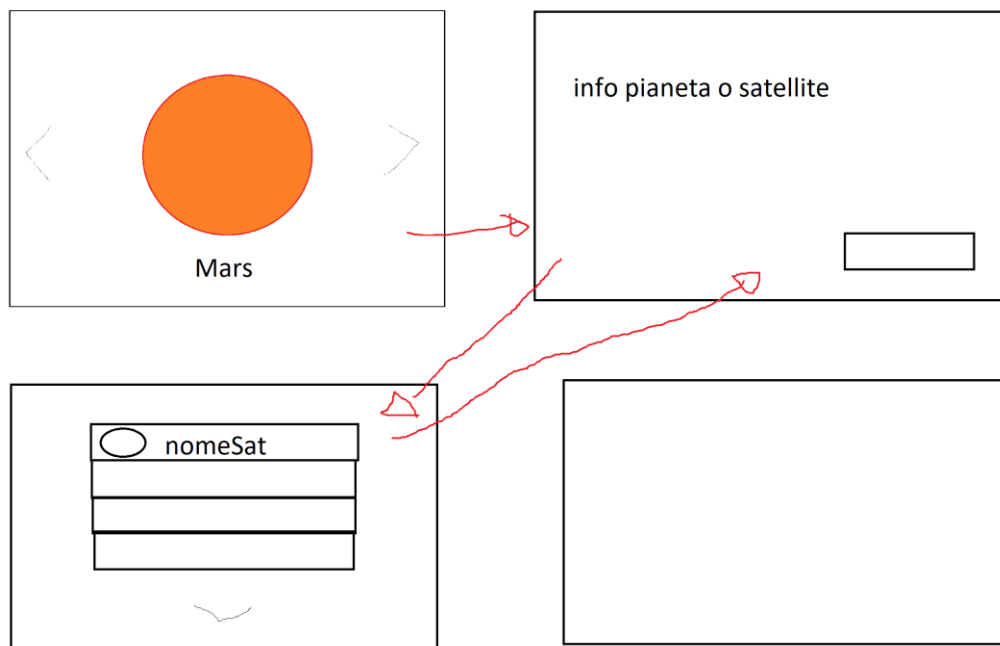


Progetto Web-service Alberti-Bertelli

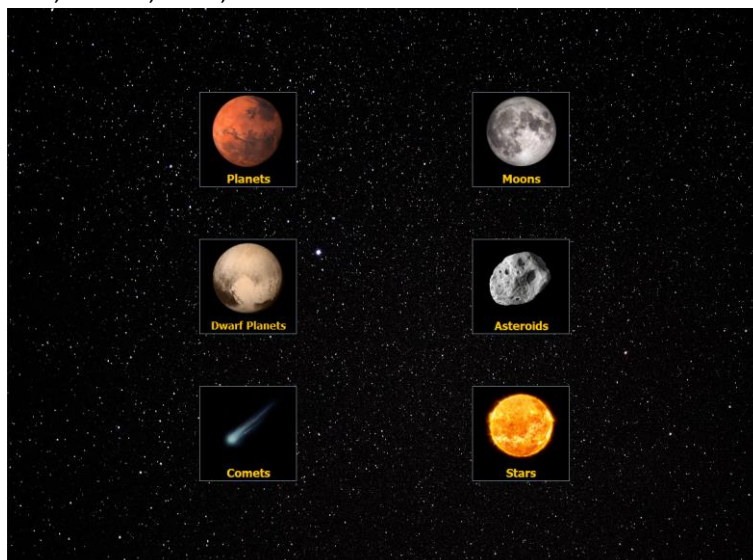
Scopo del programma: offrire una visione e descrizione dettagliata dei vari corpi celesti del sistema solare tramite un'interfaccia grafica che ne semplifica la navigazione.

Bozza del programma:

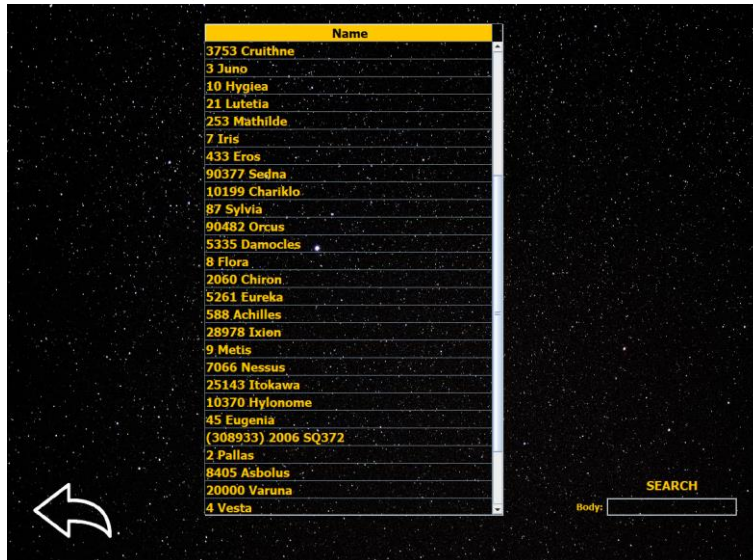


Funzionalità:

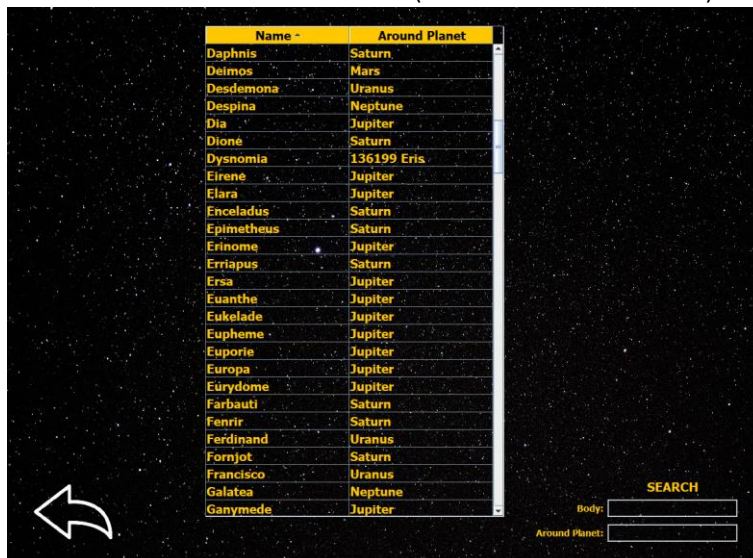
- Visualizzazione di informazioni relative ad un **preciso tipo di corpo celeste** tra: pianeti, lune, asteroidi, pianeti nani, comete, stelle;



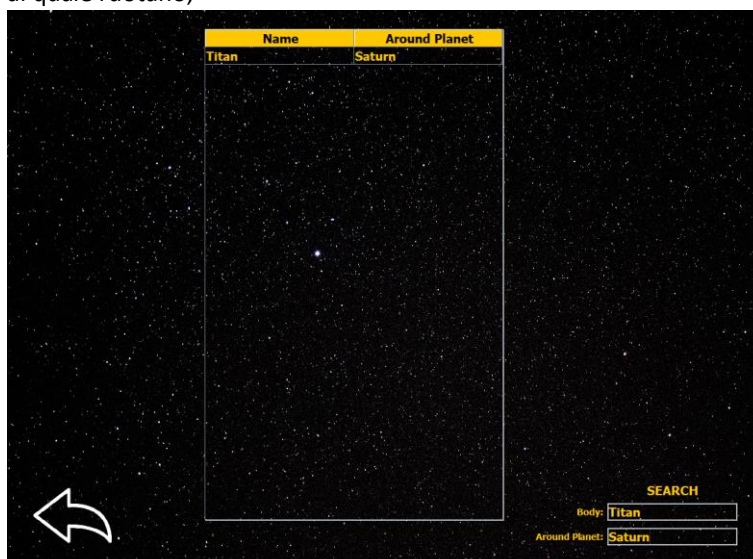
- **Lista** di tutti i corpi celesti del tipo scelto con possibile selezione di uno singolo;



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



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



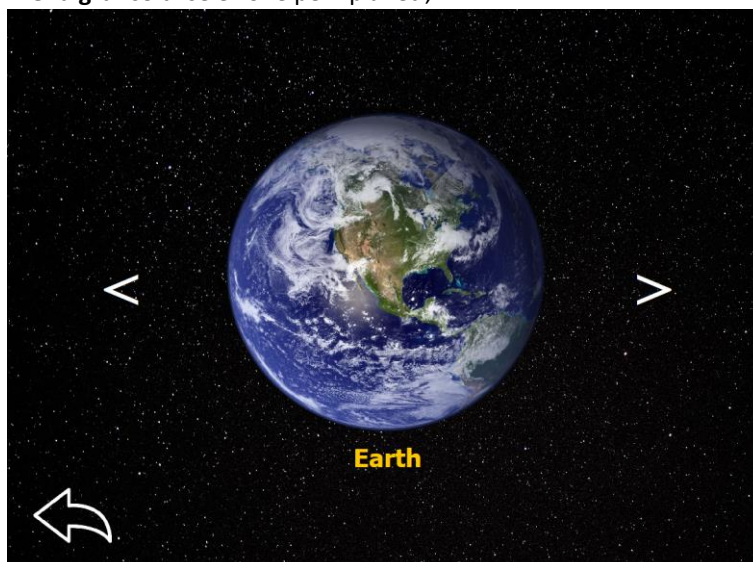
- **Visualizzazione dettagliata** delle caratteristiche fisiche, orbitali e storiche di un determinato corpo celeste;



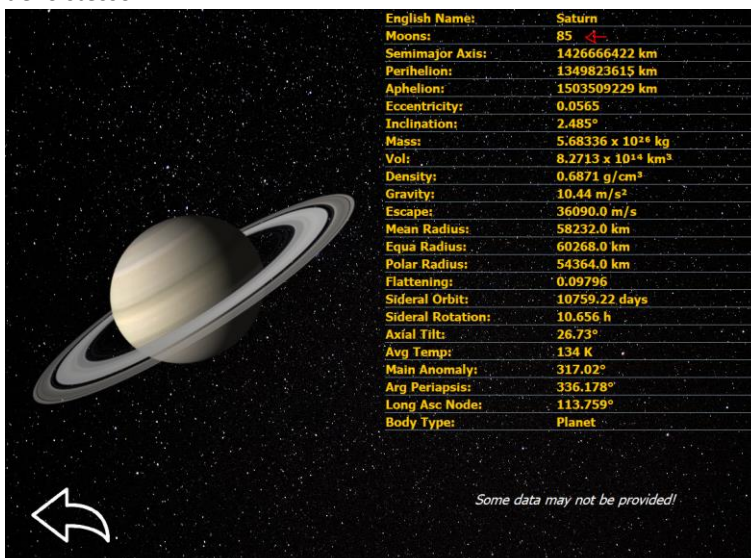
English Name:	Moon
Semimajor Axis:	384400 km
Perihelion:	363300 km
Aphelion:	405500 km
Eccentricity:	0.0549
Inclination:	5.145°
Mass:	7.346×10^{22} kg
Vol:	2.1968×10^{10} km ³
Density:	3.344 g/cm ³
Gravity:	1.62 m/s ²
Escape:	2380.0 m/s
Mean Radius:	1737.0 km
Equa Radius:	1738.1 km
Polar Radius:	1736.0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


Some data may not be provided!

- **Menu grafico di selezione** per i pianeti;



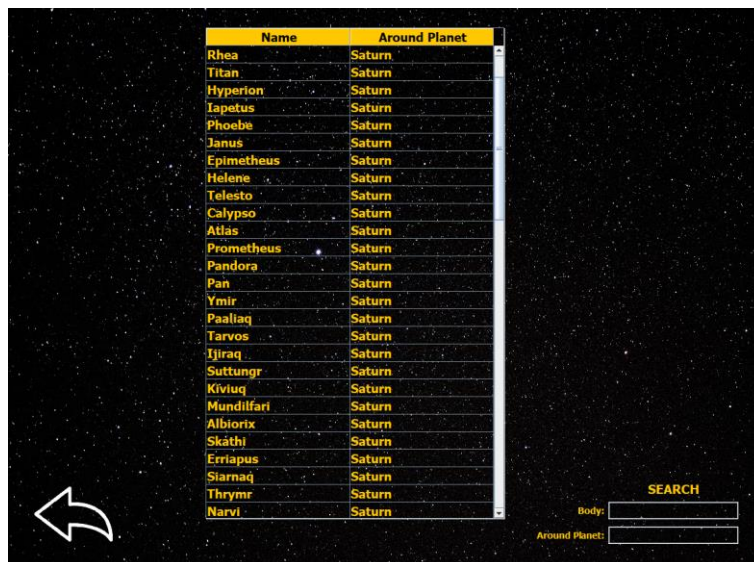
- Possibilità di visualizzare le **lune di un determinato pianeta** direttamente dalla visualizzazione dettagliata dello stesso.



English Name:	Saturn
Moons:	85 
Semimajor Axis:	1426666422 km
Perihelion:	1349823615 km
Aphelion:	1503509229 km
Eccentricity:	0.0565
Inclination:	2.485°
Mass:	5.68336×10^{26} kg
Vol:	8.2713×10^{14} km ³
Density:	0.6871 g/cm ³
Gravity:	10.44 m/s ²
Escape:	36090.0 m/s
Mean Radius:	58232.0 km
Equa Radius:	60268.0 km
Polar Radius:	54364.0 km
Flattening:	0.09796
Sideral Orbit:	10759.22 days
Sideral Rotation:	10.656 h
Axial Tilt:	26.73°
Avg Temp:	134 K
Main Anomaly:	317.02°
Arg Periaipsis:	336.178°
Long Asc Node:	113.759°
Body Type:	Planet

Some data may not be provided!





Name	Around Planet
Rhea	Saturn
Titan	Saturn
Hyperion	Saturn
Iapetus	Saturn
Phoebe	Saturn
Janus	Saturn
Epimetheus	Saturn
Helene	Saturn
Telesto	Saturn
Calypso	Saturn
Atlas	Saturn
Prometheus	Saturn
Pandora	Saturn
Pan	Saturn
Ymir	Saturn
Paaliaq	Saturn
Tarvos	Saturn
Ijiraq	Saturn
Suttungr	Saturn
Kiviuq	Saturn
Mundilfari	Saturn
Albiorix	Saturn
Skathi	Saturn
Erriapus	Saturn
Siarnaq	Saturn
Thrymr	Saturn
Narvi	Saturn

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). Esempio: <code>data=id,semimajorAxis,isPlanet</code>
2	exclude	Dati che si vogliono escludere (separati da virgola). Esempio: <code>exclude=id,isPlanet</code>
3	order	Ordinamento dato un singolo campo, specificandone il verso (separati da virgola). Esempio: <code>order=semimajorAxis,asc</code> 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). Esempio: <code>page=1,10</code> NB: Non puoi usare "page" senza "order"!
5	rowData	Trasforma gli oggetti in record. Esempio: <code>rowData=true</code> 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). Esempio: <code>filter[]=id,eq,mars</code> Operatori accettati: <ul style="list-style-type: none">▪ cs (like)▪ sw (start with)▪ ew (end with)▪ eq (equal)▪ lt (less than)▪ le (less or equal than)▪ ge (greater or equal than)▪ gt (greater than)▪ bt (between) E tutti gli operatori opposti: ncs - nsw - new - neq - nlt - nle - nge - nggt - nbt. NB: se c'è un filtro non valido, tutti i filtri sono ignorati.
7	satisfy	Tutti i filtri devono corrispondere (predefinito). Esempio: <code>satisfy=any</code> 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();
```