Zero Atmosphere.

En un mundo futuro, el agente de policía López es enviado a una remota colonia en la tercera luna de Júpiter para investigar una serie de acontecimientos en una mina.

El viaje hasta la tercera luna de júpiter, es largo y muy solitario. Según esto último se decide que el viaje se reparta en diferentes capítulos.

Capitulo I

La empresa propietaria de la mina, *Western Moon,* decide crear una aplicación que gestione todo el viaje del agente López. Como primera medida se toma la decisión de proporcionarle los siguientes elementos:

1) una nave
2) una tripulación
3) una mascota.
4) Vehículos para moverse por los valles de la luna en sus investigaciones.

La factoría de software *Fabes & Ham Corporation* ha contratado unos equipos especializados en desarrollo sw java para estos casos. Los equipos son SK Telecom, Turborrotaos, Los Diabólicos de Java, grupo guipuchi, Versión Beta y PHP. Cada equipo tendrá un coordinador.

Cada equipo desarrollará una versión de la aplicación. Además de codificar la aplicación, uno o dos miembros de cada equipo se dedicará a realizar pruebas de test de la aplicación. Estas pruebas consistirán entre otras, generar métodos para carga de los datos a mostrar cuando la aplicación se presente.

Habrá una fecha tope donde se congelarán las versiones de la aplicación y se enviarán en formato zip al aula virtual. Se tendrá que enviar un documento pdf con las pruebas y métodos desarrollados para las pruebas, además de un diagrama UML de toda la aplicación.

El agente López trabajó años atrás en una investigación referente al desempeño de los equipos en *Fabes & Ham Corporation.* Los resultados de López fueron sorprendentes. Como resultado de la investigación se determinó lo siguiente:

1. Cada capitán debe acordar con cada componente del equipo la tarea que debe hacer.
2. Se debe comunicar a la dirección de *Fabes & Ham Corporation n*ombre de componente del equipo y funciones asignadas.
3. Cada función ó tarea NO debe ser bloqueante para el resto. A modo de ejemplo si un componente se dedica ha realizar labores de testing y no las realiza, el no hacer esa tarea no bloquea procesar el resto de tareas y penaliza de forma individual a quien no haga su trabajo.
4. De igual forma, cuando cada equipo presente su trabajo a la dirección de *Fabes & Ham Corporation* si algún componente del equipo falta también le penalizará de forma individual
5. Las labores de coordinación, a pesar de no ser técnicas se trata de tareas de gestión y también puntuarán. Es decir un capitán que no sabe o desconoce las funciones de cada componente también le penalizará.
6. La investigación de López también desveló un dato escalofriante; todos aquellos que, habían utilizado algún mecanismo de IA para realizar su tarea, al final no fue un beneficio sino todo lo contrario y demostró que solo con el esfuerzo se consiguen hitos. Desde luego los descubrimientos de López no dejan indiferente a nadie

Datos adicionales:

1. Fecha de entrega de la aplicación: Por determinar
2. Fecha presentación: Por determinar
3. Capitulos II, III, IV, ……: Por determinar. Se irá avisando según surjan los acontecimientos. Se debe tener en cuenta que el viaje es largo y la nave de López puede sufrir ataques, choques con asteriodes, etc. Para ello la dirección de *Fabes & Ham Corporation* irá informando de las nuevas necesidades de la aplicación.

Datos Técnicos:

Las mascotas pueden ser gatos, perros o incluso se ha llegado a pensar en tener pájaros. Se debe tener en cuenta que las mascotas duermen, comen,se comunican mediante sonidos o bien movimientos de alguna extremidad. Las mascotas pueden tener dueño y por tanto se debe crear una clase PropietarioMascota().

Toda la tripulación tiene características comunes pero bien es cierto que todos son diferentes.
La tripulación esta compuesta (además de las mascotas) por Laura, Carmen, Federico y López. No obstante existe también un retén de soldados y de mineros.
Como medios de transporte tienen aerobikes (dos personas), aerocars (más de dos personas) y turbojets (elemento de transporte). Un vehiculo puede ser responsabilidad de un miembro del equipo por tanto se debe crear una clase *responsableVehiculo().* Cada uno de estos elementos tiene características propias y comunes. Se permite la imaginación de cada equipo para que determine cuales son los atributos de cada elemento, si bien deben tener tres elementos comunes y tres particulares.
La nave en la que viajan es de última generación y dispone de todos los adelantos hasta el momento.

Requisitos:

1. Es necesario que la aplicación disponga de los siguientes tipos de relación entre clases: herencia, asociación y agregación.
2. Cada clase abstracta que se defina deberá implementar los interfaces correspondientes como puede ser