


PLANET WARS



Fecha de realización: 09/05/2022


Integrantes:

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

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

ÍNDICE

Elaborat per: Grupo 4	Revisat per: Grupo 4	Autoritzat per: Josep Diaz
-----------------------	----------------------	----------------------------

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

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.

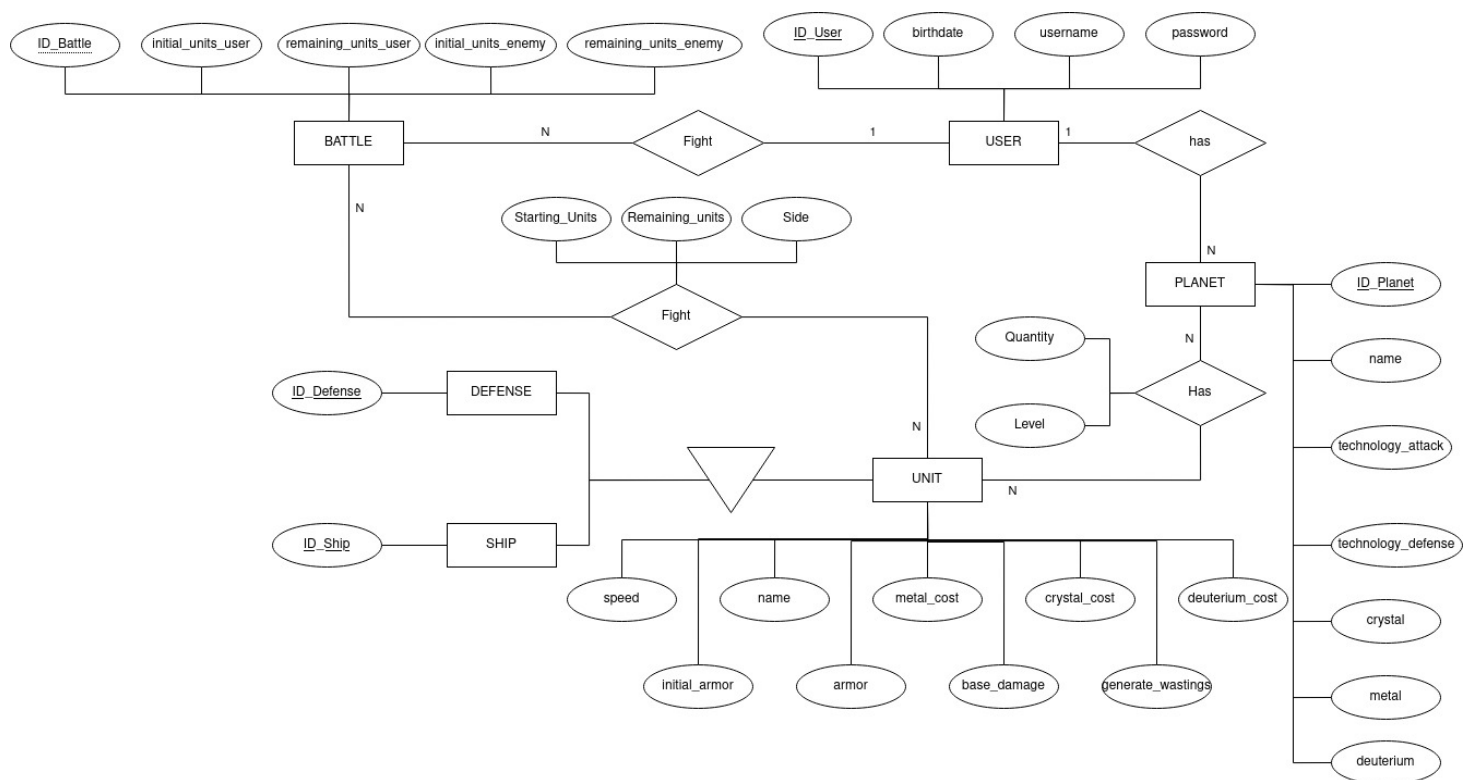
Los ataques de los enemigos serán constantes, por lo que no podrás dejar desatendido el planeta.

Elaborat per: Grupo 4	Revisat per: Grupo 4	Autoritzat per: Josep Diaz
-----------------------	----------------------	----------------------------



M2 - BASE DE DATOS

- Diagrama de Chen





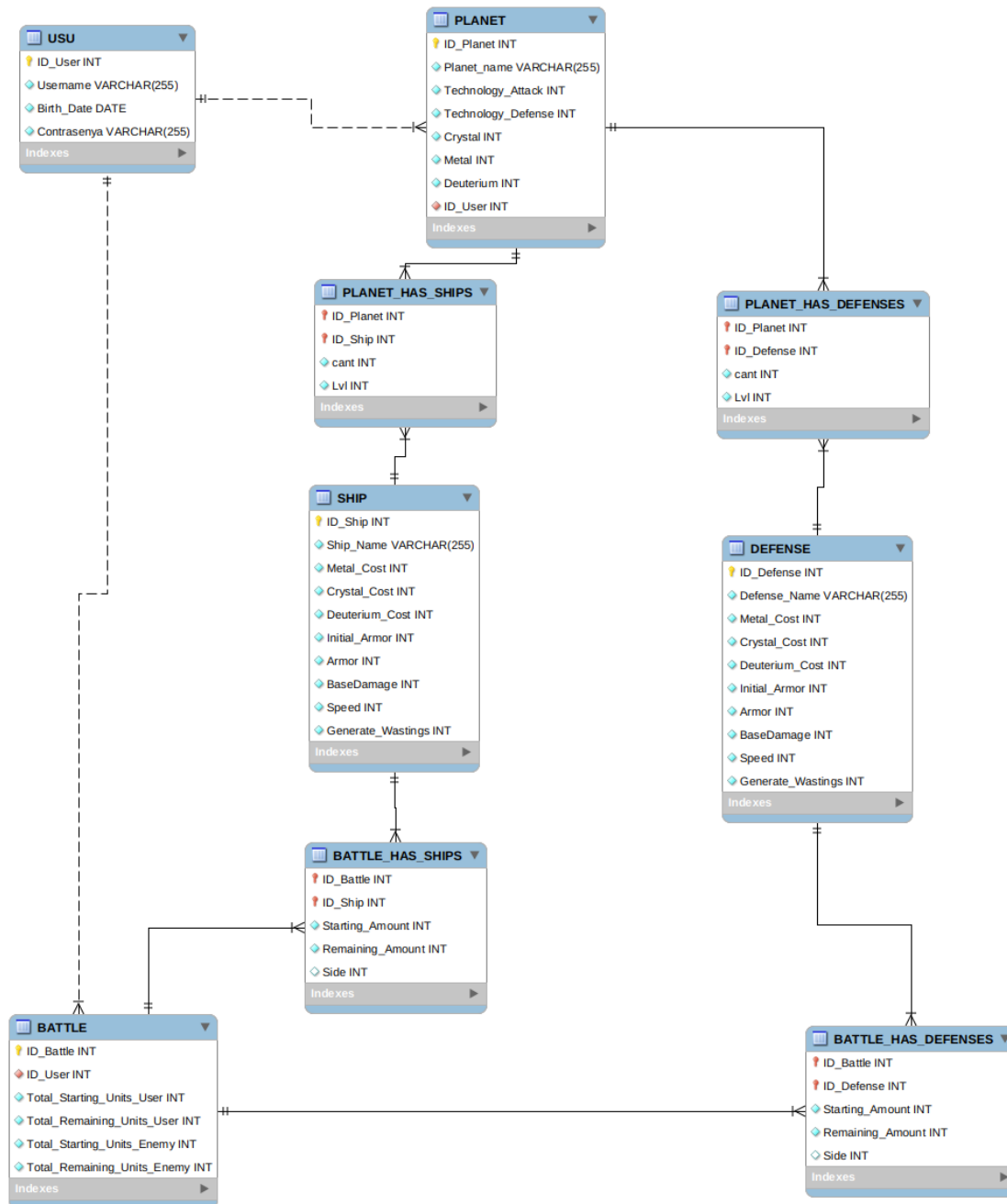
MANUAL DE DESCRIPCIÓ I ESPECIFICACIÓ DE CÀRRECS DE LA EMPRESA XYZ,CA


Pàgina N° 5

Data d'Elaboració: 09/05/2022

Data d'Aprovació:

- Diagrama relacional



	MANUAL DE DESCRIPCIÓ I ESPECIFICACIÓ DE CÀRRECS DE LA EMPRESA XYZ,CA	Pàgina N° 6
		Data d'Elaboració: 09/05/2022
		Data d'Aprovació:

- 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;
```

Elaborat per: Grupo 4	Revisat per: Grupo 4	Autoritzat per: Josep Diaz
-----------------------	----------------------	----------------------------

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

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

```

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 4	Revisat per: Grupo 4	Autoritzat per: Josep Diaz
-----------------------	----------------------	----------------------------

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

- 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 del Planeta según Username

Procedimiento para la extracción del ID del planeta según el nombre del usuario.

```
create or replace procedure get_planetusu( nombre in varchar, id_planeta
out number) as begin
select id_planet into id_planeta from planet p inner join usu u on
u.id_user = u.id_user where u.username = nombre;
end;
```

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

- Procedimiento extraer cantidad Planet_has-Ships


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_cantidad(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 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;
```

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

- 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;
```

- 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;
```

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

- 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) 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;
```

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

- 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,
    cant_metal in number,
    cant_deuterium in number,
    id_usu in number) as
    id_planeta number;
begin
select max(id_planeta) + 1 into id_planeta from planet;
if id_planeta is null then
id_planeta := 1;
end if;
insert into planet(id_planeta, 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;
```

Elaborat per: Grupo 4	Revisat per: Grupo 4	Autoritzat per: Josep Diaz
-----------------------	----------------------	----------------------------

	MANUAL DE DESCRIPCIÓ I ESPECIFICACIÓ DE CÀRRECS DE LA EMPRESA XYZ,CA	Pàgina N° 14
		Data d'Elaboració: 09/05/2022
		Data d'Aprovació:

```
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;
```

- 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;
```

	MANUAL DE DESCRIPCIÓ I ESPECIFICACIÓ DE CÀRRECS DE LA EMPRESA XYZ,CA	Pàgina N° 15
		Data d'Elaboració: 09/05/2022
		Data d'Aprovació:

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

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;

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_DEUTERIUM_COST;
ArrayList<MilitaryUnit>[] army = new ArrayList[7];

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.upgradeDefenseTechnologyDeuteriumCost+(this.upgradeDefenseTechnologyDeuteriumCost*10)/100;
    }
}

void upgradeTechnologyAttack() {
```


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

```

        if (Comprobador(0,
this.upgradeAttackTechnologyDeuteriumCost)==true) {

            this.technologyAttack++;

this.upgradeAttackTechnologyDeuteriumCost=this.upgradeAttackTechnologyDe
uteriumCost+(this.upgradeAttackTechnologyDeuteriumCost*10)/100;
        }
    }

    void newLigthHunter(int n,Connection con) {
        int conta=0;
        CallableStatement cst;
        InfoShips ship=new InfoShips();
        cst= ship.getInfoShips(con, 1);
        for (int i=0;i<n;i++) {
            try {
                if (Comprobador(cst.getInt(3),
cst.getInt(5))==true) {
                    conta++;
                    LightHunter l= new
LightHunter(technologyDefense, technologyAttack,con);
                    this.army[0].add(l);
                }
                else {
                    System.out.println("Recursos
insuficientes");
                    break;
                }
            } catch (SQLException e) {
                // TODO Auto-generated catch block
                //e.printStackTrace();
                System.out.println("Error de ejecucion");
            }
        }
        System.out.println("Se han anyadido : "+conta+" Ligth hunters");
    }

    void newHeavyHunter(int n,Connection con) {

```



```
int conta=0;
CallableStatement cst;
InfoShips ship=new InfoShips();
cst= ship.getInfoShips(con, 1);
for (int i=0;i<n;i++) {
    try {
        if (Comprobador(cst.getInt(3),
cst.getInt(5))==true) {
            conta++;
            HeavyHunter h= new
HeavyHunter(technologyDefense, technologyAtack,con);
            this.army[1].add(h);
        }
        else {
            System.out.println("Recursos
insuficientes");
            break;
        }
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
System.out.println("Se han anyadido : "+conta+" Heavy hunters");
}
void newBattleShip(int n,Connection con) {
    int conta=0;
    CallableStatement cst;
    InfoShips ship=new InfoShips();
    cst= ship.getInfoShips(con, 1);
    for (int i=0;i<n;i++) {
        try {
            if (Comprobador(cst.getInt(3),
cst.getInt(5))==true) {
                conta++;
                BattleShip b= new
BattleShip(technologyDefense, technologyAtack,con);
                this.army[2].add(b);
            }
        }
    }
}
```



```
        else {
            System.out.println("Recursos
insuficientes");

            break;
        }
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
System.out.println("Se han anyadido : "+conta+" Battleships");
}
void newArmoredShip(int n,Connection con) {
    int conta=0;
    CallableStatement cst;
    InfoShips ship=new InfoShips();
    cst= ship.getInfoShips(con, 1);
    for (int i=0;i<n;i++) {
        try {
            if (Comprobador(cst.getInt(3),
cst.getInt(5))==true) {
                conta++;
                ArmoredShip a= new
ArmoredShip(technologyDefense, technologyAtack,con);
                this.army[3].add(a);
            }
            else {
                System.out.println("Recursos
insuficientes");

                break;
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
    System.out.println("Se han anyadido : "+conta+" Armored ships");
}
```




```
//DEFENSAS-----  
-----  
  
void newMissileLauncher(int n,Connection con) {  
    int conta=0;  
    CallableStatement cst;  
    InfoDefense def=new InfoDefense();  
    cst= def.getInfoDefense(con, 1);  
    for (int i=0;i<n;i++) {  
        try {  
            if (Comprobador(cst.getInt(3),  
cst.getInt(5))==true) {  
                conta++;  
                MissileLauncher m= new  
MissileLauncher(technologyDefense, technologyAtack,con);  
                this.army[4].add(m);  
            }  
            else {  
                System.out.println("Recursos  
insuficientes");  
                break;  
            }  
        } catch (SQLException e) {  
            // TODO Auto-generated catch block  
            e.printStackTrace();  
        }  
    }  
    System.out.println("Se han anyadido : "+conta+" Missile  
Launchers");  
}  
  
void newIonCannon(int n,Connection con) {  
    int conta=0;  
    CallableStatement cst;  
    InfoDefense def=new InfoDefense();  
    cst= def.getInfoDefense(con, 2);  
    for (int i=0;i<n;i++) {  
        try {  
            if (Comprobador(cst.getInt(3),  
cst.getInt(5))==true) {
```



```
        conta++;
        IonCannon io= new
IonCannon(technologyDefense, technologyAttack,con);
        this.army[5].add(io);
    }
    else {
        System.out.println("Recursos
insuficientes");
        break;
    }
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
}
System.out.println("Se han anyadido : "+conta+" Ion Cannons");
}
void newPlasmaCannon(int n,Connection con) {
    CallableStatement cst;
    InfoDefense def=new InfoDefense();
    cst= def.getInfoDefense(con, 3);
    int conta=0;
    for (int i=0;i<n;i++) {
        try {
            if (Comprobador(cst.getInt(3),
cst.getInt(5))==true) {
                conta++;
                PlasmaCannon p= new
PlasmaCannon(technologyDefense, technologyAttack,con);
                this.army[6].add(p);
            }
            else {
                System.out.println("Recursos
insuficientes");
                break;
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```



```
    }  
}  
System.out.println("Se han anyadido : "+conta+" Plasma Cannons");  
}  
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("Ligth 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() {
```

	MANUAL DE DESCRIPCIÓ I ESPECIFICACIÓ DE CÀRRECS DE LA EMPRESA XYZ,CA	Pàgina N° 23
		Data d'Elaboració: 09/05/2022
		Data d'Aprovació:

```

        return army;
    }
    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;
    }
}

```

	MANUAL DE DESCRIPCIÓ I ESPECIFICACIÓ DE CÀRRECS DE LA EMPRESA XYZ,CA	Pàgina N° 24
		Data d'Elaboració: 09/05/2022
		Data d'Aprovació:

```

    }
    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{

```


Elaborat per: Grupo 4	Revisat per: Grupo 4	Autoritzat per: Josep Diaz
-----------------------	----------------------	----------------------------



```
CallableStatement cst;
public HeavyHunter(int tecnoDefense, int tecnoAttack, 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_LIGTHHUNTER_BY_TECHNOLOGY)*this.getInitialArmor
())/100);

this.setBaseDamage(this.getBaseDamage()+(tecnoAttack*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 {
```

	MANUAL DE DESCRIPCIÓ I ESPECIFICACIÓ DE CÀRRECS DE LA EMPRESA XYZ,CA	Pàgina N° 26
		Data d'Elaboració: 09/05/2022
		Data d'Aprovació:

```

        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_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 tecnoAttack, 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()+(tecnoAttack*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 {
```

	MANUAL DE DESCRIPCIÓ I ESPECIFICACIÓ DE CÀRRECS DE LA EMPRESA XYZ,CA	Pàgina N° 28
		Data d'Elaboració: 09/05/2022
		Data d'Aprovació:

```

        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_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_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;
    }
}
```

	MANUAL DE DESCRIPCIÓ I ESPECIFICACIÓ DE CÀRRECS DE LA EMPRESA XYZ,CA	Pàgina N° 30
		Data d'Elaboració: 09/05/2022
		Data d'Aprovació:

```

    }
}
@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();

```



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

Pàgina N° 31


Data d'Elaboració: 09/05/2022

Data d'Aprovació:

```
        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()+(tecnoAttack*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() {
```

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

```
// 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_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;
    }
}
```


	MANUAL DE DESCRIPCIÓ I ESPECIFICACIÓ DE CÀRRECS DE LA EMPRESA XYZ,CA	Pàgina N° 33
		Data d'Elaboració: 09/05/2022
		Data d'Aprovació:

```

}
public void setQuantity(int quantity) {
    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 tecnoAtack, 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()
/100);

this.setBaseDamage(this.getBaseDamage()+(tecnoAttack*PLUS_ATTACK_IONCANNON
N_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
```

	MANUAL DE DESCRIPCIÓ I ESPECIFICACIÓ DE CÀRRECS DE LA EMPRESA XYZ,CA	Pàgina N° 35
		Data d'Elaboració: 09/05/2022
		Data d'Aprovació:

```
CHANCE_GENERATING_WASTE_IONCANNON;}
}
```

- Clase PlasmaCannon

Clase PlasmaCannon hereda de Defense ya que es un tipo de nave. También implementa las interfaces MilitaryUnit y Variables porque estas 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.getInitialArmor()/100);

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

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

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

```

@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_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.

Elaborat per: Grupo 4	Revisat per: Grupo 4	Autoritzat per: Josep Diaz
-----------------------	----------------------	----------------------------



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

Pàgina N° 37

Data d'Elaboració: 09/05/2022

Data d'Aprovació:

```
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
rмор()/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;}
```

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

```

@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_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;

```

	MANUAL DE DESCRIPCIÓ I ESPECIFICACIÓ DE CÀRRECS DE LA EMPRESA XYZ,CA	Pàgina N° 39
		Data d'Elaboració: 09/05/2022
		Data d'Aprovació:

```

class VentanaLogin extends JFrame {
    private JLabel titulo,nom,pwd;
    private JTextField nombre;
    private JPasswordField password;
    private JPanel panelG,panelLogin;
    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.getPass
word()));

            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);
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();
    }
}
```

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

```

setTitle("User Registration");

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

```

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

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

        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);
```

	MANUAL DE DESCRIPCIÓ I ESPECIFICACIÓ DE CÀRRECS DE LA EMPRESA XYZ,CA	Pàgina N° 45
		Data d'Elaboració: 09/05/2022
		Data d'Aprovació:

```

        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(650,375);
    // 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 stringValue(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 DESCRIPCIÓ I ESPECIFICACIÓ DE CÀRRECS DE LA EMPRESA XYZ,CA	Pàgina N° 46
		Data d'Elaboració: 09/05/2022
		Data d'Aprovació:

- Interfaz MilitaryUnit

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

```
import java.sql.Connection;

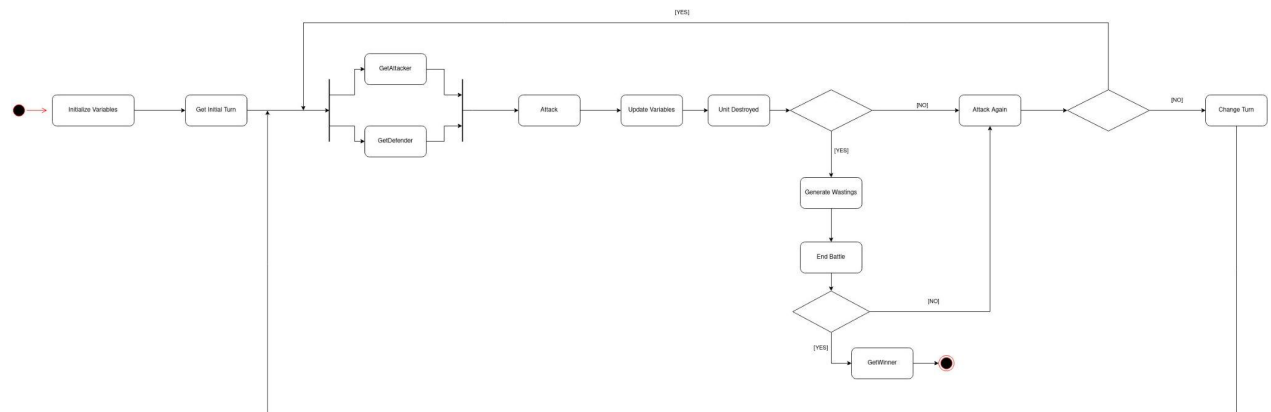
public interface MilitaryUnit {
    abstract int attack();
    abstract void takeDamage(int receivedDamage);
    abstract int getActualArmor();
    abstract int getMetalCost();
    abstract int getDeuteriumCost();
    abstract int getChanceGeneratinWaste();
    abstract int getChanceAttackAgain();
    abstract int getQuantity();
    abstract void setQuantity(int quantity);
    abstract void resetArmor();
    //take damage
}
```



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 DESCRIPCIÓ I ESPECIFICACIÓ DE CÀRRECS DE LA EMPRESA XYZ,CA	Pàgina N° 48
		Data d'Elaboració: 09/05/2022
		Data d'Aprovació:

MI - 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 4	Revisat per: Grupo 4	Autoritzat per: Josep Diaz
-----------------------	----------------------	----------------------------



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

Pàgina N° 49

Data d'Elaboració: 09/05/2022

Data d'Aprovació:

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	
PRIVATE	
FOR	
MATH.RANDOM()	
LENGTH()	
SIZE()	
INSTANCEOF	
SWITCH()	
CASE	
IF - ELSE IF - ELSE	
ADD	
MAX()	
LOGIN	
BOOLEAN	
TRUE-FALSE	
NO_DATA_FOUND	

Elaborat per: Grupo 4

Revisat per: Grupo 4

Autoritzat per: Josep Diaz



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

Pàgina N° 50

Data d'Elaboració: 09/05/2022


Data d'Aprovació:

[illegible]

Elaborat per: Grupo 4

Revisat per: Grupo 4

Autoritzat per: Josep Diaz

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

Elaborat per: Grupo 4	Revisat per: Grupo 4	Autoritzat per: Josep Diaz
-----------------------	----------------------	----------------------------