



Alumne:

### **Fitxer sondes.html:**

Estarà en la carpeta arrel.

Ja el dono fet.

Potser li falta alguna cosa...

moniato

Comprovar distancia de seguretat

Comprovar si estarà a l'abast

Waypoints en que estarà a l'abast

Veure ruta ordenada per distancia

Marcar ruta com finalitzada

### **Fitxer PalasAteneaConstants.js:**

Estarà en la carpeta arrel.

Ja té totes les constants que necessiteu.

### **Fitxer ordenacio.js:**

Estarà en la carpeta **llibries**.

Per si has de ficar algo...

### **Fitxer dates.js:**

Estarà en la carpeta **llibries**.

Per si has de ficar algo...



Alumne:

**Classe ObjecteEstelar:**Estarà dins de la carpeta **classes**.

PROPIETATS	
id	Propietat privada. String. Si l'usuari no fiques cap valor a llavors hauria de valdre "-1". Has de fer que si en l'invocació del constructor el programador no assignés cap valor al id, a llavors se li assigni com a valor per defecte el "-1".
nom	Propietat privada. String.
tipus	Propietat privada. Integer. La posició en <b>PalasAteneaConstants.TIPUSOBJECTESESTELARS</b> .
posicio	Propietat privada. Array de 2 posicions. La 1a serà la <b>coordenada X</b> i la 2a la <b>coordenada Y</b> (les 2 coordenades seran números).
visitat	Propietat privada. Boolean. Quan es crea l'objecte sempre serà <b>false</b> .

MÈTODES	
constructor(id, nom, tipus, posX, posY)	Hem d'evitar que ningú amplii l'objecte afegint/eliminant propietats i mètodes.
calcularDistancia()	Retornarà la multiplicació de les 2 coordenades que es guarden en la propietat <b>posicio</b> .
toString()	Retornarà el <b>nom</b> seguit del <b>tipus</b> (el valor que hi ha en <b>PalasAteneaConstants.TIPUSOBJECTESESTELARS</b> ) seguit de la <b>distancia</b> seguit del ID i del valor de <b>visitat</b> entre parèntesis.

En el cas que s'hagi d'afegir cap més mètode, getter, setter,... afegiu només els estrictament necessaris.



Alumne:

**Classe Sonda:**Estarà dins de la carpeta **classes**.

PROPIETATS	
id	Propietat privada. String.
nom	Propietat privada. String.
tipus	Propietat privada. La posició en <b>PalasAteneaConstants.TIPUSSONDES</b> .
dataAlta	Propietat privada. Date.
dataBaixa	Propietat privada. Date.
imatge	Propietat privada. String.
rutes	Són les rutes que tindrà en la memòria la sonda. Propietat privada. Array on cada posició serà 1 ruta. 1 ruta estarà formada per un array d'objectes de tipus <b>ObjecteEstelar</b> .

MÈTODES	
constructor(id, nom, tipus, dataAlta, dataBaixa, imatge)	Hem d'evitar que ningú amplii l'objecte afegint/eliminant propietats i mètodes. Es poden fer servir les funcions que tenim en <b>llibreries/dates.js</b> per a crear una data en funció del valor del paràmetre que rebrà el constructor.
insertarNovaRuta()	S'ha de fer una nova ruta (array amb objectes de tipus <b>ObjecteEstelar</b> ) amb els <b>ObjecteEstelar</b> rebuts i insertar-la en <b>rutes[]</b> . No sabem de quants <b>ObjectesEstelars</b> estarà formada la nova ruta. Els <b>ObjectesEstelars</b> han de quedar en la ruta en el mateix ordre en que s'han rebut per paràmetre.



Alumne:

veureTempsDeServei()	Calcular quant temps té la sonda (restar <b>dataAlta</b> a <b>dataBaixa</b> ) en anys, mesos i dies. Resteu a saco els anys, mesos i dies. És igual si dona números negatius.
toString()	Retornarà el <b>nom</b> seguit del <b>tipus</b> (el valor que hi ha en <b>PalasAteneaConstants.TIPUSSONDES</b> ) seguit del ID entre parèntesis. Afegiu també el temps de servei que té la sonda.
veureRutaOrdenadaPerDistancia()	Retorna els <b>ObjecteEstelar</b> pels quals passarà la nau en funció de la distància. No hem de modificar <b>rutes[]</b> . Has d'ordenar els <b>ObjecteEstelar</b> de l'ultima ruta que hi ha en <b>rutes[]</b> en funció de la distància (distància = multiplicació de la coordenada X per la coordenada Y).
marcarRutaFinalitzada()	Quan la sonda hagi acabat la ruta posarà <b>visitat = true</b> en tots els <b>ObjecteEstelar</b> de la ruta realitzada (la qual serà l'ultima que hi hagi dins de <b>rutes[]</b> ). Retornarà la informació de la ruta (el <b>toString()</b> ) i dels <b>ObjecteEstelar</b> (el <b>toString()</b> ) per pintar-la per pantalla.

**MÈTODES QUE S'EXECUTARAN SEMPRE CONTRA L'ÚLTIMA RUTA  
QUE TINGUI LA SONDA EN LA PROPIETAT **rutes[]****

**CADA MÈTODE FARÀ SERVIR 1 DELS MÈTODES PROPIS DELS ARRAYS I NO ELS PODEU REPETIR**

comprovarDistanciaSeguretat()	T'has d'assegurar que tots els punts que visitarà la sonda (els <b>ObjecteEstelar</b> de l'ultima ruta que hi ha en <b>rutes[]</b> ) estan com a mínim a 5000 de distància ja que a menor distància la radiació que emeten aquests <b>ObjecteEstelar</b> danyaria la sonda.
comprovarSiEstaraALabast()	Per a poder rebre informació de la sonda aquesta ha d'estar com a màxima a una distància de 25000 en algun moment. En funció de la distància als <b>ObjecteEstelar</b> de la ruta comprova si ens podem comunicar amb la sonda en cap moment.
waypointsEnQueEstaraALabast()	Retornarà la informació dels <b>ObjecteEstelar</b> de la ruta pels quals passarà la nau que estiguin a distància de poder rebre informació (distància < 25000).

En el cas que s'hagi d'afegir cap més mètode, getter, setter,... afegiu només els estrictament necessaris.



Alumne:

Pantalla de **comprovarDistanciaSeguretat()**:

Comprovar distancia de seguretat: false

Comprovar distancia de seguretat

Comprovar si estarà a l'abast

Waypoints en que estarà a l'abast

Veure ruta ordenada per distancia

Marcar ruta com finalitzada

Pantalla de **comprovarSiEstaràALAbast()**:

Comprovar si estarà a l'abast: true

Comprovar distancia de seguretat

Comprovar si estarà a l'abast

Waypoints en que estarà a l'abast

Veure ruta ordenada per distancia

Marcar ruta com finalitzada

Pantalla de **waypointsEnQueEstaràALAbast()**:

Waypoints en que estarà a l'abast:

WAYPOINT **Jupiter** de tipus PLANETA a distància 10000 (ID: P-01, visitat = false)

WAYPOINT **Marte** de tipus PLANETA a distància 2500 (ID: P-03, visitat = false)

Comprovar distancia de seguretat

Comprovar si estarà a l'abast

Waypoints en que estarà a l'abast

Veure ruta ordenada per distancia

Marcar ruta com finalitzada



Alumne:

### Pantalla de **veureRutaOrdenadaPerDistancia()**:

Veure ruta ordenada per distancia:

WAYPOINT **Marte** de tipus PLANETA a distància 2500 (ID: P-03, visitat = false) en la posició 0 i a una distància de 2500

WAYPOINT **Jupiter** de tipus PLANETA a distància 10000 (ID: P-01, visitat = false) en la posició 1 i a una distància de 10000

WAYPOINT **Saturno** de tipus PLANETA a distància 40000 (ID: P-02, visitat = false) en la posició 2 i a una distància de 40000

WAYPOINT **Cinturón de Kuiper** de tipus ASTEROIDE a distància 90000 (ID: C-01, visitat = false) en la posició 3 i a una distància de 90000

Comprovar distancia de seguretat

Comprovar si estarà a l'abast

Waypoints en que estarà a l'abast

Veure ruta ordenada per distancia

Marcar ruta com finalitzada

### Pantalla de **marcarRutaFinalitzada()**:

Marcar ruta com finalitzada:

SONDA **Voyager 1** de tipus SISTEMA SOLAR (ID: Voy-1, temps de servei 48 anys -8 mesos i -4 dies)

WAYPOINT **Marte** de tipus PLANETA a distància 2500 (ID: P-03, visitat = true)

WAYPOINT **Jupiter** de tipus PLANETA a distància 10000 (ID: P-01, visitat = true)

WAYPOINT **Saturno** de tipus PLANETA a distància 40000 (ID: P-02, visitat = true)

WAYPOINT **Cinturón de Kuiper** de tipus ASTEROIDE a distància 90000 (ID: C-01, visitat = true)

Comprovar distancia de seguretat

Comprovar si estarà a l'abast

Waypoints en que estarà a l'abast

Veure ruta ordenada per distancia

Marcar ruta com finalitzada



Alumne:

**Fitxer sondesInicialitzador.js:**1r creareu 1 sonda anomenada **Voyager** amb les següents dades:

PROPIETAT	VALOR
id	Voy-1
nom	Voyager 1
tipus	4
dataAlta	1977-08-05
dataBaixa	2025-00-01
imatge	Voyager.jpg

Després creareu 4 waypoints (objectes de tipus **ObjecteEstelar**) amb les següents dades:

OBJECTE	PROPIETAT	VALOR
Jupiter	id	P-01
	nom	Jupiter
	tipus	1
	posicio[0]	100
	posicio[1]	100
Saturno	id	P-02
	nom	Saturno
	tipus	1
	posicio[0]	200
	posicio[1]	200
Kuiper	id	C-01
	nom	Kuiper
	tipus	3
	posicio[0]	300
	posicio[1]	300



Alumne:

Marte	id	P-03
	nom	Marte
	tipus	3
	posicio[0]	50
	posicio[1]	50

Després ficareu els 4 waypoints en una ruta fent servir **Sonda.insertarNovaRuta()**.

Completa el codi de les funcions **comprovarDistanciaSeguretat()**, **comprovarSiEstaraALAbast()**, **waypointsEnQueEstaraALAbast()**, **veureRutaOrdenadaPerDistancia()** i **marcarRutaFinalitzada()** per veure en el paràgraf de **sondes.html** el resultat de l'execució de aquests mètodes de la classe **Sonda**.