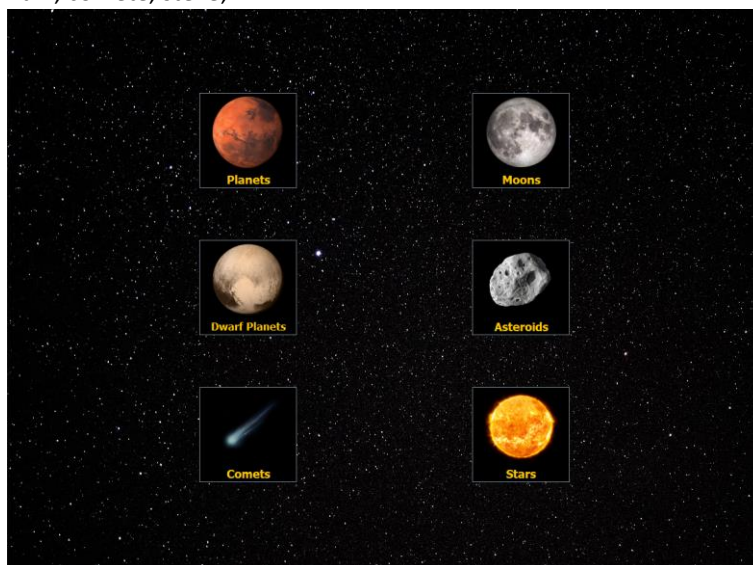


# Progetto Web-service Alberti-Bertelli

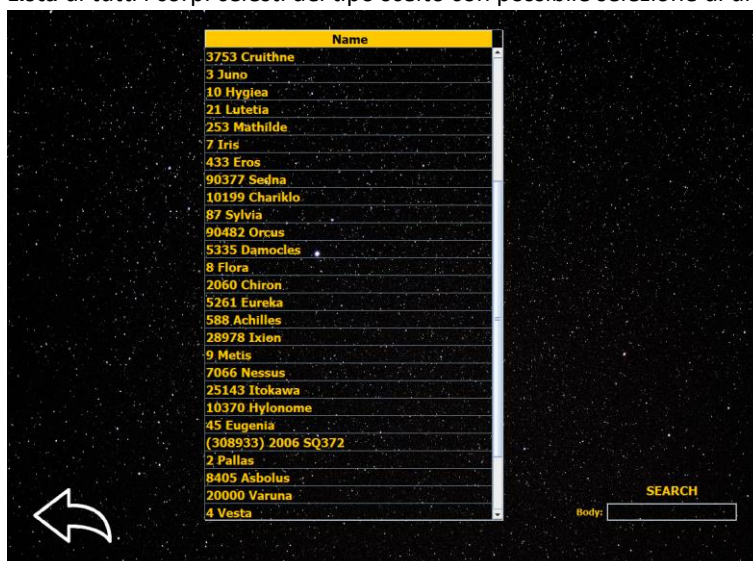
**Scopo del programma:** offrire una visione e descrizione dettagliata dei vari corpi celesti del sistema solare tramite un'interfaccia grafica che ne semplifica la navigazione.

## Funzionalità:

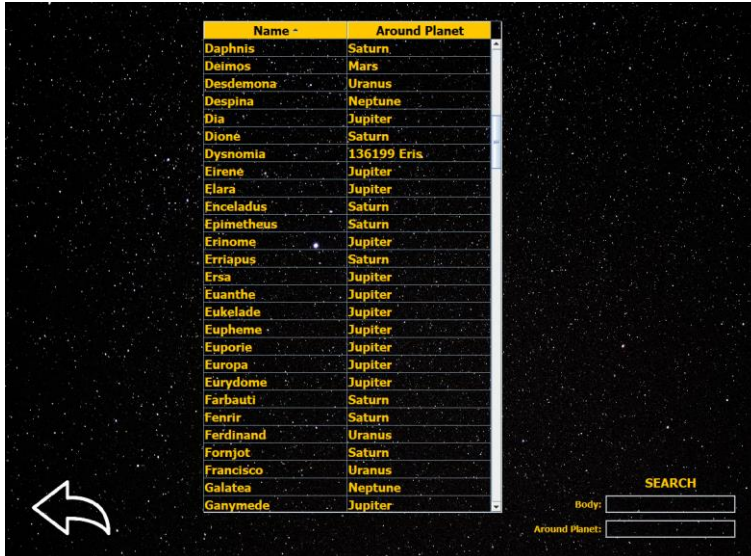
- Visualizzazione di informazioni relative ad un **preciso tipo di corpo celeste** tra: pianeti, lune, asteroidi, pianeti nani, comete, stelle;



- Lista di tutti i corpi celesti del tipo scelto con possibile selezione di uno singolo;



- Possibilità di **ordinamento** alfabetico (ascendente o discendente) della lista dei corpi del tipo scelto;



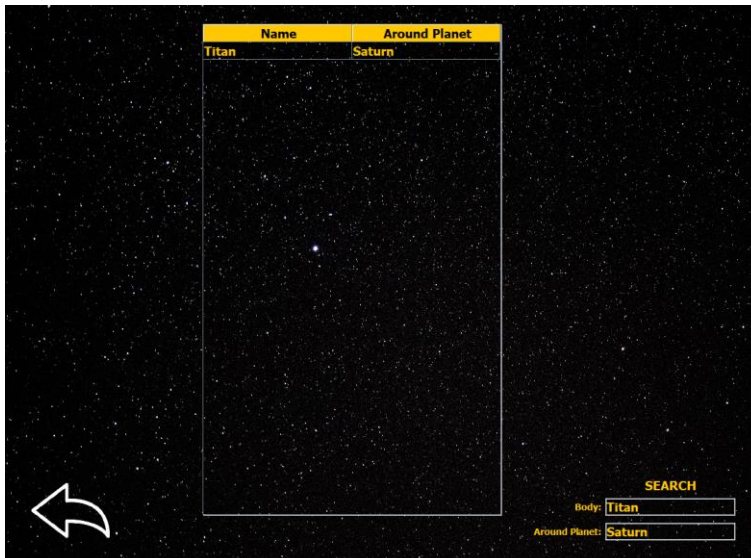
| Name       | Around Planet |
|------------|---------------|
| Daphnis    | Saturn        |
| Deimos     | Mars          |
| Desdemona  | Uranus        |
| Despina    | Neptune       |
| Dia        | Jupiter       |
| Dione      | Saturn        |
| Dysnomia   | 136199 Eris   |
| Eirene     | Jupiter       |
| Elara      | Jupiter       |
| Enceladus  | Saturn        |
| Epimetheus | Saturn        |
| Erinome    | Jupiter       |
| Erriapus   | Saturn        |
| Ersa       | Jupiter       |
| Euanthe    | Jupiter       |
| Eukelade   | Jupiter       |
| Eupheme    | Jupiter       |
| Euporie    | Jupiter       |
| Europa     | Jupiter       |
| Eurydome   | Jupiter       |
| Farbauti   | Saturn        |
| Fenrir     | Saturn        |
| Ferdinand  | Uranus        |
| Fornjot    | Saturn        |
| Francisco  | Uranus        |
| Galatea    | Neptune       |
| Ganymede   | Jupiter       |

SEARCH

Body:

Around Planet:

- Possibilità di **ricerca** di un determinato corpo celeste tramite il nome e, nel caso delle lune, del corpo attorno al quale ruotano;



| Name  | Around Planet |
|-------|---------------|
| Titan | Saturn        |

SEARCH

Body:

Around Planet:

- **Visualizzazione dettagliata** delle caratteristiche fisiche, orbitali e storiche di un determinato corpo celeste;

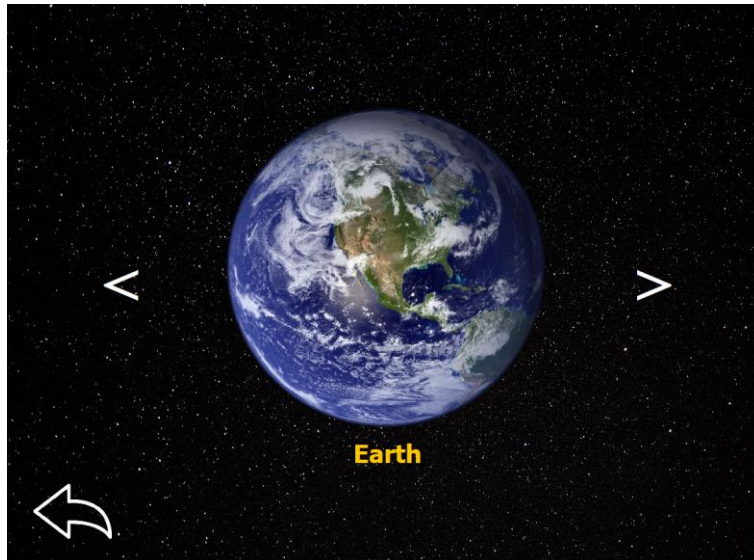


|                    |   |
|--------------------|---|
| English Name:      | Moon                                    |
| Semimajor Axis:    | 384400 km                               |
| Perihellion:       | 363300 km                               |
| Aphelion:          | 405500 km                               |
| Eccentricity:      | 0.0549                                  |
| Inclination:       | 5.145°                                  |
| Mass:              | $7.346 \times 10^{22}$ kg               |
| Vol:               | $2.1968 \times 10^{10}$ km <sup>3</sup> |
| Density:           | 3.344 g/cm <sup>3</sup>                 |
| Gravity:           | 1.62 m/s <sup>2</sup>                   |
| Escape:            | 2380.0 m/s                              |
| Mean Radius:       | 1737.0 km                               |
| Equa Radius:       | 1738.1 km                               |
| Polar Radius:      | 1736.0 km                               |
| Flattening:        | 0.0012                                  |
| Sidereal Orbit:    | 27.3217 days                            |
| Sidereal Rotation: | 655.728 h                               |
| Around Planet:     | Earth                                   |
| Axial Tilt:        | 6.68°                                   |
| Avg Temp:          | 0 K                                     |
| Main Anomaly:      | 0.0°                                    |
| Arg Periapsis:     | 0.0°                                    |
| Long Asc Node:     | 0.0°                                    |
| Body Type:         | Moon                                    |

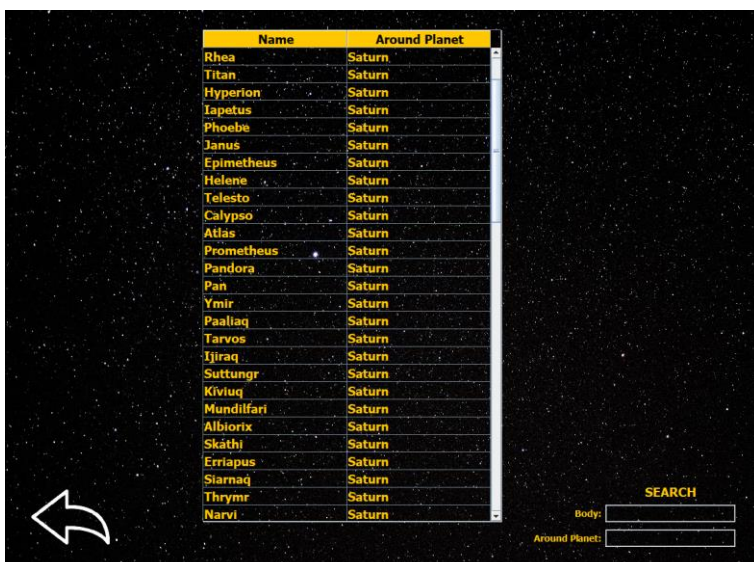
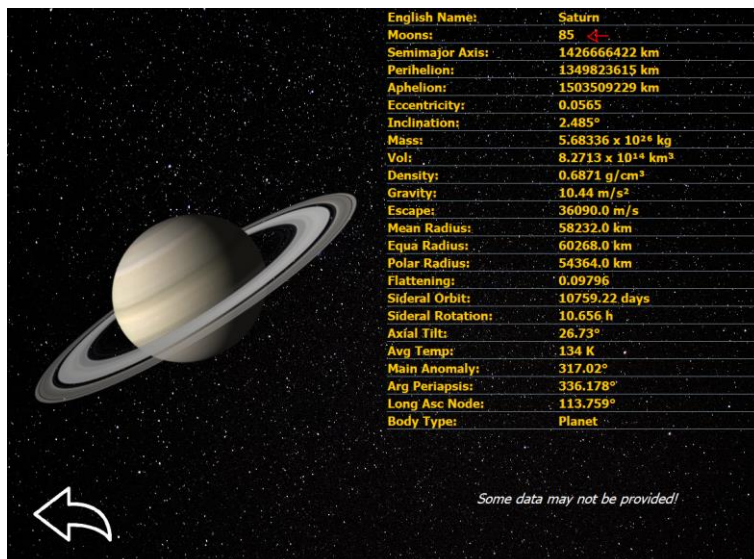
Some data may not be provided!



- Menu grafico di selezione per i pianeti;



- Possibilità di visualizzare le lune di un determinato pianeta direttamente dalla visualizzazione dettagliata dello stesso.



## ***Web-service utilizzato***

Il web service utilizzato è <https://api.le-systeme-solaire.net/>, il quale permette di ricevere dati impostando vari filtri e modalità.

### **Tutti i corpi in una sola richiesta**

Un URL ritorna tutti i corpi nel database con tutti i dati:

<https://api.le-systeme-solaire.net/rest/bodies/>

### **Corpo per corpo**

Un URL ritorna tutti i dati relativi a un singolo corpo:

<https://api.le-systeme-solaire.net/rest/bodies/{id}>

### **Tutti i numeri conosciuti**

Un URL ritorna il numero conosciuto di ogni tipo di corpo:

<https://api.le-systeme-solaire.net/rest/knowncount/>

### **Numeri conosciuti, corpo per corpo**

Un URL ritorna il numero conosciuto di un singolo tipo di corpo:

<https://api.le-systeme-solaire.net/rest/knowncount/{id}>

## Parametri e azioni

| # | Parametro | Azione   |
|---|-----------|--|
| 1 | data      | Dati che si vogliono ricevere (separati da virgola).<br>Esempio: <code>data=id,semimajorAxis,isPlanet</code>   |
| 2 | exclude   | Dati che si vogliono escludere (separati da virgola).<br>Esempio: <code>exclude=id,isPlanet</code>   |
| 3 | order     | Ordinamento dato un singolo campo, specificandone il verso (separati da virgola).<br>Esempio: <code>order=semimajorAxis,asc</code><br>NB : Solo un singolo campo è permesso.   |
| 4 | page      | Numero di pagina ( $n \geq 1$ ) e dimensione della pagina ( $dim \geq 1$ con 20 come predefinito) (separati da virgola).<br><br>Esempio: <code>page=1,10</code><br>NB: Non puoi usare "page" senza "order"!  |
| 5 | rowData   | Trasforma gli oggetti in record.<br>Esempio: <code>rowData=true</code><br>NB: Il valore predefinito è "false".   |
| 6 | filter[]  | Filtri da applicare. Ogni filtro consiste in un campo, un operatore e un valore (separati da virgola).<br>Esempio: <code>filter[]=id,eq,mars</code><br><br>Operatori accettati: <ul style="list-style-type: none"><li>▪ cs (like)</li><li>▪ sw (start with)</li><li>▪ ew (end with)</li><li>▪ eq (equal)</li><li>▪ lt (less than)</li><li>▪ le (less or equal than)</li><li>▪ ge (greater or equal than)</li><li>▪ gt (greater than)</li><li>▪ bt (between)</li></ul> E tutti gli operatori opposti: ncs - nsw - new - neq - nlt - nle - nge - nggt - nbt.<br>NB: se c'è un filtro non valido, tutti i filtri sono ignorati. |
| 7 | satisfy   | Tutti i filtri devono corrispondere (predefinito).<br>Esempio: <code>satisfy=any</code><br>NB: L'unico valore permesso è "any".  |

## XSD: spiegazione delle scelte implementative

Creazione delle classi Java dall'XSD tramite il compilatore XJC.

### Struttura delle classi:

- Oggetto radice: **Bodroot**  
Contiene un vettore di Bodies
  - o Oggetto: **Bodies**  
Contiene tutte le caratteristiche del corpo celeste sotto forma di attributi semplici, ad eccezione di 4:
    - Oggetto **Moons**  
Contiene un vettore di Element
      - Oggetto **Element**  
Contiene nome e link di raggiungimento della luna
    - Oggetto **Mass**  
Contiene il valore e l'ordine di grandezza
    - Oggetto **Vol**  
Contiene il valore e l'ordine di grandezza
    - Oggetto **AroundPlanet**  
Contiene id e link di raggiungimento del corpo

### Raggiungimento dei valori dei dati:

Metodo `makeBodies("nomeBody")` : riceve un file JSON come risultato del URL con cui si è fatta la richiesta e lo converte in un file XML su cui è possibile eseguire l'unmarshalling, che produce un oggetto di tipo `Bodroot` da ritornare

-Attributi semplici:

```
List<Serializable> attrSemplice = api.makeBodies("nomeBody").getBodies().get(posizione);  
String valore = "" + attrSemplice.getNomeAttributo();
```

-Attributi composti:

```
List<Serializable> attrComposto = api.makeBodies("nomeBody").getBodies().get(posizione).getNomeAttributo().getContent();  
String valore = "" + ((JAXBElement<attrComposto>) attrComposto.get(posizione)).getValue();
```