

# EL NOSTRE PROJECTE:

## ET-GAME



## MAIG 2022

Raúl García Sánchez  
Iker González Lumbreras  
Xavi Martínez Lozano  
Samuel Tapia Garcia

# INTRODUCCIÓN

En este proyecto hemos programado un videojuego basado en el OGame con interfaz gráfica\* y diferentes opciones tocando todo tipo de módulos\*/secciones.

Hemos participado un total de cuatro estudiantes de programación para completar este proyecto, este tenía un plazo límite de 14 días. Iniciamos el proyecto con la base de datos\* junto a entornos de desarrollo\*, después empezamos a programar ya lo que venía siendo la plantilla del programa sin interfaz gráfica. A los días cuándo teníamos prácticamente todo del programa nos pasamos a hacer la interfaz gráfica y a implementar e incluir las cosas que teníamos ya creadas anteriormente.

# ÍNDICE

M02- Base de Datos

M03- Programación

- Programación
- Interfaz Gráfica
  - View Planet Stats
  - Build
  - Build Troops
    - Ejemplo
  - Build Defenses
    - Ejemplo
  - Upgrade Technology
  - View Planet Report

M05- Entornos de Desarrollo

- GITHUB
- Diagramas
  - Diagrama De Actividades
  - Diagrama de Casos de Uso

Glossari

FAQ

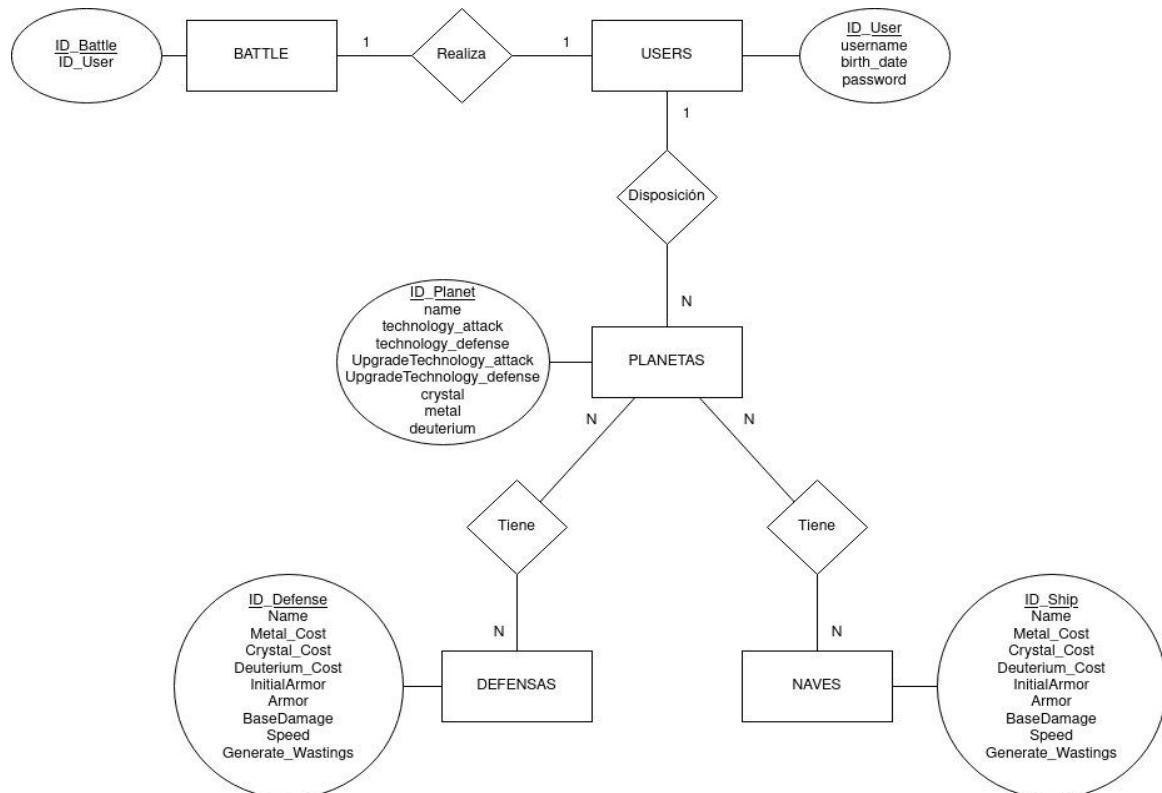
	<h1>Manual de Procedimientos informáticos</h1>	Fecha de elaboración: 09-05-2022
---	--	-------------------------------------

## M02- Base de Datos

El proyecto lo empezamos con este módulo de base de datos, ya que era lo imprescindible del programa/juego para poder empezarlo. Este se hizo en pareja para organizarse y repartir el trabajo entre los miembros del grupo.

Hicimos primero de todo el diseño del modelo relacional\* en un word para sacar las entidades\* y atributos\* que se encontraban en el programa. Después nos pusimos con el modelo relacional, así relacionamos unas tablas con otras y podíamos ver ya la base de datos clara de cómo se tenía que crear la base de datos.

Al finalizar el modelo relacional, creamos el diagrama de chen\* siguiendo los mismos pasos que con el modelo.





# Manual de Procedimientos informáticos

Fecha de elaboración:

09-05-2022

Al día siguiente creamos las tablas e insertamos los datos, después empezamos ya la base de datos creando los procedimientos de creación, eliminación de las tablas y la inserción de los datos dentro de ellas.

Y para finalizar este módulo hacer los procedimientos que nos piden dentro del proyecto.

```
create or replace NONEDITIONABLE procedure GET_SHIP(shipID IN OUT INT, nameShip OUT VARCHAR, metalShip OUT INT, crystalShip OUT INT, deuteriumShip OUT INT, initialArmor
is
existe NUMBER;
no_existe EXCEPTION;
begin
select count(*) into existe from ships where id_ships=shipID;
if existe != 0 then
select * into shipID, nameShip, metalShip, crystalShip, deuteriumShip, initialArmorship, armorShip, baseDamageShip, speedShip, generateShip from ships where id_ships=sh
else
raise no_existe;
end if;
exception
when no_existe then
dbms_output.put_line('La ID no existe');
end;
```

```
create or replace NONEDITIONABLE procedure GET_DEFENSE(defenseID IN OUT INT, nameDefense OUT VARCHAR, metalDefense OUT INT, crystalDefense OUT INT, deuteriumDefense OUT
is
existe NUMBER;
no_existe EXCEPTION;
begin
select count(*) into existe from defenses where id_defense=defenseID;
if existe = 0 then
raise no_existe;
else
select * into defenseID, nameDefense, metalDefense, crystalDefense, deuteriumDefense, initialArmorDefense, armorDefense, baseDamageDefense, speedDefense, generateDefens
end if;
exception
when no_existe then
dbms_output.put_line('La ID no existe');
end;
```

```
create or replace NONEDITIONABLE procedure INITIALIZE
is
battleNum int;
usersNum int;
planetsNum int;
shipsNum int;
defensesNum int;
shipsPlanetNum int;
defensesPlanetNum int;

begin
select count(*) into battleNum from user_tables where table_name='BATTLE';
select count(*) into usersNum from user_tables where table_name='USERS';
select count(*) into planetsNum from user_tables where table_name='PLANETS';
select count(*) into shipsNum from user_tables where table_name='SHIPS';
select count(*) into defensesNum from user_tables where table_name='DEFENSES';
select count(*) into shipsPlanetNum from user_tables where table_name='SHIPS_IN_PLANETS';
select count(*) into defensesPlanetNum from user_tables where table_name='DEFENSES_IN_PLANETS';

if battleNum + usersNum + planetsNum + shipsNum + defensesNum + shipsPlanetNum + defensesPlanetNum < 7 then
drop_table();
create_table();
insert_data();
end if;
end;
```

	<b>Manual de Procedimientos informáticos</b>	Fecha de elaboración: 09-05-2022
---	--	-------------------------------------

## M03- Programación

### Programación

Este videojuego está programado basado en objetos\* y tenemos diversas clases, tales como:

- Planet: Esta clase representará nuestro planeta, en ella guardaremos diversos datos como, nuestras tropas, defensas, recursos.... Esta clase además tiene métodos\* para mejorar las tecnologías de defensa, ataque, añadir tropas e imprimir las estadísticas de dicho planeta.
- ResourceException: Esta clase gestiona excepciones\* como cuando queramos mejorar alguna tecnología y no tengamos el deuterium suficiente, nos mostrará un mensaje.
- Ship: Esta clase será abstracta y guardaremos la armadura y el daño de nuestras unidades que no sean defensivas.
- LightHunter, Heavy Hunter, BattleShip, Armored Ship: Estas clases heredan de la clase ship y tendremos que hacer una formula para calcular la armadura en el primer constructor, el segundo constructor lo utilizaremos para crear unidades para el enemigo.
- Defense: Esta clase será abstracta y guardaremos la armadura y el daño de nuestras estructuras defensivas.
- MissileLauncher, Ion Cannon, Plasma Cannon: Estas clases heredan de defense, los constructores serán los mismos que en las clases de las unidades no defensivas.
- Battle: Esta clase es para gestionar el desarrollo de la batalla, en ella tendremos diferentes arraylist\* para guardar la flota enemiga y nuestra propia flota, en cada posición de la arraylist guardaremos el número de unidades. Esta clase tendrá un algoritmo de probabilidades para seleccionar quien ataca y quien repite el ataque, en estas batallas ganará el bando que menos pérdidas tenga.  
Esta clase además devolverá un resumen que pondremos en el apartado de reportes.
- Main: Desde esta clase podremos crear unidades, tanto defensivas como flota. Tendremos un temporizador para crear una flota enemiga que nos pueda atacar cada 3 minutos. Podremos visualizar la flota enemiga desde un método para poder prepararnos para la batalla.

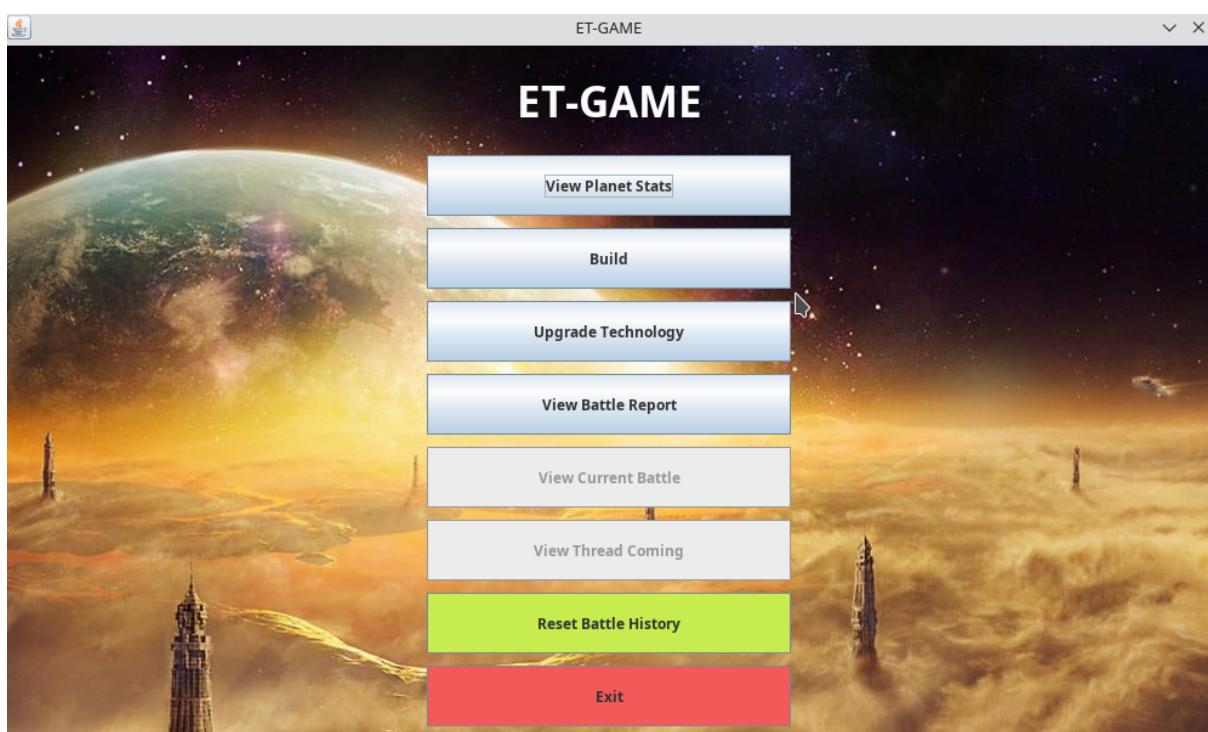
	<h1>Manual de Procedimientos informáticos</h1>	<p>Fecha de elaboración: 09-05-2022</p>
---	--	---

También tendremos interfaces\* en las cuales tendremos, métodos, variables, etc, estas serán:

- MilitaryUnit: Tendremos métodos para devolver el poder de ataque, restar a la armadura el daño, el blindaje actual, los costes, restablecer el blindaje al valor inicial, probabilidad de volver a atacar, generar residuos al acabar la batalla. Esta interfaz se implementará en todas las clases herederas de ship
- Variables: Será una interfaz para guardar muchas variables como, costes, daños, armaduras, probabilidad.

## Interfaz Gráfica

Para que el juego sea más visual y entretenido hemos implementado una interfaz gráfica la cual hace a nuestros usuarios familiarizarse más fácil con este videojuego



Empezamos creando un JFrame\* de un tamaño predeterminado y no editable (no se puede redimensionar), es un juego preparado para jugarlo en modo ventana.

En este JFrame añadimos el JPanel\* que podemos visualizar en la captura, el cual tiene una imagen de fondo ambientada en el entorno de **ET-GAME**. Este JPanel tiene un layout

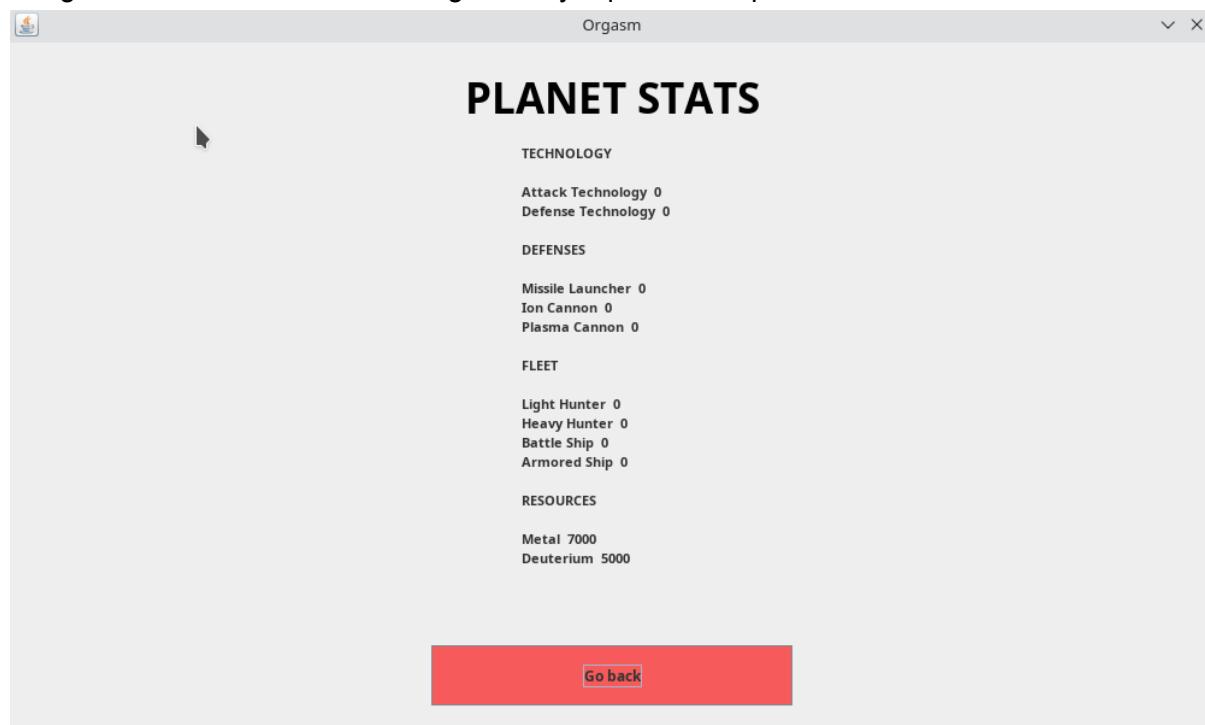
	<h1>Manual de Procedimientos informáticos</h1>	Fecha de elaboración: 09-05-2022
---	--	-------------------------------------

null para poder posicionar los botones y título donde a nosotros nos apetecía sin necesidad de añadir más JPanel, también nos facilitaba todo para añadir la imagen en el fondo.

Podemos ver que hay dos botones que están deshabilitados y eso es debido a que solo se habilitarán si viene una flota enemiga a atacarnos y cuando nos estén atacando.

## View Planet Stats

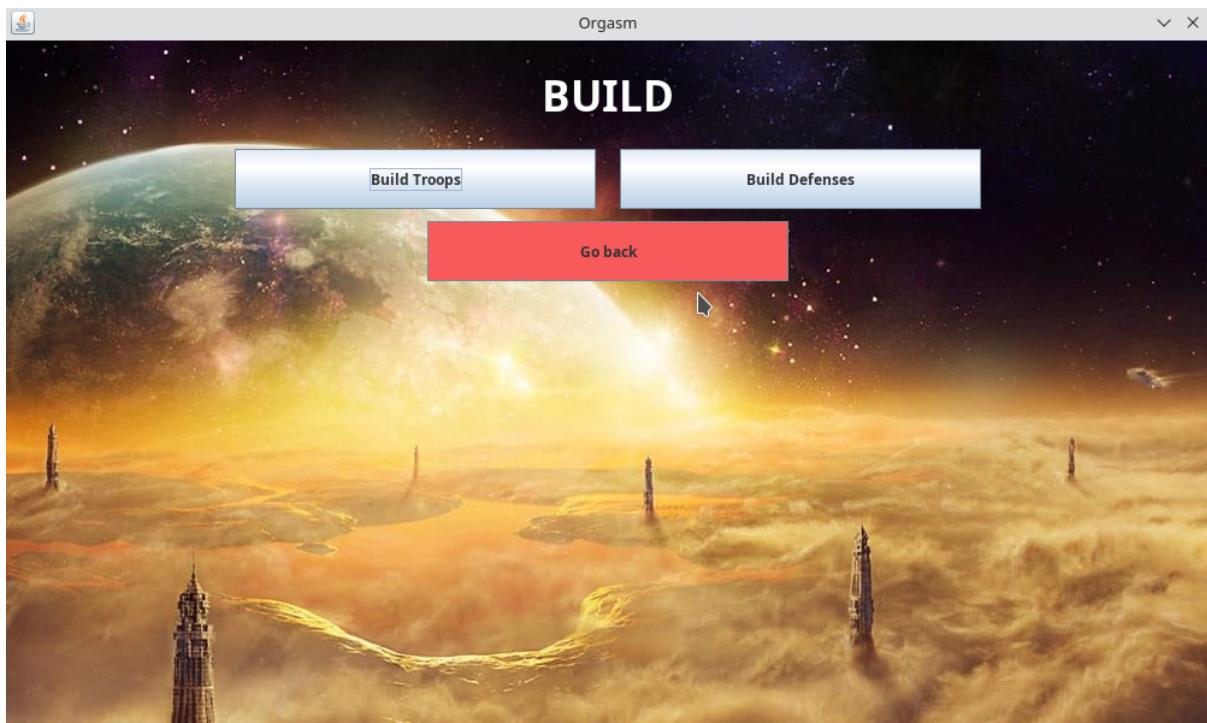
Este botón sirve para ver las estadísticas actuales de nuestro planeta, las estadísticas de la siguiente captura son las iniciales del planeta, estos datos los obtenemos de variables del código. Este JPanel no tiene background\* ya que no se apreciaba de forma correcta el texto



	<h1>Manual de Procedimientos informáticos</h1>	Fecha de elaboración: 09-05-2022
---	--	-------------------------------------

## Build

En este botón implementamos un nuevo JPanel con un pequeño menú para diferenciar la flota de las defensas, así el jugador se entenderá mejor con **ET-GAME**.



## Build Troops

En el apartado de Build Troops nos encontraremos con esta pantalla, en la cual nosotros construiremos las tropas, (Light Hunter, Heavy Hunter, Battle Ship, Armored Ship) hemos cambiado la imagen de background para que el usuario sepa que está construyendo tropas. Al presionar cualquiera de los botones de *build* se nos abrirá una pestaña en la cual pondremos el **número** de unidades que queremos construir. Si queremos confirmar la construcción de estos simplemente le daremos a **OK**, si queremos cancelar la construcción nos basta con darle a **CANCEL**.

Cabe destacar que si añadimos un **número** más alto de las unidades que queremos construir **solo** nos construirá aquellas posibles.

Y a medida que **mejoremos** la tecnología de ataque nos **costará más** cada tropa.

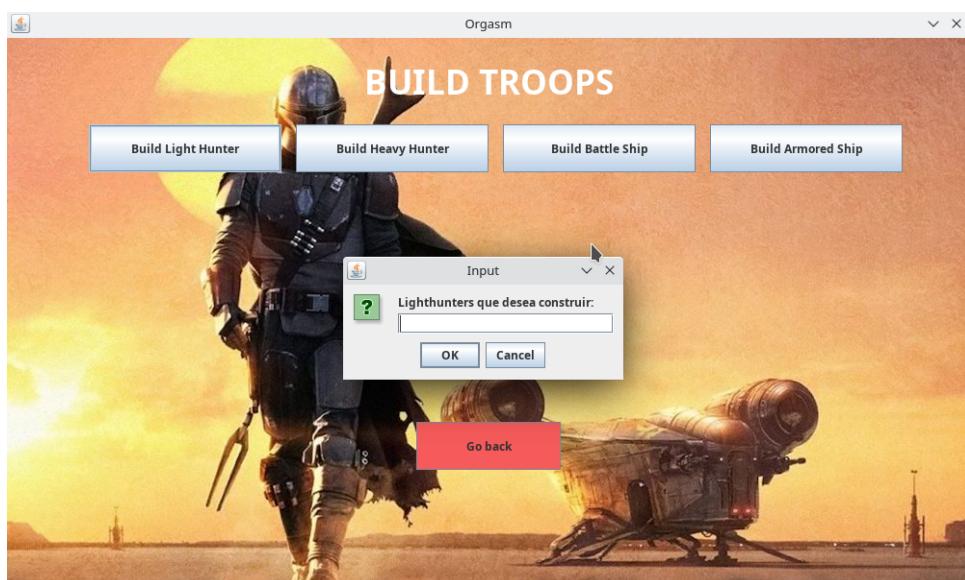
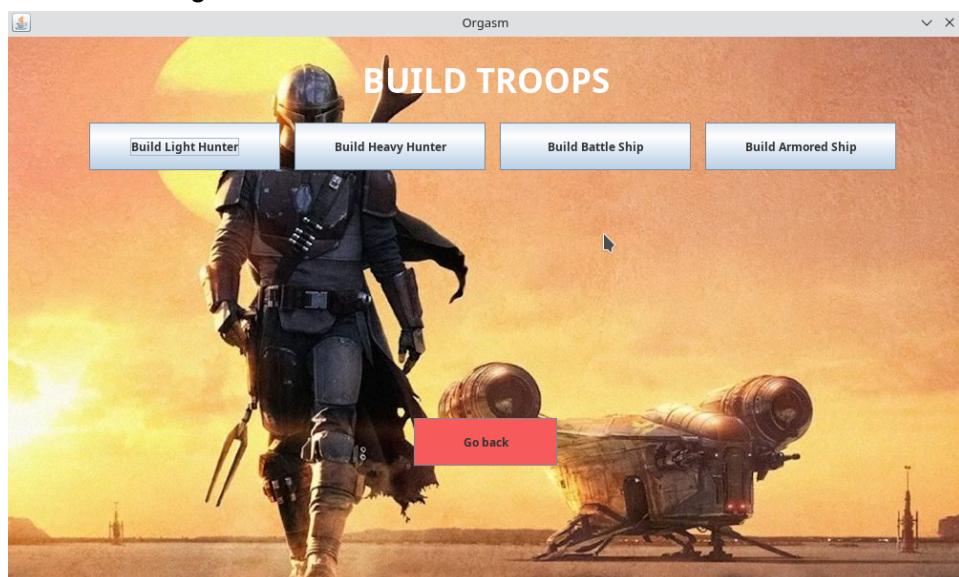
	<h1>Manual de Procedimientos informáticos</h1>	Fecha de elaboración: 09-05-2022
---	--	-------------------------------------

## Ejemplo

Tenemos 200 de metal y cada Light Hunter cuesta 100.

En el campo de Build Lighthunter ponemos el **número 5** y le damos a **OK**.

Solo nos construirá 2 Light Hunters



	<h1>Manual de Procedimientos informáticos</h1>	Fecha de elaboración: 09-05-2022
---	--	-------------------------------------

## Build Defenses

En el apartado de Build Defenses nos encontraremos con esta pantalla, en la cual nosotros construiremos las defensas, (Missile Launcher, Ion Cannon, Plasma Cannon) hemos cambiado la imagen de background para que el usuario sepa que está construyendo defensas.

Al presionar cualquiera de los botones de *build* se nos abrirá una pestaña en la cual pondremos el **número** de unidades que queremos construir. Si queremos confirmar la construcción de estos simplemente le daremos a **OK**, si queremos cancelar la construcción nos basta con darle a **CANCEL**.

Cabe destacar que si añadimos un **número** más alto de las defensas que queremos construir **solo** nos construirá aquellas posibles.

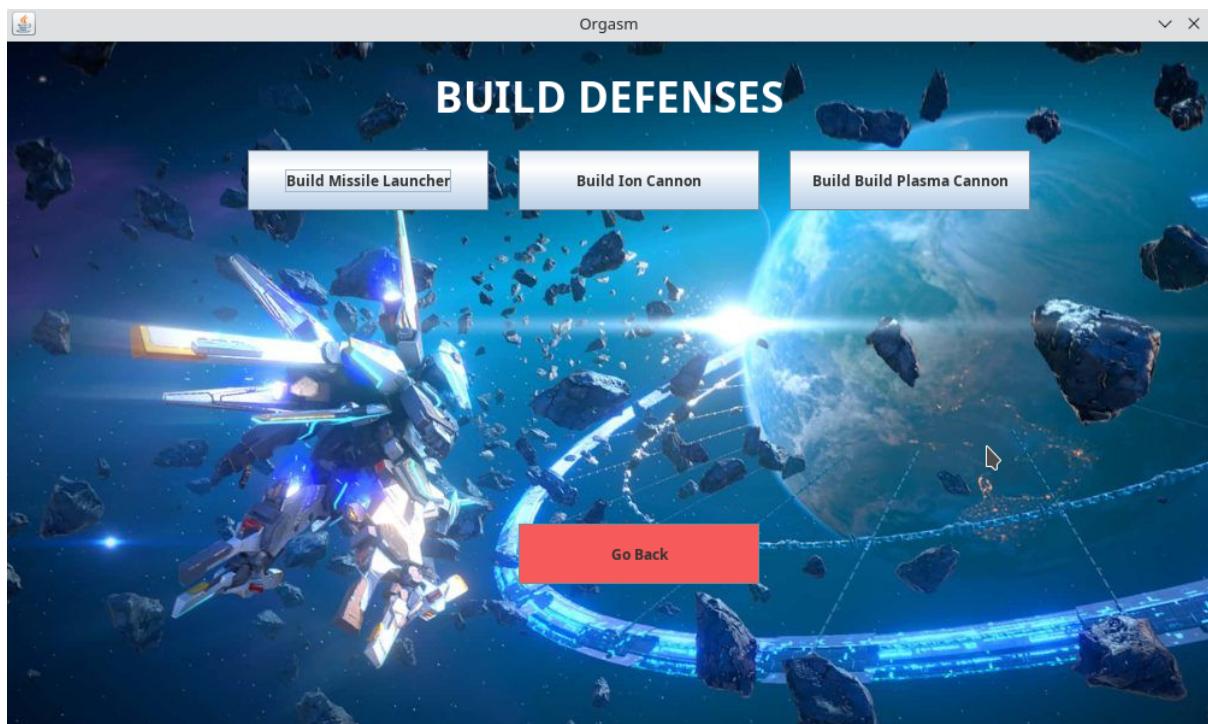
Y a medida que **mejoremos** la tecnología de defensa nos **costará más** cada defensa.

### Ejemplo

Tenemos 500 de metal y cada Missile Launcher cuesta 100.

En el campo de Missile Launcher ponemos el **número 10** y le damos a **OK**.

Solo nos construirá 5 Missile Launchers

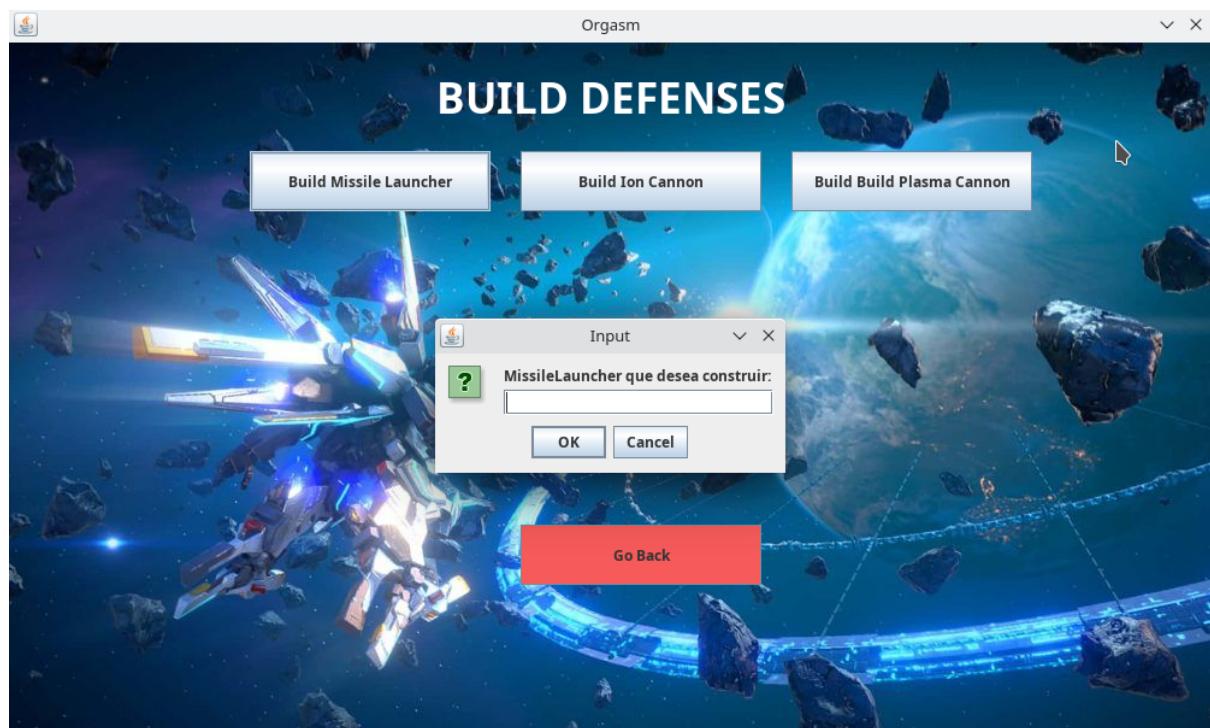




# Manual de Procedimientos informáticos

Fecha de elaboración:

09-05-2022



	<h1>Manual de Procedimientos informáticos</h1>	Fecha de elaboración: 09-05-2022
---	--	-------------------------------------

## Upgrade Technology

En Upgrade Technology podremos mejorar las tecnologías tanto de defensa como de ataque, también hemos implementado el deuterium (moneda de pago para la mejora de tecnologías) para que sepamos hasta qué punto podemos mejorar dichas tecnologías.

En los botones de mejora nos saldrá el coste de cada mejora, que es incremental, subiendo el costo a medida que subimos los niveles





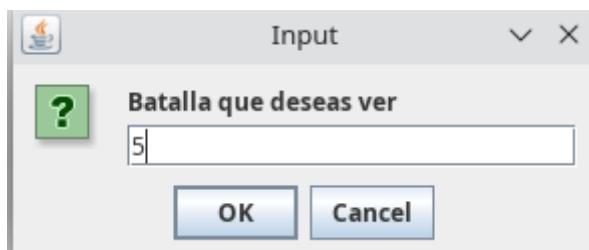
# Manual de Procedimientos informáticos

Fecha de elaboración:

09-05-2022

## View Planet Report

En Battle Report veremos un historial de las batallas con la información relevante y nosotros seleccionamos que batalla queremos ver y la tiene que reproducir.



**BATTLE 5**

Army Planet	Units	Drops	Initial Army Enemy	Units	Drops
LightHunter	2	2	Light Hunter	4	4
Heavy Hunter	0	0	Heavy Hunter	2	0
Battle Ship	0	0	Battle Ship	1	0
Armored Ship	0	0	Armored Ship	1	0
Missile Launcher	5	4			
Ion Cannon	4	4			
Plasma Cannon	0	0			

\*\*\*\*\*

Cost Army Planet	Cost Army Enemy
Metal: 32000	Metal: 100000
Deuterium: 2100	Deuterium: 22300

\*\*\*\*\*

Losses Army Planet	Losses Army Enemy
Metal: 30000	Metal: 12000
Deuterium: 2100	Deuterium: 200
Weighted: 40500	Weighted: 13000

\*\*\*\*\*

Waste Generated
Metal: 1400
Deuterium: 0

Battle Winned by Fleet enemy, don't Collecte Rubble

[Go Back](#)

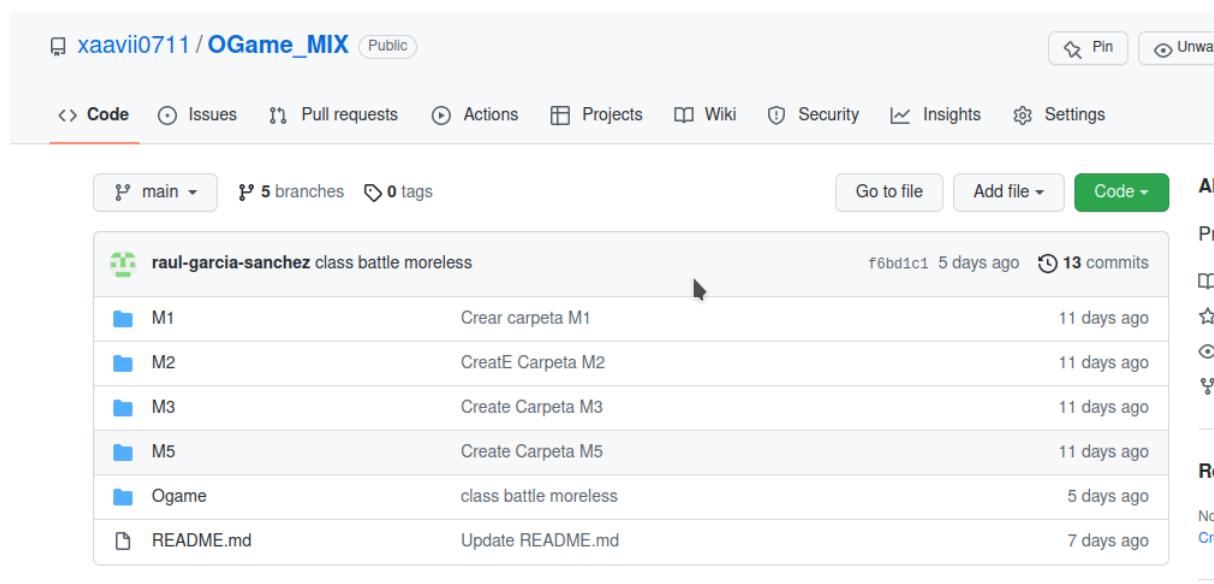
	<h1>Manual de Procedimientos informáticos</h1>	Fecha de elaboración: 09-05-2022
---	--	-------------------------------------

## M05- Entornos de Desarrollo

### GITHUB

En esta asignatura hemos empezado creando un repositorio público\* en GitHub y compartiendo con cada uno de los integrantes del grupo.

Lo primero fue crear carpetas en la carpeta Main para poder organizarnos por asignaturas



The screenshot shows a GitHub repository page for 'xaavii0711/OGame\_MIX'. The repository is public. The main branch is selected. Recent commits by 'raul-garcia-sanchez' are listed:

Commit	Message	Date
f6bd1c1	class battle moreless	5 days ago
M1	Crear carpeta M1	11 days ago
M2	CreatE Carpeta M2	11 days ago
M3	Create Carpeta M3	11 days ago
M5	Create Carpeta M5	11 days ago
Ogame	class battle moreless	5 days ago
README.md	Update README.md	7 days ago

	<h1>Manual de Procedimientos informáticos</h1>	<p>Fecha de elaboración: 09-05-2022</p>
---	--	---

Compartido con un fichero Readme el cual contiene la información del proyecto y el contacto sobre los participantes

## OGame\_MIX

---

Proyecto Final DAW Primer Año

Este proyecto consta de un videojuego de estrategia el cual es de un solo jugador, que trata de poder atacar y defender planetas con la ayuda de recursos los cuales vas ganando en las batallas, que sirven para poder construir naves, defensas y poder mejorarlas.

### Requisitos

Para poder utilizar este videojuego se necesitará tener instalado java, el programa Eclipse, tener instalado una base de datos Oracle y ejecutar nuestros procedimientos.

### CONTACTE

*Raúl García*

- Email: [rgarciasanch...cf@iesesteveterradas.cat](mailto:rgarciasanch...cf@iesesteveterradas.cat)
- Telf: 692419092

*Iker Gonzalez*

- Email: [igonzalezlumbra...cf@iesesteveterradas.cat](mailto:igonzalezlumbra...cf@iesesteveterradas.cat)
- Telf: 675959498

*Samuel Tapia*

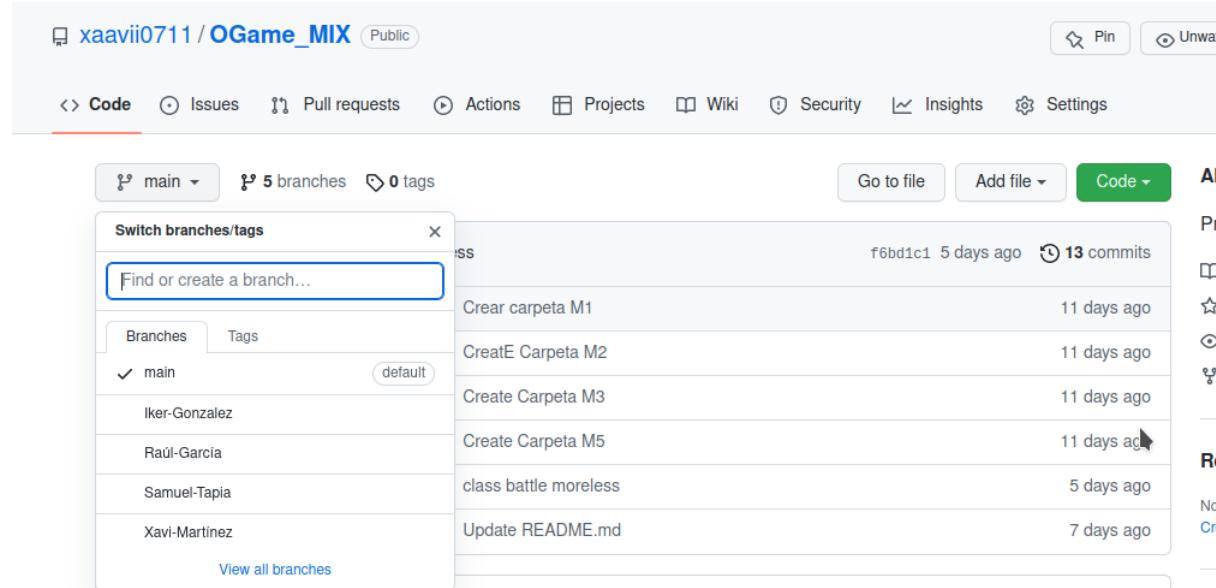
- Email: [stapiagarcia.cf@iesesteveterradas.cat](mailto:stapiagarcia.cf@iesesteveterradas.cat)
- Telf: 688925527

*Xavi Martinez*

- Email: [jmartinezlozano.cf@iesesteveterradas.cat](mailto:jmartinezlozano.cf@iesesteveterradas.cat)
- Telf: 647449529

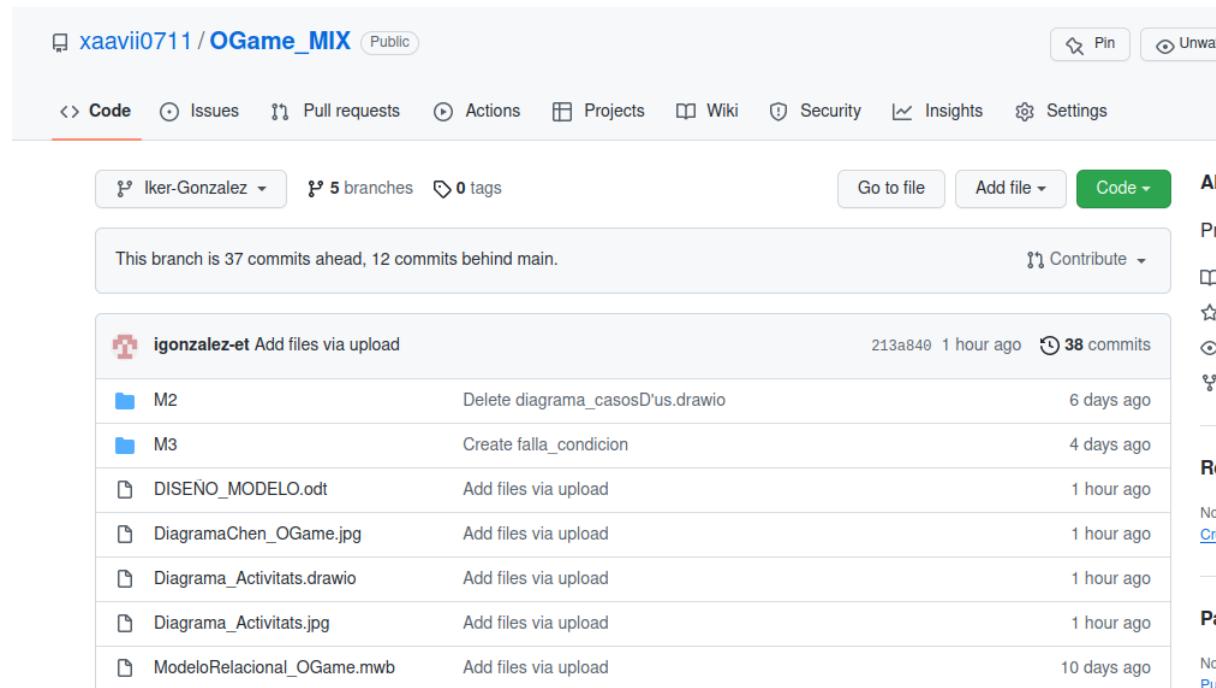
	<h1>Manual de Procedimientos informáticos</h1>	<p>Fecha de elaboración: 09-05-2022</p>
---	--	---

Una vez creadas las carpetas y el fichero Readme, creamos cuatro ramas, las cuales nos servirán para cada participante actualizar su trabajo del día a día.



The screenshot shows a GitHub repository named 'xaavii0711 / OGame\_MIX' with 5 branches and 0 tags. A dropdown menu is open over the 'main' branch, listing other branches: 'Iker-Gonzalez', 'Raúl-García', 'Samuel-Tapia', and 'Xavi-Martinez'. The 'main' branch has 13 commits, while the others have between 5 and 11 days ago.

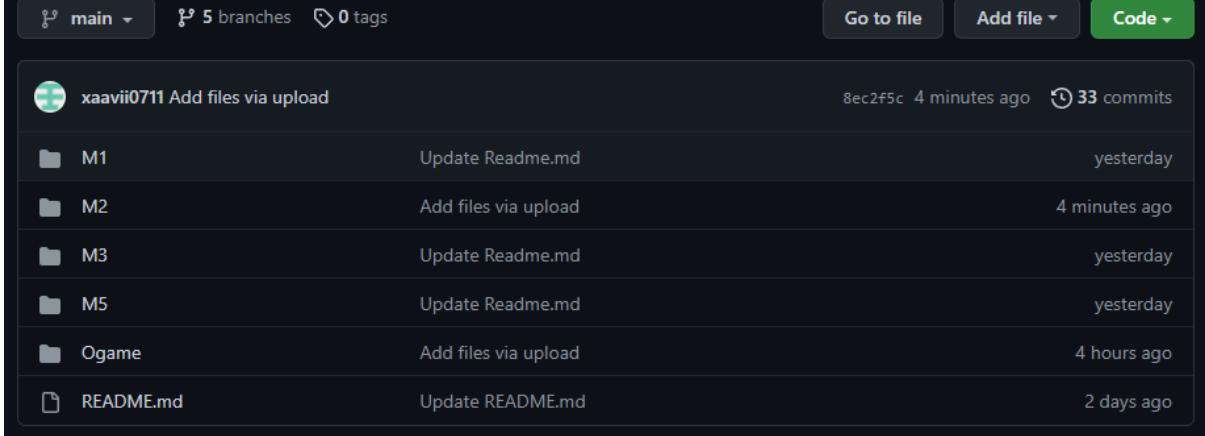
Por ejemplo, la rama de Iker



The screenshot shows the 'Iker-Gonzalez' branch of the repository. It indicates that this branch is 37 commits ahead of the main branch and 12 commits behind it. The commit list shows various file uploads by the user 'igonzalez-et', including 'M2', 'M3', and several 'Diagrama...' files.

	<h1>Manual de Procedimientos informáticos</h1>	<p>Fecha de elaboración: 09-05-2022</p>
---	--	---

Una vez finalizado el proyecto desplazamos todas las actividades de cada rama personal a la rama Main para así completar los cinco commits que necesitamos para finalizar la parte de m5.



Commit	Message	Date
M1	Update Readme.md	yesterday
M2	Add files via upload	4 minutes ago
M3	Update Readme.md	yesterday
M5	Update Readme.md	yesterday
Ogame	Add files via upload	4 hours ago
README.md	Update README.md	2 days ago



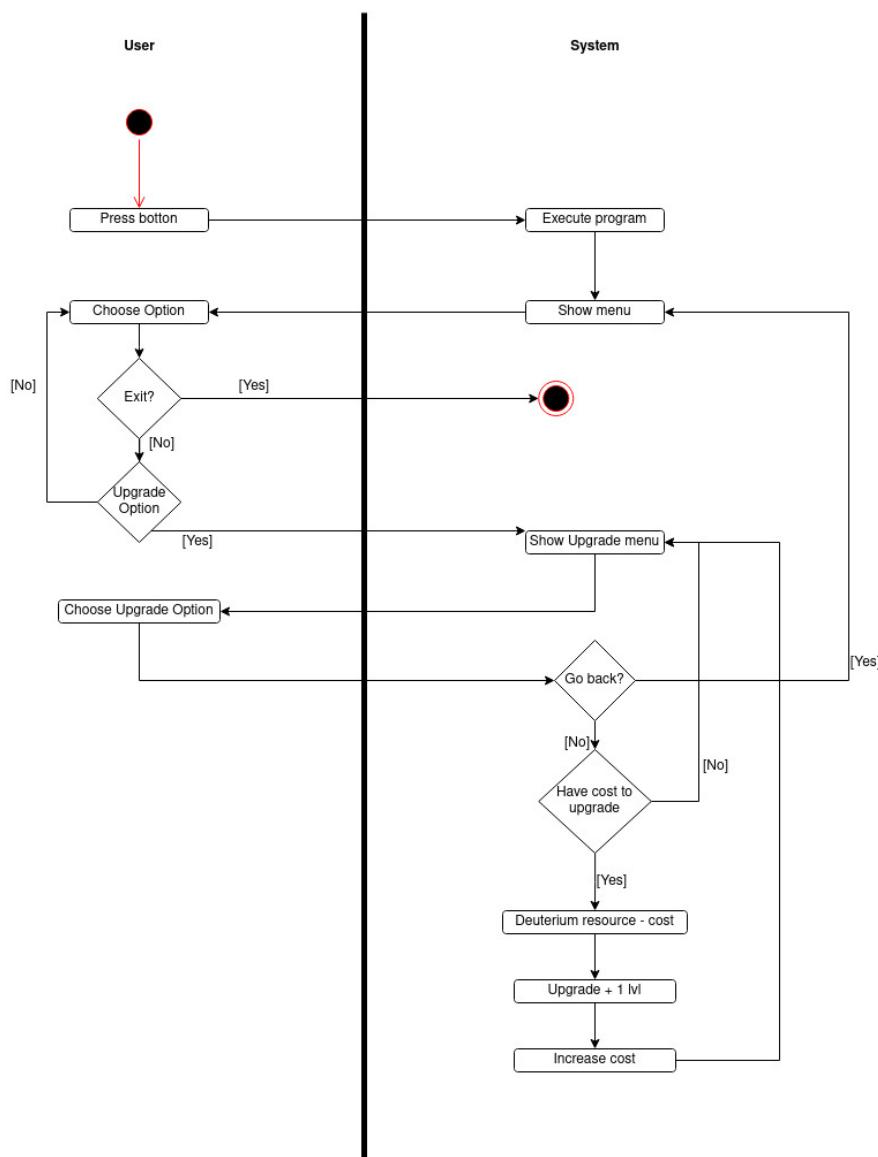
# Manual de Procedimientos informáticos

Fecha de elaboración:  
09-05-2022

## Diagramas

### Diagrama De Actividades

Para este diagrama nos pedían que hicéramos un diagrama de actividades el cual solo disponga de una opción y este es el resultado





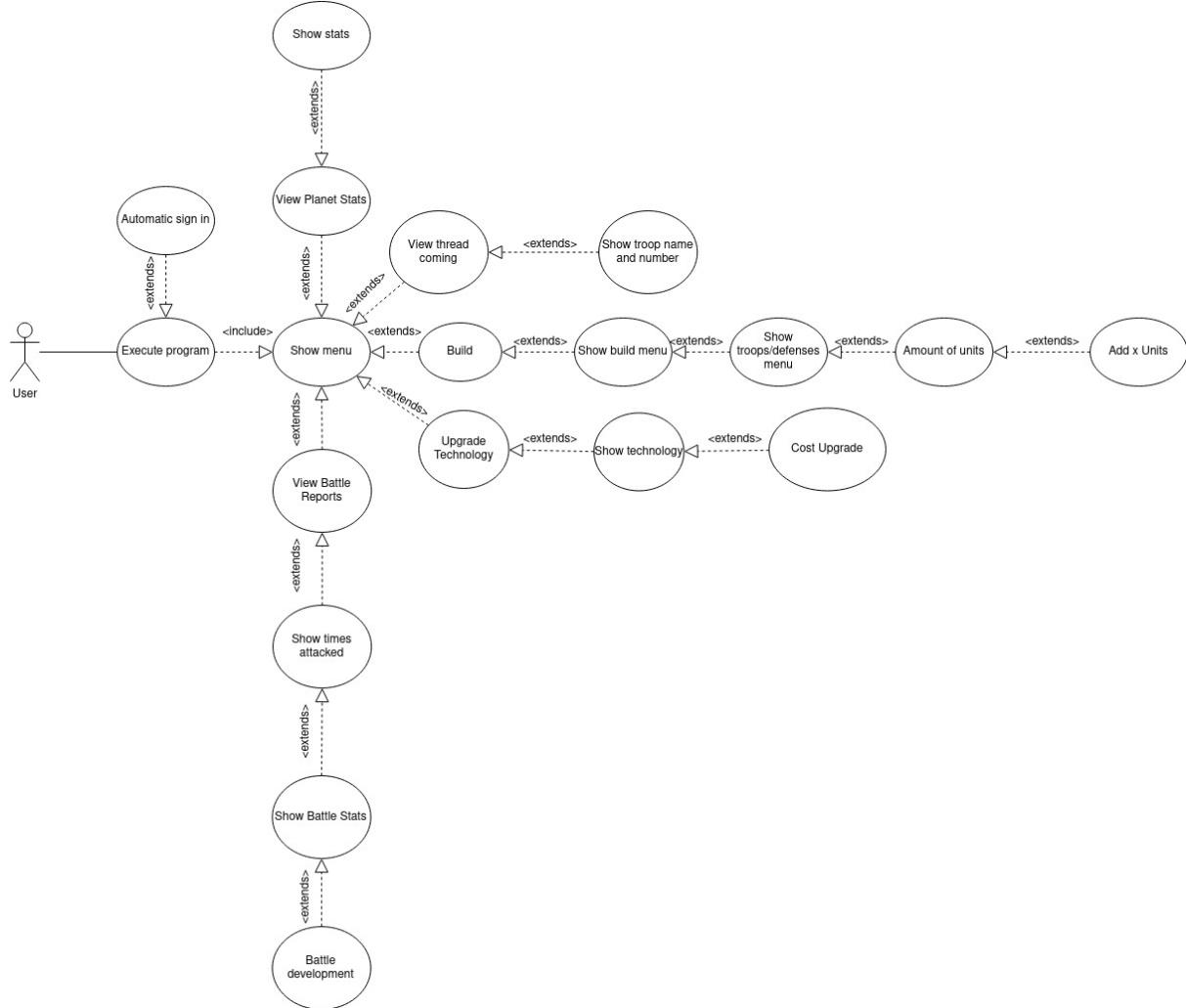
# Manual de Procedimientos informáticos

Fecha de elaboración:

09-05-2022

## Diagrama de Casos de Uso

Este es el resultado de nuestro diagrama de casos de uso del programa:



	<b>Manual de Procedimientos informáticos</b>	Fecha de elaboración: 09-05-2022
---	--	-------------------------------------

## Glossari

### A

**Atributos:**

Los **atributos**, también llamados datos o variables miembro, son porciones de información que un objeto posee o conoce de sí mismo.

**ArrayList:**

La clase **ArrayList** en **Java**, es una clase que permite almacenar datos en memoria de forma similar a los Arrays, con la ventaja de que el número de elementos que almacena, lo hace de forma dinámica, es decir, que no es necesario declarar su tamaño como pasa con los Arrays.

### B

**Base de Datos:**

Una **base de datos** es una recopilación organizada de información o **datos** estructurados, que normalmente se almacena de forma electrónica en un sistema informático.

**Background:**

Fondo de Pantalla

### D

**Diagrama de Chen:**

La notación de base de datos de Chen es útil para modelar los conceptos básicos de las entidades y relaciones, ya que presenta una vista abstracta de las asociaciones.

### E

**Entornos de Desarrollo:**

Un entorno de desarrollo es un conjunto de procedimientos y herramientas que se utilizan para desarrollar un código fuente o programa.

**Entidades:**

Una entidad en Java es un objeto de persistencia. La persistencia es la habilidad de una aplicación para mantener(persistir) y recuperar información de sistemas de almacenamiento no volátiles.

### I

**Interfaz Gráfica:**

La interfaz gráfica de usuario, conocida también como GUI (del inglés *graphical user interface*), es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.

	<h1>Manual de Procedimientos informáticos</h1>	Fecha de elaboración: 09-05-2022
---	--	-------------------------------------

## J

**JFrame:**

JFrame es una clase utilizada en Swing (biblioteca gráfica) para generar ventanas sobre las cuales añadir distintos objetos con los que podrá interactuar o no el usuario.

**JPanel:**

Los JPanel en Java son una de las herramientas más útiles que contiene el entorno Java, consisten en objetos contenedores que a su vez agrupan otros objetos tales como botones, etiquetas, selectores, campos para texto...

## M

**Módulos:**

En Java, llamamos módulo a un artefacto que puede contener código, recursos, y metadatos. Los metadatos describen dependencias con otros módulos, y regulan el acceso a los paquetes del módulo.

**Modelo Relacional:**

Este modelo intenta fundir la orientación a objetos con el modelo de base de datos relacional.

**Métodos:**

Un método en Java es un conjunto de instrucciones definidas dentro de una clase, que realizan una determinada tarea y a las que podemos invocar mediante un nombre.

## O

**Objetos:**

Un objeto en Java no es ni más ni menos que lo mismo que un objeto en el mundo real. Los objetos corresponden a cosas que se encuentran en el mundo real. Por ejemplo, un sistema de compra en línea podría tener objetos como: carrito de compras, cliente y producto.

## P

**Procedimientos:**

Los procedimientos son básicamente un conjunto de instrucciones que se ejecutan sin retornar ningún valor, hay quienes dicen que un procedimiento no recibe valores o argumentos, sin embargo en la definición no hay nada que se lo impida.

## R

**Repositorio Público:**

Los repositorios públicos de GitHub se suelen utilizar para compartir software de código abierto.

	<b>Manual de Procedimientos informáticos</b>	Fecha de elaboración: 09-05-2022
---	--	-------------------------------------

## FAQ

**Q: ¿Cómo se puede ejecutar este videojuego?**

**A:** Podréis descargar **gratuitamente** este videojuego desde nuestro repositorio de [github](#) y necesitaréis tener descargado **Java**, **Eclipse Workspace**, **SQL Developer**, importamos el proyecto y ejecutamos los scripts de Base de Datos, una vez ejecutados, ejecutaremos el archivo main.java del proyecto y ya podréis disfrutar **ET-GAME**.

**Q: ¿Cómo se ha implementado la base de datos?**

**A:** Hemos ejecutado unos procedimientos que nos crean las tablas que vamos a utilizar y dentro del programa insertamos y recuperamos datos de la base de datos

**Q: ¿Este videojuego es online?**

**A:** Por el momento no está implementado el modo Online.

**Q: ¿Cómo empiezo?**

**A:** Realmente podéis gestionar el comienzo como vosotros queráis, pero tened en cuenta que tarde o temprano os vendrán a atacar.

**Q: ¿Cuándo acaba cada batalla?**

**A:** Cuando un planeta sea atacante o defensor tenga un 20% o menos de sus tropas

## WEBGRAFIA

<https://es.stackoverflow.com/>  
<https://www.w3schools.com/>