** aiuta gli utenti attraverso le loro capacità cognitive (pensare, decidere, imparare, ricordare e conoscere)
- **Fisiche:** aiuta l'utente attraverso le sue azioni fisiche (cliccare, toccare, puntabile, gesticolare, muovere)
- **Sensoriali:** aiuta l'utente attraverso le sue azioni sensoriali (vedere, sentire e percepire)
- **Funzionali:** aiuta l'utente a realizzare il suo compito e a comprendere come usare lo strumento
- **Emotiva:** aiuta l'utente ad esperire certe emozioni

- **Sociale:** aiuta l'utente ad esperire certe funzioni sociali

Le affordance non sono esclusive, posso copartecipare all'intuizione, il problema è quando sono in contraddizione. Nel caso la progettazione non sia intuitiva gli utenti possono aggiungerle (ad esempio inserendo un cartello).

### 1.7.1 La legge di Fitt

---

#### Legge di Fitts

$$MT = a + b \log_2\left(\frac{D}{W} + 1\right)$$

dove MT è il tempo di movimento, D è la distanza dall'obiettivo e W è la grandezza dell'obiettivo

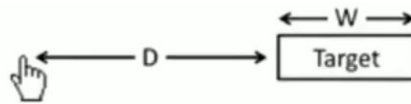


Figura 1.4: Legge di Fitt

---

Nella progettazione si può pensare di rendere più efficienti per la legge di Fitt le funzioni più frequenti e rendere meno efficienti quelli per i quali serve una forma di sicurezza.

## 1.8 Scheumorfismo

Uno scheumorfismo è un ornamento fisico o grafico apposto su un oggetto allo scopo di richiamare le caratteristiche estetiche di un altro. Un esempio è la calcolatrice digitale che cerca di

replicare gli schemi che ritroviamo su quelle fisiche, oppure all'applicazione di un e-reader che simula il girare fisicamente la pagina.



Figura 1.5: Esempio di scheumorfismo

Perchè oggi non viene più usato così pervasivamente? Perchè spesso questo approccio può essere estremamente utile per permettere all'utente un'interazione veloce ma porta con sé eventuali costi di performance. Oggi è stato sostituito da un approccio più flat.

Lo scheumorfismo può anche portare il problema di essere vincolati troppo dall'oggetto fisico.

## 1.9 Vincoli

Un vincolo può essere:

- **passivo/concettuale:** ad esempio concepisco un form con tre campi e non quattro, per vincolare l'inserimento di soli tre tipi di dato; non prevedo un campo "note", non prevedo un canale di ritorno (feedback) tra consumatore e azienda
- **attivo/funzionale:** ad esempio concepisco dei controlli per cui l'utente non può proseguire se non compila tutti i campi, o non li compila "correttamente" sulla base di regex come ad esempio per l'indirizzo mail o il codice fiscale)

I vincoli vanno sempre bene? No, se l'utente si sentisse troppo vincolato potrebbe iniziare a creare dei workaround. Questi vengono chiamati "desire path".

---

### **Desire Path**

Qualsiasi azione relativa all'esecuzione di un processo o di un compito (supportati dal sistema informatico) che non è prevista o descritta nei manuali di uso del sistema informatico e/o nei manuali che descrivono tale processo o procedura e che può bypassare l'uso del sistema o piegarlo ai propri fini.

"Percorso del desiderio", sono quei percorsi di interazione che descrivono il percorso ideale da parte dell'utente.

---