


# PLANET WARS



Fecha de realización: 09/05/2022

**Integrantes:**

- Irene Hernández
- Rafael Puerta
- Erik Tamaño
- Ismael Morillo

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 2                   |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |


# ÍNDICE

## [INTRODUCCIÓN](#)

### [M2 - BASE DE DATOS](#)

- Diagrama de Chen
- Diagrama relacional
- Procedimiento creación tablas
- Procedimiento insertar datos en tablas
- Procedimiento borrar tablas base de datos
- Procedimiento de inicialización de la aplicación
- Procedimiento extraer datos Ship
- Procedimiento extraer datos Defense
- Procedimiento extraer datos Planet
- Procedimiento extraer ID de la Batalla
- Procedimiento extraer cantidad Planet\_has\_Ship
- Procedimiento extraer cantidad Planet\_has\_Defenses
- Procedimiento extraer defensas get\_defense\_battle
- Procedimiento extraer naves get\_ship\_battle
- Procedimiento extraer pasos get\_step\_amount
- Procedimiento extraer datos User
- Procedimiento actualizar datos Planet
- Procedimiento actualizar datos Ship
- Procedimiento actualizar datos Defense
- Procedimiento añadir Battle
- Procedimiento añadir Add\_battle\_step
- Procedimiento añadir Defense\_Battle
- Procedimiento añadir Ship\_Battle
- Procedimiento añadir Defense
- Procedimiento añadir Planet
- Procedimiento añadir Ship
- Procedimiento añadir usuario
- Procedimiento comprobar usuario
- Procedimiento iniciar sesión
- Procedimiento comprobar batalla checkBattle
- Procedimiento reportes show\_reports

|                       |                      |                            |
|-----------------------|----------------------|----------------------------|
| Elaborat per: Grupo 2 | Revisat per: Grupo 2 | Autoritzat per: Josep Diaz |
|-----------------------|----------------------|----------------------------|

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 3                   |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

### M3 - PROGRAMACIÓN JAVA

- Clase Planet
- Clase Ship
- Clase HeavyHunter
- Clase LightHunter
- Clase BattleShip
- Clase ArmoredShip
- Clase Defense
- Clase IonCannon
- Clase PlasmaCannon
- Clase MissileLauncher
- Clase VentanaLogin
- Clase ConnectionBDD
- Clase VentanaInicial
- Clase ReportScreen
- Clase BattleScreen
- Clase Battle
- Clase BattleShip
- Clase Enemy
- Clase ResourceException
- Clase Main
- Interfaz MilitaryUnit
- Interfaz Variables


### M5 - ENTORNOS DE DESARROLLO

- Diagrama de Actividades
- Diagrama de casos de uso

### M1 - GLOSARIO

### WEBGRAFIA

|                       |                      |                            |
|-----------------------|----------------------|----------------------------|
| Elaborat per: Grupo 2 | Revisat per: Grupo 2 | Autoritzat per: Josep Diaz |
|-----------------------|----------------------|----------------------------|

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 4                   |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

## INTRODUCCIÓN

### Descripción del proyecto.

Aplicación basada en un popular juego de navegador, Ogame.  
El juego va sobre vivir en un mundo en el que vas consiguiendo recursos, flota de naves, defensas y 2 tipos de tecnología.

En el mismo universo, existen otros planetas que quieren conquistarnos, y tenemos que hacer lo posible por defendernos.

Durante el juego tendremos que generar unas defensas y una flota capaz de contener y derrumbar la llegada de flotas enemigas.

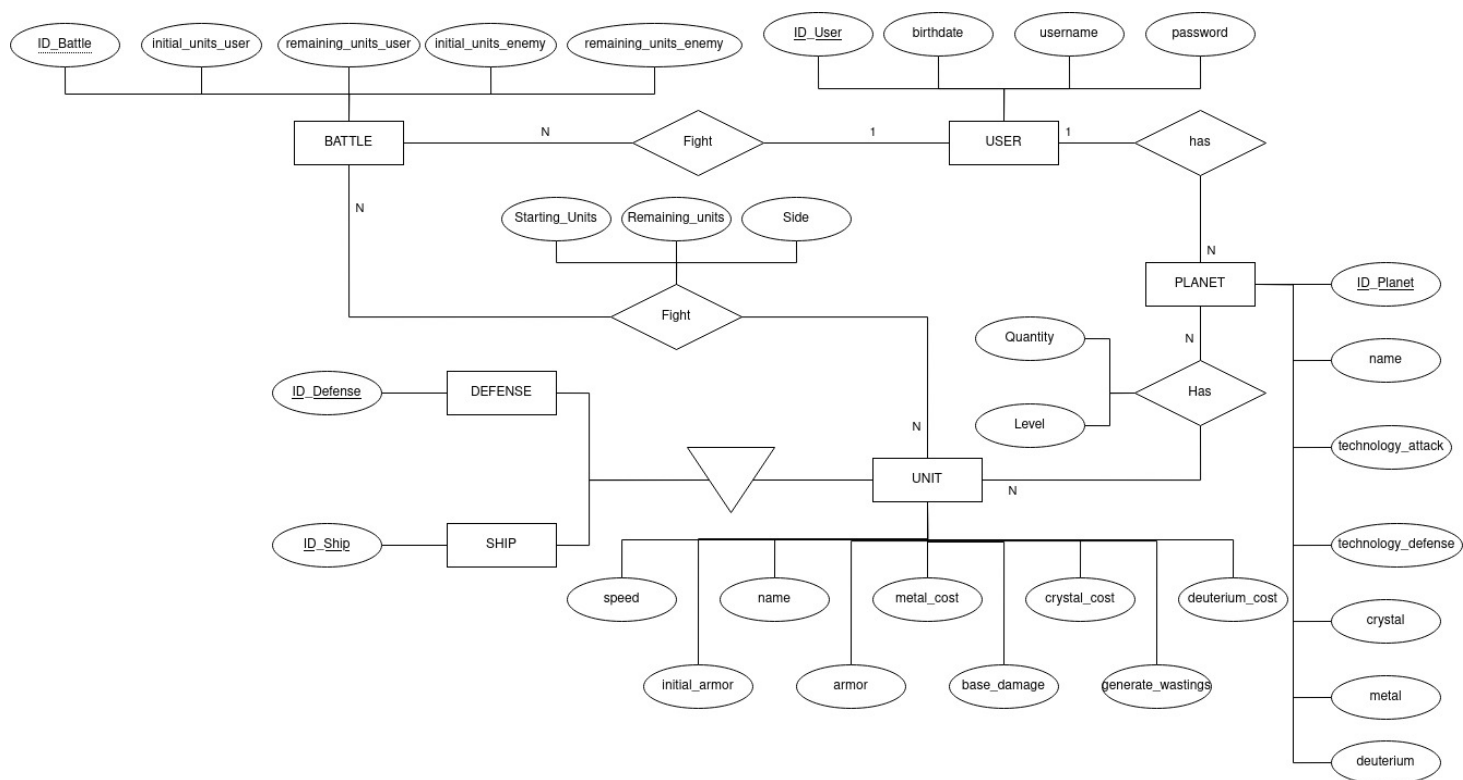
Los ataques de los enemigos serán constantes, por lo que no podrás dejar desatendido el planeta.

|                       |                      |                            |
|-----------------------|----------------------|----------------------------|
| Elaborat per: Grupo 2 | Revisat per: Grupo 2 | Autoritzat per: Josep Diaz |
|-----------------------|----------------------|----------------------------|



## M2 - BASE DE DATOS

### - Diagrama de Chen





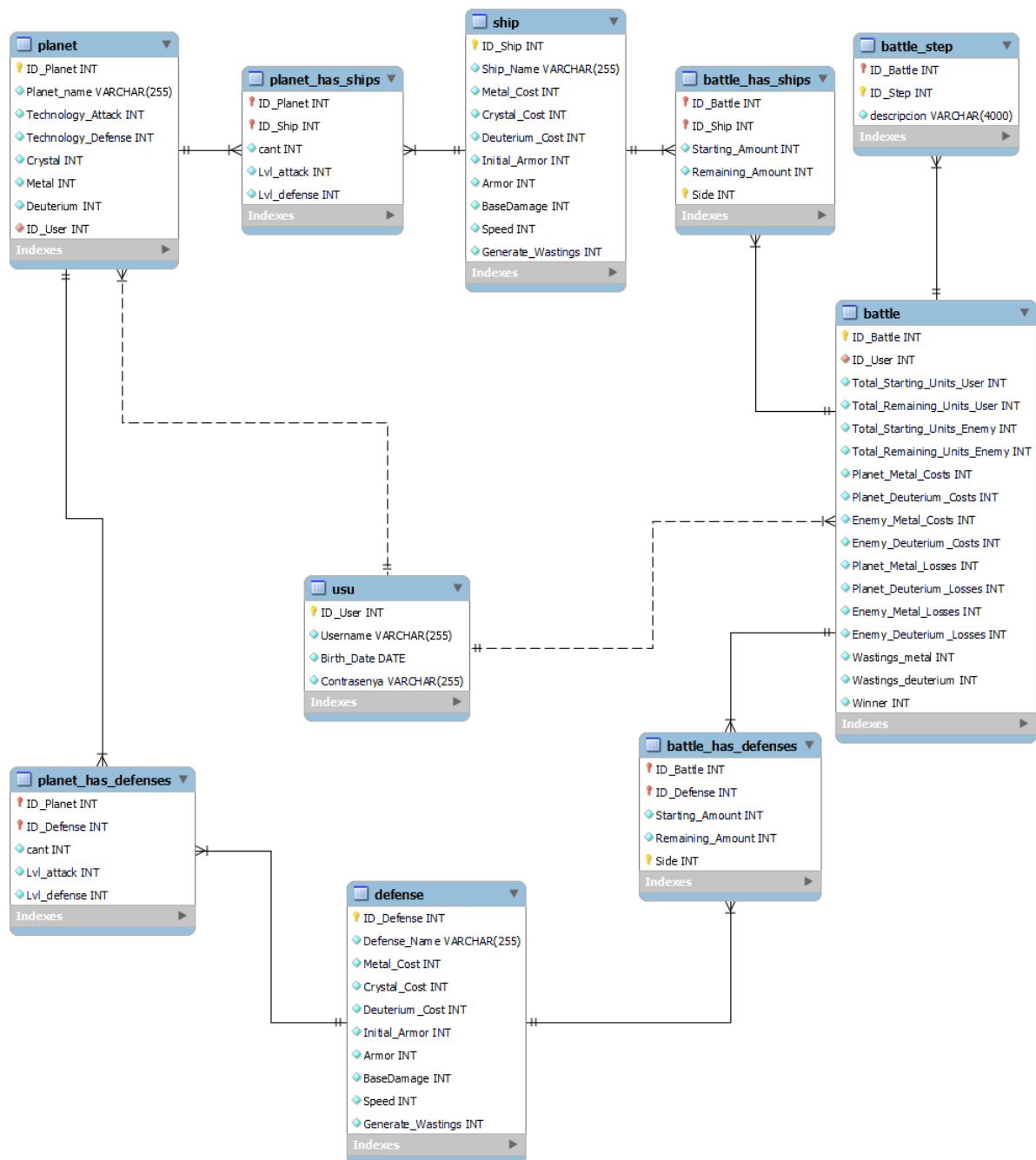
# MANUAL DE DESCRIPCIÓ I ESPECIFICACIÓ DE CÀRRECS DE LA EMPRESA XYZ,CA


Pàgina N° 6

Data d'Elaboració: 09/05/2022

Data d'Aprovació: 18/05/2022

## - Diagrama relacional



|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 7                   |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

## - Procedimiento creación tablas


Procedimiento de creación de tablas de la base de datos. Se crea una variable de tipo varchar donde le atribuimos el código en formato texto para que podamos poner el execute immediate ya que solo acepta texto. En este caso utilizamos un create table “nombre\_tabla” y todas sus especificaciones (campos, tipo de datos, claves primarias y foráneas).

```
create or replace procedure create_table as
crear varchar(32767);
begin
crear := 'create table USU (
ID_User int primary key not null,
Username varchar(255) not null unique,
Birth_Date date not null,
Password varchar(255) not null)';
execute immediate crear;
end;
```

## - Procedimiento insertar datos en tablas

Procedimiento para la inserción de datos en las tablas. Se crea una variable de tipo varchar donde le atribuimos el código en formato texto para que podamos poner el execute immediate ya que solo acepta texto. En este caso utilizamos un insert into “nombre\_tabla” values y entre paréntesis los valores a introducir.

```
create or replace procedure insert_table as
insertar varchar(255);
begin
insertar := 'insert into ship values (1, ''Light
Hunter'',3000,0,50,400,400,80,3,20)';
execute immediate insertar;
end;
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 8                   |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

## - Procedimiento borrar tablas base de datos

Procedimiento para la eliminación de las tablas de la base de datos. Se crea una variable de tipo varchar donde le atribuimos el código en formato texto para que podamos poner el execute immediate ya que solo acepta texto. En este caso utilizamos un drop table "nombre\_tabla" purge. Lo hacemos de este modo ya que si intentamos eliminar una tabla que está relacionada con otra no nos dejaría.


```
create or replace procedure drop_table as
eliminar varchar(255);
begin
eliminar := '
drop table battle_has_ships purge';
execute immediate eliminar;
end;
```

## - Procedimiento de inicialización de la aplicación

Procedimiento que se ejecutará cada vez que se inicie la aplicación. Se crea una variable de entrada de tipo number inicializada en 0 por defecto y otra contador. Con estas variables lo que vamos a controlar es la existencia de tablas al iniciar nuestro programa de manera normal o cuando el usuario le da a resetear los valores principales del juego.

```
create or replace procedure initialize (numero in number := 0)
as
contador number;
begin
if numero = 1 then
drop_table;
create_table;
insert_table;
else
select count(*) into contador from all_tables where table_name = 'USU';
if contador = 0 then
create_table;
```



|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 9                   |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

insert_table;
end if;
end if;
end;

```

## - Procedimiento extraer datos Ship

Procedimiento para la extracción de datos de la tabla Ship. Extraemos los datos y los meteremos en variables de salida que utilizaremos en la parte de programación java.

```

create or replace procedure getShip(id_nave in out number, nombre in out
varchar, metal in out number, cristal in out number, deuterium in out
number, inicial in out number, armadura in out number, base in out
number, velocidad in out number, generar in out number)
as
begin
select * into
id_nave,nombre,metal,cristal,deuterium,inicial,armadura,base,velocidad,g
enerar from ship where id_ship = id_nave;
end;

```

## - Procedimiento extraer datos Defense


Procedimiento para la extracción de datos de la tabla Defense. Extraemos los datos y los meteremos en variables de salida que utilizaremos en la parte de programación java.

```

create or replace procedure getDefense(id_nave_defensa in out number,
nombre in out varchar, metal in out number, cristal in out number,
deuterium in out number, inicial in out number, armadura in out number,
base in out number, velocidad in out number, generar in out number)
as
begin
select * into
id_nave_defensa,nombre,metal,cristal,deuterium,inicial,armadura,base,vel
ocidad,generar from defense where id_defense = id_nave_defensa;end;

```

|                       |                      |                            |
|-----------------------|----------------------|----------------------------|
| Elaborat per: Grupo 2 | Revisat per: Grupo 2 | Autoritzat per: Josep Diaz |
|-----------------------|----------------------|----------------------------|

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 10                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

## - Procedimiento extraer datos Planet


Procedimiento para la extracción de datos de la tabla Planet. Extraemos los datos y los meteremos en variables de salida que utilizaremos en la parte de programación java.

```
create or replace procedure get_planet(
    id_planeta in out number,
    nombre out varchar,
    tech_ataque out number,
    tech_defensa out number,
    cant_cristal out number,
    cant_metal out number,
    cant_deuterium out number,
    id_usuario out number) as
begin
select * into id_planeta, nombre, tech_ataque, tech_defensa,
cant_cristal, cant_metal, cant_deuterium, id_usuario from planet where
id_planet = id_planeta;
end;
```

## - Procedimiento extraer ID de la Batalla

Procedimiento para la extracción del ID de la batalla.

```
create or replace procedure get_battleid(id_bat out number) as
begin
select max(id_battle) + 1 into id_bat from battle;
if id_bat is null then
id_bat := 1;
end if;
end;
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 11                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

## - Procedimiento extraer cantidad Planet\_has\_Ship


Procedimiento para la extracción de cantidad de la tabla Planet\_has\_Ships.  
Extraemos el dato y los meteremos en una variable de salida que utilizaremos en la parte de programación java.

```
create or replace procedure get_cantidadShip(ship_id in number,planet_id
in number, nivel_ataque in number, nivel_defensa in number, cantidad in
out number)
as
begin
select cant into cantidad from planet_has_ships where id_ship = ship_id
and id_planet = planet_id and Lvl_attack = nivel_ataque and Lvl_defense
= nivel_defensa;
end;
```

## - Procedimiento extraer cantidad Planet\_has\_Defenses

Procedimiento para la extracción de cantidad de la tabla Planet\_has\_Ships.  
Extraemos el dato y los meteremos en una variable de salida que utilizaremos en la parte de programación java.

```
create or replace procedure get_cantidadDefenses(ship_id in
number,planet_id in number, nivel_ataque in number, nivel_defensa in
number, cantidad in out number)
as
begin
select cant into cantidad from planet_has_defenses where id_ship =
ship_id and id_planet = planet_id and Lvl_attack = nivel_ataque and
Lvl_defense = nivel_defensa;
end;
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 12                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

## - Procedimiento extraer defensas get\_defense\_battle


Procedimiento para la extracción de los datos de las defensas de la base de datos.

```
create or replace procedure get_defense_battle (
    id_bat in number,
    id_unit in number,
    bando in number,
    inicio out number,
    resto out number) as
begin
select starting_amount, remaining_amount into inicio, resto from
battle_has_defenses
where id_battle = id_bat and id_defense = id_unit and side =
bando;
end;
```

## - Procedimiento extraer naves get\_ship\_battle

Procedimiento para la extracción de los datos de las naves de la base de datos.

```
create or replace procedure get_ship_battle (
    id_bat in number,
    id_unit in number,
    bando in number,
    inicio out number,
    resto out number) as
begin
select starting_amount, remaining_amount into inicio, resto from
battle_has_ships
where id_battle = id_bat and id_ship = id_unit and side = bando;
end;
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 13                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

## - Procedimiento extraer pasos get\_step\_amount

Procedimiento para la extracción de los pasos de la base de datos.

```
create or replace procedure get_step_amount (
    cod_bat in number,
    num_steps out number) as
begin
select count(*) into num_steps from battle_step where id_battle =
cod_bat;
end;
```

## - Procedimiento extraer datos User

Procedimiento para la extracción de datos de la tabla User. Extraemos los datos y los meteremos en variables de salida que utilizaremos en la parte de programación java.

```
create or replace procedure get_user(nombre_usu in out varchar, cod_usu
out number, fecha_nac out date, contrasenya out varchar, existe out
number) as
begin
select * into cod_usu, nombre_usu, fecha_nac, contrasenya from USU where
username = nombre_usu;
existe := 1;
exception
when no_data_found then
existe := 0;end;
```

## - Procedimiento actualizar datos Planet

Procedimiento para la actualización de los datos de la tabla Planet.

```
create or replace procedure updatePlanet (id_usu in number,id_planetita
in number,techATK in number,techDEF in number,Iron in number,Deute in
number)
as
begin
```



```
update planet set  
technology_attack=techATK,technology_defense=TechDEF,metal=Iron,deuteriu  
m=Deute where id_user=id_usu and id_planet=id_planetita;  
end;
```

## - Procedimiento actualizar datos Ship

Procedimiento para la actualización de los datos de la tabla Ship.

```
create or replace procedure updateShips (id_planetita in number,techATK  
in number,techDEF in number,id_nave in number,canti in number)  
as  
existe number;  
begin  
select count(id_planet) into existe from planet_has_ships where  
id_planet=id_planetita and id_ship=id_nave and lvl_attack=techATK and  
lvl_defense=techDef;  
if canti > 0 then  
    if existe > 0 then  
        update planet_has_ships set CANT=canti where  
id_planet=id_planetita and id_ship=id_nave and lvl_attack=techATK and  
lvl_defense=techDef;  
    else  
        insert into planet_has_ships  
(id_planet,id_ship,cant,lvl_attack,lvl_defense) values  
(id_planetita,id_nave,canti,techATK,techDEF);  
    end if;  
end if;  
end;
```

## - Procedimiento actualizar datos Defense

Procedimiento para la actualización de los datos de la tabla Defense.

```
create or replace procedure updateDefenses (id_planetita in  
number,techATK in number,techDEF in number,id_def in number,canti in  
number)
```




```
as
existe number;
begin
select count(id_planet) into existe from planet_has_defenses where
id_planet=id_planetita and id_defense=id_def and lvl_attack=techATK and
lvl_defense=techDef;
if canti > 0 then
    if existe > 0 then
        update planet_has_defenses set CANT=canti where
id_planet=id_planetita and id_defense=id_def and lvl_attack=techATK and
lvl_defense=techDef;
    else
        insert into planet_has_defenses
(id_planet,id_defense,cant,lvl_attack,lvl_defense) values
(id_planetita,id_def,canti,techATK,techDEF);
    end if;
end if;
end;
```

## - Procedimiento añadir Battle

Procedimiento para la inserción de una nueva batalla en la base de datos.

```
create or replace procedure add_battle(
    id_usuario in number,
    inicio usuario in number,
    final_usuario in number,
    inicio enemigo in number,
    final enemigo in number) as
id batalla number;
begin
select max(id_battle) into id_batalla from battle;
if id_batalla is null then
id batalla := 1;
end if;
insert into battle(id_battle, id_user, total_starting_units_user,
total_remaining_units_user, total_starting_units_enemy,
total_remaining_units_enemy)
values (id batalla, id_usuario, inicio usuario, final usuario, inicio
enemigo, final enemigo);end;
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 16                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

## - Procedimiento añadir Add\_battle\_step

Procedimiento para la inserción de los pasos de batalla en la base de datos.

```
create or replace procedure add_battle_step(id_bat in number, id_paso
in number, descri in varchar) as
begin
insert into battle_step values(id_bat, id_paso, descri);
end;
```

## - Procedimiento añadir Defense\_Battle

Procedimiento para la inserción de nuevas defensas de batalla en la base de datos.


```
create or replace procedure add_defense_battle(
    id_batalla in number,
    id_unit in number,
    inicio in number,
    resto in number,
    bando in number)
as
begin
insert into battle_has_defenses(id_battle, id_defense, starting_amount,
remaining_amount, side)
values (id_batalla, id_unit, inicio, resto, bando);
end;
```

## - Procedimiento añadir Ship\_Battle

Procedimiento para la inserción de nuevas naves de batalla en la base de datos.

```
create or replace procedure add_ship_battle(
    id_batalla in number,
    id_unit in number,
    inicio in number,
    resto in number,
    bando in number)
```



|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 17                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

as
begin
insert into battle_has_ships(id_battle, id_ship, starting_amount,
remaining_amount, side)
values (id_batalla, id_unit, inicio, resto, bando);
end;

```

## - Procedimiento añadir Defense

Procedimiento para la inserción de una nueva defensa en la base de datos.

```

create or replace procedure add_defense(
    id_planeta in number,
    id_unit in number,
    cantidad in number,
    nivel in number)
as
existe number;
begin
select id_unit into existe from planet_has_defenses where lvl = nivel;
update planet_has_defenses set cant = cant + cantidad where
id_planet = id_planeta and id_defense = id_unit and lvl = nivel;
exception
when no_data_found then
insert into planet_has_defenses(id_planet, id_defense, cant, lvl)
values (id_planeta, id_unit, cantidad, nivel);
end;

```


## - Procedimiento añadir Planet

Procedimiento para la inserción de un nuevo planeta en la base de datos.

```

create or replace procedure add_planet (
    nombre in varchar,
    tech_ataque in number,
    tech_defensa in number,
    cant_cristal in number,

```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 18                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```


cant_metal in number,
cant_deuterium in number,
id_usu in number) as id planeta number;
begin
select max(id_planet) + 1 into id_planeta from planet;
if id_planeta is null then
id_planeta := 1;
end if;
insert into planet(id_planet, planet_name, technology_attack,
technology_defense, crystal, metal, deuterium, id_user)
values (id_planeta, nombre, tech_ataque, tech_defensa, cant_cristal,
cant_metal, cant_deuterium, id_usu);
end;
```

## - Procedimiento añadir Ship

Procedimiento para la inserción de una nueva nave en la base de datos.

```

create or replace procedure add_ship(
    id_planeta in number,
    id_unit in number,
    cantidad in number,
    nivel in number) as
existe number;
begin
select id_unit into existe from planet_has_ships where lvl = nivel;
update planet_has_ships set cant = cant + cantidad where
id_planet = id_planeta and id_ship = id_unit and lvl = nivel;
exception
when no_data_found then
insert into planet_has_ships(id_planet, id_ship, cant, lvl)
values (id_planeta, id_unit, cantidad, nivel);
end;
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 19                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

## - Procedimiento añadir usuario

Procedimiento para la inserción de un nuevo usuario en la base de datos.

```
create or replace procedure add_user(nombre in varchar, fecha in
varchar, contr in varchar) as
id_usu number;
fecha_ins date;
begin
select max(id_user) + 1 into id_usu from usu;
if id_usu is null then
id_usu := 1;
end if;
insert into usu(id_user, username, birth_date, contrasenya) values
(id_usu, nombre, fecha, contr);
end;
```

## - Procedimiento comprobar usuario


Procedimiento para la comprobación de la existencia de un usuario en la base de datos.

```
create or replace function comprobar_usuari(usuario in varchar)
return number as
existe number;
begin
select count(*) into existe from usu where username=usuario;
return existe;end;
```

## - Procedimiento iniciar sesión

Procedimiento para iniciar sesión como un usuario existente y poder acceder a los datos de esta persona.

```
create or replace function login(nombre in varchar, contr in varchar)
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 20                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

return number as
existe number;
begin
select count(*) into existe from usu where username = nombre and
contrasenya = contr;
return existe;end;

```

## - Procedimiento comprobar batalla checkBattle

Procedimiento para la comprobación de la existencia de una batalla en la BDD.

```

create or replace procedure check_battle(cod_bat in number, existe out
number) as
begin
select count(*) into existe from battle where id_battle = cod_bat;
end;

```


## - Procedimiento reportes show\_reports

Procedimiento para meter en variables los pasos de la batalla.

```

create or replace procedure show_report (
    id_bat in out number,
    id_usu out number,
    inicio_usu out number,
    final_usu out number,
    inicio_enem out number,
    final_enem out number,
    cost_metal_usu out number,
    cost_deut_usu out number,
    cost_metal_enem out number,
    cost_deut_enem out number,
    loss_metal_usu out number,
    loss_deut_usu out number,
    loss_metal_enem out number,
    loss_deut_enem out number,
    waste_metal out number,


```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 21                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

waste_deut out number,
ganador out number) as
begin
select
    id_battle,
    total_starting_units_user,
    total_remaining_units_user,
    total_starting_units_enemy,
    total_remaining_units_enemy,
    planet_metal_costs,
    planet_deuterium_costs,
    enemy_metal_costs,
    enemy_deuterium_costs,
    planet_metal_losses,
    planet_deuterium_losses,
    enemy_metal_losses,
    enemy_deuterium_losses,
    wastings_metal,
    wastings_deuterium,
    winner
into
    id_bat,
    inicio_usu,
    final_usu,
    inicio_enem,
    final_enem,
    cost_metal_usu,
    cost_deut_usu,
    cost_metal_enem,
    cost_deut_enem,
    loss_metal_usu,
    loss_deut_usu,
    loss_metal_enem,
    loss_deut_enem,
    waste_metal,
    waste_deut,
    ganador
from battle
where id_battle = id_bat;end;

```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 22                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

## M3 - PROGRAMACIÓN JAVA

### - Clase Planet

Clase Planet implementa la interfaz Variables (contiene variables estáticas con valores predeterminados de inicio o futuros cambios), contiene los métodos necesarios para subir los niveles, método para la comprobación de los recursos necesarios para la construcción de una nave en específico y todos los métodos para añadir tropas y defensas.

```
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.SQLException;
import java.util.ArrayList;

import javax.swing.JOptionPane;

public class Planet implements Variables {

    private int technologyDefense=0;
    private int technologyAttack=0;
    private int metal;
    private int deuterium;
    private int
upgradeDefenseTechnologyDeuteriumCost=UPGRADE_BASE_DEFENSE_TECHNOLOGY_DE
UTERIUM_COST;
    private int
upgradeAttackTechnologyDeuteriumCost=UPGRADE_BASE_ATTACK_TECHNOLOGY_DEUT
ERIUM_COST;
    ArrayList<MilitaryUnit>[] army = new ArrayList[7];

    public void setTechnologyDefense(int technologyDefense) {
        this.technologyDefense = technologyDefense;
    }
    public void setTechnologyAttack(int technologyAttack) {
        this.technologyAttack = technologyAttack;
    }
    /*
Army[0]  arrayList de Ligth Hunter
```



```
Army[1]  arrayList de Heavy Hunter
Army[2]  arrayList de Battle Ship
Army[3]  arrayList de Armored Ship
Army[4]  arrayList de Missile Launcher
Army[5]  arrayList de Ion Cannon
Army[6]  arrayList de Plasma Cannon
*/
public Planet () {
    for (int i=0;i<7;i++) {
        army[i]= new ArrayList<MilitaryUnit>();
    }

}
boolean Comprobador(int Metal,int Deuterium) {
    try {
        if (this.metal-Metal<0 || this.deuterium-Deuterium<0)
        {
            throw new ResourceException();
        }
        this.metal=this.metal-Metal;
        this.deuterium=this.deuterium-Deuterium;
        return true;
    } catch (ResourceException e) {
        // TODO Auto-generated catch block
        //throw new ResourceException();
        System.out.println(e.getMessage());
        return false;
    }
}
void upgradeTechnologyDefense() {

    if
(Comprobador(0,this.upgradeDefenseTechnologyDeuteriumCost)==true) {
        this.technologyDefense++;

this.upgradeDefenseTechnologyDeuteriumCost=this.upgradeDefenseTechnology
DeuteriumCost+(this.upgradeDefenseTechnologyDeuteriumCost*10)/100;

//this.upgradeDefenseTechnologyDeuteriumCost=this.upgradeDefenseTechnolo
```



MANUAL DE  
DESCRIPCIÓ I  
ESPECIFICACIÓ DE  
CÀRRECS DE LA  
EMPRESA XYZ,CA

Pàgina N° 24

Data d'Elaboració: 09/05/2022

Data d'Aprovació: 18/05/2022

```
gyDeuteriumCost+UPGRADE_PLUS_DEFENSE_TECHNOLOGY_DEUTERIUM_COST;  
  
//this.upgradeDefenseTechnologyDeuteriumCost=this.upgradeDefenseTechnolo  
gyDeuteriumCost+(UPGRADE_PLUS_DEFENSE_TECHNOLOGY_DEUTERIUM_COST*60)/100;  
    }  
    else {  
        JOptionPane.showMessageDialog(null, "ERROR:  
Insuficient resources");  
    }  
}  
void upgradeTechnologyAttack() {  
  
    if (Comprobador(0,  
this.upgradeAttackTechnologyDeuteriumCost)==true) {  
  
        this.technologyAttack++;  
  
this.upgradeAttackTechnologyDeuteriumCost=this.upgradeAttackTechnologyDe  
uteriumCost+(this.upgradeAttackTechnologyDeuteriumCost*10)/100;  
    }  
    else {  
        JOptionPane.showMessageDialog(null, "ERROR:  
Insuficient resources");  
    }  
}  
  
void newLigthHunter(int n,Connection con) {  
    int conta=0;  
    CallableStatement cst;  
    InfoShips ship=new InfoShips();  
    cst= ship.getInfoShips(con, 1);  
    String insu="";  
    for (int i=0;i<n;i++) {  
        try {  
            cst.execute();  
            if (Comprobador(cst.getInt(3),  
cst.getInt(5))==true) {  
                conta++;  
                LightHunter l= new
```





```
LightHunter(technologyDefense, technologyAttack,con);
        this.army[0].add(1);
    }
    else {
        insu=("ERROR: Insuficient resources \n");
        break;
    }
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
}
JOptionPane.showMessageDialog(null, insu+"Added : "+conta+"
Ligth hunters");
}

void newHeavyHunter(int n,Connection con) {
    int conta=0;
    CallableStatement cst;
    InfoShips ship=new InfoShips();
    cst= ship.getInfoShips(con, 2);
    String insu="";
    for (int i=0;i<n;i++) {
        try {
            cst.execute();
            if (Comprobador(cst.getInt(3),
cst.getInt(5))==true) {
                conta++;
                HeavyHunter h= new
HeavyHunter(technologyDefense, technologyAttack,con);
                this.army[1].add(h);
            }
            else {
                insu=("ERROR: Insuficient resources \n");
                break;
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```



```
    }  
    }  
    JOptionPane.showMessageDialog(null, insu+"Added : "+conta+"  
Heavy hunters");  
}  
void newBattleShip(int n,Connection con) {  
    int conta=0;  
    CallableStatement cst;  
    InfoShips ship=new InfoShips();  
    cst= ship.getInfoShips(con, 3);  
    String insu="";  
    for (int i=0;i<n;i++) {  
        try {  
            cst.execute();  
            if (Comprobador(cst.getInt(3),  
cst.getInt(5))==true) {  
                conta++;  
                BattleShip b= new  
BattleShip(technologyDefense, technologyAttack,con);  
                this.army[2].add(b);  
            }  
            else {  
                insu=("ERROR: Insuficient resources \n");  
                break;  
            }  
        } catch (SQLException e) {  
            // TODO Auto-generated catch block  
            e.printStackTrace();  
        }  
    }  
    JOptionPane.showMessageDialog(null, insu+"Added : "+conta+"  
Battleships");  
}  
void newArmoredShip(int n,Connection con) {  
    int conta=0;  
    CallableStatement cst;  
    InfoShips ship=new InfoShips();  
    cst= ship.getInfoShips(con, 4);  
    String insu="";
```



```
        for (int i=0;i<n;i++) {
            try {
                cst.execute();
                if (Comprobador(cst.getInt(3),
cst.getInt(5))==true) {
                    conta++;
                    ArmoredShip a= new
ArmoredShip(technologyDefense, technologyAttack,con);
                    this.army[3].add(a);
                }
                else {
                    insu("ERROR: Insuficient resources \n");
                    break;
                }
            } catch (SQLException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
        JOptionPane.showMessageDialog(null, insu+"Added : "+conta+"
Armored ships");
    }

//DEFENSAS-----
-----
    void newMissileLauncher(int n,Connection con) {
        int conta=0;
        CallableStatement cst;
        InfoDefense def=new InfoDefense();
        cst= def.getInfoDefense(con, 1);
        String insu="";
        for (int i=0;i<n;i++) {
            try {
                cst.execute();
                if (Comprobador(cst.getInt(3),
cst.getInt(5))==true) {
                    conta++;
                    MissileLauncher m= new
MissileLauncher(technologyDefense, technologyAttack,con);
```



```
        this.army[4].add(m);
    }
    else {
        insu("ERROR: Insuficient resources \n");
        break;
    }
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
}
OptionPane.showMessageDialog(null, insu+"Added : "+conta+"
Missile Launchers");
}
void newIonCannon(int n,Connection con) {
    int conta=0;
    CallableStatement cst;
    InfoDefense def=new InfoDefense();
    cst= def.getInfoDefense(con, 2);
    String insu="";
    for (int i=0;i<n;i++) {


        try {
            cst.execute();
            if (Comprobador(cst.getInt(3),
cst.getInt(5))==true) {
                conta++;
                IonCannon io= new
IonCannon(technologyDefense, technologyAttack,con);
                this.army[5].add(io);
            }
            else {
                insu("ERROR: Insuficient resources \n");
                break;
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```



```
    }
    JOptionPane.showMessageDialog(null, insu+"Added : "+conta+"
Ion Cannons");
}
void newPlasmaCannon(int n,Connection con) {
    CallableStatement cst;
    InfoDefense def=new InfoDefense();
    cst= def.getInfoDefense(con, 3);
    int conta=0;
    String insu="";
    for (int i=0;i<n;i++) {
        try {
            cst.execute();
            if (Comprobador(cst.getInt(3),
cst.getInt(5))==true) {
                conta++;
                PlasmaCannon p= new
PlasmaCannon(technologyDefense, technologyAttack,con);
                this.army[6].add(p);
            }
            else {
                insu=("ERROR: Insuficient resources \n");
                break;
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
    JOptionPane.showMessageDialog(null, insu+"Added : "+conta+"
Plasma Cannons");
}
public void setUpgradeDefenseTechnologyDeuteriumCost(int
upgradeDefenseTechnologyDeuteriumCost) {
    this.upgradeDefenseTechnologyDeuteriumCost =
upgradeDefenseTechnologyDeuteriumCost;
}
public void setUpgradeAttackTechnologyDeuteriumCost(int
upgradeAttackTechnologyDeuteriumCost) {
```



```
        this.upgradeAttackTechnologyDeuteriumCost =  
upgradeAttackTechnologyDeuteriumCost;  
    }  
    public void printStats() {  
        System.out.println("Planet Stats:\n\nTECHNOLOGY\n");  
        System.out.println("Attack Technology:  
"+this.technologyAttack);  
        System.out.println("Defense Technology:  
"+this.technologyDefense);  
        System.out.println("\nDEFENSES\n");  
        System.out.println("Missile Launcher:  
"+this.army[4].size());  
        System.out.println("Ion Cannon: "+this.army[5].size());  
        System.out.println("Plasma Cannon: "+this.army[6].size());  
        System.out.println("\nFLEET\n");  
        System.out.println("Light hunter: "+this.army[0].size());  
        System.out.println("Heavy hunter: "+this.army[1].size());  
        System.out.println("Battle ship: "+this.army[2].size());  
        System.out.println("Armored Ship: "+this.army[3].size());  
        System.out.println("\nRESOURCES\n");  
        System.out.println("Metal: "+this.metal+"\nDeuterium:  
"+this.deuterium);  
    }  
    //setters y getters  
    public int getMetal() {  
        return metal;  
    }  
    public void setMetal(int metal) {  
        this.metal = metal;  
    }  
    public int getDeuterium() {  
        return deuterium;  
    }  
    public void setDeuterium(int deuterium) {  
        this.deuterium = deuterium;  
    }  
    public ArrayList<MilitaryUnit>[] getArmy() {  
        return army;  
    }  
}
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 31                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

    }
    public void setArmy(ArrayList<MilitaryUnit>[] army) {
        this.army = army;
    }
    public int getTechnologyDefense() {
        return technologyDefense;
    }
    public int getTechnologyAttack() {
        return technologyAttack;
    }
    public int getUpgradeDefenseTechnologyDeuteriumCost() {
        return upgradeDefenseTechnologyDeuteriumCost;
    }
    public int getUpgradeAttackTechnologyDeuteriumCost() {
        return upgradeAttackTechnologyDeuteriumCost;
    }
}

```

## - Clase Ship

Clase abstracta Ship, esta implementa la interfaz MilitaryUnit y Variables. (La interfaz MilitaryUnit contiene los métodos necesarios para las futuras acciones de las naves y la Variables contiene variables estáticas con valores predeterminados de inicio o futuros cambios)


Esta clase aparte de heredar los métodos de MilitaryUnit tiene 3 atributos propios, armadura, armadura inicial y daño base (principales características de las naves)

```

import java.sql.Connection;
public abstract class Ship implements MilitaryUnit, Variables {
    private int armor;
    private int initialArmor;
    private int baseDamage;
    private int quantity;
    // Metodos MilitaryUnit
    public int attack() {return baseDamage;}
    public void takeDamage(int receivedDamage) {
        this.armor -= receivedDamage;
    }
}

```

|                       |                      |                            |
|-----------------------|----------------------|----------------------------|
| Elaborat per: Grupo 2 | Revisat per: Grupo 2 | Autoritzat per: Josep Diaz |
|-----------------------|----------------------|----------------------------|

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 32                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

public int getActualArmor() {return armor;}
public abstract int getMetalCost();
public abstract int getDeuteriumCost();
public abstract int getChanceGeneratinWaste();
public abstract int getChanceAttackAgain();
public void resetArmor() {armor = initialArmor;}
public int getQuantity() {
    return quantity;
}
public void setQuantity(int quantity) {
    this.quantity = quantity;
}

// Setters y Getters
public int getArmor() {return armor;}
public int getInitialArmor() {return initialArmor;}
public int getBaseDamage() {return baseDamage;}
public void setArmor(int armor) {this.armor = armor;}
public void setInitialArmor(int initialArmor) {this.initialArmor =
initialArmor;}
    public void setBaseDamage(int baseDamage) {this.baseDamage =
baseDamage;}
}

```

## - Clase HeavyHunter

Clase HeavyHunter hereda de Ship ya que es un tipo de nave. También implementa las interfaces MilitaryUnit y Variables porque estás naves necesitan los mismos métodos que Ship y se utilizan las mismas Variables estáticas para sacar los valores predefinidos.

```

import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.SQLException;

public class HeavyHunter extends Ship implements Variables,MilitaryUnit{
    CallableStatement cst;
    public HeavyHunter(int tecnoDefense, int tecnoAtack,Connection

```






```
con) {
    InfoShips ship=new InfoShips();
    cst= ship.getInfoShips(con, 2);
    try {
        cst.execute();
        super.setInitialArmor(cst.getInt(6));
        super.setBaseDamage(cst.getInt(8));
        this.setArmor(this.getInitialArmor() +
(tecnoDefense*PLUS_ARMOR_LIGHTHUNTER_BY_TECHNOLOGY)*this.getInitialArmor
())/100);

this.setBaseDamage(this.getBaseDamage()+(tecnoAtack*PLUS_ATTACK_LIGHTHUN
TER_BY_TECHNOLOGY)*this.getBaseDamage())/100);
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
@Override
public void takeDamage(int receivedDamage) {
    this.setArmor(this.getArmor()-receivedDamage);
}
@Override
public int getMetalCost() {
    // TODO Auto-generated method stub

    try {
        return cst.getInt(3);
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        return 0;
    }
}
@Override
public int getDeuteriumCost() {
    // TODO Auto-generated method stub
    try {
        return cst.getInt(5);
    } catch (SQLException e) {
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 34                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

        // TODO Auto-generated catch block
        return 0;
    }
}
@Override
public int getChanceGeneratinWaste() {
    // TODO Auto-generated method stub
    try {
        return cst.getInt(10);
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        return 0;
    }
}
@Override
public int getChanceAttackAgain() {
    // TODO Auto-generated method stub
    return this.CHANCE_ATTACK_AGAIN_HEAVYHUNTER;
}
}

```

## - Clase LightHunter

Clase LightHunter hereda de Ship ya que es un tipo de nave. También implementa las interfaces MilitaryUnit y Variables porque estás naves necesitan los mismos métodos que Ship y se utilizan las mismas Variables estáticas para sacar los valores predefinidos.

```

import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.SQLException;

//extends Ship
public class LightHunter extends Ship implements Variables,
MilitaryUnit{
    CallableStatement cst;
    public LightHunter(int tecnoDefense, int tecnoAtack,Connection con)

```



MANUAL DE  
DESCRIPCIÓ I  
ESPECIFICACIÓ DE  
CÀRRECS DE LA  
EMPRESA XYZ,CA

Pàgina N° 35

Data d'Elaboració: 09/05/2022

Data d'Aprovació: 18/05/2022

```
{
    InfoShips ship=new InfoShips();
    cst= ship.getInfoShips(con, 4);
    try {
        cst.execute();
        super.setInitialArmor(cst.getInt(6));
        super.setBaseDamage(cst.getInt(8));
        this.setArmor(this.getInitialArmor() +
(tecnoDefense*PLUS_ARMOR_LIGHTHUNTER_BY_TECHNOLOGY)*this.getInitialArmor
())/100);

this.setBaseDamage(this.getBaseDamage()+(tecnoAttack*PLUS_ATTACK_LIGHTHUN
TER_BY_TECHNOLOGY)*this.getBaseDamage())/100);
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

@Override
public void takeDamage(int receivedDamage) {
    this.setArmor(this.getArmor()-receivedDamage);
}

@Override
public int getMetalCost() {
    // TODO Auto-generated method stub


    try {
        return cst.getInt(3);
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        return 0;
    }
}

@Override
public int getDeuteriumCost() {
    // TODO Auto-generated method stub
    try {
        return cst.getInt(5);
    } catch (SQLException e) {
```

Elaborat per: Grupo 2

Revisat per: Grupo 2

Autoritzat per: Josep Diaz

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 36                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

        // TODO Auto-generated catch block
        return 0;
    }
}
@Override
public int getChanceGeneratinWaste() {
    // TODO Auto-generated method stub
    try {
        return cst.getInt(10);
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        return 0;
    }
}
@Override
public int getChanceAttackAgain() {
    // TODO Auto-generated method stub
    return this.CHANCE_ATTACK_AGAIN_LIGTHHUNTER;
}
}

```

## - Clase BattleShip

Clase BattleShip hereda de Ship ya que es un tipo de nave. También implementa las interfaces MilitaryUnit y Variables porque estás naves necesitan los mismos métodos que Ship y se utilizan las mismas Variables estáticas para sacar los valores predefinidos.

```

import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.SQLException;

public class BattleShip extends Ship implements Variables, MilitaryUnit {
    CallableStatement cst;
    public BattleShip(int tecnoDefense, int tecnoAtack, Connection con) {
        InfoShips ship = new InfoShips();
        cst = ship.getInfoShips(con, 3);
        try {


```



```
cst.execute();
super.setInitialArmor(cst.getInt(6));
super.setBaseDamage(cst.getInt(8));
this.setArmor(this.getInitialArmor() +
(tecnoDefense*PLUS_ARMOR_LIGHTHUNTER_BY_TECHNOLOGY)*this.getInitialArmor
())/100);

this.setBaseDamage(this.getBaseDamage()+(tecnoAttack*PLUS_ATTACK_LIGHTHUN
TER_BY_TECHNOLOGY)*this.getBaseDamage())/100);
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
@Override
public void takeDamage(int receivedDamage) {
    this.setArmor(this.getArmor()-receivedDamage);
}
@Override
public int getMetalCost() {
    // TODO Auto-generated method stub

    try {
        return cst.getInt(3);
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        return 0;
    }
}
@Override
public int getDeuteriumCost() {
    // TODO Auto-generated method stub
    try {
        return cst.getInt(5);
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        return 0;
    }
}
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 38                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

@Override
public int getChanceGeneratinWaste() {
    // TODO Auto-generated method stub
    try {
        return cst.getInt(10);
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        return 0;
    }
}

@Override
public int getChanceAttackAgain() {
    // TODO Auto-generated method stub
    return this.CHANCE_ATTACK_AGAIN_BATTLESHIP;
}
}

```

## - Clase ArmoredShip

Clase ArmoredShip hereda de Ship ya que es un tipo de nave. También implementa las interfaces MilitaryUnit y Variables porque estás naves necesitan los mismos métodos que Ship y se utilizan las mismas Variables estáticas para sacar los valores predefinidos.

```

import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.SQLException;

public class ArmoredShip extends Ship implements Variables,MilitaryUnit{
    CallableStatement cst;
    public ArmoredShip(int tecnoDefense, int tecnoAtack,Connection con)
    {
        InfoShips ship=new InfoShips();
        cst= ship.getInfoShips(con, 4);
        try {
            cst.execute();
            super.setInitialArmor(cst.getInt(6));
            super.setBaseDamage(cst.getInt(8));
        }
    }
}


```



```
        this.setArmor(this.getInitialArmor() +
(tecnoDefense*PLUS_ARMOR_LIGTHHUNTER_BY_TECHNOLOGY)*this.getInitialArmor
())/100);

this.setBaseDamage(this.getBaseDamage()+(tecnoAtack*PLUS_ATTACK_LIGTHHUN
TER_BY_TECHNOLOGY)*this.getBaseDamage())/100);
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
@Override
public void takeDamage(int receivedDamage) {
    this.setArmor(this.getArmor()-receivedDamage);
}
@Override
public int getMetalCost() {
    // TODO Auto-generated method stub

    try {
        return cst.getInt(3);
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        return 0;
    }
}
@Override
public int getDeuteriumCost() {
    // TODO Auto-generated method stub
    try {
        return cst.getInt(5);
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        return 0;
    }
}
@Override
public int getChanceGeneratinWaste() {
    // TODO Auto-generated method stub
    try {
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 40                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

        return cst.getInt(10);
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        return 0;
    }
}
@Override
public int getChanceAttackAgain() {
    // TODO Auto-generated method stub
    return this.CHANCE_ATTACK_AGAIN_ARMOREDSHIP;
}
}

```

## - Clase Defense

Clase abstracta Defense, esta implementa la interfaz MilitaryUnit y Variables. (La interfaz MilitaryUnit contiene los métodos necesarios para las futuras acciones de las naves y la Variables contiene variables estáticas con valores predeterminados de inicio o futuros cambios)

Esta clase aparte de heredar los métodos de MilitaryUnit tiene 3 atributos propios, armadura, armadura inicial y daño base (principales características de las naves)


```

public abstract class Defense implements Variables, MilitaryUnit {
    private int armor;
    private int initialArmor;
    private int baseDamage;
    private int quantity;

    public int attack() { return this.getBaseDamage(); }
    public void takeDamage(int receivedDamage)
    { this.setArmor(this.getActualArmor() - receivedDamage); }
    public int getActualArmor() { return this.getArmor(); }
    public void resetArmor() { armor = initialArmor; }
    public int getQuantity() {
        return quantity;
    }
    public void setQuantity(int quantity) {

```



|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 41                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

        this.quantity = quantity;
    }
    public int getArmor() {return this.armor;}
    public void setArmor(int armor) {this.armor = armor;}
    public int getInitialArmor() {return initialArmor;}
    public void setInitialArmor(int initialArmor) {this.initialArmor =
initialArmor;}
    public int getBaseDamage() {return this.baseDamage;}
    public void setBaseDamage(int baseDamage) {this.baseDamage =
baseDamage;}
}

```

## - Clase IonCannon

Clase IonCannon hereda de Defense ya que es un tipo de nave. También implementa las interfaces MilitaryUnit y Variables porque estás naves necesitan los mismos métodos que Defense y se utilizan las mismas Variables estáticas para sacar los valores predefinidos.

```

import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.SQLException;


public class IonCannon extends Defense implements MilitaryUnit{
    CallableStatement cst;

    IonCannon(int tecnoDefense, int tecnoAttack, Connection con) {
        InfoDefense def=new InfoDefense();

        cst = def.getInfoDefense(con,2);

        try {
            cst.execute();
            super.setInitialArmor(cst.getInt(6));
            super.setBaseDamage(cst.getInt(8));
            this.setArmor(this.getInitialArmor() +
(tecnoDefense*PLUS_ARMOR_IONCANNON_BY_TECHNOLOGY)*this.getInitialArmor())

```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 42                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

/100);

this.setBaseDamage(this.getBaseDamage()+(tecnoAttack*PLUS_ATTACK_IONCANNON_BY_TECHNOLOGY)*this.getBaseDamage()/100);
    }

    catch (SQLException e) {e.printStackTrace();}
}

@Override
public int attack() {return this.getBaseDamage();}

@Override
public void takeDamage(int receivedDamage)
{this.setArmor(this.getActualArmor() - receivedDamage);}

@Override
public int getActualArmor() {return this.getArmor();}


@Override
public int getChanceAttackAgain() {return
this.CHANCE_ATTACK_AGAIN_IONCANNON;}

@Override
public int getMetalCost() {
    try {return cst.getInt(3);}
    catch (SQLException e) {return 0;}
}

@Override
public int getDeuteriumCost() {
    try {return cst.getInt(5);}
    catch (SQLException e) {return 0;}
}

@Override
public int getChanceGeneratinWaste() {return
CHANCE_GENERATNG_WASTE_IONCANNON;}
}

```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 43                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

## - Clase PlasmaCannon

Clase PlasmaCannon hereda de Defense ya que es un tipo de nave. También implementa las interfaces MilitaryUnit y Variables porque estás naves necesitan los mismos métodos que Defense y se utilizan las mismas Variables estáticas para sacar los valores predefinidos.

```
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.SQLException;

public class PlasmaCannon extends Defense implements MilitaryUnit{
    CallableStatement cst;

    PlasmaCannon(int tecnoDefense, int tecnoAttack, Connection con) {
        InfoDefense def=new InfoDefense();

        cst = def.getInfoDefense(con,3);


        try {
            cst.execute();
            super.setInitialArmor(cst.getInt(6));
            super.setBaseDamage(cst.getInt(8));
            this.setArmor(this.getInitialArmor() +
(tecnoDefense*PLUS_ARMOR_PLASMACANNON_BY_TECHNOLOGY)*this.getInitialArmo
r()/100);

this.setBaseDamage(this.getBaseDamage()+(tecnoAttack*PLUS_ATTACK_PLASMACA
NNON_BY_TECHNOLOGY)*this.getBaseDamage()/100);

        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    @Override
    public int attack() {return this.getBaseDamage();}

    @Override
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 44                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

public void takeDamage(int receivedDamage)
{this.setArmor(this.getActualArmor() - receivedDamage);}

@Override
public int getActualArmor() {return this.getArmor();}

@Override
public int getChanceAttackAgain() {return
this.CHANCE_ATTACK_AGAIN_PLASMACANNON;}

@Override
public int getMetalCost() {
    try {return cst.getInt(3);}
    catch (SQLException e) {return 0;}
}

@Override
public int getDeuteriumCost() {
    try {return cst.getInt(5);}
    catch (SQLException e) {return 0;}
}

@Override
public int getChanceGeneratinWaste() {return
CHANCE_GENERATNG_WASTE_PLASMACANNON;}
}

```

## - Clase MissileLauncher

Clase MissileLauncher hereda de Defense ya que es un tipo de nave. También implementa las interfaces MilitaryUnit y Variables porque estás naves necesitan los mismos métodos que Defense y se utilizan las mismas Variables estáticas para sacar los valores predefinidos.

```

import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.SQLException;

```



```
public class MissileLauncher extends Defense implements MilitaryUnit{
    CallableStatement cst;

    MissileLauncher(int tecnoDefense, int tecnoAttack, Connection con) {
        InfoDefense def=new InfoDefense();

        cst = def.getInfoDefense(con,1);

        try {
            cst.execute();
            super.setInitialArmor(cst.getInt(6));
            super.setBaseDamage(cst.getInt(8));
            this.setArmor(this.getInitialArmor() +
            (tecnoDefense*PLUS_ARMOR_MISSILELAUNCHER_BY_TECHNOLOGY)*this.getInitialA
            rmor()/100);
            this.setBaseDamage(this.getBaseDamage()+(tecnoAttack*PLUS_ATTACK_MISSILEL
            AUNCHER_BY_TECHNOLOGY)*this.getBaseDamage()/100);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }


    @Override
    public int attack() {return this.getBaseDamage();}

    @Override
    public void takeDamage(int receivedDamage)
    {this.setArmor(this.getActualArmor() - receivedDamage);}

    @Override
    public int getActualArmor() {return this.getArmor();}

    @Override
    public int getChanceAttackAgain() {return
    this.CHANCE_ATTACK_AGAIN_MISSILELAUNCHER;}

    @Override
    public int getMetalCost() {
        try {return cst.getInt(3);}
        catch (SQLException e) {return 0;}
    }
}
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 46                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

    }
    @Override
    public int getDeuteriumCost() {
        try {return cst.getInt(5);}
        catch (SQLException e) {return 0;}
    }
    @Override
    public int getChanceGeneratinWaste() {return
CHANCE_GENERATNG_WASTE_MISSILELAUNCHER;}
}

```

## - Clase VentanaLogin

Clase VentanaLogin extiende de JFrame, sirve para que el usuario pueda iniciar sesión en el juego, todos sus datos almacenados en la base de datos serán cargados en la Ventana Inicial. Una vez has iniciado sesión puede empezar a jugar.


```

import org.jdatepicker.impl.*;

import javax.swing.*;
import javax.swing.border.EmptyBorder;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.Properties;

class VentanaLogin extends JFrame {
    private JLabel titulo,nom,pwd;
    private JTextField nombre;
    private JPasswordField password;
    private JPanel panelG,panelLogin;

```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 47                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

private JButton login,register;
InfoUsers infoU;
CallableStatement cst;

VentanaLogin(Connection con){
    iniciar();
    setTitle("Inicio de sesión");

    infoU = new InfoUsers();

    panelLogin = new JPanel();
    panelLogin.setOpaque(false);

    panelG = new JPanel(new FlowLayout());
    panelG = new JPanelConFondo(new
ImageIcon("space2.gif").getImage());

    GridLayout grid = new GridLayout(3,1);
    grid.setHgap(25);
    grid.setVgap(25);

    panelLogin.setLayout(grid);
    panelLogin.setBorder(new EmptyBorder(35, 35, 35, 35));

    titulo = new JLabel("Planet Wars");
    titulo.setFont(new Font("Aharoni",Font.BOLD,40));
    titulo.setForeground(Color.white);

    nom = new JLabel("Nombre: "); nombre = new JTextField(14);
    pwd = new JLabel("Contrasenya: "); password = new
JPasswordField(14);
    login = new JButton("Iniciar sesión");register = new
JButton("Registrarse");


    login.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            if (nombre.getText().isEmpty() ||
password.getPassword().length == 0){

```



```
JOptionPane.showMessageDialog(null, "Rellené los  
campos", "Inicio de sesión", JOptionPane.WARNING_MESSAGE, null);  
    } else {  
  
        int correcto =  
infoU.compruebaUser(con,nombre.getText(),String.valueOf(password.getPassword()));  
  
        if (correcto == 1){  
            JOptionPane.showOptionDialog(null, "Has iniciado  
sesión correctamente","Inicio de sesión",  
JOptionPane.DEFAULT_OPTION,JOptionPane.INFORMATION_MESSAGE, null, new  
Object[] {}, null);  
  
            dispose();  
            new VentanaInicial(con);  
        } else {  
            JOptionPane.showMessageDialog(  
                null,  
                "Error | Usuario o contraseña  
incorrecto",  
  
                "Inicio de sesión",  
                JOptionPane.INFORMATION_MESSAGE,  
                null);  
        }  
    }  
});  
register.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        dispose();  
        new VentanaRegister(con);  
    }  
});  
  
nom.setForeground(Color.white);pwd.setForeground(Color.white);  
  
panelLogin.add(nom);panelLogin.add(nombre);
```



|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 49                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

        panelLogin.add(pwd);panelLogin.add(password);
        panelLogin.add(register);panelLogin.add(login);

        panelG.add(titulo);
        panelG.add(panelLogin);

        add(panelG,BorderLayout.CENTER);
        this.setVisible(true);
    }

    public void iniciar(){

        this.setIconImage(new ImageIcon("ico.png").getImage());
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(650,350);
        // this.setResizable(false);
        this.setLocationRelativeTo(null);
    }
}

class VentanaRegister extends JFrame{


    private JLabel titulo,nom,fecha,pwd,planet;
    private JTextField nombre,planetName;
    private JPasswordField password;
    private JPanel panelG,panelLogin;
    private JButton register,login;
    InfoUsers infoU = new InfoUsers();
    CallableStatement cst;

    VentanaRegister(Connection con){
        iniciar();

        setTitle("User Registration");

        panelLogin = new JPanel();    panelG = new JPanel(new
FlowLayout());
        panelLogin.setOpaque(false); panelG = new JPanelConFondo(new

```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 50                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

ImageIcon("space2.gif").getImage());

// Calendario
UtilDateModel model = new UtilDateModel();
Properties p = new Properties();
p.put("text.today", "Today");p.put("text.month",
"Month");p.put("text.year", "Year");

JDatePanelImpl datePanel = new JDatePanelImpl(model, p);
JDatePickerImpl datePicker = new JDatePickerImpl(datePanel, new
DateLabelFormatter());

// Grid del Login
GridLayout grid = new GridLayout(5,1);
grid.setHgap(25);
grid.setVgap(25);

panelLogin.setLayout(grid);


// Padding del panel
panelLogin.setBorder(new EmptyBorder(35, 35, 35, 35));

// Labels y TextFields
titulo = new JLabel("Planet Wars");
titulo.setFont(new Font("Aharoni",Font.BOLD,40));
titulo.setForeground(Color.white);

planet = new JLabel("Planet Name: ");planetName = new
JTextField(14);
nom = new JLabel("Name: "); nombre = new JTextField(14);
pwd = new JLabel("Password: "); password = new
JPasswordField(14);
fecha = new JLabel("Birth Date: ");
register = new JButton("Register");login = new JButton("Login");

// Action Listeners
register.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {

```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 51                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

        java.util.Date selectedDate = (Date)
datePicker.getModel().getValue();
        DateFormat formatter = new
SimpleDateFormat("dd/MM/yyyy");

        if (selectedDate == null) {
            JOptionPane.showMessageDialog(null, "Fill up the
blanks", "User Registration", JOptionPane.WARNING_MESSAGE, null);
        } else {
            String today = formatter.format(selectedDate);

            SimpleDateFormat format = new
SimpleDateFormat("dd/MM/yyyy");
            try {
                Date parsed = format.parse(today);
                java.sql.Date sql = new
java.sql.Date(parsed.getTime());

                if (nombre.getText().isBlank() || selectedDate
== null || String.valueOf(password.getPassword()).isBlank()) {
                    JOptionPane.showMessageDialog(null, "Fill up
the blanks", "User Registration", JOptionPane.WARNING_MESSAGE, null);
                }
                else {
                    int existe = infoU.existeUser(con,
nombre.getText());

                    if (existe == 1) {
                        JOptionPane.showMessageDialog(null, "The
user already exists", "User Registration", JOptionPane.WARNING_MESSAGE,
null);
                    } else {
                        infoU.anyadeUser(con, nombre.getText(),
sql, String.valueOf(password.getPassword()));

                        int id =
infoU.getIdUser(con,nombre.getText());

```



```
infoU.anyadePlaneta(con,planetName.getText(),id);

                                JOptionPane.showMessageDialog(null, "
You have registred correctly | WELCOME ABOARD", "User Registration",
JOptionPane.WARNING_MESSAGE, null);
                                dispose();
                                new VentanaInicial(con);
                                }
                                }
                                }
                                catch (ParseException ex) {ex.printStackTrace();}
                                }
                                }


));
login.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        dispose();
        new VentanaLogin(con);
    }
});

planet.setForeground(Color.white);nom.setForeground(Color.white);
pwd.setForeground(Color.white);fecha.setForeground(Color.white);

// Añadir Objetos
panelLogin.add(planet);panelLogin.add(planetName);
panelLogin.add(nom);panelLogin.add(nombre);
panelLogin.add(fecha);panelLogin.add(datePicker);
panelLogin.add(pwd);panelLogin.add(password);
panelLogin.add(login);panelLogin.add(register);

panelG.add(titulo);
panelG.add(panelLogin);

add(panelG, BorderLayout.CENTER);
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 53                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

        this.setVisible(true);}

public void iniciar(){
    this.setIconImage(new ImageIcon("ico.png").getImage());
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setSize(650,375);
    // this.setResizable(false);
    this.setLocationRelativeTo(null);
}
}

class DateLabelFormatter extends JFormattedTextField.AbstractFormatter {


    private String datePattern = "dd/MM/yyyy";
    private SimpleDateFormat dateFormatter = new
SimpleDateFormat(datePattern);

    @Override
    public Object stringToValue(String text) throws ParseException {
        return dateFormatter.parseObject(text);
    }

    @Override
    public String valueToString(Object value) throws ParseException {
        if (value != null) {
            Calendar cal = (Calendar) value;
            return dateFormatter.format(cal.getTime());
        }

        return "";
    }
}

```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 54                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

## - Clase ConnectionBDD

Clase ConnectionBDD contiene la conexión a la base de datos y los métodos necesarios para realizar todas las operaciones necesarias en la base de datos, importaciones, exportaciones y cambios.

```
import java.sql.*;

public class ConnectionBDD {
    private Connection conex;
    String url;
    String user;
    String password;
    public ConnectionBDD() {
        // Carga el driver de oracle
        try {
            DriverManager.registerDriver(new
oracle.jdbc.driver.OracleDriver());

            // Conecta con la base de datos orcl con el usuario system y la
contrasea password
            conex =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:ORCLEBASE"
, "PLANETWARS", "PLANETWARS");
            System.out.println("Conectado MAQUINAAA");
        }

        catch (SQLException e) {System.out.println("Error de conexion");}
    }
    public Connection getConex() {
        return conex;
    }
}

class InfoShips {
    CallableStatement getInfoShips(Connection con, int id_nave) {
```



```
CallableStatement cst = null;
try {
    cst = con.prepareCall("{call getship
(?,?,?,?,?,?,?,?,?,?)}");

    // Parametro 1 del procedimiento almacenado
    cst.setInt(1, id_nave);

    // Definimos los tipos de los parametros de salida del
procedimiento almacenado
    cst.registerOutParameter(2, java.sql.Types.VARCHAR); //nombre
    cst.registerOutParameter(3, java.sql.Types.INTEGER); //metal
cost
    cst.registerOutParameter(4,
java.sql.Types.INTEGER); //cristal
    cst.registerOutParameter(5,
java.sql.Types.INTEGER); //deuterium
    cst.registerOutParameter(6,
java.sql.Types.INTEGER); //initial armor
    cst.registerOutParameter(7, java.sql.Types.INTEGER); //armor
    cst.registerOutParameter(8, java.sql.Types.INTEGER); //base
dmg
    cst.registerOutParameter(9, java.sql.Types.INTEGER); // speed
    cst.registerOutParameter(10, java.sql.Types.INTEGER); //
wastings
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

return cst;
}

}

class InfoPlanet {
    CallableStatement getInfoPlanet(Connection con, int id_planet) {
        CallableStatement cst = null;
        try {
            cst = con.prepareCall("{call get_planet
```



```
(?,?,?, ?, ?, ?, ?, ?)}");
    // Parametro 1 del procedimiento almacenado
    cst.setInt(1, id_planet);

    // Definimos los tipos de los parametros de salida del
procedimiento almacenado
    cst.registerOutParameter(2, java.sql.Types.VARCHAR);//name
    cst.registerOutParameter(3, java.sql.Types.INTEGER);//tech
atk
    cst.registerOutParameter(4, java.sql.Types.INTEGER);//tech
def
    cst.registerOutParameter(5,
java.sql.Types.INTEGER);//cristal
    cst.registerOutParameter(6, java.sql.Types.INTEGER);//metal
    cst.registerOutParameter(7,
java.sql.Types.INTEGER);//deuterio
    cst.registerOutParameter(8, java.sql.Types.INTEGER);//id
user
    } catch (SQLException e) {
        e.printStackTrace();
    }

    return cst;
}

int getInfoQuantityShips(Connection con, int id_planet,int
id_ship,int nivelAtk,int nivelDef) {
    CallableStatement cst = null;
    try {
        cst = con.prepareCall("{? = call
get_cantidadShip(?,?,?,?)}");

        // Parametro 1 del procedimiento almacenado
        cst.registerOutParameter(1,Types.INTEGER);
        cst.setInt(2, id_ship);
        cst.setInt(3, id_planet);
        cst.setInt(4, nivelAtk);
        cst.setInt(5, nivelDef);
        // Definimos los tipos de los parametros de salida del
```





procedimiento almacenado

```
        cst.execute();
        int fin=cst.getInt(1);
        cst.close();
        return fin;

    } catch (SQLException e) {

        return 0;
    }
}

int getInfoQuantityDefenses(Connection con, int id_planet,int
id_defense,int nivelAtk,int nivelDef) {
    CallableStatement cst = null;
    try {
        cst = con.prepareCall("{? = call get_cantidadDefenses
(?,?,?,?)}");
        // Parametro 1 del procedimiento almacenado
        cst.registerOutParameter(1,
java.sql.Types.INTEGER);//cantidad
        cst.setInt(2, id_defense);
        cst.setInt(3, id_planet);
        cst.setInt(4, nivelAtk);
        cst.setInt(5, nivelDef);

        // Definimos los tipos de los parametros de salida del
procedimiento almacenado
        cst.execute();
        int fin=cst.getInt(1);
        cst.close();
        return fin;

    } catch (SQLException e) {

        return 0;
    }
}

class InfoDefense {
```



```
CallableStatement getInfoDefense(Connection con, int id_nave) {
    CallableStatement cst = null;
    try {
        cst = con.prepareCall("{call getdefense
(?,?,?,?,?,?,?,?,?,?)}");
        // Parametro 1 del procedimiento almacenado
        cst.setInt(1, id_nave);

        // Definimos los tipos de los parametros de salida del
procedimiento almacenado
        cst.registerOutParameter(2, java.sql.Types.VARCHAR);
        cst.registerOutParameter(3, java.sql.Types.INTEGER);
        cst.registerOutParameter(4, java.sql.Types.INTEGER);
        cst.registerOutParameter(5, java.sql.Types.INTEGER);
        cst.registerOutParameter(6, java.sql.Types.INTEGER);
        cst.registerOutParameter(7, java.sql.Types.INTEGER);
        cst.registerOutParameter(8, java.sql.Types.INTEGER);
        cst.registerOutParameter(9, java.sql.Types.INTEGER);
        cst.registerOutParameter(10, java.sql.Types.INTEGER);
    } catch (SQLException e) {
        e.printStackTrace();
    }


    return cst;
}

class InfoUsers{
    int compruebaUser(Connection con, String name, String password) {
        CallableStatement cst = null;

        try {
            cst = con.prepareCall("{? = call login (?,?)}");

            cst.registerOutParameter(1, Types.INTEGER);
            cst.setString(2, name);
            cst.setString(3, password);

            cst.execute();
        }
    }
}
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 59                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

    }
    catch (SQLException e) {e.printStackTrace();}

    try {return cst.getInt(1);}
    catch (SQLException e){e.printStackTrace();}

    return 0;
}
int existeUser(Connection con, String name) {
    CallableStatement cst = null;

    try {
        cst = con.prepareCall("{? = call comprovar_usuari (?) }");
        cst.registerOutParameter(1, Types.INTEGER);
        cst.setString(2, name);
        cst.execute();
    }
    catch (SQLException e) {e.printStackTrace();}

    try {return cst.getInt(1);}
    catch (SQLException e){e.printStackTrace();}

    return 0;
}
void anyadeUser(Connection con, String name, java.sql.Date
fecha,String password) {
    CallableStatement cst = null;

    try {
        cst = con.prepareCall("{call add_user (?,?,?) }");
        cst.setString(1,name);
        cst.setDate(2,fecha);
        cst.setString(3, password);
        cst.execute();
    }
    catch (SQLException e) {e.printStackTrace();}
}
int getIdUser(Connection con, String name){

```



```
CallableStatement cst = null;

try {
    cst = con.prepareCall("{? = call getid (?) }");
    cst.registerOutParameter(1,Types.INTEGER);
    cst.setString(2,name);
    cst.execute();
}
catch (SQLException e) {e.printStackTrace();}

try {return cst.getInt(1);}
catch (SQLException e){e.printStackTrace();}
return 0;
}

void anyadePlaneta(Connection con, String name,int id_user) {
    CallableStatement cst = null;

    try {
        cst = con.prepareCall("{call add_planet (?, ?, ?, ?, ?, ?, ?, ?)
    }");

        cst.setString(1,name);
        cst.setInt(2,0);
        cst.setInt(3,0);
        cst.setInt(4,0);
        cst.setInt(5,0);
        cst.setInt(6,0);
        cst.setInt(7, id_user);
        cst.execute();

    }
    catch (SQLException e) {e.printStackTrace();}
}

CallableStatement getInfoPlanetId(Connection con, String usuario)
{
    CallableStatement cst = null;
    try {
        cst = con.prepareCall("{call get_planetusu (?,?) }");

        // Parametro 1 del procedimiento almacenado
```



```
cst.setString(1, usuario);

// Definimos los tipos de los parametros de salida del
procedimiento almacenado

cst.registerOutParameter(2, java.sql.Types.INTEGER);

} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}


return cst;
}
}
class UpdateGame {
    public UpdateGame(Connection con,int id_usu,int id_planetita,int
tecnoAtk,int tecnoDef,int metalito,int deuterito) {
        CallableStatement cst = null;
        try {
            cst = con.prepareCall("{call updatePlanet (?, ?, ?, ?, ?, ?)
}");

            // Parametro 1 del procedimiento almacenado
            cst.setInt(1, id_usu);
            cst.setInt(2, id_planetita);
            cst.setInt(3, tecnoAtk);
            cst.setInt(4, tecnoDef);
            cst.setInt(5, metalito);
            cst.setInt(6, deuterito);

            cst.execute();
            cst.close();
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```



```
    }  
}  
  
public void UpdateShips(Connection con,int id_planetita,int  
id_shipy,int tecnoAtk,int tecnoDef,int canti) {  
    CallableStatement cst = null;  
    try {  
        cst = con.prepareCall("{call updateShips (?,?,?,?,?) }");  
  
        // Parametro 1 del procedimiento almacenado  
        cst.setInt(1, id_planetita);  
        cst.setInt(2, tecnoAtk);  
        cst.setInt(3, tecnoDef);  
        cst.setInt(4, id_shipy);  
        cst.setInt(5, canti);  
  
        cst.execute();  
        cst.close();  
    } catch (SQLException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
}  
  
public void UpdateDefenses(Connection con,int id_planetita,int  
id_defensy,int tecnoAtk,int tecnoDef,int canti) {  
    CallableStatement cst = null;  
    try {  
        cst = con.prepareCall("{call updateDefenses (?,?,?,?,?)  
}");  
  
        // Parametro 1 del procedimiento almacenado  
        cst.setInt(1, id_planetita);  
        cst.setInt(2, tecnoAtk);  
        cst.setInt(3, tecnoDef);  
        cst.setInt(4, id_defensy);  
        cst.setInt(5, canti);  
  
        cst.execute();  
        cst.close();  
    } catch (SQLException e) {
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 63                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
}

```

## - Clase VentanaInicial

Clase VentanaInicial es la interfaz que contiene todo lo relacionado con la gestión de nuestro planeta, al principio carga todos los datos de la BDD y tiene timers para hacer guardados automáticos, subir los recursos y para la probabilidad de que nos ataquen.

```

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.sql.*;
import java.util.ArrayList;

import javax.swing.*;
import javax.swing.border.EmptyBorder;
import javax.swing.event.ChangeEvent;
import javax.swing.event.ChangeListener;

public class VentanaInicial extends JFrame implements Variables{

    private JTabbedPane pestanyas;
    private JTabbedPane build;
    private JPanel inp1,panel2, panel3,panel4; //paneles principales
    private JPanel panelB1,panelB2,panelBp1,panelBp2,panelBp10;
    //subpaneles

    private JPanel panelFinalP1,panelFinalP2,panelFinalP3,panelFinalP4;
    // Paneles Inp1

```



```
private JTextField[] textoP1= new JTextField[5];
private JLabel[] etiquetaP1=new JLabel[10];
// build
private JButton[] btnBuild=new JButton[5];
private JButton[] btnBuild2=new JButton[3];
private JLabel[] etiquetaBuild=new JLabel[8];
private JLabel[] etiquetaBuild2=new JLabel[8];
private ArrayList<JLabel>[] labelsBuild=new ArrayList[5];
private ArrayList<JLabel>[] labelsBuild2=new ArrayList[3];
private JTextField[] buildText=new JTextField[4];
private JTextField[] buildText2=new JTextField[3];
private JLabel[] images=new JLabel[7];
private JPanel[] textoformatado=new JPanel[4];
private JPanel[] textoformatado2=new JPanel[4];
private JLabel[] etiquetaP3=new JLabel[6];

private JButton botonP11, botonP12,salir;

private JLabel
deuterium,metal,lgtCant,hvyCant,btlCant,missileCant,ionCant,plasmaCant,a
rshipCant;

private CallableStatement cst;
InfoShips ship= new InfoShips();
InfoUsers usu= new InfoUsers();
InfoPlanet plane= new InfoPlanet();
InfoDefense defe=new InfoDefense();

private JLabel[] buildTroopsdef=new JLabel[4];
private JLabel[] buildTroopsatk=new JLabel[4];

private JLabel attackTech,defenseTech;
private JLabel planetGIF;

private Planet planeta;
private int IDplanetita;
private int compro=0;

ArrayList<MilitaryUnit>[] armyInicial = new ArrayList[7];
```





MANUAL DE  
DESCRIPCIÓ I  
ESPECIFICACIÓ DE  
CÀRRECS DE LA  
EMPRESA XYZ,CA

Pàgina N° 65

Data d'Elaboració: 09/05/2022

Data d'Aprovació: 18/05/2022

```
public void aumentadorRecursos(int metalito,int deuterito) {
    metal.setText(String.valueOf(metalito));
    deuterium.setText(String.valueOf(deuterito));
}

public VentanaInicial(Connection con,String usuario) {
    for (int i=0;i<7;i++) {
        armyInicial[i]= new ArrayList<MilitaryUnit>();
    }
    /*
        Army[0]  arrayList de Ligth Hunter
        Army[1]  arrayList de Heavy Hunter
        Army[2]  arrayList de Battle Ship
        Army[3]  arrayList de Armored Ship
        Army[4]  arrayList de Missile Launcher
        Army[5]  arrayList de Ion Cannon
        Army[6]  arrayList de Plasma Cannon
    */

    //INICIAR PLANETA
    planeta=new Planet();

    try {

        cst=usu.getInfoPlanetId(con, usuario);
        cst.execute();
        IDplanetita=cst.getInt(2);
        cst=plane.getInfoPlanet(con, IDplanetita);
        cst.execute();

        planeta.setTechnologyAtack(cst.getInt(3));
//tecnoatk
        planeta.setTechnologyDefense(cst.getInt(4)); //tecno
def
        planeta.setDeuterium(cst.getInt(7));
        planeta.setMetal(cst.getInt(6));

        //for nave del 0 a 4 for tecno1 for tecno 2 si no es 0
```



```
for anyadir naves
    for (int i=0;i<4;i++) {

        for (int
j=0;j<planeta.getTechnologyAttack()+1;j++) {

            for (int
k=0;k<planeta.getTechnologyDefense()+1;k++) {
                int
canti_defense=plane.getInfoQuantityDefenses(con, IDplanetita, i+1, j,
k);

                int canti =
plane.getInfoQuantityShips(con, IDplanetita , i+1, j, k);
                if(canti>0) {
                    for (int l=0; l < canti ;l++) {
                        if (i==0) {

armyInicial[0].add(new LightHunter(planeta.getTechnologyDefense(),
planeta.getTechnologyAttack(), con));

                        }
                        else if(i==1) {

armyInicial[1].add(new HeavyHunter(planeta.getTechnologyDefense(),
planeta.getTechnologyAttack(), con));

                        }
                        else if(i==2) {

armyInicial[2].add(new BattleShip(planeta.getTechnologyDefense(),
planeta.getTechnologyAttack(), con));

                        }
                        else if(i==3) {

armyInicial[3].add(new ArmoredShip(planeta.getTechnologyDefense(),
planeta.getTechnologyAttack(), con));

                        }
                    }}
                if(canti_defense>0) {
                    for (int
h=0;h<canti_defense;h++) {
```



MANUAL DE  
DESCRIPCIÓ I  
ESPECIFICACIÓ DE  
CÀRRECS DE LA  
EMPRESA XYZ,CA

Pàgina N° 67

Data d'Elaboració: 09/05/2022

Data d'Aprovació: 18/05/2022

```

        if (i==0) {
            armyInicial[4].add(new MissileLauncher(planeta.getTechnologyDefense(),
            planeta.getTechnologyAttack(), con));
        }
        else if(i==1) {
            armyInicial[5].add(new IonCannon(planeta.getTechnologyDefense(),
            planeta.getTechnologyAttack(), con));
        }
        else if(i==2) {
            armyInicial[6].add(new PlasmaCannon(planeta.getTechnologyDefense(),
            planeta.getTechnologyAttack(), con));
        }
    }
}

planeta.setArmy(armyInicial);


for (int k=0;k<planeta.getTechnologyAttack();k++) {
    planeta.setUpgradeAttackTechnologyDeuteriumCost(planeta.getUpgradeAttack
    TechnologyDeuteriumCost()+planeta.getUpgradeAttackTechnologyDeuteriumCos
    t()*10/100);
}
for (int g=0;g<planeta.getTechnologyDefense();g++) {
    planeta.setUpgradeDefenseTechnologyDeuteriumCost(planeta.getUpgradeDefen
    seTechnologyDeuteriumCost()
    +planeta.getUpgradeDefenseTechnologyDeuteriumCost()*10/100);
} catch (SQLException e2) {
    // TODO Auto-generated catch block
    e2.printStackTrace();
}

```

Elaborat per: Grupo 2

Revisat per: Grupo 2

Autoritzat per: Josep Diaz

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 68                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

    }
    compro=0;
    Timer countdown = new Timer (180000,new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent e) {
            // TODO Auto-generated method stub
            guardado(con, usuario);
            System.out.println("SAVEPOINT");
            compro=1;
            dispose();
            new BattleScreen(planeta, new
Enemy(con).getEnemyArmy(), con, usu.getIdUser(con, usuario),usuario);

        }
    });
    Timer batalla =new Timer (65000,new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent e) {
            // TODO Auto-generated method stub
            double rnd=Math.floor(Math.random()*10);
            System.out.println(rnd);
            if (rnd==5.0) {
                countdown.start();
                JOptionPane.showMessageDialog(null, "ENEMY SHIPS INCOMING!\nYOU HAVE 3
MINUTES TO PREPARE THE DEFENSES"); }
            }
    });
    //TIMERS
    //timer de recursos
    Timer timer = new Timer (1000, new ActionListener ()
    {
        public void actionPerformed(ActionEvent e)
        {
            // Aqu el cdigo que queremos ejecutar.
            planeta.setDeuterium(planeta.getDeuterium()+100);
            planeta.setMetal(planeta.getMetal()+1000);
            aumentadorRecursos(planeta.getMetal(),

```



```
planeta.getDeuterium());

    }

});
timer.start();
Timer savepoint=new Timer (60000,new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        guardado(con, usuario);
        System.out.println("SAVEPOINT");
    }

});
savepoint.start();
Timer Stopper=new Timer (500,new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        if (countdown.isRunning()==true) {
            batalla.stop();
            timer.stop();
            savepoint.stop();
            if (compro==1) {
                countdown.stop();
            }
        }
    }

});
Stopper.start();
batalla.start();
inicializarTextos(con,usuario);
this.setSize(800, 600);
setUndecorated(true);
//undecorated para eliminar la barra de maximizar y cerrar
Image imagen = Toolkit.getDefaultToolkit().getImage("eve.jpg");
Image imagen2 =
Toolkit.getDefaultToolkit().getImage("space.jpg");
```



MANUAL DE  
DESCRIPCIÓ I  
ESPECIFICACIÓ DE  
CÀRRECS DE LA  
EMPRESA XYZ,CA

Pàgina N° 70

Data d'Elaboració: 09/05/2022

Data d'Aprovació: 18/05/2022

```
panel2=new JPanelConFondo(imagen);
panel3=new JPanelConFondo(imagen);
panel4=new JPanelConFondo(imagen);
panelB1=new JPanelConFondo(imagen2);

panelB2=new JPanelConFondo(imagen2);
panelBp1=new JPanelConFondo(imagen2);
panelBp2=new JPanelConFondo(imagen2);
panelBp10=new JPanelConFondo(imagen2);
panelBp10.setLayout(new GridLayout(2,1));


// Tocar estos grid layouts para agregar cosas
finales-----
panelB1.setLayout(new GridLayout(5,8)); //
iamgen,nombre,armor,dmg, cose metal, coste deuterio, texfield, boton
panelB2.setLayout(new GridLayout(4,8));

// arriba tocar
etiquetaP1[0].setPreferredSize(new Dimension(700,50));
//etiqueta gigante para redimensionar el layout y que el
jtabbedpane salga bien

//NO TOCAR ABAJO
panelBp10.add(panelBp1);

panelBp10.add(etiquetaP1[0]); // etiqueta de formato
pestanias=new JTabbedPane();
build=new JTabbedPane();
panelBp1.add(panelB1);
panelBp2.add(panelB2);

inp1=new JPanelConFondo(imagen);
GridLayout grid = new GridLayout(2,2);
grid.setHgap(15);
grid.setVgap(15);
inp1.setLayout(grid);
inp1.setBorder(new EmptyBorder(10, 10, 10, 10));
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 71                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

pestanyas.addTab("PLANET STATS",inp1);
pestanyas.addTab("BUILD",panel2);
pestanyas.addTab("UPGRADE TECHNOLOGY",panel3);
pestanyas.addTab("VIEW BATTLE REPORTS",panel4);
pestanyas.addTab("EXIT",salir);

pestanyas.setForeground(Color.BLUE);
pestanyas.setBackground(Color.BLACK);

build.addTab("TROOPS", panelBp10);
build.addTab("DEFENSES", panelBp2);

build.setForeground(Color.BLUE);
build.setBackground(Color.BLACK);


panel2.add(build);
this.add(pestanyas);

//boton salir del jpane, cuando el index es el de la pestanya
salir, salimos del programa con idex 4

pestanyas.addChangeListener(new ChangeListener() {

    @Override
    public void stateChanged(ChangeEvent e) {
        // TODO Auto-generated method stub
        if (pestanyas.getSelectedIndex()==4) {
            pestanyas.setSelectedIndex(0);
            int opc=JOptionPane.showOptionDialog(rootPane,"ARE
YOU SURE TO EXIT THE GAME?","Exit",JOptionPane.YES_NO_OPTION,
JOptionPane.WARNING_MESSAGE, null, new Object[] {"Yes","No"}, "No");
            switch (opc) {
                case 0:
                    guardado(con, usuario);
                    System.out.println("SAVEPOINT EXIT");
                    System.exit(EXIT_ON_CLOSE);
                    break;}}
            else if (pestanyas.getSelectedIndex()==1) {
                pestanyas.setSelectedIndex(1);

```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 72                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

        buildReset(con, usuario);
    }
    else if (pestanyas.getSelectedIndex()==0) {
        pestanyas.setSelectedIndex(0);
        armyInicial=planeta.getArmy();

lgtCant.setText(String.valueOf(armyInicial[0].size()));

hvyCant.setText(String.valueOf(armyInicial[1].size()));

btlCant.setText(String.valueOf(armyInicial[2].size()));

arshipCant.setText(String.valueOf(armyInicial[3].size()));

missileCant.setText(String.valueOf(armyInicial[4].size()));

ionCant.setText(String.valueOf(armyInicial[5].size()));

plasmaCant.setText(String.valueOf(armyInicial[6].size()));

attackTech.setText(String.valueOf(planeta.getTechnologyAtack()));

defenseTech.setText(String.valueOf(planeta.getTechnologyDefense()));

    }
}

});

this.setTitle("PLANET WARS");
this.setResizable(false);
//arriba para evitar redimensionar ventana , permitido por
defecto

Toolkit pantalla=Toolkit.getDefaultToolkit();
Image img=pantalla.getImage("ico.png");
this.setIconImage(img);

this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

```





```
// arriba para terminar programa al cerrar ventana

Dimension d=pantalla.getScreenSize();
int ancho=d.width;
int alto=d.height;
this.setLocation(ancho/2-this.getWidth()/2,
alto/2-this.getHeight()/2);
//arriba para centrar , independientemente de la pantalla

//ABAJO SE PUEDE TOCAR

// Pestanya 1 Planet Stats
-----
=====

// Panel global de Planet Stats
panelFinalP1=new JPanel();panelFinalP1.setOpaque(false);
// GIF del planeta giratorio
planetGIF = new JLabel(new
ImageIcon(Toolkit.getDefaultToolkit().getImage("planet_rotate.gif").getS
caledInstance(250,150,Image.SCALE_DEFAULT)));

etiquetaP1[7].setForeground(Color.white);etiquetaP1[7].setFont(new
Font("Aliens.ttf",Font.BOLD,40));
panelFinalP1.add(etiquetaP1[7]);
panelFinalP1.add(planetGIF);

panelFinalP2 = new JPanel();panelFinalP2.setOpaque(false);
// Image Icons de deuterio y metal
JLabel buildMtl = new JLabel(new
ImageIcon(Toolkit.getDefaultToolkit().getImage("metal.png").getScaledIns
tance(25, 25, Image.SCALE_DEFAULT)));
JLabel buildDtm = new JLabel(new
ImageIcon(Toolkit.getDefaultToolkit().getImage("deuterium.png").getScale
dInstance(25, 25, Image.SCALE_DEFAULT)));
// JPanel para los recursos del planeta
JPanel resources = new JPanel(new
GridLayout(1,5));resources.setBackground(new Color(247,235,232));
```



```
resources.setPreferredSize(new Dimension(175,50));
resources.setBorder(BorderFactory.createBevelBorder(1,new
Color(229,75,75),new Color(30,30,36)));

// JLabels de los recursos
deuterium = new
JLabel(String.valueOf(planeta.getDeuterium()));
metal = new JLabel(String.valueOf(planeta.getMetal()));
metal.setForeground(Color.black);
deuterium.setForeground(Color.black);


// Anyadir elementos
resources.add(buildMtl);resources.add(metal);
resources.add(buildDtm);resources.add(deuterium);

JPanel technology = new JPanel(new
GridLayout(2,3));technology.setBackground(new Color(247,235,232));
technology.setPreferredSize(new Dimension(325,100));
technology.setBorder(BorderFactory.createBevelBorder(1,new
Color(229,75,75),new Color(30,30,36)));

attackTech = new
JLabel(String.valueOf(planeta.getTechnologyAttack()));
defenseTech = new
JLabel(String.valueOf(planeta.getTechnologyDefense()));
JLabel buildTechAtk = new JLabel(new
ImageIcon(Toolkit.getDefaultToolkit().getImage("techAtk.png").getScaledI
nstance(35,35,Image.SCALE_DEFAULT)));
JLabel buildTechDef = new JLabel(new
ImageIcon(Toolkit.getDefaultToolkit().getImage("techDef.png").getScaledI
nstance(35,35,Image.SCALE_DEFAULT)));

etiquetaP1[5].setForeground(Color.black);attackTech.setForeground(new
Color(255,169,135));

etiquetaP1[6].setForeground(Color.black);defenseTech.setForeground(new
Color(255,169,135));
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 75                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```
technology.add(buildTechAtk);technology.add(etiquetaP1[5]);technology.add(attackTech);
```

```
technology.add(buildTechDef);technology.add(etiquetaP1[6]);technology.add(defenseTech);
```

```
panelFinalP2.add(resources);
panelFinalP2.add(technology);
```

```
panelFinalP3 = new JPanel();panelFinalP3.setLayout(new
BoxLayout(panelFinalP3, BoxLayout.X_AXIS));
panelFinalP3.setBorder(BorderFactory.createBevelBorder(1,new
Color(229,75,75),new Color(30,30,36)));
panelFinalP3.setBackground(new Color(247,235,232));
```

```
JPanel fleetP = new JPanel(new GridLayout(5,3));
fleetP.setOpaque(false);fleetP.setBorder(new
EmptyBorder(10,10,10,10));
JLabel fleetIMG = new JLabel(new
ImageIcon(Toolkit.getDefaultToolkit().getImage("fleet.png").getScaledIns
tance(35,35,Image.SCALE_DEFAULT)));
```

```
JLabel lightL = new JLabel("Light Hunter");lgtCant = new
JLabel(String.valueOf(armyInicial[0].size()));lgtCant.setForeground(new
Color(255,169,135));
```

```
JLabel heavyL = new JLabel("Heavy Hunter");hvyCant = new
JLabel(String.valueOf(armyInicial[1].size()));hvyCant.setForeground(new
Color(255,169,135));
```


```
JLabel battleL = new JLabel("Battle Ship");btlCant = new
JLabel(String.valueOf(armyInicial[2].size()));btlCant.setForeground(new
Color(255,169,135));
```

```
JLabel ArShipL = new JLabel("Armored Ship");arshipCant = new
JLabel(String.valueOf(armyInicial[3].size()));arshipCant.setForeground(n
ew Color(255,169,135));
```

```
etiquetaP1[8].setForeground(Color.black);etiquetaP1[8].setFont(new
Font("Arial",Font.BOLD,20));
```

```
fleetP.add(fleetIMG);fleetP.add(etiquetaP1[8]);
```

|                       |                      |                            |
|-----------------------|----------------------|----------------------------|
| Elaborat per: Grupo 2 | Revisat per: Grupo 2 | Autoritzat per: Josep Diaz |
|-----------------------|----------------------|----------------------------|

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 76                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

fleetP.add(lightL);fleetP.add(lgtCant);
fleetP.add(heavyL);fleetP.add(hvyCant);
fleetP.add(ArShipL);fleetP.add(arshipCant);
fleetP.add(battleL);fleetP.add(btlCant);

panelFinalP3.add(fleetP);

panelFinalP4 = new JPanel();panelFinalP4.setLayout(new
BoxLayout(panelFinalP4, BoxLayout.X_AXIS));
panelFinalP4.setBorder(BorderFactory.createBevelBorder(1,new
Color(229,75,75),new Color(30,30,36)));
panelFinalP4.setBackground(new Color(247,235,232));

JPanel defenseP = new JPanel(new
GridLayout(4,2));defenseP.setOpaque(false);defenseP.setBorder(new
EmptyBorder(10,10,10,10));
JLabel defenseIMG = new JLabel(new
ImageIcon(Toolkit.getDefaultToolkit().getImage("cannon.png").getScaledIn
stance(35,35,Image.SCALE_DEFAULT)));

JLabel missileL = new JLabel("Missile Launcher");missileCant
= new
JLabel(String.valueOf(armyInicial[4].size()));missileCant.setForeground(
new Color(255,169,135));
JLabel ionL = new JLabel("ION Cannon");ionCant = new
JLabel(String.valueOf(armyInicial[5].size()));ionCant.setForeground(new
Color(255,169,135));
JLabel plasmaL = new JLabel("Plasma Cannon");plasmaCant =
new
JLabel(String.valueOf(armyInicial[6].size()));plasmaCant.setForeground(n
ew Color(255,169,135));

etiquetaP1[9].setForeground(Color.black);etiquetaP1[9].setFont(new
Font("Arial",Font.BOLD,20));

defenseP.add(defenseIMG);defenseP.add(etiquetaP1[9]);

```

|                       |                      |                            |
|-----------------------|----------------------|----------------------------|
| Elaborat per: Grupo 2 | Revisat per: Grupo 2 | Autoritzat per: Josep Diaz |
|-----------------------|----------------------|----------------------------|



```
defenseP.add(missileL);defenseP.add(missileCant);
defenseP.add(ionL);defenseP.add(ionCant);
defenseP.add(plasmaL);defenseP.add(plasmaCant);

panelFinalP4.add(defenseP);

inp1.add(panelFinalP1);inp1.add(panelFinalP3);
inp1.add(panelFinalP2);inp1.add(panelFinalP4);

// Pestanya 2 BUILD
-----
// Troops
// Imagenes de build
Image ligh = Toolkit.getDefaultToolkit().getImage("light.png");
ImageIcon lgth= new ImageIcon(ligh.getScaledInstance(65, 65,
Image.SCALE_DEFAULT));
JLabel buildLight= new JLabel(lgth);
Image heavy = Toolkit.getDefaultToolkit().getImage("heavy.png");
ImageIcon hvy= new ImageIcon(heavy.getScaledInstance(65, 65,
Image.SCALE_DEFAULT));
JLabel buildHeavy = new JLabel(hvy);
Image armored =
Toolkit.getDefaultToolkit().getImage("armored.jpg");
ImageIcon armo= new ImageIcon(armored.getScaledInstance(65, 65,
Image.SCALE_DEFAULT));
JLabel buildArmored= new JLabel(armo);
Image battleshp =
Toolkit.getDefaultToolkit().getImage("battle.jpg");
ImageIcon bt1= new ImageIcon(battleshp.getScaledInstance(65,
65, Image.SCALE_DEFAULT));
JLabel buildBattle=new JLabel(bt1);
Image ion = Toolkit.getDefaultToolkit().getImage("ion.jpg");
ImageIcon ioncan= new ImageIcon(ion.getScaledInstance(65, 65,
Image.SCALE_DEFAULT));
JLabel buildIon=new JLabel(ioncan);
Image plasma =
```



```
Toolkit.getDefaultToolkit().getImage("plasma.jpg");
    ImageIcon plas= new ImageIcon(plasma.getScaledInstance(65, 65,
Image.SCALE_DEFAULT));
    JLabel buildPlasma=new JLabel(plas);
    Image missile =
Toolkit.getDefaultToolkit().getImage("missile.jpg");
    ImageIcon missl= new ImageIcon(missile.getScaledInstance(65, 65,
Image.SCALE_DEFAULT));
    JLabel buildMissile=new JLabel(missl);

    //-----cabecera
troops-----

    for (int i=0;i<etiquetaBuild.length;i++)
{panelB1.add(etiquetaBuild[i]);}

    // Declaracion de jlabels build
    InfoShips ship=new InfoShips();
    cst= ship.getInfoShips(con, 1);
    try {cst.execute();}
    catch (SQLException e1) {e1.printStackTrace();}
    //-----
    for (int l=0;l<6;l++) {images[l]=new JLabel();}

images[0]=buildLight;images[1]=buildHeavy;images[2]=buildArmored;
    images[3]=buildBattle;images[5]=buildIon;images[6]=buildPlasma;
    images[4]=buildMissile;

    for (int q=0;q<4;q++) {
        panelB1.add(images[q]);
        for (int k=0;k<5;k++) {
            panelB1.add(labelsBuild[q].get(k));
            labelsBuild[q].get(k).setForeground(Color.WHITE);
        }panelB1.add(textoformatado[q]);panelB1.add(btnBuild[q]);}

    // Botones tropa build

    btnBuild[0].addActionListener(new ActionListener() {
```



```
        public void actionPerformed(ActionEvent e) {
            try {

planeta.newLigthHunter(Integer.parseInt(buildText[0].getText()), con);}

            catch (NumberFormatException e1) {
                JOptionPane.showMessageDialog(null, "ERROR: must be
a number");
            }
        });
        btnBuild[1].addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                try {
                    planeta.newHeavyHunter
(Integer.parseInt(buildText[1].getText()), con);}

                    catch (NumberFormatException e1) {
                        JOptionPane.showMessageDialog(null, "ERROR: must
be a number");
                    }
                });
        btnBuild[2].addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                try {

planeta.newBattleShip(Integer.parseInt(buildText[2].getText()), con);}

                    catch (NumberFormatException e1) {
                        JOptionPane.showMessageDialog(null, "ERROR: must
be a number");
                    }
                });
        btnBuild[3].addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                try {

planeta.newArmoredShip(Integer.parseInt(buildText[3].getText()), con);}

                    catch (NumberFormatException e1) {
```



```
JOptionPane.showMessageDialog(null, "ERROR: must  
be a number");  
  
    }}  
});  
  
// Defenses  
for (int i=0;i<etiquetaBuild2.length;i++)  
{panelB2.add(etiquetaBuild2[i]);}  
for (int q=0;q<3;q++) {  
    panelB2.add(images[q+4]);  
    for (int k=0;k<5;k++) {  
        panelB2.add(labelsBuild2[q].get(k));  
        labelsBuild2[q].get(k).setForeground(Color.WHITE);  
    }  
    panelB2.add(textoformatado2[q]);  
    panelB2.add(btnBuild2[q]);}  
// botones def  
btnBuild2[0].addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        try {  
  
planeta.newMissileLauncher(Integer.parseInt(buildText2[0].getText()),  
con);}  
  
        catch (NumberFormatException e1) {  
            JOptionPane.showMessageDialog(null, "ERROR: must be  
a number");  
        }  
    }  
});  
btnBuild2[1].addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        try {  
  
planeta.newIonCannon(Integer.parseInt(buildText2[1].getText()), con);}  
  
        catch (NumberFormatException e1) {  
            JOptionPane.showMessageDialog(null, "ERROR: must be  
a number");  
        }  
    }  
});
```





```
        }}
    });
    btnBuild2[2].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            try {

planeta.newPlasmaCannon(Integer.parseInt(buildText2[2].getText()),
con);}

                catch (NumberFormatException e1) {
                    JOptionPane.showMessageDialog(null, "ERROR: must be
a number");
                }}
    });

//pestanya 3
technology-----
-----=====

    JPanel marcador = new JPanel();marcador.setOpaque(false);
    marcador.setBorder(new EmptyBorder(30,30,30,30));

    JPanel unitgrid_tecnoAttack = new
JPanel();unitgrid_tecnoAttack.setOpaque(false);


    JPanel unitgrid_tecnoDefense = new
JPanel();unitgrid_tecnoDefense.setOpaque(false);

    JPanel buttons = new JPanel(new
FlowLayout());buttons.setOpaque(false);

    JButton boton_attack = new JButton(new
ImageIcon(Toolkit.getDefaultToolkit().getImage("atkPlus.png").getScaledI
nstance(65,65,Image.SCALE_DEFAULT)));
    boton_attack.setBorderPainted(false);
    boton_attack.setContentAreaFilled(false);
    boton_attack.setFocusPainted(false);
    boton_attack.setOpaque(false);
```



```
        JButton boton_defense = new JButton(new  
        ImageIcon(Toolkit.getDefaultToolkit().getImage("defPlus.png").getScaledI  
nstance(65,65,Image.SCALE_DEFAULT)));  
        boton_defense.setBorderPainted(false);  
        boton_defense.setContentAreaFilled(false);  
        boton_defense.setFocusPainted(false);  
        boton_defense.setOpaque(false);  
  
        for (int i=0;i<etiquetaP3.length;i++){  
            etiquetaP3[i].setForeground(Color.black);  
        }  
        unitgrid_tecnoAtack.add(etiquetaP3[2]); //Titulo Coste Ataque  
        unitgrid_tecnoAtack.add(etiquetaP3[3]); //Titulo Nivel Ataque  
  
        JLabel DeuteriumCostAttack = new  
        JLabel(Integer.toString(planeta.getUpgradeAttackTechnologyDeuteriumCost(  
        )));DeuteriumCostAttack.setForeground(new Color(255,169,135));  
        JLabel lvlAttack = new  
        JLabel(Integer.toString(planeta.getTechnologyAtack()));lvlAttack.setFore  
ground(new Color(255,169,135));  
  
        unitgrid_tecnoAtack.add(DeuteriumCostAttack); //coste  
        unitgrid_tecnoAtack.add(lvlAttack); // nivel ataque  
  
        unitgrid_tecnoDefense.add(etiquetaP3[4]); //Titulo Coste Defense  
        unitgrid_tecnoDefense.add(etiquetaP3[5]); // Titulo Nivel Defense  
  
        JLabel DeuteriumCostDefense = new  
        JLabel(Integer.toString(planeta.getUpgradeDefenseTechnologyDeuteriumCost  
        ()));DeuteriumCostDefense.setForeground(new Color(255,169,135));  
        JLabel lvlDefense = new  
        JLabel(Integer.toString(planeta.getTechnologyDefense()));lvlDefense.setF  
oreground(new Color(255,169,135));  
  
        unitgrid_tecnoDefense.add(DeuteriumCostDefense); //coste defense  
        unitgrid_tecnoDefense.add(lvlDefense); //nivel defense  
  
        marcador.setLayout(new GridLayout(2, 3));
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 83                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

unitgrid_tecnoAttack.setLayout(new GridLayout(2, 2));
unitgrid_tecnoDefense.setLayout(new GridLayout(2, 2));

marcador.add(etiquetaP3[0]);
marcador.add(etiquetaP3[1]);
marcador.add(unitgrid_tecnoAttack);
marcador.add(unitgrid_tecnoDefense);

boton_attack.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        planeta.upgradeTechnologyAttack();

lvlAttack.setText(String.valueOf(planeta.getTechnologyAttack()));

DeuteriumCostAttack.setText(String.valueOf(planeta.getUpgradeAttackTechn
ologyDeuteriumCost()));
    }
});

boton_defense.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        planeta.upgradeTechnologyDefense();

lvlDefense.setText(String.valueOf(planeta.getTechnologyDefense()));

DeuteriumCostDefense.setText(String.valueOf(planeta.getUpgradeDefenseTec
hnologyDeuteriumCost()));
    }
});

buttons.add(boton_attack);
buttons.add(boton_defense);

JPanel panelGeneral3 = new JPanel();panelGeneral3.setLayout(new
GridLayout(2, 1));panelGeneral3.setBackground(new Color(247,235,232));
    panelGeneral3.setBorder(BorderFactory.createBevelBorder(1,new
Color(229,75,75),new Color(30,30,36)));

```



```
panelGeneral3.setPreferredSize(new Dimension(600,250));
panelGeneral3.add(marcador);
panelGeneral3.add(buttons);

panel3.add(panelGeneral3);
//pestanya 4 battle
reports-----
-----=====

panel4.add(new ReportScreen(con));

this.setVisible(true);
}

//metodo actualizador de dmg y defense en build
public void buildReset(Connection con ,String usuari) {
    for (int i=0;i<4;i++) {
        cst= ship.getInfoShips(con, i+1);
        try {
            cst.execute();
            int basedmg=cst.getInt(8);
            int basedef=cst.getInt(7);

buildTroopsdef[i].setText(Integer.toString(basedef+
(planeta.getTechnologyDefense()*PLUS_ARMOR_MISSILELAUNCHER_BY_TECHNOLOGY
)));

buildTroopsatk[i].setText(Integer.toString(basedmg+
(planeta.getTechnologyAtack()*PLUS_ATTACK_MISSILELAUNCHER_BY_TECHNOLOGY
)));

        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```



```
//buildTroopsatk[1].setText(getName());
}
public void inicializarTextos(Connection con,String usuari) {
    //Falta config
    textoP1[0]=new JTextField(""); // etiqueta reservada para el
formatado
    textoP1[1]=new JTextField("");
    textoP1[2]=new JTextField("0");
    textoP1[3]=new JTextField("0");
    textoP1[4]=new JTextField("1");
    etiquetaP1[0]=new JLabel(" ");
    etiquetaP1[1]=new JLabel("Descripcion: ");
    etiquetaP1[3]=new JLabel("Puntos: ");
    etiquetaP1[4]=new JLabel("Nivel: ");
    etiquetaP1[5]=new JLabel("Attack Tech: ");
    etiquetaP1[6]=new JLabel("Defense Tech: ");

    try {
        cst = usu.getInfoPlanetId(con,usuari);
        cst.execute();

        cst = plane.getInfoPlanet(con,cst.getInt(2));
        cst.execute();
        etiquetaP1[7]=new JLabel(cst.getString(2));}
    catch (SQLException e) {e.printStackTrace();}
    etiquetaP1[8] = new JLabel("Fleet");
    etiquetaP1[9] = new JLabel("Defenses");
    // iamgen,nombre,armor,dmg, cose metal, coste deuterio,
texfield, boton
    etiquetaBuild[0]=new JLabel(" "); // cabecera imagen
    etiquetaBuild[1]=new JLabel("Name");
    etiquetaBuild[4]=new JLabel("    Damage");
    etiquetaBuild[5]=new JLabel("Armor");
    etiquetaBuild[2]=new JLabel("Metal cost");
    etiquetaBuild[3]=new JLabel("Deuterium cost");
    etiquetaBuild[6]=new JLabel("    Quantity");
    etiquetaBuild[7]=new JLabel(" "); //cabecera boton build
```



```
// BUILD SHIPS
for (int i=0;i<5;i++) {
    labelsBuild[i]= new ArrayList<JLabel>();
}
for (int i=0;i<4;i++) {
    cst= ship.getInfoShips(con, i+1);
    try {
        cst.execute();
        labelsBuild[i].add(new JLabel(cst.getString(2)));//
nombre
        labelsBuild[i].add(new
JLabel(Integer.toString(cst.getInt(3))));//metal cost
        labelsBuild[i].add(new
JLabel(Integer.toString(cst.getInt(5))));//deuterium cost
        int basedmg=cst.getInt(8);
        int basedef=cst.getInt(7);

        labelsBuild[i].add(buildTroopsatk[i]=new
JLabel(Integer.toString(basedmg+
(planeta.getTechnologyAtack()*PLUS_ATTACK_MISSILELAUNCHER_BY_TECHNOLOGY
)))); // dmg
        labelsBuild[i].add(buildTroopsdef[i]= new
JLabel(Integer.toString(basedef+
(planeta.getTechnologyDefense()*PLUS_ARMOR_MISSILELAUNCHER_BY_TECHNOLOGY
)))); // armor
    } catch (SQLException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
}

//-----

// defensas:
for (int i=0;i<8;i++) {
    etiquetaBuild2[i]=new JLabel(etiquetaBuild[i].getText());
}
for (int i=0;i<3;i++) {
    labelsBuild2[i]= new ArrayList<JLabel>();
}
```



```
for (int i=0;i<3;i++) {
    cst= defe.getInfoDefense(con, i+1);
    try {
        cst.execute();
        labelsBuild2[i].add(new JLabel(cst.getString(2)));//
nombre
        labelsBuild2[i].add(new JLabel("
"+Integer.toString(cst.getInt(3))));//metal cost
        labelsBuild2[i].add(new
JLabel(Integer.toString(cst.getInt(5))));//deuterium cost
        int basedmg=cst.getInt(8);
        int basedef=cst.getInt(7);

        labelsBuild2[i].add(buildTroopsatk[i]=new
JLabel(Integer.toString(basedmg+
(planeta.getTechnologyAtack()*PLUS_ATTACK_MISSILELAUNCHER_BY_TECHNOLOGY
)))); // dmg
        labelsBuild2[i].add(buildTroopsdef[i]= new
JLabel(Integer.toString(basedef+
(planeta.getTechnologyDefense()*PLUS_ARMOR_MISSILELAUNCHER_BY_TECHNOLOGY
)))); // armor
    } catch (SQLException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
}
// for de inicializar
textfield-----prueba
    Image fondito =
Toolkit.getDefaultToolkit().getImage("space.jpg");
    for (int i=0;i<buildText.length;i++) {
        buildText[i]=new JTextField("0",3);
        textoformatado[i]=new JPanelConFondo(fondito);
        textoformatado[i].setLayout(new GridLayout(3,1));
        textoformatado[i].add(new JLabel(" "));
        textoformatado[i].add(buildText[i]);
        textoformatado[i].add(new JLabel(" "));
    }
}
```



```
for (int i=0;i<buildText2.length;i++) {
    buildText2[i]=new JTextField("0",3);
    textoformatado2[i]=new JPanelConFondo(fondito);
    textoformatado2[i].setLayout(new GridLayout(3,1));
    textoformatado2[i].add(new JLabel(" "));
    textoformatado2[i].add(buildText2[i]);
    textoformatado2[i].add(new JLabel(" "));
}
//botonP11=new JButton("REINICIAR");
botonP12=new JButton("CONFIRMAR");
deuterium=new JLabel("0");
metal=new JLabel("0");

metal.setForeground(Color.green);
deuterium.setForeground(Color.green);
//Abajo anyado botones con aspecto personalizado, para poder
cambiar el tamanyo de la imagen he tenido que pasarlo a imagen y luego
otra vez a icon
ImageIcon btn = new ImageIcon("btn.png");
Image trans = btn.getImage();
Image newimg = trans.getScaledInstance(65, 65,
java.awt.Image.SCALE_SMOOTH);
btn = new ImageIcon(newimg);
for (int i=0;i<4;i++) {
    btnBuild[i] = new JButton("",btn);
    btnBuild[i].setPreferredSize( new Dimension(65, 65));
    btnBuild[i].setBorderPainted(false);
    btnBuild[i].setContentAreaFilled(false);
    btnBuild[i].setFocusPainted(false);
    btnBuild[i].setOpaque(false);
}
for (int i=0;i<3;i++) {
    btnBuild2[i] = new JButton("",btn);
    btnBuild2[i].setPreferredSize( new Dimension(65, 65));
    btnBuild2[i].setBorderPainted(false);
    btnBuild2[i].setContentAreaFilled(false);
    btnBuild2[i].setFocusPainted(false);
    btnBuild2[i].setOpaque(false);
}
```





```
//Label de upgrade

etiquetaP3[0]=new JLabel("TECHNOLOGY ATTACK");

etiquetaP3[1]=new JLabel("TECHNOLOGY DEFENSE");

etiquetaP3[2]=new JLabel("COST");
etiquetaP3[3]=new JLabel("LEVEL");

etiquetaP3[4]= new JLabel("COST");
etiquetaP3[5]=new JLabel("LEVEL");
//abajo etiquetas que importan valores de la base de datos
try {
    etiquetaP1[2]=new JLabel(Integer.toString(cst.getInt(1)));
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
//cambiar toda label de la array a color verde
for (int i=1;i<etiquetaP1.length;i++) {
    etiquetaP1[i].setForeground(Color.GREEN);
}
for (int i=0;i<etiquetaBuild.length;i++) {
    etiquetaBuild[i].setForeground(Color.GREEN);
    etiquetaBuild2[i].setForeground(Color.GREEN);
}
}


public void guardado(Connection con,String usuario) {

    UpdateGame upda=new UpdateGame(con,usu.getIdUser(con,usuario ) ,
IDplanetita, planeta.getTechnologyAttack(),
planeta.getTechnologyDefense(), planeta.getMetal(),
planeta.getDeuterium());
    for (int w=0;w<4;w++) { //id nave

        for (int i=0;i<planeta.getTechnologyAttack()+1;i++) {
            for (int j=0;j<planeta.getTechnologyDefense()+1;j++) {
                int quantity=0;
```



```
for (int o =0;o<planeta.getArmy()[w].size();o++)
{
    MilitaryUnit shipAux=new LightHunter(j, i,
con);
    if (w==0) {
        shipAux=new LightHunter(j, i, con);
    }
    else if (w==1) {
        shipAux=new HeavyHunter(j, i, con);
    }
    else if (w==2) {
        shipAux=new BattleShip(j, i, con);
    }
    else if(w==3) {
        shipAux=new ArmoredShip(j, i, con);
    }
    if
(planeta.getArmy()[w].get(o).attack()==shipAux.attack() &&
planeta.getArmy()[w].get(o).getActualArmor()==shipAux.getActualArmor())
{
        quantity=quantity+1;
    }
}
upda.UpdateShips(con, IDplanetita, w+1, i, j,
quantity);
}
}
for (int w=0;w<3;w++) { //id def
    for (int i=0;i<planeta.getTechnologyAtack()+1;i++) {
        for (int j=0;j<planeta.getTechnologyDefense()+1;j++) {
            int quantity=0;
            for (int o
=0;o<planeta.getArmy()[w+4].size();o++) {
                MilitaryUnit defAux=new MissileLauncher(j,
i, con);
                if (w==0) {
                    defAux=new MissileLauncher(j, i,
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 91                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

con);

        }
        else if (w==1) {
            defAux=new IonCannon(j, i, con);
        }
        else if (w==2) {
            defAux=new PlasmaCannon(j, i, con);
        }
        if
(planeta.getArmy()[w+4].get(o).attack()==defAux.attack() &&
planeta.getArmy()[w+4].get(o).getActualArmor()==defAux.getActualArmor())
{
            quantity=quantity+1;
        }

    }
    upda.UpdateDefenses(con, IDplanetita, w+1, i, j,
quantity);
    }
}
}

}

//Fondo
class JPanelConFondo extends JPanel {

    private Image imagen;


    //...

    @Override
    public void paint(Graphics g) {

        g.drawImage(imagen, 0, 0, getWidth(), getHeight(),
            this);

        setOpaque(false);

```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 92                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

        super.paint(g);
    }
    public JPanelConFondo(Image imagenInicial) {
        if (imagenInicial != null) {
            imagen = imagenInicial;
        }

        //...
    }
}

```

## - Clase ReportScreen

Clase ReportScreen contiene el código para crear una interfaz gráfica donde se mostrará los pasos de la batalla.

```

import java.awt.Color;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import javax.swing.BoxLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.ScrollPaneConstants;

```



```
public class ReportScreen extends JPanel {

    Connection con;

    JPanel panel1, panel2, panel3;
    JLabel label_bat = new JLabel("Battle ID: ");
    JTextField id_bat;
    JTextArea terminal;
    JButton boton_resumen, boton_completo;

    int total_starting_units_planet, total_final_units_planet,
    total_starting_units_enemy, total_final_units_enemy,
    planet_metal_costs, planet_deuterium_costs, enemy_metal_costs,
    enemy_deuterium_costs, planet_metal_losses,
    planet_deuterium_losses, enemy_metal_losses,
    enemy_deuterium_losses, metal_wastings, deuterium_wastings, winner;

    String complete_rep;

    int[] starting_units_planet = new int[7];
    int[] final_units_planet = new int[7];


    int[] starting_units_enemy = new int[4];
    int[] final_units_enemy = new int[4];

    JScrollPane scroll;

    public ReportScreen(Connection con) {

        // guarda la connexio a la bbdd com a un atribut
        this.con = con;

        // panel principal que conte tots els elements i els altres
panels
        panel1 = new JPanel();
        panel1.setLayout(new BorderLayout(panel1, BorderLayout.Y_AXIS));
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 94                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

        // panel que conte la JLabel de id de batalla, un textfield
        per a insertar la id de batalla i un panel amb dos botons per a mostrar
        els reports
        panel2 = new JPanel();
        panel2.setLayout(new BorderLayout(panel2, BorderLayout.X_AXIS));

        // panel que conte els botons per a mostrar els reports
        panel3 = new JPanel();
        panel3.setLayout(new BorderLayout(panel3, BorderLayout.Y_AXIS));

        label_bat.setForeground(Color.WHITE);

        // textfield per a insertar la id de batalla
        id_bat = new JTextField();
        id_bat.setPreferredSize(new Dimension(1, 1));

        // textarea per a mostrar els reports
        terminal = new JTextArea();
        terminal.setEditable(false);
        terminal.setFont(new Font(Font.MONOSPACED, Font.PLAIN, 11));

        scroll = new JScrollPane(terminal);
        scroll.setPreferredSize(new Dimension(750, 500));

        this.setBackground(new Color(0, 0, 0, 0));
        panel1.setBackground(new Color(0, 0, 0, 0));
        panel2.setBackground(new Color(0, 0, 0, 0));
        panel3.setBackground(new Color(0, 0, 0, 0));
        this.setPreferredSize(new Dimension(800, 600));

        // botons per a mostrar els reports i els seus respectius
listeners
        boton_resumen = new JButton("Show Battle Summary");
        boton_resumen.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent arg0) {

```



```
try {  
    // primer es comprova que la ID sigui  
correcta  
    CallableStatement cst =  
con.prepareCall("{call check_battle (?, ?)}");  
    cst.setInt(1,  
Integer.parseInt(id_bat.getText()));  
    cst.registerOutParameter(2,  
java.sql.Types.INTEGER);  
    cst.execute();  
    System.out.println("comprobacion");  
    int existe = cst.getInt(2);  
    if (existe == 0) {  
        JOptionPane.showMessageDialog(null,  
"ERROR: battle ID does not exist");  
    } else {  
        System.out.println("a");  
        // si existeix la batalla es genera  
el report resumit  
setVariables(Integer.parseInt(id_bat.getText()));  
terminal.setText(getSummary(Integer.parseInt(id_bat.getText())));  
    }  
} catch (SQLException e) {  
    // TODO Auto-generated catch block  
    System.out.println("ERROR");  
    e.printStackTrace();  
} catch (NumberFormatException e) {  
    JOptionPane.showMessageDialog(null, "ERROR:  
battle ID must be a number");  
    }  
}  
  
});  
  
boton_completo = new JButton("Show Battle Development");  
boton_completo.addActionListener(new ActionListener() {
```



```
@Override
public void actionPerformed(ActionEvent e) {
    try {
        CallableStatement cst =
con.prepareStatement("{call check_battle (?, ?)}");
        cst.setInt(1,
Integer.parseInt(id_bat.getText()));
        cst.registerOutParameter(2,
java.sql.Types.INTEGER);

        cst.execute();
        if (cst.getInt(2) == 0) {
            JOptionPane.showMessageDialog(null,
"ERROR: battle ID does not exist");
        } else {

setVariables(Integer.parseInt(id_bat.getText()));

setCompleteReport(Integer.parseInt(id_bat.getText()));
            terminal.setText(complete_rep);
        }

    } catch (SQLException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    } catch (NumberFormatException e1) {
        JOptionPane.showMessageDialog(null, "ERROR:
battle ID must be a number");
    }
}

});

// s'aniden els panels i s'afageixen a la finestra
panel1.add(panel2);
panel1.add(scroll);
panel2.add(label_bat);
panel2.add(id_bat);
panel2.add(panel3);
panel3.add(boton_resumen);
```






```
panel13.add(boton_completo);

add(panel1);

}

private String getSummary(int id_battle) {
    String msg = ("\n" +
        "BATTLE ID: " + id_battle + "\n" +
        "BATTLE STATISTICS" + "\n\n" +
        "Planet Army" + repetirChar(20-"Planet
Army".length(), ' ') + "Starting Units"
        + repetirChar(15-"Starting Units".length(), ' ')
+ "Remaining Units" + repetirChar(20-"Remaining Units".length(),
' ') + "Enemy Army" + repetirChar(20-"Enemy
Army".length(), ' ') + "Starting Units"
        + repetirChar(15-"Starting Units".length(), ' ')
+ "Remaining Units" + "\n" +
        "Light Hunter" + repetirChar(20-"Light
Hunter".length(), ' ') + starting_units_planet[0] +
        repetirChar(15-String.valueOf(starting_units_planet[0]).length(), ' ')
+ final_units_planet[0] +
        repetirChar(20-String.valueOf(final_units_planet[0]).length(), ' ') +
        "Light Hunter" + repetirChar(20-"Light Hunter".length(), ' ') +
        starting_units_enemy[0] +
        repetirChar(15-String.valueOf(starting_units_enemy[0]).length(), ' ') +
        final_units_enemy[0] + "\n" +
        "Heavy Hunter" + repetirChar(20-"Heavy
Hunter".length(), ' ') + starting_units_planet[1] +
        repetirChar(15-String.valueOf(starting_units_planet[1]).length(), ' ')
+ final_units_planet[1] +
        repetirChar(20-String.valueOf(final_units_planet[1]).length(), ' ') +
        "Heavy Hunter" + repetirChar(20-"Heavy Hunter".length(), ' ') +
        starting_units_enemy[1] +
        repetirChar(15-String.valueOf(starting_units_enemy[1]).length(), ' ') +
        final_units_enemy[1] + "\n" +
        "Battle Ship" + repetirChar(20-"Battle
Ship".length(), ' ') + starting_units_planet[2] +
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 98                  |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

repetirChar(15-String.valueOf(starting_units_planet[2]).length(), ' ')
+ final_units_planet[2] +
repetirChar(20-String.valueOf(final_units_planet[2]).length(), ' ') +
"Battle Ship" + repetirChar(20-"Battle Ship".length(), ' ') +
starting_units_enemy[2] +
repetirChar(15-String.valueOf(starting_units_enemy[2]).length(), ' ') +
final_units_enemy[2] + "\n" +
        "Armored Ship" + repetirChar(20-"Armored
Ship".length(), ' ') + starting_units_planet[3] +
repetirChar(15-String.valueOf(starting_units_planet[3]).length(), ' ')
+ final_units_planet[3] +
repetirChar(20-String.valueOf(final_units_planet[3]).length(), ' ') +
"Armored Ship" + repetirChar(20-"Armored Ship".length(), ' ') +
starting_units_enemy[3] +
repetirChar(15-String.valueOf(starting_units_enemy[3]).length(), ' ') +
final_units_enemy[3] + "\n" +
        "Missile Launcher" + repetirChar(20-"Missile
Launcher".length(), ' ') + starting_units_planet[4] +
repetirChar(15-String.valueOf(starting_units_planet[4]).length(), ' ')
+ final_units_planet[4] + "\n" +
        "Ion Cannon" + repetirChar(20-"Ion
Cannon".length(), ' ') + starting_units_planet[5] +
repetirChar(15-String.valueOf(starting_units_planet[5]).length(), ' ')
+ final_units_planet[5] + "\n" +
        "Plasma Cannon" + repetirChar(20-"Plasma
Cannon".length(), ' ') + starting_units_planet[6] +
repetirChar(15-String.valueOf(starting_units_planet[6]).length(), ' ')
+ final_units_planet[6] + "\n" +
        "TOTAL" +
repetirChar(20-"TOTAL".length(), ' ') +
total_starting_units_planet +
repetirChar(15-String.valueOf(total_starting_units_planet).length(), '
') + total_final_units_planet +
repetirChar(20-String.valueOf(total_final_units_planet).length(), ' ') +
"TOTAL" + repetirChar(20-"TOTAL".length(), ' ') +
total_starting_units_enemy +
repetirChar(15-String.valueOf(total_starting_units_enemy).length(), ' ')
+ total_final_units_enemy + "\n" +

```

|                       |                      |                            |
|-----------------------|----------------------|----------------------------|
| Elaborat per: Grupo 2 | Revisat per: Grupo 2 | Autoritzat per: Josep Diaz |
|-----------------------|----------------------|----------------------------|



```
        repetirChar(106, '*') + "\n" +
        "Planet Army's Costs" +
repetirChar(55-"Planet Army's Costs".length(), ' ') +
"Enemy Army's Costs" + "\n" +
        "Metal: " + planet_metal_costs +
repetirChar(55-("Metal: " + planet_metal_costs).length(), ' ') +
"Metal: " + enemy_metal_costs + "\n" +
        "Deuterium: " + planet_deuterium_costs +
repetirChar(55-("Deuterium: " + planet_deuterium_costs).length(), ' ') +
"Deuterium: " + enemy_deuterium_costs + "\n" +

        repetirChar(106, '*') + "\n" +
        "Planet Army's Losses"
+ repetirChar(55-"Planet Army's Losses".length(), ' ')
+ "Enemy Army's Losses" + "\n" +
        "Metal: " + planet_metal_losses
+ repetirChar(55-("Metal: " + planet_metal_losses).length(), ' ')
+ "Metal: " + enemy_metal_losses + "\n" +
        "Deuterium: " + planet_deuterium_losses
+ repetirChar(55-("Deuterium: " + planet_deuterium_losses).length(), ' ')
+ "Deuterium: " + enemy_deuterium_losses + "\n" +
        "Weighted: " + (planet_metal_losses +
planet_deuterium_losses*5) + repetirChar(55-("Weighted: " +
(planet_metal_losses + planet_deuterium_losses*5)).length(), ' ') +
"Weighted: " + (enemy_metal_losses + enemy_deuterium_losses*5)
+ "\n" +

        repetirChar(106, '*') + "\n" +
        "Wastings Generated" + "\n" +
        "Metal: " + metal_wastings + "\n" +
        "Deuterium: " + deuterium_wastings + "\n\n"
    );
    if (winner == 0) {
        msg += "Battle won by Planet Army";
    } else {
        msg += "Battle won by Enemy Army";
    }
    return msg;
```



```
}

private String repetirChar(int repeticiones, char caracter) {
    String msg = new String(new
char[repeticiones]).replace('\0', caracter);
    return msg;
}

private void setUnits(int id_battle, int index) {
    try {
        CallableStatement cst1 = con.prepareCall("{call
get_ship_battle (?, ?, ?, ?, ?)}");
        CallableStatement cst2 = con.prepareCall("{call
get_defense_battle (?, ?, ?, ?, ?)}");

        cst1.setInt(1, id_battle);

        cst1.registerOutParameter(4, java.sql.Types.INTEGER);
        cst1.registerOutParameter(5, java.sql.Types.INTEGER);

        cst1.setInt(2, index+1);
        cst1.setInt(3, 0);

        cst1.execute();

        starting_units_planet[index] = cst1.getInt(4);
        final_units_planet[index] = cst1.getInt(5);

        cst1.setInt(3, 1);

        cst1.execute();

        starting_units_enemy[index] = cst1.getInt(4);
        final_units_enemy[index] = cst1.getInt(5);

        if (index < 3) {
            cst2.setInt(1, id_battle);
            cst2.setInt(2, index+1);
            cst2.setInt(3, 0);
```



```
        cst2.registerOutParameter(4,  
java.sql.Types.INTEGER);  
        cst2.registerOutParameter(5,  
java.sql.Types.INTEGER);  
        cst2.execute();  
        starting_units_planet[index+3] = cst2.getInt(4);  
        final_units_planet[index+3] = cst2.getInt(5);  
    }  
  
    } catch (SQLException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
}  
  
private void setVariables(int id_batalla) {  
    try {  
        CallableStatement cst = con.prepareCall("{call  
SHOW_REPORT (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)}");  
        cst.setInt(1, id_batalla);  
  
        for (int i = 1; i < 18; i++) {  
            cst.registerOutParameter(i,  
java.sql.Types.INTEGER);  
        }  
        System.out.println("parametros variable");  
  
        cst.execute();  
  
        total_starting_units_planet = cst.getInt(3);  
        total_final_units_planet = cst.getInt(4);  
        total_starting_units_enemy = cst.getInt(5);  
        total_final_units_enemy = cst.getInt(6);  
        planet_metal_costs = cst.getInt(7);  
        planet_deuterium_costs = cst.getInt(8);  
        enemy_metal_costs = cst.getInt(9);  
        enemy_deuterium_costs = cst.getInt(10);  
        planet_metal_losses = cst.getInt(11);
```



```
planet_deuterium_losses = cst.getInt(12);
enemy_metal_losses = cst.getInt(13);
enemy_deuterium_losses = cst.getInt(14);
metal_wastings = cst.getInt(15);
deuterium_wastings = cst.getInt(16);
winner = cst.getInt(17);

System.out.println("iniciando units");
for (int i = 0; i < 4; i++) {
    setUnits(id_batalla, i);
}
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

}

private void setCompleteReport(int id_batalla) {
    try {
        CallableStatement cst = con.prepareCall("{call
GET_STEP_AMOUNT (?, ?)}");


        cst.setInt(1, id_batalla);

        cst.registerOutParameter(2, java.sql.Types.INTEGER);

        cst.execute();
        System.out.println("a");
        String msg = "";
        int num_steps = cst.getInt(2);
        cst.close();
        for (int i = 1; i <= num_steps; i++) {

            cst = con.prepareCall("{call GETBATTLESTEP (?, ?,
?)}");

            cst.setInt(1, id_batalla);
            cst.setInt(2, i);
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 103                 |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

                                cst.registerOutParameter(3,
java.sql.Types.VARCHAR);
                                cst.execute();

                                msg += "\n" + cst.getString(3);
                                cst.close();
                                }

                                complete_rep = msg;
                                } catch (SQLException e) {
                                    // TODO Auto-generated catch block
                                    e.printStackTrace();
                                }
                                }
                                }

```

## - Clase BattleScreen

Clase BattleScreen contiene el código para crear una interfaz gráfica que contendrá ReportScreen y Battle.

```

import java.awt.Color;
import java.awt.Component;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.GridLayout;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.SQLException;
import java.util.ArrayList;

import javax.swing.BoxLayout;
import javax.swing.JButton;

```



```
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextArea;
import javax.swing.border.Border;
import javax.swing.plaf.BorderUIResource;

public class BattleScreen extends JFrame {

    private Battle battle;

    private JPanel marcador = new JPanel();
    private JPanel unitgrid_player = new JPanel();
    private JPanel unitgrid_enemy = new JPanel();

    private JLabel tit_jugador = new JLabel("PLAYER");
    private JLabel tit_enemy = new JLabel("ENEMY");

    private JLabel lightHunter_player = new JLabel();
    private JLabel lightHunter_enemy = new JLabel();

    private JLabel heavyHunter_player = new JLabel();
    private JLabel heavyHunter_enemy = new JLabel();

    private JLabel battleShip_player = new JLabel();
    private JLabel battleShip_enemy = new JLabel();

    private JLabel armoredShip_player = new JLabel();
    private JLabel armoredShip_enemy = new JLabel();

    private JLabel missileLauncher_player = new JLabel();

    private JLabel ionCannon_player = new JLabel();

    private JLabel plasmaCannon_player = new JLabel();

    private JTextArea desc = new JTextArea();

    private JButton boton = new JButton("CONTINUE");
```





```
private ActionListener continuar, acabar;

public BattleScreen(Planet planet, ArrayList<MilitaryUnit>[]
enemyArmy, Connection con, int idUser,String username) {

    battle = new Battle(planet.getArmy(), enemyArmy);
    marcador.setBackground(Color.BLACK);
    unitgrid_player.setBackground(Color.BLACK);
    unitgrid_enemy.setBackground(Color.BLACK);
    this.setBackground(Color.BLACK);
    desc.setBackground(Color.BLACK);
    desc.setForeground(Color.GREEN);
    desc.setFont(new Font(Font.MONOSPACED, Font.PLAIN, 11));
    tit_jugador.setFont(new Font(Font.MONOSPACED, Font.PLAIN,
11));
    tit_enemy.setFont(new Font(Font.MONOSPACED, Font.PLAIN,
11));
    lightHunter_player.setFont(new Font(Font.MONOSPACED,
Font.PLAIN, 11));
    lightHunter_enemy.setFont(new Font(Font.MONOSPACED,
Font.PLAIN, 11));
    heavyHunter_player.setFont(new Font(Font.MONOSPACED,
Font.PLAIN, 11));
    heavyHunter_enemy.setFont(new Font(Font.MONOSPACED,
Font.PLAIN, 11));
    battleShip_player.setFont(new Font(Font.MONOSPACED,
Font.PLAIN, 11));
    battleShip_enemy.setFont(new Font(Font.MONOSPACED,
Font.PLAIN, 11));
    armoredShip_player.setFont(new Font(Font.MONOSPACED,
Font.PLAIN, 11));
    armoredShip_enemy.setFont(new Font(Font.MONOSPACED,
Font.PLAIN, 11));
    missileLauncher_player.setFont(new Font(Font.MONOSPACED,
```



```
Font.PLAIN, 11));
    ionCannon_player.setFont(new Font(Font.MONOSPACED,
Font.PLAIN, 11));
    plasmaCannon_player.setFont(new Font(Font.MONOSPACED,
Font.PLAIN, 11));

    tit_jugador.setForeground(Color.GREEN);
    tit_enemy.setForeground(Color.GREEN);
    lightHunter_player.setForeground(Color.GREEN);
    lightHunter_enemy.setForeground(Color.GREEN);
    heavyHunter_player.setForeground(Color.GREEN);
    heavyHunter_enemy.setForeground(Color.GREEN);
    battleShip_player.setForeground(Color.GREEN);
    battleShip_enemy.setForeground(Color.GREEN);
    armoredShip_player.setForeground(Color.GREEN);
    armoredShip_enemy.setForeground(Color.GREEN);
    missileLauncher_player.setForeground(Color.GREEN);
    ionCannon_player.setForeground(Color.GREEN);
    plasmaCannon_player.setForeground(Color.GREEN);

    updateLabels();

    unitgrid_player.add(lightHunter_player);
    unitgrid_player.add(heavyHunter_player);
    unitgrid_player.add(battleShip_player);
    unitgrid_player.add(armoredShip_player);
    unitgrid_player.add(missileLauncher_player);
    unitgrid_player.add(ionCannon_player);
    unitgrid_player.add(plasmaCannon_player);

    unitgrid_enemy.add(lightHunter_enemy);
    unitgrid_enemy.add(heavyHunter_enemy);
    unitgrid_enemy.add(battleShip_enemy);
    unitgrid_enemy.add(armoredShip_enemy);

    desc.setEditable(false);
    desc.setPreferredSize(new Dimension(500, 500));

    String texto = battle.startBattle();
```



```
int winner = battle.getWinner();
if (winner != 0) {
    boton.setText("END BATTLE");
    if (winner == 1) {
        texto += "\n\n" + "Planet Army won the battle!";
        planet.setMetal(planet.getMetal() +
battle.getWasteMetalDeuterium()[0]);
        planet.setDeuterium(planet.getDeuterium() +
battle.getWasteMetalDeuterium()[1]);
    } else if (winner == 2) {
        texto += "\n\n" + "Enemy Army won the battle!";
    }
    planet.setArmy(battle.getPlanetArmy());
}

desc.setText(texto);

this.setLayout(new BorderLayout(this.getContentPane(),
BoxLayout.Y_AXIS));
marcador.setLayout(new GridLayout(2, 3));
unitgrid_player.setLayout(new GridLayout(4, 2));
unitgrid_enemy.setLayout(new GridLayout(4, 2));

marcador.add(tit_jugador);
marcador.add(tit_enemy);
marcador.add(unitgrid_player);
marcador.add(unitgrid_enemy);

acabar = new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        battle.saveBattleData(con, idUser);
        dispose();
        //Falta guardado
        new VentanaInicial(con, username);
    }
};

continuar = new ActionListener() {
```



```
public void actionPerformed(ActionEvent arg0) {
    updateLabels();
    String texto = battle.continueBattle();
    int winner = battle.getWinner();
    if (winner != 0) {
        boton.setText("END BATTLE");
        boton.addActionListener(acabar);
        boton.removeActionListener(continuar);
        if (winner == 1) {
            texto += "\n\n" + "Planet Army won
the battle!";
            planet.setMetal(planet.getMetal() +
battle.getWasteMetalDeuterium()[0]);

planet.setDeuterium(planet.getDeuterium() +
battle.getWasteMetalDeuterium()[1]);
        } else if (winner == 2) {
            texto += "\n\n" + "Enemy Army won the
battle!";
        }
    }

    desc.setText(texto);
}

};


boton.addActionListener(continuar);

add(marcador);
add(desc);
add(boton);

setSize(900,700);
setVisible(true);
}
```



```
private void updateLabels() {  
  
    // Actualitza les JLabels de les dades de cada exercit  
    int[] units_planet = battle.getActualNumberUnitsPlanet();  
    int[] units_enemy = battle.getActualNumberUnitsEnemy();  
  
    lightHunter_player.setText("Light Hunter - " +  
units_planet[0]);  
    lightHunter_enemy.setText("Light Hunter - " +  
units_enemy[0]);  
  
    heavyHunter_player.setText("Heavy Hunter - " +  
units_planet[1]);  
    heavyHunter_enemy.setText("Heavy Hunter - " +  
units_enemy[1]);  
  
    battleShip_player.setText("Battle Ship - " +  
units_planet[2]);  
    battleShip_enemy.setText("Battle Ship - " + units_enemy[2]);  
  
    armoredShip_player.setText("Armored Ship - " +  
units_planet[3]);  
    armoredShip_enemy.setText("Armored Ship - " +  
units_enemy[3]);  
  
    missileLauncher_player.setText("Missile Launcher - " +  
units_planet[4]);  
  
    ionCannon_player.setText("Ion Cannon - " + units_planet[5]);  
  
    plasmaCannon_player.setText("Plasma Cannon - " +  
units_planet[6]);  
  
    }  
}
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 110                 |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

## - Clase Battle

Clase Battle contiene el código para crear una interfaz gráfica donde se mostrará la batalla.

```
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.SQLException;
import java.util.ArrayList;

public class Battle implements Variables {
    private ArrayList<MilitaryUnit>[] planetArmy = new ArrayList[7];
    private ArrayList<MilitaryUnit>[] enemyArmy = new ArrayList[4];
    private ArrayList[][] armies = new ArrayList[2][7];
    private ArrayList<String> battleDevelopment = new
ArrayList<String>();
    int[][] initialCostFleet = new int[2][2];
    int initialNumberUnitsPlanet, initialNumberUnitsEnemy;
    int[] wasteMetalDeuterium = new int[2];
    int[][] resourcesLosses = new int[2][3];
    int[][] initialArmies = new int[2][7];
    int[] actualNumberUnitsPlanet = new int[7];
    int[] actualNumberUnitsEnemy = new int[7];

    int turn;

    public Battle(ArrayList<MilitaryUnit>[] planetArmy,
ArrayList<MilitaryUnit>[] enemyArmy) {
        initVariables(planetArmy, enemyArmy);
    }

    // Funcio que retorna la quantitat total d'unitats d'un mateix
exercit
    public int getTotalUnitsFleet(ArrayList<MilitaryUnit>[] fleet) {
        int total = 0;
        for (int i = 0; i < fleet.length; i++) {
            total += fleet[i].size();
        }
        return total;
    }
}
```



```
}
private ArrayList<MilitaryUnit> initSquad(ArrayList<MilitaryUnit>
squad) {
    ArrayList<MilitaryUnit> returnable = new
ArrayList<MilitaryUnit>();
    for (int i = 0; i < squad.size(); i++) {
        returnable.add(squad.get(i));
    }
    return returnable;
}

private void initVariables(ArrayList<MilitaryUnit>[]
planetInitialArmy, ArrayList<MilitaryUnit>[] enemyInitialArmy) {
    initialNumberUnitsPlanet =
getTotalUnitsFleet(planetInitialArmy);
    initialNumberUnitsEnemy =
getTotalUnitsFleet(enemyInitialArmy);
    int total_cost_metal_planet = 0;
    int total_cost_deuterium_planet = 0;
    int total_cost_metal_enemy = 0;
    int total_cost_deuterium_enemy = 0;

    // Recorre les arrays d'exercits per a comptar el nombre
total de tropes (objectes) que hi ha a cada exercit
    for (int i = 0; i < planetInitialArmy.length; i++) {
        initialArmies [0][i] = planetInitialArmy[i].size();
        actualNumberUnitsPlanet[i] =
planetInitialArmy[i].size();
        planetArmy[i] = initSquad(planetInitialArmy[i]);
        armies[0][i] = planetArmy[i];
        for (int j = 0; j < planetInitialArmy[i].size(); j++)
        {
            total_cost_metal_planet += METAL_COST_UNITS[i];
            total_cost_deuterium_planet +=
DEUTERIUM_COST_UNITS[i];
        }
    }
}
```



```
        for (int i = 0; i < enemyInitialArmy.length; i++) {
            initialArmies[1][i] = enemyInitialArmy[i].size();
            actualNumberUnitsEnemy[i] =
enemyInitialArmy[i].size();
            enemyArmy[i] = initSquad(enemyInitialArmy[i]);
            armies[1][i] = enemyArmy[i];
            for (int j = 0; j < enemyInitialArmy[i].size(); j++) {
                total_cost_metal_enemy += METAL_COST_UNITS[i];
                total_cost_deuterium_enemy +=
DEUTERIUM_COST_UNITS[i];
            }
        }

        initialCostFleet[0][0] = total_cost_metal_planet;
        initialCostFleet[0][1] = total_cost_deuterium_planet;
        initialCostFleet[1][0] = total_cost_metal_enemy;
        initialCostFleet[1][1] = total_cost_deuterium_enemy;

        getStartingTurn();
    }

    private void getStartingTurn() {
        // Retorna un numero enter aleatori entre 0 i 1
        // Si retorna 0 es el torn del jugador, si retorna 1 es el
torn de l'enemic
        turn = (int) (Math.random());
    }

    private MilitaryUnit getAttacker() {
        // S'escull un nombre aleatori entre 0 i 100 perquè
        // els valors totals de les Arrays de probabilitat, tant les
de jugador com les d'enemic,
        // equivalen a 100
        int attacking_squad = (int) (Math.random()*100);
        if (turn == 0) {
            for (int i = 0; i < planetArmy.length; i++) {
                System.out.println("inicio: " + i);
                // Es resta la probabilitat d'atacar de cada grup
al nombre escollit aleatoriament
```





```
        attacking_squad -= CHANCE_ATTACK_PLANET_UNITS[i];
        // Si el nombre resultant a la resta es igual o
menor a 0 i n'hi al menys una tropa en el grup, s'escull una tropa
aleatoria

        // entre les de l'ArrayList del grup
        if (attacking_squad <= 0 && planetArmy[i].size()
> 0) {

            return planetArmy[i].get((int)
(Math.random()*planetArmy[i].size()));

        } else if (attacking_squad <= 0 && i ==
planetArmy.length-1) {

            i = -1;
        }
        System.out.println("final: " + i);
    }
} else {
    for (int i = 0; i < enemyArmy.length; i++) {
        // Es resta la probabilitat d'atacar de cada grup
al nombre escollit aleatoriament
        attacking_squad -= CHANCE_ATTACK_ENEMY_UNITS[i];
        // Si el nombre resultant a la resta es igual o
menor a 0, s'escull una tropa aleatoria
        // entre les de l'ArrayList del grup
        if (attacking_squad <= 0 && enemyArmy[i].size() >
0) {

            return enemyArmy[i].get((int)
(Math.random()*enemyArmy[i].size()));
        } else if (attacking_squad <= 0 && i ==
enemyArmy.length-1) {

            i = -1;
        }
    }
}
// retorna un valor nul per defecte
return null;
}

private MilitaryUnit getDefender() {
```



```
        if (turn == 0) {
            // Obte el nombre total d'unitats en l'exercit per a
            calcular les probabilitats
            int totalChances = getTotalUnitsFleet(enemyArmy);

            // S'escull un nombre aleatori entre 0 i el nombre
            total d'unitats
            int defending_squad = (int)
            (Math.random()*totalChances);
            System.out.println(enemyArmy.length);
            for (int i = 0; i < enemyArmy.length; i++) {
                // Es resta la quantitat de cada grup al nombre
                escollit aleatoriament
                defending_squad -= enemyArmy[i].size();
                if (defending_squad <= 0 && enemyArmy[i].size() >
0) {
                    return enemyArmy[i].get((int)
                    (Math.random()*enemyArmy[i].size()));
                } else if (defending_squad <= 0 && i ==
                    enemyArmy.length-1) {
                        i = -1;
                    }
                }
            } else {
                // Obte el nombre total d'unitats en l'exercit per a
                calcular les probabilitats
                int totalChances = getTotalUnitsFleet(planetArmy);

                // S'escull un nombre aleatori entre 0 i el nombre
                total d'unitats
                int defending_squad = (int)
                (Math.random()*totalChances);

                for (int i = 0; i < planetArmy.length; i++) {
                    // Es resta la quantitat de cada grup al nombre
                    escollit aleatoriament
                    defending_squad -= planetArmy[i].size();
                    if (defending_squad <= 0 && planetArmy[i].size()
```



```
> 0) {  
    return planetArmy[i].get((int)  
(Math.random()*planetArmy[i].size()));  
} else if (defending_squad <= 0 && i ==  
planetArmy.length-1) {  
    i = -1;  
}  
}  
}  
return null;  
}  
  
private String attack(MilitaryUnit attacker, MilitaryUnit  
defender) {  
    String message = attacker.getClass().getSimpleName() + "  
attacks " + defender.getClass().getSimpleName();  
    int dmg = attacker.attack();  
    message += "\n" + attacker.getClass().getSimpleName() + "  
deals " + dmg + " damage";  
    defender.takeDamage(dmg);  
    message += "\n" + defender.getClass().getSimpleName() + "  
has " + defender.getActualArmor() + " remaining armor";  
  
    if (defender.getActualArmor() <= 0) {  
        int chance_waste = (int) (Math.random()*100);  
        int type;  
        switch (defender.getClass().getSimpleName()) {  
            case "LightHunter":  
                type = 0;  
                break;  
            case "HeavyHunter":  
                type = 1;  
                break;  
            case "BattleShip":  
                type = 2;  
                break;  
            case "ArmoredShip":  
                type = 3;  
                break;  
        }  
    }  
}
```



```
        case "MissileLauncher":
            type = 4;
            break;
        case "IonCannon":
            type = 5;
            break;
        default:
            type = 6;
            break;
    }

    message += "\n" + loseShip(defender, type);
    if (chance_waste <= CHANCE_GENERATNG_WASTE_LIGHTHUNTER
    && type == 0) {
        message += "\n" + generateWastings(type);
    } else if (chance_waste <=
    CHANCE_GENERATNG_WASTE_HEAVYHUNTER && type == 1) {
        message += "\n" + generateWastings(type);
    } else if (chance_waste <=
    CHANCE_GENERATNG_WASTE_BATTLESHIP && type == 2) {
        message += "\n" + generateWastings(type);
    } else if (chance_waste <=
    CHANCE_GENERATNG_WASTE_ARMOREDSHIP && type == 3) {
        message += "\n" + generateWastings(type);
    } else if (chance_waste <=
    CHANCE_GENERATNG_WASTE_MISSILELAUNCHER && type == 4) {
        message += "\n" + generateWastings(type);
    } else if (chance_waste <=
    CHANCE_GENERATNG_WASTE_IONCANNON && type == 5) {
        message += "\n" + generateWastings(type);
    } else if (chance_waste <=
    CHANCE_GENERATNG_WASTE_PLASMACANNON && type == 6) {
        message += "\n" + generateWastings(type);
    }

    }
    return message;
}
```



```
public int getWinner() {
    if (getTotalUnitsFleet(planetArmy) <
initialNumberUnitsPlanet*0.2 || getTotalUnitsFleet(enemyArmy) <
initialNumberUnitsPlanet*0.2) {
        if (resourcesLosses[0][2] < resourcesLosses[1][2]) {
            System.out.println("winner: 1");
            return 1;
        }
        System.out.println("winner: 2");
        return 2;
    }
    System.out.println("no winner");
    return 0;
}

public String generateWastings(int type) {
    wasteMetalDeuterium[0] =
METAL_COST_UNITS[type]*PERCENTATGE_WASTE;
    wasteMetalDeuterium[1] =
DEUTERIUM_COST_UNITS[type]*PERCENTATGE_WASTE;
    return METAL_COST_UNITS[type]*PERCENTATGE_WASTE + " Metal
and " + DEUTERIUM_COST_UNITS[type]*PERCENTATGE_WASTE + " generated as
Wastings";
}

public String loseShip(MilitaryUnit ship, int type) {
    resourcesLosses[turn][0] += METAL_COST_UNITS[type];
    resourcesLosses[turn][1] += DEUTERIUM_COST_UNITS[type];
    resourcesLosses[turn][2] += METAL_COST_UNITS[type] +
5*DEUTERIUM_COST_UNITS[type];

    if (turn == 0) {
        enemyArmy[type].remove(ship);
        System.out.println(actualNumberUnitsEnemy);
        actualNumberUnitsEnemy[type] -= 1;
        System.out.println(actualNumberUnitsEnemy);
    } else {
        planetArmy[type].remove(ship);
        actualNumberUnitsPlanet[type] -= 1;
    }
}
```



```
    }

    return ship.getClass().getSimpleName() + " has been
destroyed";
}

public String startBattle() {
    String msg = "\n\n" + "***** START
BATTLE *****" + "\n";
    if (turn == 0) {
        msg += "Planet attacks: " + "\n";
    } else {
        msg += "Enemy Fleet attacks: " + "\n";
    }
    MilitaryUnit attacker = getAttacker();
    MilitaryUnit defender = getDefender();

    msg += attack(attacker, defender);
    battleDevelopment.add(msg);

    if (turn == 0) {
        turn = 1;
    } else {
        turn = 0;
    }

    return msg;
}

public String continueBattle() {
    String msg = "\n\n" + "***** CHANGE
ATTACKER *****" + "\n";
    if (turn == 0) {
        msg += "Planet attacks: " + "\n";
    } else {
        msg += "Enemy Fleet attacks: " + "\n";
    }
    MilitaryUnit attacker = getAttacker();
    MilitaryUnit defender = getDefender();
```



```
msg += attack(attacker, defender);
battleDevelopment.add(msg);

if (turn == 0) {
    turn = 1;
} else {
    turn = 0;
}

return msg;
}

public void saveBattleData(Connection con, int idUser) {
    try {
        System.out.println(idUser);
        CallableStatement get_id = con.prepareCall("{call
get_battleid (?)}");
        get_id.registerOutParameter(1,
java.sql.Types.VARCHAR);
        get_id.execute();
        int id = get_id.getInt(1);

        CallableStatement add_battle = con.prepareCall("{call
add_battle (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)}");
        add_battle.setInt(1, id);
        add_battle.setInt(2, idUser);
        add_battle.setInt(3, initialNumberUnitsPlanet);
        add_battle.setInt(4, getTotalUnitsFleet(planetArmy));
        add_battle.setInt(5, initialNumberUnitsEnemy);
        add_battle.setInt(6, getTotalUnitsFleet(enemyArmy));
        add_battle.setInt(7, initialCostFleet[0][0]);
        add_battle.setInt(8, initialCostFleet[0][1]);
        add_battle.setInt(9, initialCostFleet[1][0]);
        add_battle.setInt(10, initialCostFleet[1][1]);
        add_battle.setInt(11, resourcesLosses[0][0]);
        add_battle.setInt(12, resourcesLosses[0][1]);
        add_battle.setInt(13, resourcesLosses[1][0]);
        add_battle.setInt(14, resourcesLosses[1][1]);
```



```
add_battle.setInt(15, wasteMetalDeuterium[0]);
add_battle.setInt(16, wasteMetalDeuterium[1]);
add_battle.setInt(17, getWinner());
add_battle.execute();

CallableStatement add_ship = con.prepareCall("{call
add_ship_battle (?, ?, ?, ?, ?)}");
CallableStatement add_defenses =
con.prepareCall("{call add_defense_battle (?, ?, ?, ?, ?)}");
add_ship.setInt(1, id);
add_defenses.setInt(1, id);

for (int i = 0; i < enemyArmy.length; i++) {

    add_ship.setInt(2, i+1);
    add_ship.setInt(3, initialArmies[0][i]);
    add_ship.setInt(4, actualNumberUnitsPlanet[i]);
    add_ship.setInt(5, 0);

    add_ship.execute();

    add_ship.setInt(3, initialArmies[1][i]);
    add_ship.setInt(4, actualNumberUnitsEnemy[i]);
    add_ship.setInt(5, 1);

    add_ship.execute();
}

for (int i = 4; i < planetArmy.length; i++) {

    add_defenses.setInt(2, i-3);
    add_defenses.setInt(3, initialArmies[0][i]);
    add_defenses.setInt(4,
actualNumberUnitsPlanet[i]);
    add_defenses.setInt(5, 0);
    add_defenses.execute();
}
```





```
CallableStatement add_step = con.prepareCall("{call  
add_battle_step (?, ?, ?)}");  
    for (int i = 0; i < battleDevelopment.size(); i++) {  
        add_step.setInt(1, id);  
        add_step.setInt(2, i+1);  
        add_step.setString(3, battleDevelopment.get(i));  
        add_step.execute();  
    }  
} catch (SQLException e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}  
}  
  
public int getTurn() {  
    return turn;  
}  
  
public void setTurn(int turn) {  
    this.turn = turn;  
}  
  
public ArrayList<MilitaryUnit>[] getPlanetArmy() {  
    return planetArmy;  
}  
  
public void setPlanetArmy(ArrayList<MilitaryUnit>[] planetArmy) {  
    this.planetArmy = planetArmy;  
}  
  
public ArrayList<MilitaryUnit>[] getEnemyArmy() {  
    return enemyArmy;  
}  
  
public void setEnemyArmy(ArrayList<MilitaryUnit>[] enemyArmy) {  
    this.enemyArmy = enemyArmy;  
}
```



```
public ArrayList[][] getArmies() {  
    return armies;  
}  
  
public void setArmies(ArrayList[][] armies) {  
    this.armies = armies;  
}  
  
public int[][] getInitialCostFleet() {  
    return initialCostFleet;  
}  
  
public void setInitialCostFleet(int[][] initialCostFleet) {  
    this.initialCostFleet = initialCostFleet;  
}  
  
public int getInitialNumberUnitsPlanet() {  
    return initialNumberUnitsPlanet;  
}  
  
public void setInitialNumberUnitsPlanet(int  
initialNumberUnitsPlanet) {  
    this.initialNumberUnitsPlanet = initialNumberUnitsPlanet;  
}  
  
public int getInitialNumberUnitsEnemy() {  
    return initialNumberUnitsEnemy;  
}  
  
public void setInitialNumberUnitsEnemy(int  
initialNumberUnitsEnemy) {  
    this.initialNumberUnitsEnemy = initialNumberUnitsEnemy;  
}  
  
public int[] getWasteMetalDeuterium() {  
    return wasteMetalDeuterium;  
}  
  
public void setWasteMetalDeuterium(int[] wasteMetalDeuterium) {
```



```
        this.wasteMetalDeuterium = wasteMetalDeuterium;
    }

    public int[][] getResourcesLosses() {
        return resourcesLosses;
    }

    public void setResourcesLosses(int[][] resourcesLosses) {
        this.resourcesLosses = resourcesLosses;
    }

    public int[][] getInitialArmies() {
        return initialArmies;
    }


    public void setInitialArmies(int[][] initialArmies) {
        this.initialArmies = initialArmies;
    }

    public int[] getActualNumberUnitsPlanet() {
        return actualNumberUnitsPlanet;
    }

    public void setActualNumberUnitsPlanet(int[]
actualNumberUnitsPlanet) {
        this.actualNumberUnitsPlanet = actualNumberUnitsPlanet;
    }

    public int[] getActualNumberUnitsEnemy() {
        return actualNumberUnitsEnemy;
    }

    public void setActualNumberUnitsEnemy(int[]
actualNumberUnitsEnemy) {
        this.actualNumberUnitsEnemy = actualNumberUnitsEnemy;
    }
}
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 124                 |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

## - Clase BattleShip

Clase BattleShip extiende de Ship e implementa las interfaces Variables y MilitaryUnit donde nos conectamos a la base de datos y le pasamos la información de Ships.

```
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.SQLException;

public class BattleShip extends Ship implements Variables, MilitaryUnit {
    CallableStatement cst;

    public BattleShip(int tecnoDefense, int tecnoAtack, Connection con) {
        InfoShips ship = new InfoShips();
        cst = ship.getInfoShips(con, 3);
        try {
            cst.execute();
            super.setInitialArmor(cst.getInt(6));
            super.setBaseDamage(cst.getInt(8));
            this.setArmor(this.getInitialArmor() +
                (tecnoDefense * PLUS_ARMOR_LIGTHHUNTER_BY_TECHNOLOGY) * this.getInitialArmor() / 100);

            this.setBaseDamage(this.getBaseDamage() + (tecnoAtack * PLUS_ATTACK_LIGTHHUNTER_BY_TECHNOLOGY) * this.getBaseDamage() / 100);
            cst.close();
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    @Override
    public void takeDamage(int receivedDamage) {
        this.setArmor(this.getArmor() - receivedDamage);
    }

    @Override
    public int getMetalCost() {
        // TODO Auto-generated method stub
    }
}
```




```
        try {
            return cst.getInt(3);
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            return 0;
        }
    }

    @Override
    public int getDeuteriumCost() {
        // TODO Auto-generated method stub
        try {
            return cst.getInt(5);
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            return 0;
        }
    }

    @Override
    public int getChanceGeneratinWaste() {
        // TODO Auto-generated method stub
        try {
            return cst.getInt(10);
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            return 0;
        }
    }

    @Override
    public int getChanceAttackAgain() {
        // TODO Auto-generated method stub
        return this.CHANCE_ATTACK_AGAIN_BATTLESHIP;
    }
}
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 126                 |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

## - Clase Enemy

Clase Enemy representa al enemigo que nos ataca cada cierto tiempo, este tiene su propio ejército con sus naves.

```
import java.sql.Connection;
import java.util.ArrayList;

public class Enemy {

    ArrayList<MilitaryUnit>[] enemyArmy = new ArrayList[4];

    public Enemy(Connection con) {
        enemyArmy[0] = addLightHunters(con);
        enemyArmy[1] = addHeavyHunters(con);
        enemyArmy[2] = addBattleShips(con);
        enemyArmy[3] = addArmoredShips(con);
    }

    public ArrayList<MilitaryUnit>[] getEnemyArmy() {
        return enemyArmy;
    }

    private ArrayList<MilitaryUnit> addLightHunters(Connection con) {
        ArrayList<MilitaryUnit> lightHunters = new
ArrayList<MilitaryUnit>();
        int[] num_units = new int[5];
        // total de 30 unidades que se reparten aleatoriamente en
grupos de diferentes niveles de tecnologia
        num_units[0] = (int) (Math.random()*30);
        num_units[1] = (int) (Math.random()*(30-num_units[0]));
        num_units[2] = (int)
(Math.random()*(30-num_units[1]-num_units[0]));
        num_units[3] = (int)
(Math.random()*(30-num_units[2]-num_units[1]-num_units[0]));
        num_units[4] = (int)
```



```
(Math.random()*(30-num_units[3]-num_units[2]-num_units[1]-num_units[0]))
;

    for (int i = 0; i < num_units.length; i++) {
        for (int j = 1; j <= num_units[i]; j++) {
            lightHunters.add(new LightHunter(i, i, con));
        }
    }
    return lightHunters;
}

private ArrayList<MilitaryUnit> addHeavyHunters(Connection con) {
    ArrayList<MilitaryUnit> heavyHunters = new
ArrayList<MilitaryUnit>();
    int[] num_units = new int[5];
    // total de 15 unidades que se reparten aleatoriamente en
grupos de diferentes niveles de tecnologia
    num_units[0] = (int) (Math.random()*15);
    num_units[1] = (int) (Math.random()*(15-num_units[0]));
    num_units[2] = (int)
(Math.random()*(15-num_units[1]-num_units[0]));
    num_units[3] = (int)
(Math.random()*(15-num_units[2]-num_units[1]-num_units[0]));
    num_units[4] = (int)
(Math.random()*(15-num_units[3]-num_units[2]-num_units[1]-num_units[0]));
;

    for (int i = 0; i < num_units.length; i++) {
        for (int j = 1; j <= num_units[i]; j++) {
            heavyHunters.add(new HeavyHunter(i, i, con));
        }
    }
    return heavyHunters;
}

private ArrayList<MilitaryUnit> addBattleShips(Connection con) {
    ArrayList<MilitaryUnit> battleShips = new
ArrayList<MilitaryUnit>();
    int[] num_units = new int[5];
```




```
// total de 6 unidades que se reparten aleatoriamente en
grupos de diferentes niveles de tecnologia
num_units[0] = (int) (Math.random()*6);
num_units[1] = (int) (Math.random()*(6-num_units[0]));
num_units[2] = (int)
(Math.random()*(6-num_units[1]-num_units[0]));
num_units[3] = (int)
(Math.random()*(6-num_units[2]-num_units[1]-num_units[0]));
num_units[4] = (int)
(Math.random()*(6-num_units[3]-num_units[2]-num_units[1]-num_units[0]));

    for (int i = 0; i < num_units.length; i++) {
        for (int j = 1; j <= num_units[i]; j++) {
            battleShips.add(new BattleShip(i, i, con));
        }
    }
    return battleShips;
}

private ArrayList<MilitaryUnit> addArmoredShips(Connection con) {
    ArrayList<MilitaryUnit> armoredShips = new
ArrayList<MilitaryUnit>();
    int[] num_units = new int[5];
    // total de 3 unidades que se reparten aleatoriamente en
grupos de diferentes niveles de tecnologia
    num_units[0] = (int) (Math.random()*3);
    num_units[1] = (int) (Math.random()*(3-num_units[0]));
    num_units[2] = (int)
(Math.random()*(3-num_units[1]-num_units[0]));
    num_units[3] = (int)
(Math.random()*(3-num_units[2]-num_units[1]-num_units[0]));
    num_units[4] = (int)
(Math.random()*(3-num_units[3]-num_units[2]-num_units[1]-num_units[0]));

    for (int i = 0; i < num_units.length; i++) {
        for (int j = 1; j <= num_units[i]; j++) {
            armoredShips.add(new ArmoredShip(i, i, con));}}
    return armoredShips;
}}
```



|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 129                 |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

## - Clase ResourceException

Clase ResourceException nos lanza un mensaje si los recursos son insuficientes.

```
public class ResourceException extends Exception{
    public ResourceException() {
        // TODO Auto-generated constructor stub
        super("¡Recursos insuficientes!");
    }
    public ResourceException(String error) {
        super(error);
    }
}
```

## - Clase Main

Clase Main inicializa toda la aplicación.

```
public class Main {

    public static void main(String[] args) {
        ConnectionBDD c = new ConnectionBDD();
        Connection con= c.getConex();


        new VentanaLogin(con);
    }
}
```

## - Interfaz MilitaryUnit

Interfaz MilitaryUnit, contiene varios métodos que comparten las naves y las defensas.

```
import java.sql.Connection;

public interface MilitaryUnit {
    abstract int attack();
    abstract void takeDamage(int receivedDamage);
}
```

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 130                 |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

```

abstract int getActualArmor();
abstract int getMetalCost();
abstract int getDeuteriumCost();
abstract int getChanceGeneratinWaste();
abstract int getChanceAttackAgain();
abstract int getQuantity();
abstract void setQuantity(int quantity);
abstract void resetArmor();
//take damage
}

```

## - Interfaz Variables

Interfaz Variables, contiene variables estáticas que comparten las naves y las defensas.

```

public interface Variables {
    // resources available to create the first enemy fleet
    public final int DEUTERIUM_BASE_ENEMY_ARMY = 26000;
    public final int METAL_BASE_ENEMY_ARMY = 180000;

    // percentage increase of resources available to create enemy
    fleet
    public final int ENEMY_FLEET_INCREASE = 6;

    // resources increment every minute
    public final int PLANET_DEUTERIUM_GENERATED = 1500;
    public final int PLANET_METAL_GENERATED = 5000;

    // TECHNOLOGY COST
    public final int UPGRADE_BASE_DEFENSE_TECHNOLOGY_DEUTERIUM_COST =
2000;
    public final int UPGRADE_BASE_ATTACK_TECHNOLOGY_DEUTERIUM_COST =
2000;
    public final int UPGRADE_PLUS_DEFENSE_TECHNOLOGY_DEUTERIUM_COST =
60;
    public final int UPGRADE_PLUS_ATTACK_TECHNOLOGY_DEUTERIUM_COST =
60;
}

```

|                       |                      |                            |
|-----------------------|----------------------|----------------------------|
| Elaborat per: Grupo 2 | Revisat per: Grupo 2 | Autoritzat per: Josep Diaz |
|-----------------------|----------------------|----------------------------|



```
// COST SHIPS
public final int METAL_COST_LIGTHHUNTER = 3000;
public final int METAL_COST_HEAVYHUNTER = 6500;
public final int METAL_COST_BATTLESHIP = 45000;
public final int METAL_COST_ARMOREDSHIP = 30000;
public final int DEUTERIUM_COST_LIGTHHUNTER = 50;
public final int DEUTERIUM_COST_HEAVYHUNTER = 50;
public final int DEUTERIUM_COST_BATTLESHIP = 7000;
public final int DEUTERIUM_COST_ARMOREDSHIP = 15000;

// COST DEFENSES

public final int DEUTERIUM_COST_MISSILELAUNCHER = 0;
public final int DEUTERIUM_COST_IONCANNON = 500;
public final int DEUTERIUM_COST_PLASMACANNON = 5000;
public final int METAL_COST_MISSILELAUNCHER = 2000;
public final int METAL_COST_IONCANNON = 4000;
public final int METAL_COST_PLASMACANNON = 50000;

// array units costs
public final int[] METAL_COST_UNITS =
{METAL_COST_LIGTHHUNTER,METAL_COST_HEAVYHUNTER,METAL_COST_BATTLESHIP,METAL_COST_ARMOREDSHIP,METAL_COST_MISSILELAUNCHER,METAL_COST_IONCANNON,METAL_COST_PLASMACANNON};
public final int[] DEUTERIUM_COST_UNITS =
{DEUTERIUM_COST_LIGTHHUNTER,DEUTERIUM_COST_HEAVYHUNTER,DEUTERIUM_COST_BATTLESHIP,DEUTERIUM_COST_ARMOREDSHIP,DEUTERIUM_COST_MISSILELAUNCHER,DEUTERIUM_COST_IONCANNON,DEUTERIUM_COST_PLASMACANNON};

// BASE DAMAGE SHIPS
public final int BASE_DAMAGE_LIGTHHUNTER = 80;
public final int BASE_DAMAGE_HEAVYHUNTER = 150;
public final int BASE_DAMAGE_BATTLESHIP = 1000;
public final int BASE_DAMAGE_ARMOREDSHIP = 700;

// BASE DAMAGE DEFENSES

public final int BASE_DAMAGE_MISSILELAUNCHER = 80;
```



```
public final int BASE_DAMAGE_IONCANNON = 250;
public final int BASE_DAMAGE_PLASMACANNON = 2000;

// REDUCTION_DEFENSE
public final int REDUCTION_DEFENSE_IONCANNON = 100;

// ARMOR SHIPS
public final int ARMOR_LIGTHHUNTER = 400;
public final int ARMOR_HEAVYHUNTER = 1000;
public final int ARMOR_BATTLESHIP = 6000;
public final int ARMOR_ARMOREDSHIP = 8000;

// ARMOR DEFENSES
public final int ARMOR_MISSILELAUNCHER = 200;
public final int ARMOR_IONCANNON = 1200;
public final int ARMOR_PLASMACANNON = 7000;

//fleet armor increase percentage per tech level
public final int PLUS_ARMOR_LIGTHHUNTER_BY_TECHNOLOGY = 5;
public final int PLUS_ARMOR_HEAVYHUNTER_BY_TECHNOLOGY = 5;
public final int PLUS_ARMOR_BATTLESHIP_BY_TECHNOLOGY = 5;
public final int PLUS_ARMOR_ARMOREDSHIP_BY_TECHNOLOGY = 5;

// defense armor increase percentage per tech level
public final int PLUS_ARMOR_MISSILELAUNCHER_BY_TECHNOLOGY = 5;
public final int PLUS_ARMOR_IONCANNON_BY_TECHNOLOGY = 5;
public final int PLUS_ARMOR_PLASMACANNON_BY_TECHNOLOGY = 5;

// fleet attack power increase percentage per tech level
public final int PLUS_ATTACK_LIGTHHUNTER_BY_TECHNOLOGY = 5;
public final int PLUS_ATTACK_HEAVYHUNTER_BY_TECHNOLOGY = 5;
public final int PLUS_ATTACK_BATTLESHIP_BY_TECHNOLOGY = 5;
public final int PLUS_ATTACK_ARMOREDSHIP_BY_TECHNOLOGY = 5;

// Defense attack power increase percentage per tech level
public final int PLUS_ATTACK_MISSILELAUNCHER_BY_TECHNOLOGY = 5;
```



```
public final int PLUS_ATTACK_IONCANNON_BY_TECHNOLOGY = 5;
public final int PLUS_ATTACK_PLASMACANNON_BY_TECHNOLOGY = 5;

// fleet probability of generating waste
public final int CHANCE_GENERATNG_WASTE_LIGHTHUNTER = 55;
public final int CHANCE_GENERATNG_WASTE_HEAVYHUNTER = 65;
public final int CHANCE_GENERATNG_WASTE_BATTLESHIP = 80;
public final int CHANCE_GENERATNG_WASTE_ARMOREDSHIP = 90;

// Defense probability of generating waste
public final int CHANCE_GENERATNG_WASTE_MISSILELAUNCHER = 55;
public final int CHANCE_GENERATNG_WASTE_IONCANNON = 65;
public final int CHANCE_GENERATNG_WASTE_PLASMACANNON = 75;

// fleet chance to attack again
public final int CHANCE_ATTACK_AGAIN_LIGHTHUNTER = 3;
public final int CHANCE_ATTACK_AGAIN_HEAVYHUNTER = 7;
public final int CHANCE_ATTACK_AGAIN_BATTLESHIP = 45;
public final int CHANCE_ATTACK_AGAIN_ARMOREDSHIP = 70;

//Defense chance to attack again
public final int CHANCE_ATTACK_AGAIN_MISSILELAUNCHER = 5;
public final int CHANCE_ATTACK_AGAIN_IONCANNON = 12;
public final int CHANCE_ATTACK_AGAIN_PLASMACANNON = 30;

// CHANCE ATTACK EVERY UNIT
// LIGHTHUNTER, HEAVYHUNTER, BATTLESHIP, ARMOREDSHIP,
// MISSILELAUNCHER, IONCANNON, PLASMACANNON
public final int[] CHANCE_ATTACK_PLANET_UNITS =
{5,10,15,40,5,10,15};
// LIGHTHUNTER, HEAVYHUNTER, BATTLESHIP, ARMOREDSHIP
public final int[] CHANCE_ATTACK_ENEMY_UNITS = {10,20,30,40};

// percentage of waste that will be generated with respect to the
// cost of the units
public final int PERCENTATGE_WASTE = 70;

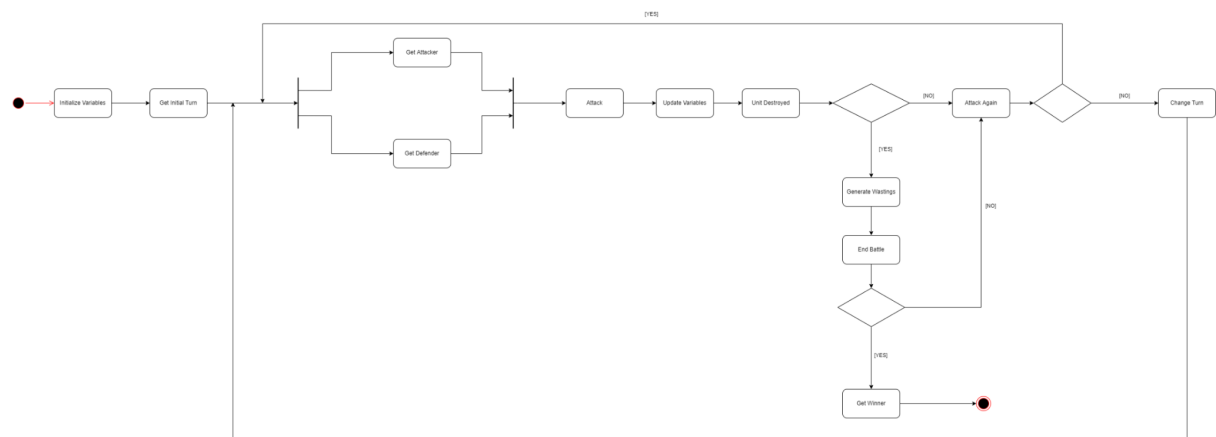
}
```



## M5 - ENTORNOS DE DESARROLLO

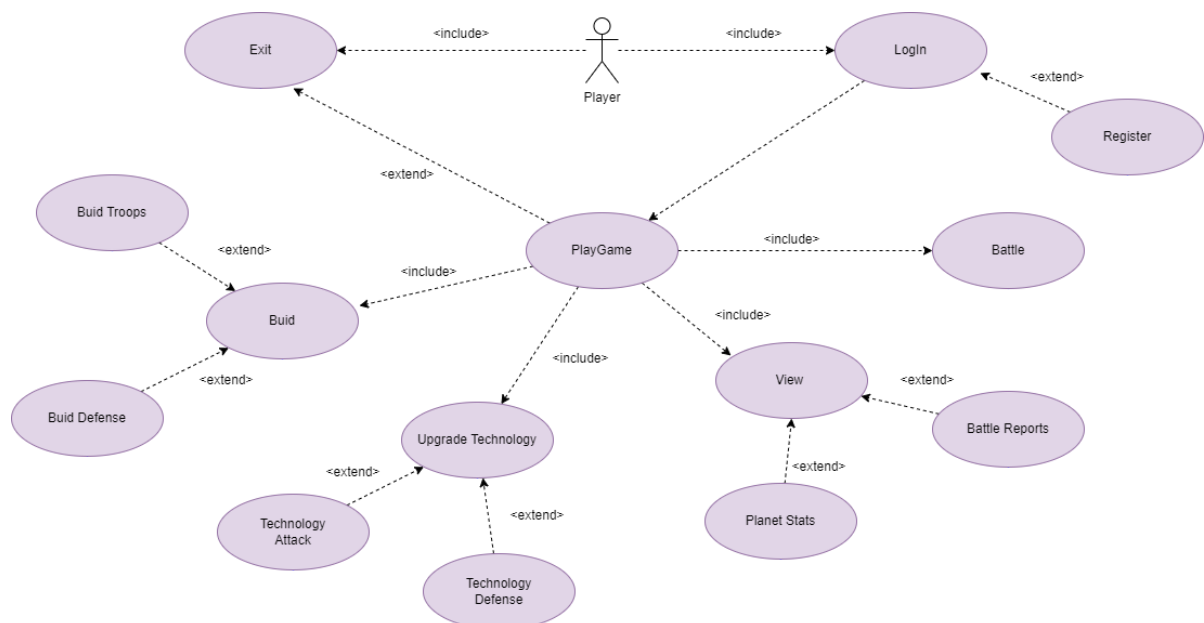
### - Diagrama de Actividades


Diagrama de Actividades de la batalla. Funcionamiento de la batalla, sus pasos y posibilidades.



### - Diagrama de casos de uso

Diagrama de casos de uso de la aplicación completa. Funcionamiento de la aplicación y sus pasos.




|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 135                 |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

## M1 - GLOSARIO

| PALABRA           | DESCRIPCIÓN   |
|-------------------|---|
| PROCEDURE         | Procedimiento.  |
| PRIMARY KEY       | Clave primaria.   |
| NOT NULL          | Dato no nulo.   |
| VARCHAR           | Dato de tipo texto.   |
| UNIQUE            | Dato único, que no se puede repetir.                                      |
| EXECUTE IMMEDIATE | Ejecutar inmediatamente algo.   |
| INSERT            | Insertar.   |
| VALUES            | Datos.  |
| DROP              | Borrar.   |
| PURGE             | Purgar, eliminar de forma que den igual las relaciones entre tablas.      |
| SELECT            | Seleccionar algo.   |
| COUNT             | Contar algo.  |
| IN OUT            | Datos de entrada y salida.  |
| BEGIN - END       | Atributos para especificar que dentro de estos está el cuerpo del código. |
| FOREIGN KEY       | Clave foránea o ajena.  |
| IMPLEMENTS        | Implementar.  |
| INT               | Tipo de dato. Numérico entero.  |
| RETURN            | Retorno de datos.   |
| SET               | Cambiar algo.   |
| GET               | Coger algo.   |


|                       |                      |                            |
|-----------------------|----------------------|----------------------------|
| Elaborat per: Grupo 2 | Revisat per: Grupo 2 | Autoritzat per: Josep Diaz |
|-----------------------|----------------------|----------------------------|

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 136                 |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

|                     |   |
|---------------------|---|
| STRING              | Tipo de dato, texto.  |
| VOID                | Atributo de una función que no retorna datos.   |
| EXTENDS             | Extiende de una clase.  |
| OVERRIDE            | Sobreescribir.  |
| ABSTRACT CLASS      | Clase abstracta.  |
| ARRAYLIST           | Clase que permite almacenar datos en memoria.   |
| PRIVATE             | Atributo privado, solo se puede acceder desde la clase.   |
| FOR                 | Bucle iterativo.  |
| MATH.RANDOM()       | Librería de Java que saca números random.   |
| LENGTH()            | Método que retorna la longitud de un objeto.  |
| SIZE()              | Método de similar comportamiento al anterior.   |
| INSTANCEOF          | "Instancia de" Determina si el objeto al que apunta una referencia dada es una instancia de una clase o interfaz concretas. |
| SWITCH()            | Es una manera de tomar una decisión a partir de un valor dado, con varios resultados posibles.                              |
| CASE                | Donde se definirán los diferentes resultados.   |
| IF - ELSE           | Condicional. Si pasa esto o lo otro.  |
| ADD                 | Añadir.   |
| MAX()               | Función que te retorna el valor máximo.   |
| LOGIN               | Iniciar sesión.   |
| BOOLEAN(TRUE-FALSE) | Verdadero-Falso. Valores binarios.  |
| NO_DATA_FOUND       | Excepción. Aparece cuando no hay datos en la tabla.   |
| INTO                | Comando que se utiliza para meter datos en una o varias variables.  |


|                       |                      |                            |
|-----------------------|----------------------|----------------------------|
| Elaborat per: Grupo 2 | Revisat per: Grupo 2 | Autoritzat per: Josep Diaz |
|-----------------------|----------------------|----------------------------|



|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 137                 |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |


|                            |   |
|----------------------------|---|
| WHERE                      | Condicional. Comando que se utiliza para especificar condiciones de búsqueda.       |
| NUMBER                     | Tipo de dato. Número.   |
| AS                         | Comando que se utiliza para designar un sobrenombre.                                |
| FROM                       | Comando que se utiliza para especificar de donde viene algo.                        |
| EXCEPTION                  | Excepcion.  |
| DATE                       | Tipo de dato. Fecha.  |
| PUBLIC                     | Clase o método público, al que se puede acceder desde cualquier lado.               |
| TRY-CATCH                  | Sentencia que revisa una posibilidad y recoge cualquier error que pueda saltar.     |
| JAVA.SQL.CALLABLESTATEMENT | Sentencia que nos permite llamar a procedimientos o funciones alojadas en el PLSQL. |
| JAVA.SQL.CONNECTION        | Librería de SQL que nos permite conectarnos a la base de datos.                     |
| JAVA.SQL.SQLEXCEPTION      | Librería de SQL que nos permite controlar excepciones.                              |
| JFRAME                     | Elemento de la librería Swing de Java que lanza una ventana emergente.              |
| JLABEL                     | Componente heredado de JFrame, muestra una etiqueta en la ventana.                  |
| JTEXTFIELD                 | Componente heredado de JFrame, muestra una campo de texto en la ventana.            |
| JPANEL                     | Componente heredado de JFrame, muestra una panel en la ventana.                     |
| JBUTTON                    | Componente heredado de JFrame, muestra un botón en la ventana.                      |
| JOPTIONPANE                | Componente heredado de JFrame, muestra un   |

|                       |                      |                            |
|-----------------------|----------------------|----------------------------|
| Elaborat per: Grupo 2 | Revisat per: Grupo 2 | Autoritzat per: Josep Diaz |
|-----------------------|----------------------|----------------------------|

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 138                 |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |


|                         |  |
|-------------------------|--|
|                         | mensaje emergente del tipo que sea.  |
| JPASSWORDFIELD          | Componente heredado de JFrame, muestra un campo de texto ocultando aquello que escribes en él. |
| JDATEPANEL_IMPL         | Te muestra un panel con un calendario.   |
| FLOWLAYOUT              | Tipo de despliegue de un panel, que ubica los componentes de forma horizontal.                 |
| GRIDLAYOUT              | Tipo de despliegue de un panel, que ubica los componentes en forma de grilla.                  |
| EMPTYBORDER             | Tipo de borde, despliega un borde vacío.   |
| FONT                    | Objeto de tipo Fuente, para modificar el tipo de letra.  |
| BOLD                    | Peso de la letra.  |
| SET                     | Colocar.   |
| ACTION_LISTENER         | Clase que escucha la acción que un usuario hace sobre un objeto en el JFrame.                  |
| ACTION_EVENT            | Clase que realiza eventos.   |
| VALUE_OF                | Valor de.  |
| NEW                     | Nuevo.   |
| DISPOSE                 | Prescindir de algo.  |
| SHOWMESSAGEDIALOG       | Mostrar mensajes de diálogo.   |
| FOREGROUND              | Primer plano.  |
| BORDER_LAYOUT           | Para definir posición en el Layout (Ejemplo:BorderLayout.center).                              |
| DEFAULT_CLOSE_OPERATION | Operación de cierre predeterminada.  |
| EXIT_ON_CLOSE           | Cerrar al salir. Se acabará el programa una vez cerremos la ventana.                           |

|                       |                      |                            |
|-----------------------|----------------------|----------------------------|
| Elaborat per: Grupo 2 | Revisat per: Grupo 2 | Autoritzat per: Josep Diaz |
|-----------------------|----------------------|----------------------------|

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 139                 |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

|                     |  |
|---------------------|--|
| RESIZABLE           | Que se puede redimensionar.  |
| OPAQUE              | Que es opaco.  |
| UTIL_DATA_MODEL     | Modelo de datos útiles.  |
| PROPERTIES          | Propiedades.   |
| PUT                 | Poner.   |
| DATE_PICKER         | Selector de fechas.  |
| SIMPLE_DATE_FORMAT  | Formato simple de fechas.  |
| WARNING_MESSAGE     | Mensaje de advertencia.  |
| FORMAT              | Formato de algo.   |
| SHOW_MESSAGE_DIALOG | Muestra un POP-UP con un mensaje que podemos personalizar.         |
| PARSE               | Cambiar tipo de dato.  |
| STACK_TRACE         | Devuelve datos de errores.   |
| ICON_IMAGE          | Objeto que sirve para trabajar imágenes con Labels.                |
| DATE_PATTERN        | Fecha patrón.  |
| THROWS              | Indica que puede devolver un error.                                |
| PARSE_EXCEPTION     | Objeto contiene información sobre las excepciones.                 |
| TO_STRING           | Transforma un valor a String(texto).                               |
| JTABBEDPANE         | Es un panel gráfico que nos permite tener pestañas independientes. |
| ARRAYLIST           | Es una lista de objetos o variables.                               |
| UNDECORATED         | Sirve para eliminar decoraciones.                                  |
| PREFERRED_SIZE      | Define el tamaño por defecto de un objeto o ventana.               |
| CHANGE_LISTENER     | Se activa cuando se cambia la pestaña.                             |

|                       |                      |                            |
|-----------------------|----------------------|----------------------------|
| Elaborat per: Grupo 2 | Revisat per: Grupo 2 | Autoritzat per: Josep Diaz |
|-----------------------|----------------------|----------------------------|

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 140                 |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

|                        |  |
|------------------------|--|
| SELECTED_INDEX         | Indica el index seleccionado del JTabbedPane.    |
| EXIT                   | Sirve para salir.                                |
| DEFAULT_TOOLKIT        | Herramienta que usamos para importar imágenes.   |
| SCREEN_SIZE            | Se usa para redimensionar la ventana.            |
| SET_LOCATION           | Se usa para reubicar objetos o ventanas.         |
| GET_WIDTH              | Devuelve el ancho del objeto o ventana.          |
| GET_HEIGHT             | Devuelve el alto del objeto o ventana.           |
| GET_SCALED_INSTANCE    | Cambiar dimensiones de una imagen.               |
| SCALE_SMOOTH           | Redimensionado suave.                            |
| REGISTER_DRIVER        | Driver para el registro.                         |
| ORACLE_DRIVER          | Driver para operar con una base de datos Oracle. |
| DRIVER_MANAGER         | Manager de drivers.                              |
| GET_CONNECTION         | Importamos la conexión.                          |
| REGISTER_OUT_PARAMETER | El valor indicado va a ser devuelto.             |
| TYPES                  | Especifica el tipo de entrada o salida.          |
| PREPARE_CALL           | Prepara una llamada a la base de datos.          |
| EXECUTE                | Ejecuta un código ya preparado con anterioridad. |


## WEBGRAFIA

<https://lefunes.wordpress.com/2008/11/22/incluyendo-una-imagen-de-fondo-en-un-jpanel/>

<https://stackoverflow.com/>

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

|                       |                      |                            |
|-----------------------|----------------------|----------------------------|
| Elaborat per: Grupo 2 | Revisat per: Grupo 2 | Autoritzat per: Josep Diaz |
|-----------------------|----------------------|----------------------------|

|   |  |                               |
|---|--|-------------------------------|
|  | MANUAL DE<br>DESCRIPCIÓ I<br>ESPECIFICACIÓ DE<br>CÀRRECS DE LA<br>EMPRESA XYZ,CA | Pàgina N° 141                 |
|   |  | Data d'Elaboració: 09/05/2022 |
|   |  | Data d'Aprovació: 18/05/2022  |

<https://www.codejava.net/java-se/swing/how-to-use-jdatepicker-to-display-calendar-component>

[mtortos6@xtec.cat](mailto:mtortos6@xtec.cat)

[raracil1@xtec.cat](mailto:raracil1@xtec.cat)

[jdiazalbero@gmail.com](mailto:jdiazalbero@gmail.com)

[jsales8@xtec.cat](mailto:jsales8@xtec.cat)

[jgarc29@xtec.cat](mailto:jgarc29@xtec.cat)