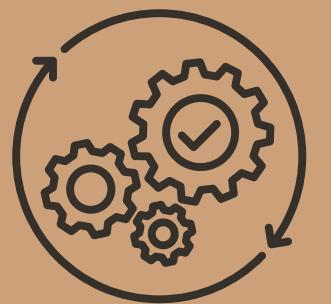


# **PROJECTE FINAL JAVAFX**

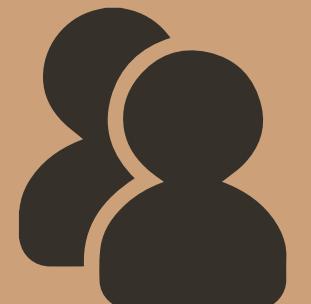
# CONTINGUT

- 01 Introducció
- 02 Distribucció en el GitHub
- 03 Connexió BD
- 04 Interfaç Gràfica
- 05 PixelArt
- 06 Joc de la Vida
- 07 Pescamines
- 08 Wordle
- 09 Conclusió

# EL NOSTRE PROJECTE



Anem a introduir el nostre projecte final, explicat pas per pas el que hem fet i el resultat d'aquest.



L'equip esta format per: Jorge, Georgi, Jaume, Ivan i Pablo



# DISTRIBUCIÓ GITHUB

Projecte Equip03

Add status update ·

Projecte · + New view

Filter by keyword or by field · Discard

**Todo** 0 · ...  
This item hasn't been started

**In Progress** 9 · ...  
This is actively being worked on

**Done** 16 · ...  
This has been completed

**SPRINT** 1 · ...  
Contingut dels puch setmanals

+ Add item

+ Add item

+ Add item

+ Add item

Projecte\_E03 #10 · Codi Registre

Projecte\_E03 #25 · Full d'estils de la interfície (Pablo)

Projecte\_E03 #24 · Disseny gràfic de la interfície

Projecte\_E03 #38 · Disseny gràfic

Projecte\_E03 #9 · Bases de Dades

Projecte\_E03 #28 ·

Projecte\_E03 #26 · Menú d'entrada

Projecte\_E03 #4 · CONSULTES A RAÜL

Projecte\_E03 #15 · Codi Pescamines

Projecte\_E03 #6 · Codi Wordle

Projecte\_E03 #16 · Codi Woodle

Projecte\_E03 #11 ·

# DISTRIBUCIÓ GITHUB

**In Progress** 9

This is actively being worked on

- Projecte\_E03 #9 Bases de Dades
- Projecte\_E03 #28 Interfície crear usuari
- Projecte\_E03 #5 Codi Pescamines
- Projecte\_E03 #7 Codi Joc de la Vida
- Projecte\_E03 #8 Codi Pixelart

**Done** 16

This has been completed

- Projecte\_E03 #16 Codi Woodle
- Projecte\_E03 #11 Asociar col·laboradors
- Projecte\_E03 #12 1er SPRINT
- Projecte\_E03 #13 2on SPRINT
- Projecte\_E03 #17 Prova Classe Casella, Tauler general
- Projecte\_E03 #24
- Projecte\_E03 #34 Codi JocdelaVida
- Projecte\_E03 #37 Vídeo explicatiu
- Projecte\_E03 #27 Interfície login
- Projecte\_E03 #29 Interfície elegir joc
- Projecte\_E03 #33 Codi PixelArt
- Projecte\_E03 #35

# CONNEXIÓ BD

```
public static boolean connectarScriptBD(String direccio) {  
    String url = "jdbc:mysql://localhost:3306/";  
    String script = llegirFitxerBD(direccio);  
    String[] aux = script.split(";");
    String usuari = "root";
    String contrasenya = "root";

    try (Connection c = DriverManager.getConnection(url, usuari, contrasenya));
Statement s = c.createStatement(); {
        ConnexioBD.conexio = c;
        Class.forName("com.mysql.cj.jdbc.Driver");
        for (String line : aux) {
            line = line + ";";
            if (!line.matches("^\\s*#.*$") || !line.matches("^\\s*--.*$")) {
                int execucio = s.executeUpdate(line);
            }
        }
        System.out.println("Base de dades creada amb script.");
        return true;
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
        System.err.println("Error en la class de l'script BD: " + e);
        return false;
    } catch (SQLException e) {
        System.err.println("Error en la connexió de l'script BD");
        e.printStackTrace();
        return false;
    }
}
```

```
public static String llegirFitxerBD(String direc) {
    String script = "";
    String aux = "";
    File f = new File(direc);

    if (f.exists() && f.isFile()) {

        try (FileReader fr = new FileReader(f)) {
            BufferedReader br = new BufferedReader(fr);

            while (br.ready()) {
                aux = br.readLine();
                if (!aux.matches("^--.*$"))
                    script += aux;
            }
            fr.close();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    } else
        System.out.println("Error: la ruta indicada no és un arxiu o no existeix.");
    return script;
}
```

# DADES “SINGLETON” DESPRÉS DE LES DE BASES DE DADES

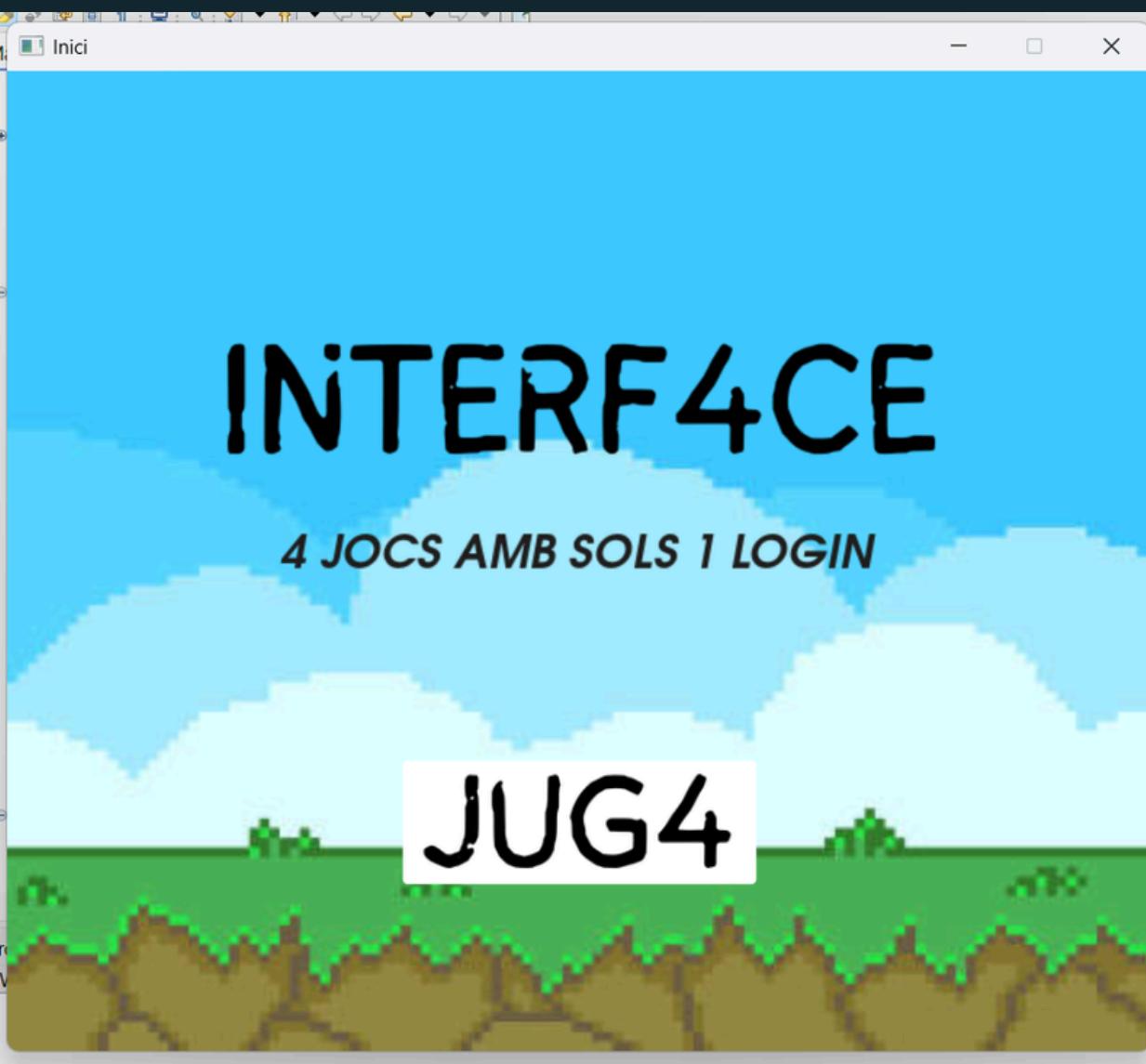
```
public class DadesSingleton {  
    private String cadenaCompartida;  
    private File partidaCompartida;  
    private int tamanyCompartit;
```

# CONNEXIÓ BD

```
public static int insertarDades(String taula, byte[] arxiu, String[] camps, String[] valors) {  
    int resultat = 0;  
    int[] tipusCamps = tipusCamp(taula, camps);  
    System.out.println("insertarDades");  
    if (valors.length == 0 || camps.length == 0) {  
        return 0;  
    }  
  
    String sentencia = "INSERT INTO " + taula + " (" + Arrays.toString(camps).replaceAll("\\[|\\]", "")  
        + ") VALUES (" + "?,".repeat(camps.length - 1) + "?);";  
  
    try (PreparedStatement ps = connexio.prepareStatement(sentencia, Statement.RETURN_GENERATED_KEYS)) {  
  
        for (int o = 0; o < valors.length; o++) {  
            if (!valors[o].isBlank())  
                conversioDadesBD(ps, tipusCamps[o], o + 1, valors[o]);  
            else  
                ps.setBytes(o + 1, arxiu);  
        }  
  
        ps.executeUpdate();  
  
        ResultSet r = ps.getGeneratedKeys();  
        if (r.next()) {  
            resultat = r.getInt(1);  
        }  
        System.out.println("S'han inserit " + resultat + " dades.");  
        return resultat;  
    } catch (SQLException e) {  
        e.printStackTrace();  
        return resultat;  
    }  
}
```

```
public static String obtencioDadesBD(ResultSet rs, int tipus, String col) throws SQLException {  
    // https://stackoverflow.com/questions/12367828/how-can-i-get-different-datatypes-from-resultsetmetadata-in-java  
    // Taula feta en base a un excel tret de Gemini.ia :  
    // https://docs.google.com/spreadsheets/d/1pl1vdyujL0XSci0Mn-u80nAfBCb6mboEB0alzrkZibU/edit?usp=sharing  
  
    switch (tipus) {  
        case Types.INTEGER:  
        case Types.SMALLINT:  
        case Types.TINYINT:  
            return "" + rs.getInt(col);  
        case Types.BIGINT:  
            return "" + rs.getLong(col);  
        case Types.BOOLEAN:  
        case Types.BIT:  
            return "" + rs.getBoolean(col);  
        case Types.NUMERIC:  
        case Types.DECIMAL:  
        case Types.DOUBLE:  
            return "" + rs.getDouble(col);  
        case Types.REAL:  
        case Types.FLOAT:  
            return "" + rs.getFloat(col);  
        case Types.NCHAR:  
        case Types.NVARCHAR:  
        case Types.LONGNVARCHAR:  
        case Types.VARCHAR:  
        case Types.CHAR:  
        case Types.LONGVARCHAR:  
            return rs.getString(col);  
        case Types.DATE:  
            return "" + rs.getDate(col);  
        case Types.TIME:  
            return "" + rs.getTime(col);  
    }  
    default:  
        System.err.println("Error: de tipus de daders.");  
    }  
}
```

# INTERFAÇ GRÀFICA



Inici

LOGIN

Usuari

Contraseña

Registrar Login

Inici

FORMULARI DE REGISTRE

Nom

Cognoms

Població

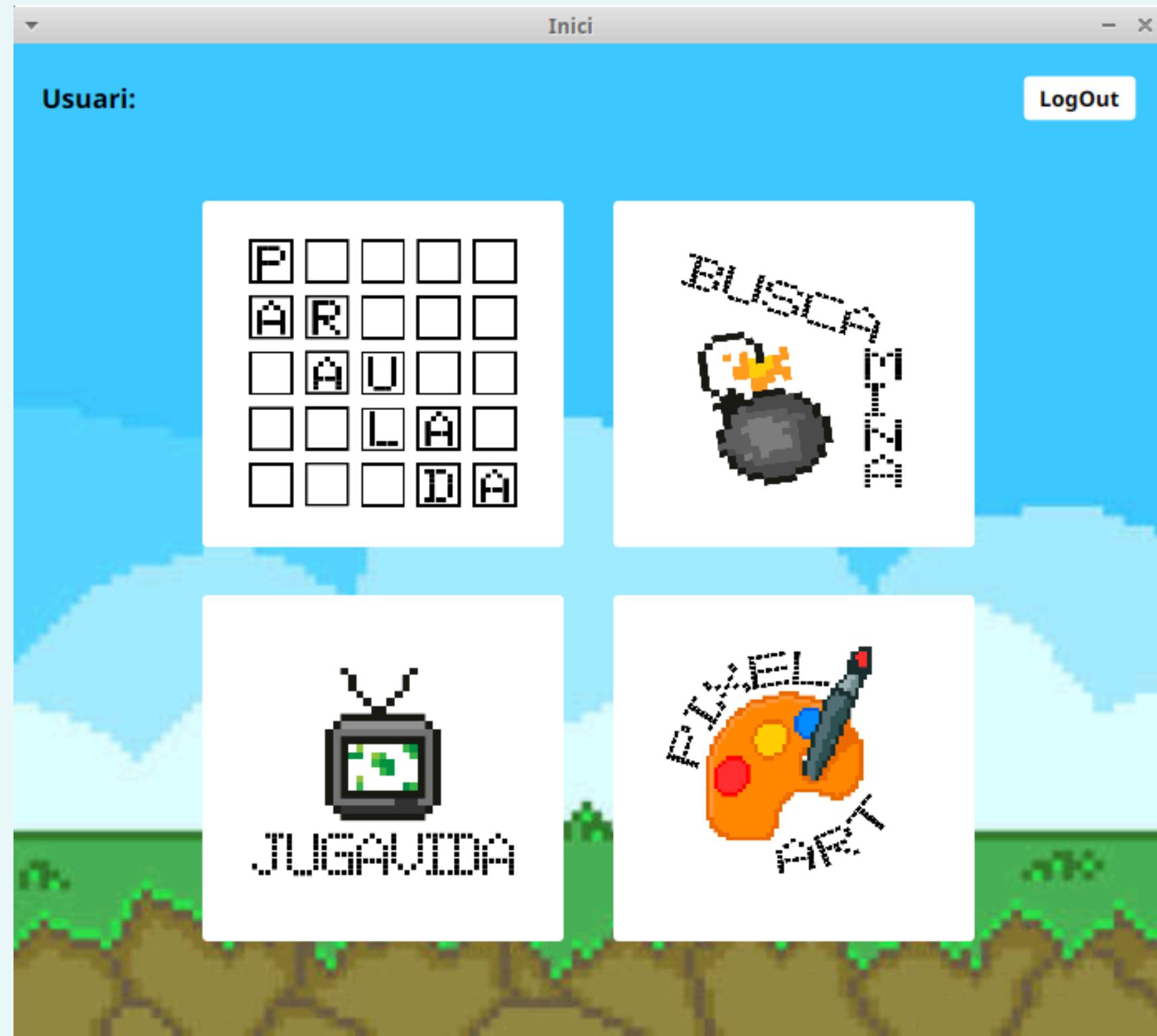
Correu electrònic

Nom de la imatge (ex: avatar.png)

Contraseña

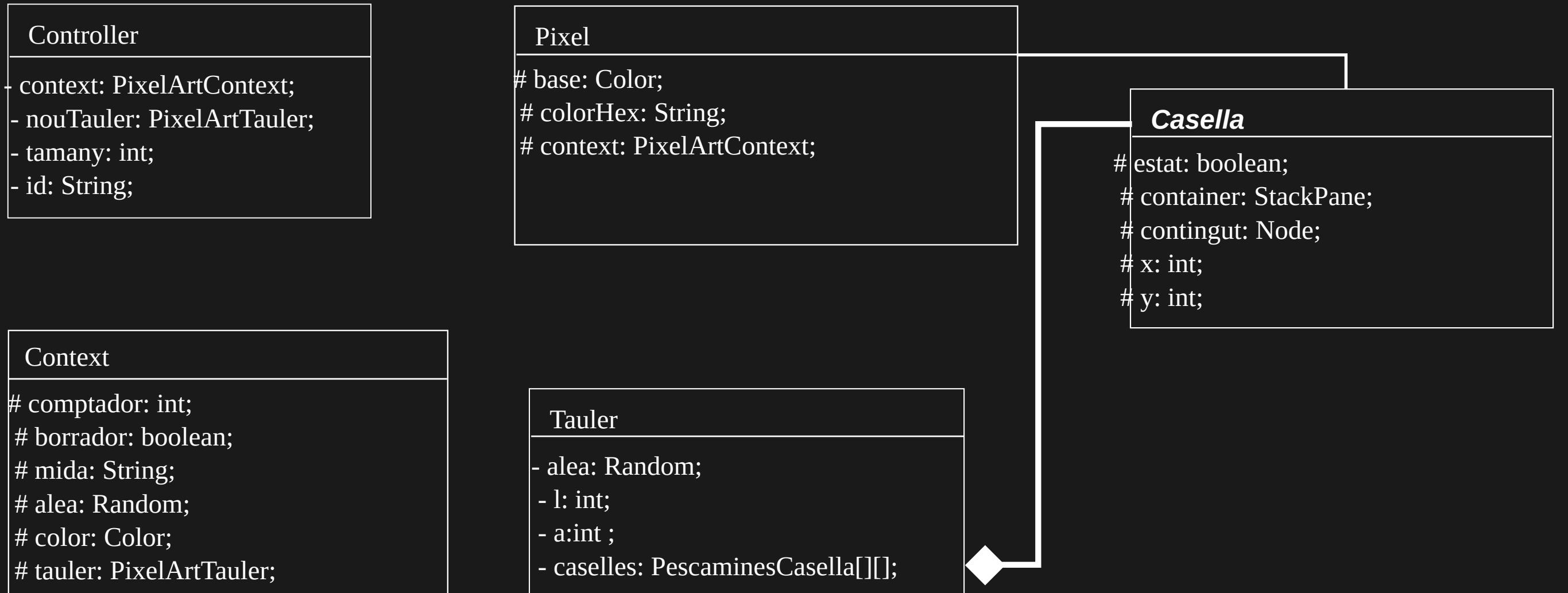
Registrar Tornar

# INTERFAÇ GRÀFICA



# PIXELART

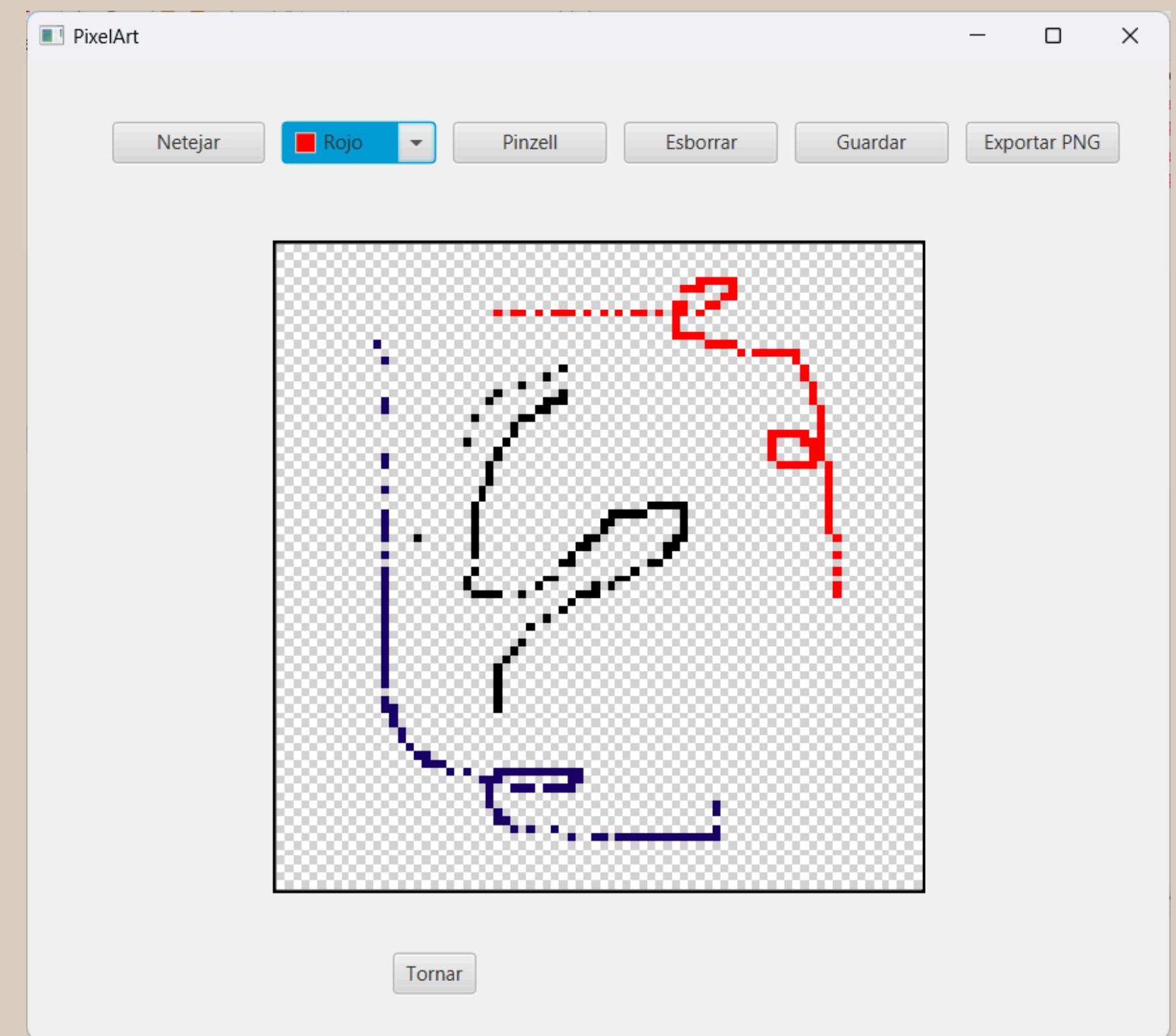
## Pixel Art



# PIXELART

```
public void pintar(Node n, PixelPixelArt l) {
    n.setOnMousePressed(e -> {
        if (borrador) {
            n.setStyle("-fx-background-color: " + conversioAHex(l.getBase()) + ";");
            l.setColorHex("");
        } else {
            if (e.getButton() == MouseButton.PRIMARY) {
                String c = conversioAHex(color);
                l.setColorHex(c);
                n.setStyle("-fx-background-color: " + c + ";");
            } else if (e.getButton() == MouseButton.SECONDARY) {
                n.setStyle("-fx-background-color: " + conversioAHex(l.getBase()) + ";");
                l.setColorHex("");
            }
        }
    });
    n.setOnDragDetected(e -> {
        n.startFullDrag();
        e.consume();
    });
    n.setOnMouseDragEntered(e -> {
        if (borrador) {
            n.setStyle("-fx-background-color: " + conversioAHex(l.getBase()) + ";");
            l.setColorHex("");
        } else {
            if (e.isPrimaryButtonDown()) {
                String c = conversioAHex(color);
                l.setColorHex(c);
                n.setStyle("-fx-background-color: " + c + ";");
            } else if (e.isSecondaryButtonDown()) {
                n.setStyle("-fx-background-color: " + conversioAHex(l.getBase()) + ";");
                l.setColorHex("");
            }
        }
        e.consume();
    });
}
```

# PIXELART



# JOC DE LA VIDA

```
public void cicleDeLaVida() {  
  
    int[][] aux = seleccioNatural(cellules);  
  
    Cellula[][] nova = new Cellula[cellules.length][cellules.length];  
  
    for (int i = 0; i < aux.length; i++) {  
        for (int j = 0; j < aux[0].length; j++) {  
  
            nova[i][j] = new Cellula();  
            //nova[i][j].setEstat("morta");  
  
            if (cellules[i][j].getEstat().equals("moribunda")) {  
                nova[i][j].setEstat("morta");  
            } else if (cellules[i][j].getEstat().equals("naixement")) {  
                nova[i][j].setEstat("viva");  
                comptCel++;  
            } else  
                if (aux[i][j] == 1) {  
  
                    if (cellules[i][j].getEstat().equals("morta") || cellules[i][j].getEstat().equals("moribunda")) {  
                        nova[i][j].setEstat("naixement");  
                        naixements++;  
                        comptCel++;  
                    } else {  
                        nova[i][j].setEstat("viva");  
                    }  
                } else {  
                    if (cellules[i][j].getEstat().equals("viva")) {  
                        nova[i][j].setEstat("moribunda");  
                        comptMor++;  
                    } else {  
                        comptMor++;  
                        nova[i][j].setEstat("morta");  
                    }  
                }  
            }  
        }  
    }  
}
```

```
// Comparar l'actual matriu amb la que se li envia  
public boolean compararCaselles(Cellula[][] c, int n) {  
    boolean result = true;  
    // https://  
    // www.lawebdelprogramador.com/foros/Java/674725-Saber-si-un-objeto-existe.html  
    for (int i = 0; i < c.length; i++) {  
        for (int j = 0; j < c[0].length; j++) {  
  
            // Comparació en profunditat  
            Cellula a = this.cellules[i][j];  
            Cellula b = c[i][j];  
  
            if (!a.getEstat().equals(b.getEstat()))  
                return false;  
        }  
    }  
    return result;  
}  
  
public boolean compararCaselles(Cellula[][] c) {  
    boolean result = true;  
    // https://  
    // www.lawebdelprogramador.com/foros/Java/674725-Saber-si-un-objeto-existe.html  
    for (int i = 0; i < c.length; i++) {  
        for (int j = 0; j < c[0].length; j++) {  
            Cellula a = this.cellules[i][j];  
            Cellula b = c[i][j];  
            if (!a.getEstat().equals(b.getEstat()))  
                return false;  
        }  
    }  
    return result;  
}
```

# JOC DE LA VIDA

```
// creació de plantilla d'enteros per a indicar qui mor i qui viu
public int[][] seleccioNatural(Cellula[][] c) {
    int[][] aux = new int[c.length][c.length];
    int veines;
    for (int i = 0; i < c.length; i++) {
        for (int j = 0; j < c[0].length; j++) {
            veines = 0;
            veines = estatVida(c, i, j);

            if (c[i][j].getEstat().equals("viva")) {
                if (veines == 2 || veines == 3)
                    aux[i][j] = 1;
            } else if (c[i][j].getEstat().equals("mortua")) {
                if (veines == 3)
                    aux[i][j] = 1;
            } else
                aux[i][j] = 0;
        }
    }
    return aux;
}
```

```
// mètode de creació de cel·lules
public Cellula[][] crearVida(Cellula[][] c, Random alea) {
    int a = alea.nextInt(c.length);
    int b = alea.nextInt(c.length);

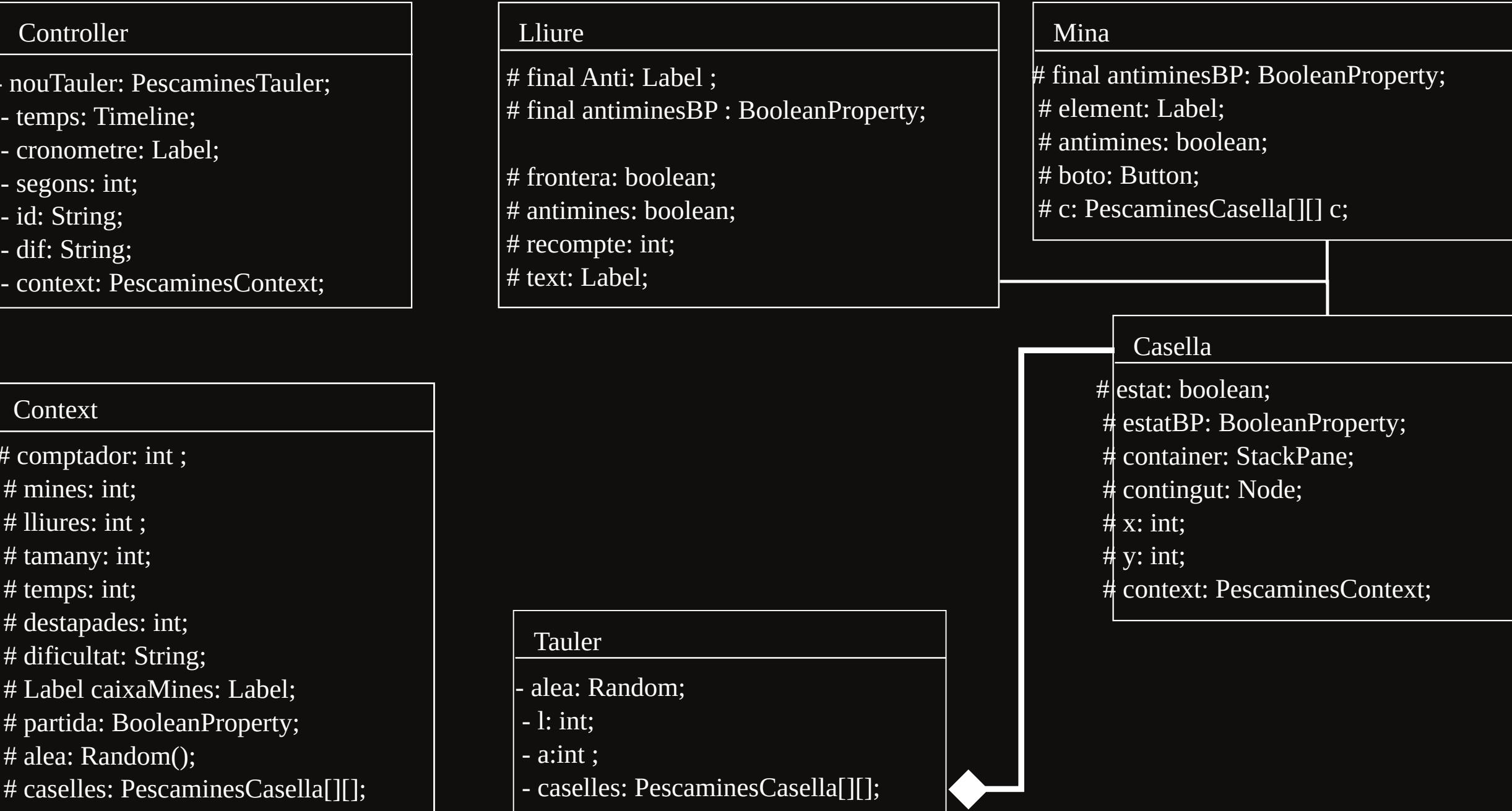
    if (c[a][b].getEstat().equals("mortua")) {
        c[a][b].setEstat("viva");
        comptCel++;
    } else
        crearVida(c, alea);
}
}
```

# JOC DE LA VIDA



# PESCAMINES

## Pescamines



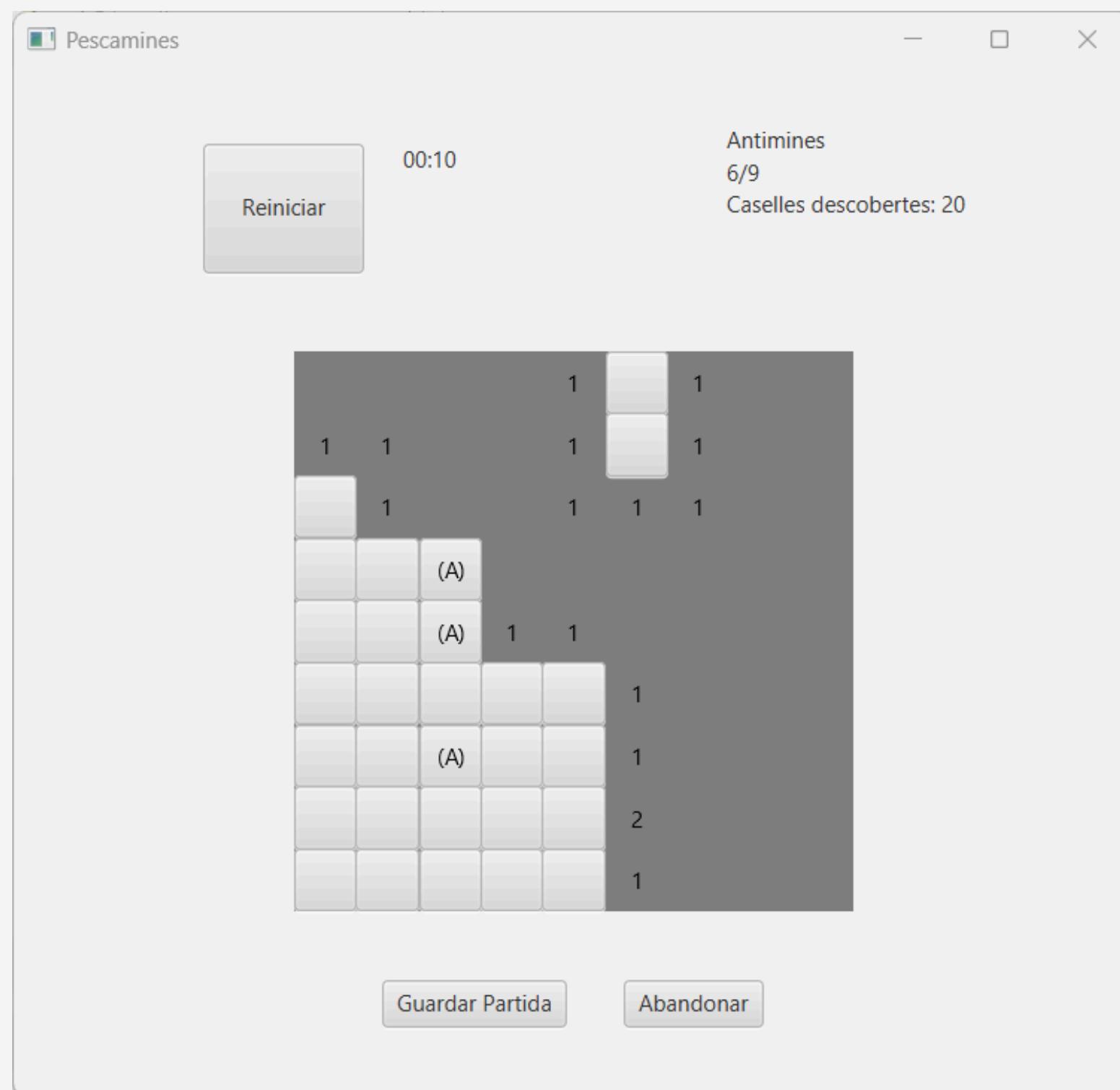
# PESCAMINES

```
@Override  
public void reaccio() {  
    boto.setOnMouseClicked(null);  
    Anti.setOnMouseClicked(null);  
  
    boto.setOnMouseClicked(new EventHandler<MouseEvent>() {  
        @Override  
        public void handle(MouseEvent e) {  
            if (e.getButton() == MouseButton.SECONDARY) {  
                if (!antimines && context.disminuirComptador()) {  
                    antimines = !antimines;  
                    Anti.setVisible(antimines);  
                    e.consume();  
                } else if (antimines && context.aumentarComptador()) {  
                    antimines = !antimines;  
                    Anti.setVisible(antimines);  
                    e.consume();  
                }  
            }  
  
            if ((e.getButton() == MouseButton.PRIMARY) && !antimines) {  
                despejar(Lliure.this, Lliure.this.c);  
  
                e.consume();  
            }  
        }  
    });  
}  
  
});
```

```
context.getPartida().addListener((obs, newVal, newVal) -> {  
    if (!newVal) {  
        acabarPartida();  
    }  
});  
}
```

```
public Lliure despejar(Lliure l, PescaminesCasella[][] c) {  
  
    if (l == null || !l.descobrir(l)) {  
        return l;  
    }  
    if (l.recompte == 0) {  
        if (l.getX() < c.length - 1) {  
            despejar((Lliure) c[l.getX() + 1][l.getY()], c);  
        }  
        if (l.getX() > 0) {  
            despejar((Lliure) c[l.getX() - 1][l.getY()], c);  
        }  
        if (l.getY() < c[0].length - 1) {  
            despejar((Lliure) c[l.getX()][l.getY() + 1], c);  
        }  
        if (l.getY() > 0) {  
            despejar((Lliure) c[l.getX()][l.getY() - 1], c);  
        }  
        return l;  
    }  
    return l;  
}  
  
public boolean descobrir(Lliure l) {  
    boolean descoberta = false;  
  
    if (l instanceof Lliure && l.isEstat() == true && !l.antimines) {  
        descoberta = true;  
  
        l.setEstat(false);  
        l.boto.setVisible(l.isEstat());  
        l.text.setVisible(true);  
        l.Anti.setVisible(false);  
        context.setLliures(--context.lliuress);  
        context.getCaixaMines().setText("Antimines\n" + context.getComptador() + "/" + context.getTamany(  
escobertes: " + (context.getGrandaria()-context.getLliures()));  
  
        context.comprovarPartida();  
    }  
    return descoberta;  
}
```

# PESCAMINES



# WORDLE

```
@FXML
private void gestionarIntent() {
    String intent = entrada.getText().toLowerCase();
    entrada.clear();

    if (intent.length() != 5) {
        pista.setText("La paraula ha de tindre exactament 5 lletres.");
        return;
    }

    String normalObjectiu = eliminarAccents(paraulaObjectiu);
    String normalIntent = eliminarAccents(intent);
    HBox filaIntent = new HBox(5);
    filaIntent.setAlignment(Pos.CENTER);
    boolean encertTotal = true;

    for (int i = 0; i < 5; i++) {
        char lletra = intent.charAt(i);
        Label casella = new Label(String.valueOf(Character.toUpperCase(lletra)));
        casella.setMinSize(40, 40);
        casella.setStyle("-fx-font-size: 18px; -fx-font-weight: bold; -fx-alignment: center; -fx-border-color: black;");

        char lletraNormal = normalIntent.charAt(i);
        if (normalObjectiu.charAt(i) == lletraNormal) {
            casella.setStyle(casella.getStyle() + " -fx-background-color: green; -fx-text-fill: white;");
        } else if (normalObjectiu.contains(String.valueOf(lletraNormal))) {
            casella.setStyle(casella.getStyle() + " -fx-background-color: gold; -fx-text-fill: black;");
            encertTotal = false;
        } else {
            casella.setStyle(casella.getStyle() + " -fx-background-color: gray; -fx-text-fill: white;");
            encertTotal = false;
        }
        filaIntent.getChildren().add(casella);
    }
    root.getChildren().add(filaIntent);
    intentsFets++;

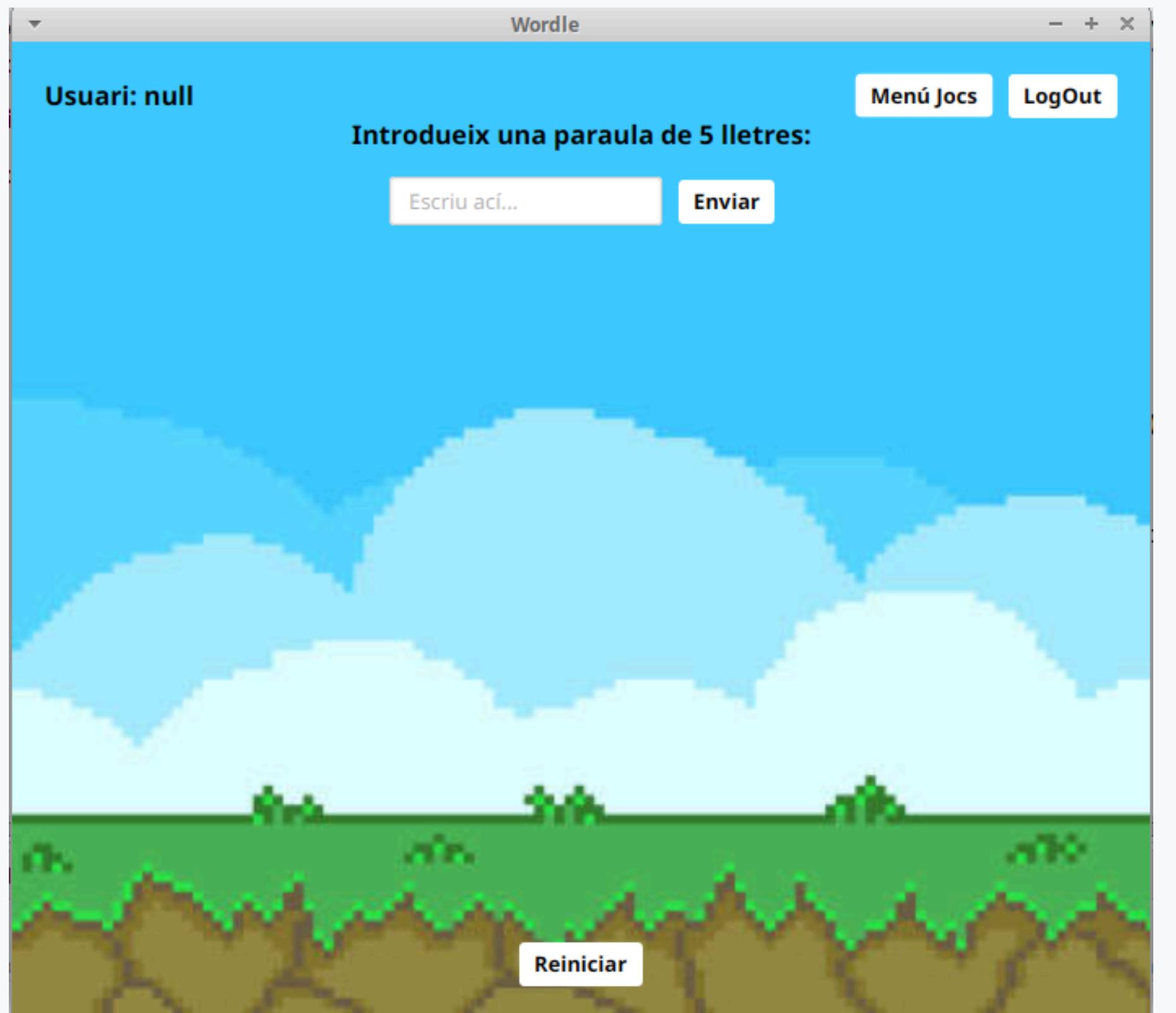
    if (encertTotal) {
        pista.setText("✿ Enhorabona! Has encertat la paraula!");
        entrada.setDisable(true);
        enviar.setDisable(true);
    } else if (intentsFets >= 5) {
        pista.setText("✖ Has esgotat els intents! La paraula era: " + paraulaObjectiu.toUpperCase());
        entrada.setDisable(true);
        enviar.setDisable(true);
    } else {
        pista.setText("Intent " + intentsFets + " de 5");
    }
}
```

```
private void carregarParaules() throws IOException {
    InputStreamReader isr = new InputStreamReader(getClass().getResourceAsStream("/paraules.txt"));
    BufferedReader br = new BufferedReader(isr);
    paraules = br.lines()
        .map(String::toLowerCase)
        .filter(p -> p.length() == 5)
        .collect(Collectors.toList());
    br.close();
}

private String eliminarAccents(String text) {
    return Normalizer.normalize(text, Normalizer.Form.NFD)
        .replaceAll("\\p{InCombiningDiacriticalMarks}+", "");
}

private void mostrarError(String missatge) {
    Alert alert = new Alert(Alert.AlertType.ERROR, missatge, ButtonType.OK);
    alert.showAndWait();
    System.exit(1);
}
```

# WORDLE



# CONCLUSIONS

Errors:

1. Dependència de tasques (de programació de codi a aspecte visual i gestió del projecte)
2. Amuntonament de càrrega de feina al final dels sprints.
3. Mala planificació inicial de tasques (major presència de micro)
4. Mal plantejament teòric del codi
5. Mal repartiment de tasques (d'allò macro resulta responsabilitat sobrecarregada i risc de sobrevaloració de la força de treball.

## Abstracció vs. Utilitat

Encerts:

1. Sprint final generalment molt efectiu
2. Implicació general en el moment crític
3. Acompliment, en gran part, dels objectius
4. Efectivitat de les decisions

Abstracció

Informació d'acord amb conceptes  
Reutilització de codi  
Comprendsió esquemàtica  
Optimització de

Utilitat

Lògica computacional (comprendre el funcionament de la migració i flux de dades l'ordinador)  
Fluïdesa de funcionament

Elements  
IG

Elements  
Prog. lògica